

Mathematics in Industry 27

The European Consortium for Mathematics in Industry

Luca Ghezzi

Dietmar Hömberg

Chantal Landry *Editors*

Math for the Digital Factory

Editors

Hans Georg Bock
Frank de Hoog
Avner Friedman
Arvind Gupta
André Nachbin
Tohru Ozawa
William R. Pulleyblank
Torgeir Rusten
Fadil Santosa
Jin Keun Seo
Anna-Karin Tornberg

THE EUROPEAN CONSORTIUM
FOR MATHEMATICS IN INDUSTRY

SUBSERIES

Managing Editor
Michael Günther

Editors
Luis L. Bonilla
Otmar Scherzer
Wil Schilders

More information about this series at <http://www.springer.com/series/4650>

Luca Ghezzi • Dietmar Hömberg • Chantal Landry
Editors

Math for the Digital Factory

 Springer


EUROPEAN CONSORTIUM FOR
MATHEMATICS IN INDUSTRY

Editors

Luca Ghezzi
ABB Italy S.p.A.
Mathematical Modeling & Numerical
Simulations
Vittuone, MI, Italy

Dietmar Hömberg
Weierstrass Institute for Applied Analysis
and Stochastics
Berlin, Germany

Chantal Landry
Zurich University of Applied Sciences
Winterthur, Switzerland

ISSN 1612-3956 ISSN 2198-3283 (electronic)
Mathematics in Industry
The European Consortium for Mathematics in Industry
ISBN 978-3-319-63955-0 ISBN 978-3-319-63957-4 (eBook)
DOI 10.1007/978-3-319-63957-4

Library of Congress Control Number: 2017955618

Mathematics Subject Classification (2010): 35Q80, 37N40, 49M37, 90C90

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

Manufacturing industry has, for decades, been a key element in the prosperity of regions and countries around the world, and, more precisely, in Europe. The competitiveness of our strong industry, which leads the global competition in diverse sectors, is grounded on its technological strength. New technologies and their industrialisation are closely linked to the application and the market in the knowledge life cycle. However, if we take a closer look at the basics, the fundamentals of these innovative technologies, we can easily identify the contribution of mathematics: industrial cybersecurity, machine learning, manufacturing process monitoring and diagnosis, product and process design and analysis tools, navigation systems for mobile devices, robotics, artificial vision systems—all of them are based on diverse mathematical fields: analysis, algebra, statistics, probability, topology, geometry, numeric methods, etc. This book provides a brilliant example of cooperation between the scientific and industrial worlds, and I would like to welcome it on behalf of the European “Factories of the Future” research community. Readers will find articles that clearly explain the contribution of mathematics to solving diverse problems in industry, or to providing better tools for the manufacturing sector. In the age of the digitalisation of factories, the smartisation of the productive sector, or, as it is called in some countries, the fourth industrial revolution, let us celebrate the contribution from mathematics, even in the naming of the whole new paradigm, 4.0!

Donostia/San Sebastián, Spain
June 2017

Dr. Rikardo Bueno¹

¹Dr. Rikardo Bueno is Co-chair of the Factories of the Future private-public partnership board and director of the advanced manufacturing area at Tecnalia, Mikeletegi 7, 20009 Donostia/San Sebastián.

Preface

The digital factory represents a network of digital models, as well as simulation and 3D visualisation methods for the holistic planning, realisation, control and ongoing improvement of all factory processes related to a specific product. In the last ten years, all industrialised countries have launched initiatives to realise this vision, sometimes also referred to as Industry 4.0 (in Europe) or Smart Manufacturing (in the United States). Its main goals are

- reconfigurable, adaptive and evolving factories capable of small-scale production
- high performance production, combining flexibility, productivity, precision and zero defect
- energy and resource efficiency in manufacturing

None of these goals can be achieved without the development of new concepts for the mathematical modelling, simulation and optimisation of all aspects of manufacturing on multiple scales, ranging from individual production processes to work cells and large-scale manufacturing chains with hundreds of robots, together with new concepts for bridging these in a multi-scale framework for manufacturing. To foster collaboration between mathematics and industry in this area, the European Consortium for Mathematics in Industry (ECMI) founded a new special interest group on Math for the Digital Factory (M4DiFa).

This volume compiles a selection of review papers from the M4DiFa kick-off meeting, which took place from 7 to 9 May 2014 at Weierstrass Institute for Applied Analysis and Stochastics in Berlin, Germany. Mathematicians and industry practitioners discussed the essential role of mathematics in the competitiveness of manufacturing companies. The topics addressed include state-of-the-art mathematical models and simulation tools to minimise production costs, reduce the time to market, as well as increase energy and resource efficiency. The book includes 15 articles and is divided into 3 sections. The first is concerned with the planning and scheduling of production systems and presents innovative mathematical techniques such as mixed integer linear programs (T. Kis and M. Drótos), continuous partial differential equations (S. Göttlich et al.) and $(\max,+)$ -algebra to model and optimise production processes (X. David-Henriet et al.). The paper by L. Ghezzi offers

new insights into the optimal storage of physical goods. The last contribution (A. Lüder and N. Schmidt) shows the crucial link between mechatronical engineering and mathematics for digital factories. The second section covers the optimisation of production lines. The first two papers (F. Damrath et al. and P. Burget et al.) discuss the optimisation of energy consumption in a robotic cell, while T. Hajba et al., D. Hömberg et al., and K. Palagachev and M. Gerdts make new contributions to the task assignment and scheduling of jobs to robots. In addition, these last two papers provide innovative solutions to the crucial issue of the collision-free coordination of robots. The last section describes several production technologies. First, F. Edelvik et al. discuss the full simulation of electrostatic spray painting and sealing. D. Brander et al. provide insights into the use of robots to cut expanded polystyrene for the building industry. M. Krüger et al. present a self-correcting strategy for a metal forming process, while J. Linn and K. Dressler provide realistic simulations of large deformations of slender flexible structures. Last but not least, C. Leithäuser and R. Pinnau model, simulate and optimise the entire process chain for the production of non-woven materials. We hope this volume will be of interest to researchers in applied mathematics as well as practitioners and engineers from the manufacturing sector. Even though many contributions have been written at an advanced mathematical level, efforts have been made to make the aim and the impact on digital factories in each paper understandable to a wide audience. Lastly, we would like to express our gratitude to the Research Center MATHEON, the Committee for Mathematical Modeling, Simulation and Optimization (KoMSO), the European Consortium for Mathematics in Industry (ECMI) and the Weierstrass Institute, whose financial support made the workshop possible.

Milano, Italy
Berlin, Germany
Winterthur, Switzerland
May 2017

Luca Ghezzi
Dietmar Hömberg
Chantal Landry

Contents

Part I Planning and Scheduling of Production Systems

Hard Planning and Scheduling Problems in the Digital Factory	3
Tamás Kis and Márton Drótos	
Modeling of Material Flow Problems	21
Simone Göttlich, Michael Herty, and Melanie Luckert	
Max-Plus-Linear Systems for Modeling and Control of Manufacturing Problems	37
Xavier David-Henriet, Laurent Hardouin, and Jörg Raisch	
Stochastic Optimal Sizing of a Warehouse	61
Luca Ghezzi	
Challenges of Mechatronical Engineering of Production Systems: An Automation System Engineering View	93
Arndt Lüder and Nicole Schmidt	

Part II Optimization of Production Lines

Physics-Based Simulation for Energy Consumption Optimization of Automated Assembly Systems in the Automotive Industry	117
Felix Damrath, Anton Strahilov, Thomas Bär, and Michael Vielhaber	
Optimisation of Power Consumption for Robotic Lines in Automotive Industry	135
Pavel Burget, Libor Bukata, Přemysl Šůcha, Martin Ron, and Zdeněk Hanzálek	
Production Line Optimization with Model Based Methods	163
T. Hajba, Z. Horváth, C. Kiss-Tóth, and J. Jósvai	

Automatic Reconfiguration of Robotic Welding Cells	183
Dietmar Hömberg, Chantal Landry, Martin Skutella, and Wolfgang A. Welz	
Numerical Approaches Towards Bilevel Optimal Control Problems with Scheduling Tasks	205
Konstantin D. Palagachev and Matthias Gerdtz	
Part III Selected Production Technologies	
Math-Based Algorithms and Software for Virtual Product Realization Implemented in Automotive Paint Shops	231
Fredrik Edelvik, Andreas Mark, Niklas Karlsson, Tomas Johnson, and Johan S. Carlson	
Hot Blade Cuttings for the Building Industries	253
David Brander, Andreas Bærentzen, Anton Evgrafov, Jens Gravesen, Steen Markvorsen, Toke Bjerger Nørbjerg, Peter Nørtoft, and Kasper Steenstrup	
Model-Based Design of Self-Correcting Forming Processes	273
M. Krüger, M. Borzykh, U. Damerow, M. Gräler, and A. Trächtler	
Discrete Cosserat Rod Models Based on the Difference Geometry of Framed Curves for Interactive Simulation of Flexible Cables	289
Joachim Linn and Klaus Dreßler	
The Production of Filaments and Non-woven Materials: The Design of the Polymer Distributor	321
Christian Leithäuser and René Pinnau	
Author Index	341
Subject Index	343

Part I
Planning and Scheduling
of Production Systems

Hard Planning and Scheduling Problems in the Digital Factory

Tamás Kis and Márton Drótos

Abstract Production planning and scheduling with the aid of software tools in today's manufacturing industries have become a common practice which is indispensable for providing high level customer service, and at the same time to utilize the production resources, like workforce, machine tools, raw materials, energy, etc., efficiently. To meet the new requirements, problem modeling tools, optimization techniques, and visualization of data and results have become part of the software packages. In this chapter some recent developments in problem modeling and optimization techniques applied to important and challenging industrial planning and scheduling problems are presented. We will focus on new problem areas which are still at the edge of current theoretical research, but they are motivated by practical needs. On the one hand, we will discuss project based production planning, and on the other hand, we will tackle a resource leveling problems in a machine environment. We will present the problems, some modeling and solution approaches, and various extensions and applications.

1 Introduction

Digital factories encompass the digital model of most of the technical and business processes of physical production systems [17]. Running such a model in parallel with the real-world factory can help decrease the time required for the realization of products as well as production [6]. An integral part of digital factories is the modeling of planning and scheduling activities, as they are largely responsible for meeting customer's deadlines, and minimizing production costs. Therefore, software tools that are capable to obtain suboptimal production plans, and schedules are fundamental components of digital factory solutions [23]. In this chapter we overview some of the recent developments in automatic planning and scheduling of complex manufacturing processes. We will consider problems that in our experience frequently occur in practice, but they are not so well studied like several single

T. Kis (✉) • M. Drótos

Institute for Computer Science and Control, Kende str. 13-17, Budapest H-1111, Hungary

e-mail: kis.tamas@sztaki.mta.hu; drotos.marton@sztaki.mta.hu

machine scheduling problems, or the makespan minimization problem in job shops [5, 7].

We will introduce two problem areas in detail, and for each problem area we provide a problem formulation, some theoretical background, sketch a solution approach and summarize computational results. We will also provide references to the literature offering further results and extensions.

2 Project Based Production Planning

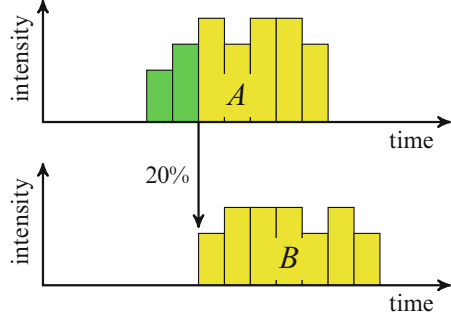
Traditionally, production planning is concerned with determining production quantities of final products and that of their subcomponents over time based on manufacturing lead times and the bill-of-materials. This approach may be inadequate when dealing with a large variety of products manufactured in small quantities. For instance, when a customer orders e.g., a complex product made of several components, which have to be designed, manufactured, assembled, tested, and finally delivered to the customer, then determining production quantities is just not the right way of making a feasible plan, not mentioning that a manufacturing company may have several big projects running concurrently, which have to be controlled separately.

In order to build a planning model, we will use the terminology of project scheduling [9]. A project consists of *activities* needing various *resources*, and connected by *precedence constraints*. Consider for instance a manufacturing firm which produces complex products for its customers. Each customer's order becomes a project, where an activity represents a major step in the project, like design, various manufacturing steps, assembly, etc. There is a natural precedence relation between the project activities. Design must precede manufacturing, which, in turn, is a prerequisite to assembly, and testing. As for the resources, the design activity requires engineers making the blueprint of the parts to be manufactured. The manufacturing steps may require CNC work centers, or milling/turning machines, heat treatment, etc. In the following we assume that the projects are broken down to some main steps, and for each step the key resources are known. So far, we can build a network of activities representing the main steps of the project which in the end delivers the final product to the customer.

Since production plans are made for a longer time horizon, e.g., 26 or 52 weeks, it is desirable that activities are also at the right aggregation level, e.g., the design of the project is represented by a few major design activities, which have a time span of several weeks. Resource are also aggregated, like chef-designers, or a group of identical CNC machines is considered as a single cumulative resource. The processing capacity of a cumulative resource equals the sum of the processing capacities of the resources grouped together.

In practice, the intensity of aggregated activities vary over time. The intensity increases gradually to a maximum level permitted by technological constraints, then it is run for a while at maximum, or close to maximum level, For instance, if a project needs 100h of CNC machining, only 10% can be done daily, since

Fig. 1 Variable intensity activities connected by a feeding precedence constraint



several operations must be done on the same part. On the other hand, the activities may overlap in time. For example, design provides blueprints to manufacturing, and as manufacturing of parts progresses, more and more components can be assembled. Therefore, we connect the activities of the project by *feeding precedence constraints*. Such a constraint specifies that, say, 20% of activity A must be completed before activity B may start, and then B cannot progress faster than A, for an illustration see Fig. 1.

It is natural to assume that resource consumption of variable intensity activities is proportional to their intensity over time. For renewable resources, like machine tools, or workforce, if the intensity of an activity i is x_t^i in time period t , then its resource consumption from some renewable resource R_k is $q_k^i \cdot x_t^i$, where q_k^i is the total demand of activity i from resource k . Of course, one may consider more general functions for computing the resource usage of activities depending on their intensity, however, we consider only linear functions in this paper.

In the rest of this section, we describe a mathematical model and discuss some solution approaches. Finally, we overview some possible extensions of the model.

2.1 Modeling by a Mixed-Integer Linear Program

Firstly, we introduce formally the problem data and the objective function, and then describe a mixed-integer linear program for solving it to optimality. There is a set N of activities, and a set RR of renewable resources. The time horizon is divided into time periods $1, \dots, T$, and any changes in the project can occur only at the border of two consecutive time periods. Each activity i has a time window $[r_i, d_i]$ in which it has to be completed, a maximum intensity $a^i > 0$, and resource requirements $q_k^i \geq 0$ for $k \in RR$. The activities are connected by feeding precedence constraints given by triples (i, j, f_{ij}) meaning that an f_{ij} fraction of activity i must be completed before starting activity j . Each renewable resource $k \in RR$ has a time varying capacity b_{kt} , and an additional external capacity which can be purchased at the expense of additional costs.

The objective is to find an intensity assignment to the activities such that each activity is entirely processed in its time window, the nonrenewable resource

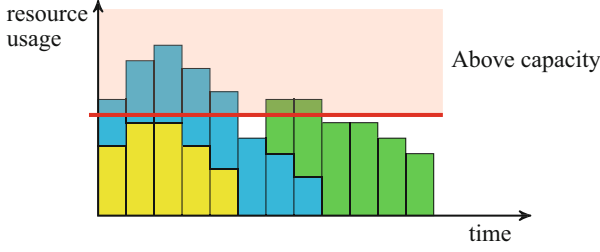


Fig. 2 Resource usage above internal capacity. The internal capacity is indicated by *horizontal thick line*

constraints are respected, and the cost of external resource usage is minimized. The external resource usage from some renewable resource $k \in RR$ can be measured as the total resource usage above the internal capacities, i.e., $\max\{\sum_{t=1}^T \sum_{i \in N} q_k^i x_t^i - b_{kt}, 0\}$. This is illustrated in Fig. 2. The motivation for this objective function is the fact that companies usually have their internal workforce and other production capacities from renewable resources, and in case of bottlenecks, they are willing to subcontract or hire some extra resources.

Now we are ready to present a mixed-integer linear program for modelling our problem. The decision variables are

x_t^i = intensity of activity i in time period t ,

y_{kt} = resource usage above internal capacity from resource $k \in RR$ in time period t ,

$z_t^{if} = \begin{cases} 0 & \text{if } f\text{-fraction of activity } i \text{ has been completed before time period } t, \\ 1 & \text{otherwise.} \end{cases}$

The meaning of other symbols in the following problem formulation are as follows:

\mathcal{N} = set of activities,

r^i, d^i = earliest and latest time periods when activity i can be processed,

N_t = set of activities with $r^i \leq t \leq d^i$,

a^i = maximum intensity of activity i ,

c_{kt} = cost of external resource usage from resource $k \in RR$ in time period t ,

b_{kt} = internal capacity of resource $k \in RR$ in time period t ,

\bar{b}_{kt} = additional external capacity of resource $k \in RR$ in time period t ,

E^i = set of precedence relations of the form (i, j, f) ,

F^i = set of fractions that occur in precedence constraints F^i ,

p^{if} = minimum number of periods to finish an f fraction of activity i

$$\min \sum_{t=1}^T \sum_{k \in RR} c_{kt} y_{kt} \quad (1)$$

subject to

$$\sum_{t=r^i}^{d^i} x_t^i = 1, \quad i \in \mathcal{N}, \quad (2)$$

$$\sum_{t=r^i}^{\ell-1} x_t^i + f z_\ell^{if} \geq f, \quad i \in \mathcal{N}, f \in F^i, \ell \in \{r^i + p^{if}, \dots, d^i\}, \quad (3)$$

$$x_t^j + a^j z_t^{jf} \leq a^j, \quad i \in \mathcal{N}, (i, j, f) \in E^i, t \in \{r^i + p^{if}, \dots, d^i\} \quad (4)$$

$$z_t^{if} - z_{t+1}^{if} \geq 0, \quad i \in \mathcal{N}, f \in F^i, t \in \{r^i + p^{if}, \dots, d^i - 1\}, \quad (5)$$

$$\sum_{t=r^i}^{\ell} x_t^i - \sum_{t=r^j}^{\ell} x_t^j \geq 0, \quad i \in \mathcal{N}, (i, j, f) \in E^i, \quad (6)$$

$$\ell \in \{\max\{r^i, r^j\}, \dots, \min\{d^i, d^j\}\},$$

$$\sum_{i \in \mathcal{N}_i} q_k^i \cdot x_t^i - y_{kt} \leq b_{kt}, \quad k \in RR, t \in \{1, \dots, T\}, \quad (7)$$

$$0 \leq x_t^i \leq a^i, \quad i \in \mathcal{N}, t \in \{r^i, \dots, d^i\}, \quad (8)$$

$$0 \leq y_{kt} \leq \bar{b}_{kt}, \quad k \in RR, t \in \{1, \dots, T\}, \quad (9)$$

$$z_t^{if} \in \{0, 1\}, \quad i \in \mathcal{N}, f \in F^i, t \in \{r^i + p^{if}, \dots, d^i\}, \quad (10)$$

The objective function (1) aims at the minimization of resource hiring/subcontracting costs. Constraints (2) ensure that each activity is totally processed in its time window. The precedence constraints between pairs of activities are described by inequalities (3) through (5). In particular, (3) ensures that z_t^{if} can be zero only if an f -fraction of activity i is completed up to time period $t - 1$. Moreover, (4) makes sure that a the successor j of activity i can only start if an f -fraction of activity i is completed. Inequalities (5) ensure that there is only one point in time when the z_t^{if} switches from 1 to 0. The feeding aspect is captured by (6). The external renewable resource usage is expressed by (7), since y_{kt} is non-negative and it has a positive weight in the objective function, thus strict inequality holds in (7) only if the resource usage is below the internal capacity, in which case $y_{kt} = 0$. Finally, the remaining constraints specify the feasible domains of the variables.

2.2 Solution Approaches

There are various exact and heuristic methods for solving the problem (1)–(10). As for the exact methods, we sketch the main ideas of one based on branch-and-cut, and another based on branch-and-price. Both methods use a linear programming formulation of the problem along with restricting a subset of variables to take integral values only. In the former one, the LP is a natural and more or less straightforward formulation of the problem. In the latter one, the original problem is reformulated by encoding parts of the feasible solutions as columns of a huge linear program. Both methods guarantee to find an optimal solution for the problem at the expense of possibly large computation times. In contrast, heuristics work much faster than exact methods, but they do not guarantee optimal solutions. They may use some LP formulation, but they could also be based on proprietary data structures and techniques.

Solution by Branch-and-Cut This approach is an extension of branch-and-bound in which the linear programming relaxation of a MIP is solved and strengthened by inequalities valid for the convex hull of integer solutions, but violated by the (fractional) solution of the LP relaxation [22]. The new inequalities are added to the problem in the root node, and also in search tree nodes. The inequalities are generated by so-called separation procedures, which may be exact or heuristic, and they are designed to find inequalities in a class that are violated by the optimal LP solution of the search tree node.

A special case of the problem in which there can be no overlap between pairs of activities connected by a precedence constraint is discussed in [15]. In that paper, a polyhedral approach is pursued, and an exact branch-and-cut type method is proposed. The crux of the method is a polyhedral characterization of the convex hull of points satisfying (3)–(6), along with (8)–(10). The polyhedral characterization consists of providing the inequalities giving the convex hull of points with *integer* z coordinates satisfying the constraints (3)–(6). The inequalities are used in a branch-and-cut method in which the formulation is strengthened by the new family of inequalities, and they proved very effective in solving the problem with non-overlapping activities to optimality. These results are generalized to overlapping activities in [16]. The computational results for the latter problem show that if we allow more overlap between activities, but we do not change other problem parameters, then the resulting instance is easier to solve. This is plausible, since overlapping of activities connected by a precedence constraint can be seen as a relaxation of the problem without any overlaps between activities connected by precedences.

Solution by Branch-and-Price This approach is suitable for solving a reformulation in which parts of the feasible solutions are encoded as columns of a huge LP. Hans [14] proposed a problem reformulation in which the columns represented the supports of feasible activity assignments of the activities. The possible executions of project activities are modeled by a set of binary vectors $\{\beta^h \in \{0, 1\}^{|\mathcal{A}| \times T} \mid h \in \Pi\}$,

Π being a suitable set of indices, consisting of the supports of all feasible intensity assignments to the activities. Notice that a binary vector $\beta \in \{0, 1\}^{|\mathcal{N}| \times T}$ is the support of a feasible intensity assignment if and only if $\sum_{t=1}^T \beta_{i,t} \geq p^{i,1}$, $\min\{t \mid \beta_{i,t} = 1\} \geq r^i$, $\max\{t \mid \beta_{i,t} = 1\} \leq d^i$, and if (i, j) is a pair of activities connected by a precedence relation, then $\max\{t \mid \beta_{i,t} = 1\} < \min\{t \mid \beta_{j,t} = 1\}$. For solving the problem, precisely one vector β^h must be chosen. To this end, Hans introduced new binary variables z_h , $h \in \Pi$, together with the following constraints:

$$\begin{aligned} \sum_{h \in \Pi} z_h &= 1, \\ z_h &\in \{0, 1\}, \quad h \in \Pi, \\ 0 \leq x_t^i &\leq a^i \left(\sum_{h \in \Pi} \beta_{i,t}^h z_h \right), \quad i \in \mathcal{N}, \quad t \in \{r^i, \dots, T\}. \end{aligned}$$

The first two constraints ensure that exactly one vector β^h is chosen. The third one specifies that x_t^i is either 0, or is between 0 and a^i , depending on whether $\beta_{i,t}^h$ is 0 or 1. Hans' model incorporates resource constraints similar to ours, although instead of (9) it has $y_t^k \geq 0$, and $\sum_k y_{kt} \leq \bar{b}_k$, for all t .

Since there are millions of combinations in Π , it is not convenient to store all the columns in memory. For solving LPs with millions of variables, branch-and-price is the method of choice. In such a method, there is an initial linear program containing only a fraction of the columns of the entire linear program, just enough to have a feasible solution. Then, new columns are inserted gradually using the standard pricing technique of the primal simplex method. The crux of the method is the subroutine for solving the pricing problem efficiently, i.e., given the values of the dual variables associated with the rows of the restricted primal program, one has to find a new column with negative reduced cost (in case of minimization type of problems), or verify that no such column exists in the full linear program, in which case the current LP basis is optimal. When embedded in a Branch-and-Bound method with appropriate branching rules, we get Branch-and-Price, see Barnhart et al. [3].

Hans proposed various algorithms for solving the pricing problem, and to branch on the right variables. However, the computational results are inferior to those with Branch-and-Cut, see [15] for a comparison.

Heuristics Heuristical methods usually provide a feasible solution fast, but there is no guarantee for optimality, or even to get solutions close to the optimum. Gademann and Schutten [12] divide the heuristics for our problem into three categories: (1) constructive heuristics, (2) heuristics that start with infeasible solutions and convert these to feasible ones, and (3) heuristics that improve feasible solutions.

De Boer and Schutten [8] propose algorithms in the first two categories, and Gademann and Schutten [12] present algorithms that fall in the second and third class. Wullink [24] propose new heuristics and provide a very detailed comparison of the various exact and heuristical methods.

2.3 Extensions and Further Developments

The model discussed above can be extended in various ways. For instance, in some applications non-renewable resources, like raw materials, or money are to be taken into account when making feasible project plans. Let NR be the set of non-renewable resources. Each non-renewable resource $k \in NR$ has an initial supply (or stock level) of s_{k0} , and there are further supplies arriving at known moments of time, i.e., for each non-renewable resource $k \in NR$, we have a sequence of u_k supplies with supplied quantities $s_{kh} > 0$ in time points $t_{kh} \in \{1, \dots, T\}$, $h = 1, \dots, u_k$. The consumption of activity i from some $k \in NR$ until the end of time period t can be computed as $q_k^i \sum_{\tau=0}^t x_\tau^i$.

The following set of constraints can be added to the model (1)–(10):

$$\sum_{i \in N} \sum_{t=1}^{t_{k,(\ell+1)}-1} q_k^i x_t^i \leq s_{k0} + \sum_{h=1}^{\ell} s_{kh}, \quad \ell = 1, \dots, u_k, k \in NR, \quad (11)$$

where we define $t_{k,u_{k+1}} := T + 1$. This constraint ensures that the total amount of resource $k \in NR$ that is used until the $(\ell + 1)$ th supply event is not more than the total supply over the first ℓ supply events in addition to the initial stock.

Another possible direction is to consider further variants of the precedence constraints. In Alfieri et al. [1] and Bianco and Caramia [4] the following four types of constraints are considered:

- (a) %Completed-to-Start (CtS) precedence: successor activity j can start its processing only when, in time period t , the fraction of predecessor activity i that has been processed is greater than or equal to f_{ij} (Fig. 3a).

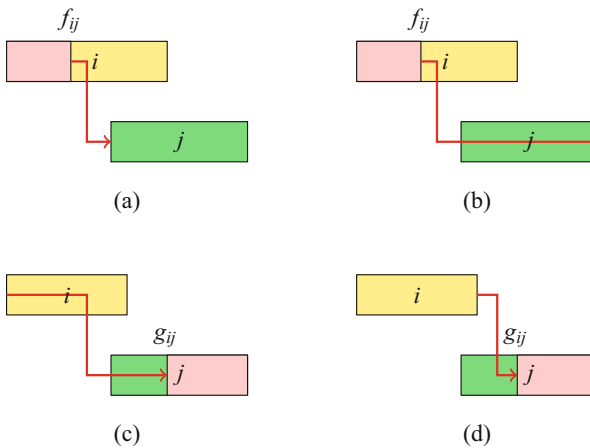


Fig. 3 Illustration of precedence relations: (a) CtS, (b) CtF, (c) StC, (d) FtC

- (b) %Completed-to-Finish (CtF) precedence: successor activity j can be completed only when, in time period t , the fraction of predecessor activity i that has been processed is greater than or equal to f_{ij} (Fig. 3b).
- (c) Start-to-%Completed (StC) precedence: the fraction execution of successor activity j , in time period t , can be greater than g_{ij} only if the execution of predecessor activity i has already started (Fig. 3c).
- (d) Finish-to-%Completed (FtC) precedence: the fraction execution of successor activity j , in time period t , can be greater than g_{ij} only if the execution of predecessor activity i has been completed (Fig. 3d).

The first one in the list is the same as that defined in Sect. 2.1, while the other three are new. Alfieri et al. provide two problem formulations; in the first one binary variables are used to mark the start and finish of activities over the time horizon, whereas the second one is much like ours, where binary variables are used as execution masks as in Sect. 2.1. A detailed computational evaluation shows the superiority of the second model in terms of solution time. Bianco and Caramia [4] in turn develop a new Lagrangian relaxation based lower bound for the makespan minimization problem with feeding precedence constraints, where the resource usage is bounded by a constant.

When preemption of activities is not allowed, but a flexible resource usage per activity is desirable, the models discussed above need to be extended by additional constraints to ensure that once an activity started, its intensity does not become zero until it is completely finished. Such formulations are proposed and thoroughly evaluated in Naber and Kolisch [18]. One of their main findings is that the modeling of precedence constraints by the system (3)–(5) is a key ingredient of a strong formulation.

3 Resource Leveling

Resource leveling problems aim at finding schedules in which the resource usage is *leveled*, or *smooth* over time. Such problems are well studied in project scheduling (see e.g., [2, 9, 19, 21]), but there are only a few results for machine scheduling problems, see e.g., Rager et al [20]. Notice that in machine scheduling, machines are unary resources that can process one job at a time, while in the more general project scheduling problems each resource can process multiple activities at the same time. Moreover, in project scheduling activities are usually connected by precedence constraints, while in machine scheduling problems this is not always the case [5, 7]. In this section we will study resource leveling problems in a machine environment, where each job is dedicated to a single machine, and may require one or more additional resources whose usage must be leveled.

Consider a scenario where the tasks are already assigned to machines, and the time windows where individual tasks can be processed are already known (e.g. based on precedence constraints, due dates, etc.). Each task may require a

given amount of some resources (such as skilled workers, some tools, etc.). For each resource, the available amount is known. Most companies are willing to rent temporary resources (e.g. hiring temporary workers) in order to complete their orders on time, however they want to minimize the extra cost (recall that this was also the motivation for the objective function in Sect. 2). Another related application is the classical resource leveling problem: the company wants to minimize the variation of resource usage over time.

Beside the above applications, resource leveling problems occur in scheduling problems where a balanced use of energy is one of the main objectives [20], in construction engineering [11], and in production planning [2, 13].

In this section, based on [10], an efficient solution approach is presented for a resource leveling problem in a machine environment as described above. There are m parallel machines, and n_i tasks are assigned to machine i . Preemption of tasks is not allowed, and no machine can process more than one task at a time. Task j has a time window $[e_j, d_j]$ in which it has to be processed for p_j time units. Furthermore there are L types of renewable resources, each resource ℓ having a target level C_ℓ . Task j requires an amount of $b_{j\ell}$ from resource ℓ . An illustrative example is shown in Fig. 4.

The goal is to minimize some function of the deviation of the resource usage from prespecified values. We consider objective functions of the following form:

$$\sum_{\ell=1}^L \sum_{t=0}^D f_\ell(y_{\ell t}, C_\ell),$$

where $y_{\ell t}$ is the total usage of resource ℓ at time t , D is the time horizon, and $f_\ell : \mathbb{Q}_+ \times \mathbb{Q}_+ \rightarrow \mathbb{Q}_+$ satisfy $f_\ell(x, y - z) = f_\ell(x + z, y)$. Note that f_ℓ may be different for different types of resources.

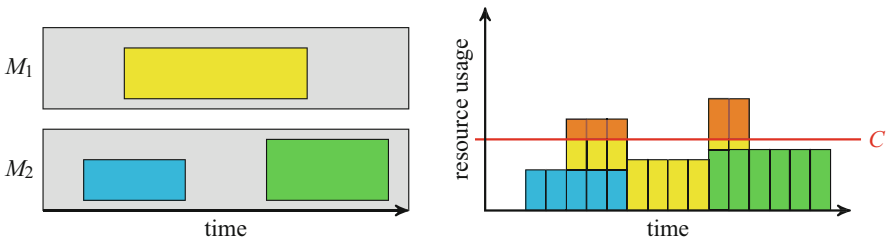


Fig. 4 Example with two machines, three tasks, and one resource. The height of the tasks is proportional to their resource requirement. The chart on the *right hand side* shows the resource profile of the schedule for each time unit; the resource overuse is marked above the target level C of the resource

3.1 Modeling by a Mixed Integer Linear Program

The following notations will be used to describe the problem and a solution approach:

m = number of machines

L = number of resource types

C_ℓ = target level of resource ℓ

J = set of tasks

J_i = set of tasks pre-assigned to machine M_i

$n_i = |J_i|$, the number of tasks pre-assigned to machine i

p_j = processing time of task j

e_j = release time of task j

d_j = deadline of task j

$b_{j\ell}$ = amount of resource ℓ required by task j

D = time horizon

The optimization problem can be formulated as a mixed integer linear program:

$$OPT = \min \sum_{\ell=1}^L \sum_{t=0}^D f_\ell(y_{\ell t}, C_\ell) \quad (12)$$

subject to

$$\sum_{t=0}^D x_{jt} = 1, \quad \forall j \in J, \quad (13)$$

$$\sum_{j \in J_i} \sum_{\tau=t-p_j+1}^t x_{j\tau} \leq 1, \quad \forall t \in \{0, \dots, D\}, i \in \{1, \dots, m\} \quad (14)$$

$$\sum_{j \in J} \sum_{\tau=t-p_j+1}^t b_{j\ell} x_{j\tau} - y_{\ell t} = 0, \quad \forall t \in \{0, \dots, D\}, \ell \in \{1, \dots, L\} \quad (15)$$

$$x_{jt} \in \{0, 1\}, \quad \forall j \in J, t \in \{e_j, \dots, d_j - p_j\}. \quad (16)$$

The decision variables are $x_{jt} \in \{0, 1\}, j \in J, t \in \{e_j, \dots, d_j - p_j\}$, indicating the start times of the tasks, and $y_{\ell t} \in \mathbb{Q}_+, j \in J, t \in \{0, \dots, D\}$ representing the resource usage in each time period from each resource. As each task must be processed in its

time window, if $\tau < e_j$ or $\tau > d_j - p_j$, then $x_{j\tau}$ is not defined and the corresponding term is omitted.

The set of equations (13) ensures that every task is started in precisely one time point $t \in [e_j, d_j - p_j] \cap \mathbb{Z}$. The set of constraints (14) prescribes that no two tasks on the same machine may overlap. Finally, the resource usage is computed in (15).

3.2 Calculation of Lower Bound by Lagrangian Relaxation

In order to compute lower bounds for an optimization problem, a standard technique is to apply Lagrangian duality (see, e.g., [22]), where some nasty constraints of a problem formulation are moved to the objective function so that the resulting problem becomes a *relaxation* of the original, and at the same time is easier to solve.

By dualizing the constraints (15) we obtain the following Lagrangian relaxation of the problem:

$$LB(\boldsymbol{\lambda}) = \sum_{i=1}^m LB_i(\boldsymbol{\lambda}) + \min_y \sum_{\ell=1}^L \sum_{t=0}^D (f_\ell(y_{\ell t}, C_\ell) - \lambda_{\ell t} y_{\ell t}), \quad (17)$$

where

$$LB_i(\boldsymbol{\lambda}) = \min \sum_{\ell=1}^L \sum_{t=0}^D \sum_{j \in J_i} \sum_{\tau=t-p_j+1}^t \lambda_{\ell t} b_{j\ell} x_{j\tau} \quad (18)$$

subject to

$$\sum_{t \in \{e_j, \dots, d_j - p_j\}} x_{jt} = 1, \quad \forall j \in J_i, \quad (19)$$

$$\sum_{j \in J_i} \sum_{\tau=t-p_j+1}^t x_{j\tau} \leq 1, \quad \forall t \in \{0, \dots, D\} \quad (20)$$

$$x_{jt} \in \{0, 1\}, \quad \forall j \in J_i, t \in \{e_j, \dots, d_j - p_j\}. \quad (21)$$

From the theory of Lagrangian duality it is known that

$$\max_{\boldsymbol{\lambda}} LB(\boldsymbol{\lambda}) \leq OPT$$

where OPT is the optimum value of (12)–(16).

In [10] it is shown that the subproblems (18)–(21) can be solved efficiently for $f(x, y) := \max\{x - y, 0\}$, and $f(x, y) := (x - y)^2$.

Note that by using this relaxation, the original problem is decomposed into independent single machine problems that can be solved concurrently. Another

advantage of this approach is that the structure and the size of the subproblems (identified by (18)–(21)) are independent of the number of resources and the objective function.

3.3 A Branch&Bound Method

The nodes in the Branch&Bound search tree represent a constrained version of the original problem, where the time windows of the tasks are narrowed, and the root node represents the original problem. In each node, the following calculations are performed:

1. **Constraint propagation.** Some well known single machine constraint propagation methods are applied on each machine in order to narrow the time windows of the tasks.
2. **Calculation of lower bound.** By using the subgradient method, the Lagrangian multipliers λ are determined for the actual subproblem, and a lower bound is calculated for the actual node. The formulation presented in Sect. 3.2 is used, however instead of solving the IP-s, their LP-relaxations are considered.
3. **Shaving.** Concurrently for each machine, a shaving procedure is applied. For each task, the lower bound of the objective function is calculated for each possible start time, again using the Lagrangian relaxation. This method may improve the lower bound, and may also narrow the time windows. For an overview of shaving techniques, the reader is referred to [5].
4. **Calculation of upper bound.** By using the solution of the Lagrangian relaxation and applying some heuristics, a solution is sought for the problem represented by the actual node of the Branch&Bound tree.
5. **Branching.** A task is chosen heuristically, and its time window is partitioned into sub-windows. By using the results of shaving, a lower bound can be determined for each child node without extra calculations.

A best-first search is used to traverse the search tree according to the predicted lower bounds of the unvisited nodes, ensuring that the promising combination of time windows are evaluated first. Furthermore, the minimal lower bound among the unvisited nodes represents a lower bound for the original problem.

3.4 Test Results

The effectiveness of the presented Branch&Bound method was demonstrated using randomly generated test instances of different sizes, for the following objective functions:

$$f_{\text{lin}}(y_{\ell t}, C_{\ell}) = w_{\ell} \max(0, y_{\ell t} - C_{\ell}) \quad (22)$$

$$f_{\text{quad}}(y_{\ell t}, C_{\ell}) = (y_{\ell t} - C_{\ell})^2 \quad (23)$$

f_{lin} represents the minimization of total weighted resource overuse, while f_{quad} represents the resource leveling problem (i.e. the minimization of the variation of the resource usage over time).

A series of test instances were used with $m = 5, 10, 20$ machines, $t = 10, 15$ and 20 tasks per machine, giving a total of $n = 10m, 15m$ and $20m$ tasks, respectively. Each test class with parameters (m, t) contained 10 instances.

The results of the Branch&Bound method were compared to those obtained by the commercial solver ILOG CPLEX 11.2 using the MIP formulation of the resource leveling problem (12)–(16). For both programs, a time limit of 1800 s was set, and the best lower- and upper bound was recorded at the end of each run.

The average optimality gap (defined as $UB/LB - 1$, expressed in percents) in each case is shown in Tables 1 and 2 for the linear and quadratic objective function, respectively. For the test instances with quadratic objective function, CPLEX was only able to compute lower or upper bounds for the smallest instances within the time limit.

3.5 Computation with Multiple CPUs

Our Branch&Bound procedure offers several opportunities for parallel computing. We have investigated the processing of search-tree nodes by multiple CPU cores on a shared-memory computer. Our goal with the tests has been to measure the speed-up that could be gained by parallel processing, using the natural decomposition of the

Table 1 Average optimality gap for the linear objective function with $C_\ell = \lfloor \sum_{j \in J} b_{\ell j} p_j / D \rfloor$, and three resources

	m5		m10		m20		Avg	
	BB	CPX	BB	CPX	BB	CPX	BB	CPX
t10	6.16%	3.10%	0.74%	0.24%	0.36%	0.37%	2.42%	1.24%
t15	12.94%	11.28%	5.08%	5.96%	0.41%	1.62%	6.14%	6.29%
t20	18.39%	17.15%	5.41%	7.48%	2.19%	10.39%	8.66%	11.67%
Avg	12.49%	10.51%	3.74%	4.56%	0.99%	4.13%	5.74%	6.40%

Table 2 Average optimality gap for the quadratic objective function with $C_\ell = 0$, and three resources

	m5		m10		m20		Avg	
	BB	CPX	BB	CPX	BB	CPX	BB	CPX
t10	2.31%	1.51%	0.85%	–	0.24%	–	1.13%	–
t15	3.77%	–	1.20%	–	0.60%	–	1.86%	–
t20	4.31%	–	2.71%	–	0.37%	–	2.46%	–
Avg	3.46%	–	1.59%	–	0.40%	–	1.82%	–

Table 3 Effects of using multiple CPU cores

(a) Average ratio of CPU time and wall clock time with different number of CPU cores					(b) Average speed of the algorithm relative to execution with a single CPU				
		m10	m20	Avg			m10	m20	Avg
t10	CPU5	2.65	2.58	2.62	t10	CPU5	1.88	2.01	1.95
	CPU10	3.18	3.24	3.21		CPU10	2.14	2.42	2.28
t15	CPU5	3.09	3.17	3.13	t15	CPU5	2.77	2.82	2.79
	CPU10	4.27	4.37	4.32		CPU10	3.81	3.79	3.8
t20	CPU5	3.29	3.33	3.31	t20	CPU5	3.08	2.91	3
	CPU10	4.81	4.92	4.87		CPU10	4.41	4.2	4.31
Avg	CPU5	3.01	3.03	3.02	Avg	CPU5	2.58	2.58	2.58
	CPU10	4.09	4.18	4.13		CPU10	3.45	3.47	3.46

problem as described in Sect. 3.2. Note that other approaches of parallel processing would also be possible, such as e.g. evaluating the search-tree nodes parallelly.

As the type of the problem and the actual test environment (server load, tasks with high priority, etc) may influence the results, the parallel execution was evaluated with two different methods. The first is the ratio of the elapsed CPU time and the wall clock time (see Table 3a). Recall that the notation t10, t15, t20 means that there are 10, 15, and 20, respectively, tasks to be scheduled on a *single* machine, so, in the cell, say, t20-m20, we provide speed-up for instances with 20×20 tasks (20 tasks per machine). We use the notation CPU n to indicate that a computation is using n CPU cores.

An ideally parallel execution would use $n \cdot t$ CPU seconds with n CPU cores during t seconds wall clock time in an ideal environment. However the wall clock time is still passing even when some CPUs are waiting for synchronizing with the others, and therefore the ratio of total CPU time vs wall clock time is usually worse (smaller) than n .

The other method is to calculate the average number of nodes evaluated in a second, which can be considered the speed of the algorithm. For each instance the speed of the multi-core test runs was calculated relative to the single-CPU one (see Table 3b). This measure may be less accurate than the previous one because the nodes of the branch-and-bound tree may require different amount of calculation. This is the consequence of the fact that the nodes of the search-tree may represent problems with different complexity.

4 Conclusions

In this chapter we have described a planning and a scheduling problem which have recently gained more and more attention in the academic research, but which frequently occur in practice and need proper solution techniques so that they could

be routinely solved by future generation manufacturing planning and scheduling softwares.

We have described some techniques to optimally solve those problems, but the methods mentioned could be turned into heuristics by standard techniques, like truncated branch-and-bound, or beam search.

We believe that variants of these problems do occur in several real-world applications, and a deeper understanding and further work is needed in order to solve them properly in industrial practice.

Acknowledgements The authors are grateful to the editors, and to a referee for comments that helped improve the paper. This work has been supported by the OTKA grant K112881, and by the grant GINOP-2.3.2-15-2016-00002 of the Ministry of National Economy of Hungary. The research of Tamás Kis has been supported by the János Bolyai research grant BO/00412/12/3 of the Hungarian Academy of Sciences.

References

1. Alfieri, A., Tolio, T., Urgo, M.: A project scheduling approach to production planning with feeding precedence relations. *Int. J. Prod. Res.* **49**, 995–1020 (2011)
2. Ballestín, F., Schwindt, C., Zimmermann, J.: Resource leveling in make-to-order production: modeling and heuristic solution method. *Int. J. Oper. Res.* **4**, 50–62 (2007)
3. Barnhart, C., Johnson, E.E., Nemhauser, G.E., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: column generation for solving huge integer programs. *Oper. Res.* **46**, 316–329 (1998)
4. Bianco, L., Caramia, M.: Minimizing the completion time of a project under resource constraints and feeding precedence relations: a Lagrangian relaxation based lower bound. *4OR-Q. J. Oper. Res.* **9**, 371–389 (2011)
5. Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Weglarz, J.: *Handbook on Scheduling. From Theory to Applications*. Springer, Berlin (2007)
6. Bracht, U., Masurat, T.: The digital factory between vision and reality. *Comput. Ind.* **56** 325–333 (2005)
7. Brucker, P.: *Scheduling Algorithms*, 5th edn. Springer, Berlin (2007)
8. de Boer, R., Schutten, J.M.J.: Multi-project rough-cut capacity planning. In: Ashayeri, J., Sullivan, W.G., Munir Ahmad, M. (eds.) *Flexible Automation and Intelligent Manufacturing*, pp. 631–644, Begell House, New York (1999)
9. Demeulemeester, E.L., Herroelen, W.S.: *Project Scheduling: A Research Handbook*. International Series in Operations Research & Management Science, vol. 49 (Kluwer Academic, Dordrecht, 2002)
10. Drótos, M., Kis, T.: Resource leveling in a machine environment. *Eur. J. Oper. Res.* **212**, 12–21 (2011)
11. Easa, S.M.: Resource leveling in construction by optimization. *J. Constr. Eng. Manag.* **115**, 302–316 (1998)
12. Gademann, N., Schutten, M.: Linear-programming-based heuristics for project capacity planning. *IIE Trans.* **37**, 153–165 (2005)
13. Gahm, C., Dünwald, B., Sahamie, R.: A multi-criteria master production scheduling approach for special purpose machinery. *Int. J. Prod. Econ.* **149**, 89–101 (2014)
14. Hans, E.W.: *Resource loading by branch-and-price techniques*. Ph.D. Thesis, Twente University Press (2001)
15. Kis, T.: A branch-and-cut algorithm for scheduling of projects with variable intensity activities. *Math. Program.* **103**, 515–539 (2005)

16. Kis, T.: RCPS with variable intensity activities and feeding precedence constraints, In: Józefowska, J., Weglarz, J. (eds.) *Perspectives in Modern Project Scheduling*, pp. 105–129. Springer, New York (2006)
17. Monostori, L.: Cyber-physical production systems: roots, expectations and R&D challenges. *Proc. CIRP* **17** 9–13 (2014)
18. Naber, A., Kolisch, R.: MIP models for resource-constrained project scheduling with flexible resource profiles. *Eur. J. Oper. Res.* **239**, 335–348 (2014)
19. Neumann, K., Zimmermann, J.: Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *Eur. J. Oper. Res.* **127**, 425–443 (2000)
20. Rager, M., Gahm, C., Denz, F.: Energy-oriented scheduling based on evolutionary algorithms. *Comput. Oper. Res.* **54**, 218–231 (2015)
21. Ranjbar, M.: A path-relinking metaheuristic for the resource levelling problem. *J. Oper. Res. Soc.* **64**, 1071–1078 (2013)
22. Wolsey, L.A.: *Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, New York (1998)
23. Wörn, H., Frey, D., Keitel, J.: Digital factory – planning and running enterprises of the future. In: *Proceedings of IECON 2000, 26th Annual symposium of the IEEE, Nagoya, 22 Oct 2000–28 Oct 2000*, vol 2, pp. 1286–1291
24. Wullink, G.: *Resource loading under uncertainty*. PhD Thesis, Twente University Press (2005)

Modeling of Material Flow Problems

Simone Göttlich, Michael Herty, and Melanie Luckert

Abstract In this article we discuss the description of modern manufacturing or production problems using continuous models. Instead of a detailed description of the production process, a mathematical formulation is used based on transport equations. The challenge is to derive novel and nonstandard approaches that allow to incorporate detailed nonlinear dynamic behavior, which is currently not possible with the widely applied linear or mixed integer linear approaches. Starting from discrete event simulations as a basic description we explore the relation between the product density and the flow of parts (also known as clearing function). Data-fitting procedures help to identify the underlying parameters. We show the relationships between discrete event simulations, queuing models and transport model-based methods, and present several applications.

1 Introduction and Literature Overview

Manufacturing systems are studied in the literature on either a discrete level (using time recursions) or on a macroscopic level (using a continuum description based on differential equations for transport processes). In recent years, continuous or fluid-like models have been particularly introduced to model high-volume production [3–5, 10–13, 17, 21]. Those dynamics are often inspired by discrete event simulations (DES), see [9]. In the current work we aim on bridging the discrete and continuous level by presenting a suitable hierarchy of models with reasonable transitions.

S. Göttlich (✉)

Department of Mathematics, University of Mannheim, Mannheim, Germany

e-mail: goettlich@uni-mannheim.de

M. Herty

Department of Mathematics, RWTH Aachen University, Aachen, Germany

e-mail: herty@igpm.rwth-aachen.de

M. Luckert

Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany

e-mail: M.Luckert@wzl.rwth-aachen.de

An approach is proposed similar to gas dynamics. In physics, discrete events and discrete parts are considered as fundamental units used to describe microscopic phenomena. Those time-dependent individual dynamics are typically governed by ordinary differential equations. They provide an accurate description of the underlying process. At the same time, the system as a whole shows pattern formation such as jams in traffic flow, flocking in swarming behavior or shock waves in aerodynamics. A similar approach for production processes is considered. Here, the detailed dynamic is the description of the production process of individual parts. However, often there is only interest in the global phenomena of the system like overloads, queuing or mean production rates. We may argue that there are different scales also present in production. Therefore, a similar approach as in gas dynamics is suitable in order to understand the pattern formation in production.

The individual dynamics are described by a discrete event simulation (DES) in production processes. DES is a stochastic simulation tool for individual parts. The corresponding continuous equations are fluid-like models. On a different scale the latter describe production flow in an aggregate way leading to coarse-grained models. Due to the reduced dimension they are expected to be computationally efficient. Typically, there is only one conserved quantity in production being the total number of produced items. Therefore, the proposed continuous models are conservation laws for the product density $\rho(x, t)$ at production stage $x \in [0, 1]$ and time $t \geq 0$. Here, the flux function f is usually called clearing function. Starting with Graves [19] and Karmarkar [23] monotone, concave clearing functions have been proposed. They are now used in production engineering, see for example [7, 8, 25]. Other approaches to derive clearing functions are mean field limit considerations [3, 5], comparisons with observed behavior [6] or queuing theory under steady-state assumptions. Examples for all these possible options can be found in Sect. 2 (Fig. 1).

In the case of a single unlimited buffer, Poisson processes for the arrival of products and a Poisson process for the production time lead to $f = \frac{\mu W}{1+W}$, where $W = \int_0^1 \rho(x, t) dx$ is the (total) Work in Progress (WIP), see for example [20], and μ is the maximal production rate. In queuing theory this is known as an M/M/1 queue, see also the discussion in Sect. 2. We may use ρ and W equivalently whenever ρ is constant in x due to a production stage of at most $x = 1$. In [25] a clearing function for an M/G/1 queue (here service times obey general distributions) is proposed including parameters that may adjusted to given data. The resulting clearing function is again a steady-state consideration and in general for models based on product flow no transient clearing function model has been derived

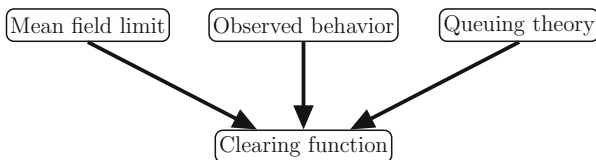


Fig. 1 Different ways to derive a clearing function f

yet [26]. It has been observed that in different production periods different clearing functions may be suitable. We refer to [2, 13, 24] for an overview.

We also propose different approaches using (real) data to establish the clearing function, i.e., the fundamental relation between f and ρ . We present new continuous models based on realistic data in order to predict production behavior. As already mentioned non-stationary queuing theory predicts that there is no fixed functional relationship between product density and flux [26]. Therefore we are concerned with the detailed data-fitted modeling of the flux function f and its application to conservation laws of the form

$$\partial_t \rho(x, t) + \partial_x f(\rho(x, t)) = 0. \quad (1)$$

The model (1) is based on the assumption that the amount of products and the number of production stages justifies a continuous model. A prototype of a production process consists of a machine with associate buffer and no limit on the storage capacity. As we will see in Sect. 2 there are several ways to establish the fundamental relation between ρ and f . Such a relation is required to obtain a closed model by Eq. (1).

2 Data-Driven Differential Equations for Production

2.1 Mean Field Limit Approach

The following model was originally introduced by Armbruster et al. [5] in 2006. It was the starting point for the description of a high-volume multi-stage production line by partial differential equations. Detailed explanations and reasonable extensions regarding this model can be found in [13, 16, 17]. The key modeling idea was and is still today to use a discrete description, a so-called discrete event simulation (DES), for the small scale effects and a continuous model to describe large scale phenomena. It can be really shown that both approaches lead to the same results in case of mass production. In the following we present the main ingredients of these models since they are the basic framework for all further considerations.

Discrete event simulations models (DES) provide a powerful tool for an accurate description of the underlying production process. The main idea of these models is to track parts through the whole production process so that information on all part arrival times is fully available. These times obey internal production and order policies but can be given in the case of a *first come-first serve* policy in a straightforward manner.

In the sequel, we assume that the amount of parts is conserved, i.e. no parts are lost or gained during the production process. The parts have to undergo different production steps where there is the possibility to store parts inbetween. For the first consideration the inventories or buffers have infinite size. Parameters defined by production are the processing velocity and a maximal capacity for each entity.

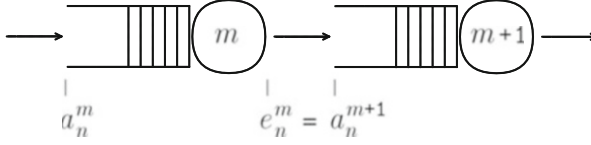


Fig. 2 A serial production line

To derive a discrete model we consider the particular situation of a serial production line consisting of M_P production units. The output of one unit is directly fed into the next one, i.e. machine m ships all parts to the next machine $m + 1$ as in Fig. 2.

Every machine is characterized by the processing time $T(m)$ and the maximal capacity $\mu(m)$ measured in parts per unit time. The processing time $T(m)$ is the time which is needed to finish a single production step. In this first attempt, the production line should be reliable, i.e. sudden shut-downs of machines are ignored for the moment. However, since machines have possibly different capacities, it may happen that parts have to wait until the next operations can be performed. Therefore, inventories or buffers are installed between production units.

The evolution of parts through the system is now determined by the computation of arrival times. We define the arrival time of part n at the buffer of machine m as a_n^m . The total amount of parts in the system is denoted by N_P . After successful production, the leaving time e_n^m denotes when part n leaves machines m and arrives at machine $m + 1$, see Fig. 2.

The computation of arrival times a_n^m obviously depends on the current buffer load, i.e. either the buffer is empty or filled. If the buffer is empty, part n is immediately passed into production. Once the part is released for production, the leaving time e_n^m can be determined by adding the processing time $T(m)$. In the other case the part has to wait. If N parts arrive at the same time t at the machine having an empty buffer, the model (2) yields the departure time of the i th part as $T(m) + (i - 1)/\mu(m)$, $i = 1, \dots, N$. Hence, within a unit length of time the machine produces $\mu(m)$ parts. Therefore, a buffer will be build up if the inflow per unit time exceeds $\mu(m)$. This buffer may grow to infinity if the inflow to system exceeds $\mu(m)$ for all times.

We end up with a time recursion for the computation of all arrival times:

$$e_n^m = \max \left\{ a_n^m + T(m), e_{n-1}^m + \frac{1}{\mu(m)} \right\} \quad m = 0, 1, \dots, \quad n \geq 1. \quad (2)$$

As evaluation measures for (2) we use curves of cumulative counts, so-called Newell-curves, as already successfully applied in traffic engineering, see [27]. The idea of Newell-curves $U(m, t)$ is to count all parts that have been arrived at machine m up to any fixed time t :

$$U(m, t) = \sum_{n=0}^N H(t - e_n^m), \quad m = 0, \dots, M_P, \quad t > 0, \quad (3)$$

where $H(\cdot)$ is the Heaviside function

$$H(t - e_n^m) = \begin{cases} 0, & \text{if } t < e_n^m \\ 1, & \text{if } t \geq e_n^m \end{cases}.$$

Hence, the Newell-curve $U(m, t)$ provides the total number of parts passing from machine $m - 1$ to machine m up to time t . The difference of two consecutive Newell-curves is the number of parts actually processed in unit m including the parts in the buffer as well. This difference is known as Work In Progress (WIP) and is denoted by $W(m, t)$:

$$W(m, t) = U(m, t) - U(m + 1, t), \quad m = 0, \dots, M_P - 1. \quad (4)$$

Although DES models reflect the most accurate way of modeling a time-varying production process, the computational complexity highly depends on the number of parts being considered. An alternative simulation approach might be continuous equations. These kind of equations arise whenever the relationship between changing quantities (modeled by functions) and their rates of change (expressed as derivatives) is known. For the special scenario depicted above, a continuous model can be directly derived from the DES, see [5] for a detailed proof. The idea is to investigate the continuum limit ($M_P, N_P \rightarrow \infty$) and to analyze in which sense an approximate density and flux satisfy a conservation law for the part density.

The continuous model describes the evolution of the part density $\rho(x, t)$ at x in time t . The space variable x can be interpreted as the degree of completion. For instance, $x \in [0, 1]$ does not represent a physical position but rather the degree of completion or stage of production. The manufacturing system has a prescribed inflow $\lambda(t)$ over time t at $x = 0$ and an outflow at $x = 1$ of finished products. The density $\rho(x, t)$ is transported with velocity $v(x)$ if the flow of parts is less than the maximal capacity $\mu(x)$, i.e., ρ satisfies the transport equation or mass conservation law

$$\partial_t \rho(x, t) + \partial_x f(\rho(x, t)) = 0, \quad \rho(x, 0) = \rho_0(x), \quad (5)$$

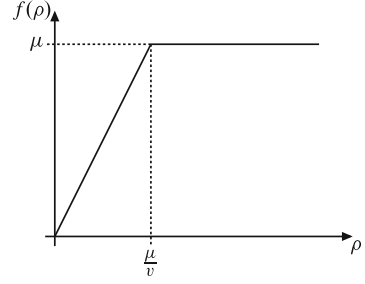
where the relation between flux and density is given by

$$f(\rho(x, t)) = \min\{v(x)\rho(x, t), \mu(x)\}, \quad (6)$$

and $\rho_0(x)$ describes the initial state of the line, see also Fig. 3. This relation is also known as clearing function in the production literature.

Equation (5) is the continuous analogue to Eq. (4) and hence the Work In Progress (WIP) the discrete representation of the part density $\rho(x, t)$. The main difficulty is that the flux function (6) can become discontinuous due to the assumption that processors may have different maximal capacities. For instance, if machine x_m has higher capacity than machine x_{m+1} , i.e. $\mu(x_m) > \mu(x_{m+1})$, δ -distributions occur

Fig. 3 Example of a clearing function given by Eq. (6)



in the density at his point since mass has to be conserved. Obviously, the limiting density will be a distribution and not a classical function. This corresponds to the fact having buffers in front of machines.

Finally, we present computational results for the discrete (2) and the continuous model (5). We consider a production line consisting of two processors, i.e. $M_P = 2$. The capacities and processing times of the two machines are $\mu_1 = 2, T_1 = 1$ and $\mu_2 = 1, T_2 = 1$. The discrete model (2)–(4) can be straightforward implemented using

$$\lambda(\tau(0, n)) = \frac{1}{\tau(0, n+1) - \tau(0, n)} \quad (7)$$

$$\tau(m+1, 0) = \tau(m, 0) + T(m). \quad (8)$$

as initial conditions. Here, the function λ denotes the total inflow into the system, see Fig. 4. Furthermore, we discretize the system (5) in space m and time i using an Upwind-scheme for the conservation law:

$$\begin{aligned} \rho(x_m, t_{i+1}) &= \rho(x_m, t_i) - \frac{\Delta t}{\Delta x} (f(x_{m+1}, t_i) - f(x_m, t_i)), \quad m = 0, 1, 2, \\ f(x_m, t_n) &= \begin{cases} \min\{\mu(x_m), v(x_m)\rho(x_m, t_i)\} & m = 1, 2 \\ \lambda(t_i), & m = 0. \end{cases} \end{aligned} \quad (9)$$

The time steps Δt are constant and satisfy the CFL condition $\Delta t \leq \Delta x/v$. We assume an empty line in the beginning, i.e. $\rho_{x_m,0} = 0$, and a randomly disturbed initial profile $\lambda(t)$ such that the maximum capacity of the machines is exceeded, see upper part of Fig. 4. We compute the arrival times according to (7). Both machines have length one and are divided into ten cells. We compare the WIP from the recursion (2) and the discretized conservation law (9). Figure 4 also shows the corresponding WIP of each machine in the production line. The red line is computed from the time recursion for the transition times while the blue dots are computed from the conservation law. The WIP of machine one is computed as $\int_0^1 \rho(x, t) dx$.

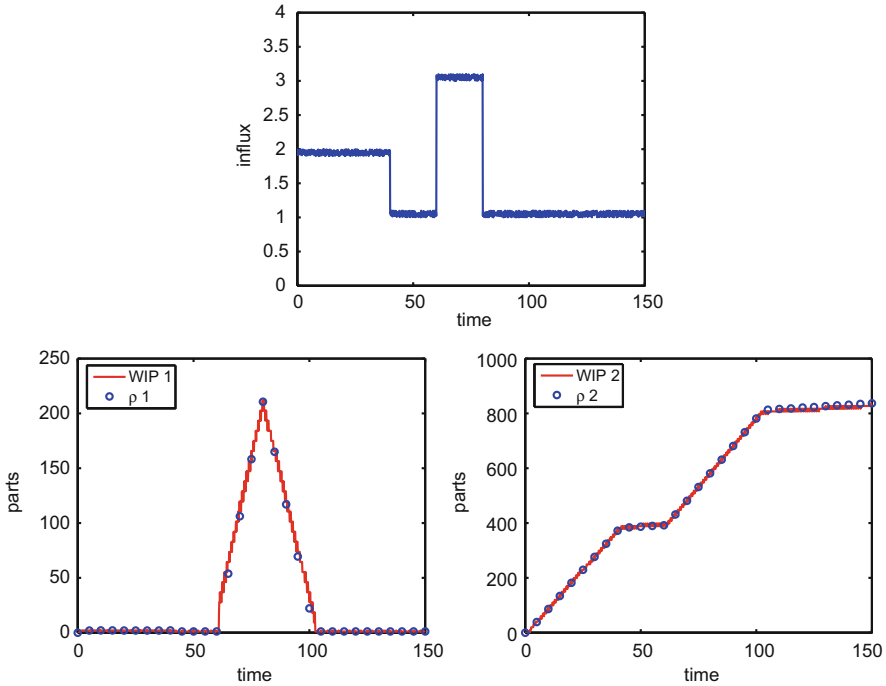


Fig. 4 Inflow profile $\lambda(t)$ prescribed as initial data (*above*) and work in progress versus part density (*below*)

2.2 Observed Behavior and Phenomenological Approach

In this section we illustrate a phenomenological approach to modeling with macroscopic equations. The goal is to derive equations based on observations of DES simulations. This approach can be employed when the detailed description of the DES equations and its setup is not available.

To exemplify, we consider experiments conducted by Gossens [15] using the χ (or Chi) language during her Master-Thesis at TU Eindhoven. The data for the DES description was collected in semiconductor production with limited storage capacity. From the DES simulation several interesting observations have been obtained. A single production line with exponentially distributed interarrival times for the inflow has been considered, cf. Fig. 2. It has been assumed that the processing rate is $\mu(x)$ but with the crucial difference that the storage capacities buffers are limited by a quantity $\rho_{\max} > 0$. The following scenario has been taken from the semiconductor factory and analyzed numerically using a DES simulator. In above case the χ Simulator developed by Beek and Rooda at TU Eindhoven has been used. The description of the experiments and simulations are summarized as follows:

1. A production line of $M_p = 100$ stations and time horizon of $T = 11,000$ is considered.
2. We start with an initially empty factory, where the arrival rate is

$$\min_{t \in [0, T]} \lambda(t) < \min_{x \in [0, M_p]} \mu(x).$$

The inflow is ramped up until a steady state formation of the part density within the factory is achieved.

3. After the system runs in steady state there is a shutdown $\mu(M_p) = 0$ of the last machine immediately leading to a bottleneck situation. Buffers of downstream machines are filled step by step since production is blocked by the unavailable last machine. Due to the finite size of the buffers the production process stops at some time t_0 .
4. At time t_1 the last machine is again operational at same capacity as before. The production starts again. The congestion is slowly moving and buffer sizes are reduced until the system approaches its steady state.

We are interested in a continuous equation having the same wave pattern as observed in the DES simulation, see Fig. 7. A suggestion has been proposed in [6] and [22]. In [6] a conservation law has been derived taking into account limited capacities of buffers and non-homogeneous steady state behavior starting from observations only. Since parts are still conserved during production a conservation law similar to (5) has been proposed. However, the design of the clearing function is more involved due to the maximum part density ρ_{\max} characterizing the buffer limits. The key difference to the previous model is that the production might be interrupted and jams may occur. The latter move backwards within the production line. In particular, the observation described in step 2 motivates therefore a non-monotone and discontinuous clearing function. We introduce a discontinuity at ρ_{\max} such that information propagates extremely fast towards the downstream machines. The final relation is given by Eq. (10)

$$f(\rho, x) = \begin{cases} \frac{\mu\rho}{1+\rho+k\rho(1-x)} & \text{for } \rho < \rho_{\max} \\ 0 & \text{for } \rho \geq \rho_{\max} \end{cases} \quad (10)$$

with $k > 0$ being the decay rate of the processing capacity along x . An example is depicted in Fig. 6.

The discontinuous clearing function involves several numerical challenges due to the high speed of wave propagation. The simplest remedy is to smooth the function (10). Unfortunately, this implies severe restrictions to the time step size Δt . An alternative is the embedding of the clearing function into a second order model [18, 22].

For the experiments we parameterize the workstations by $x \in [0, 1]$ and a maximal density of $\rho_{\max} = 1$. The production capacity is constant $\mu(x) = 2$ for

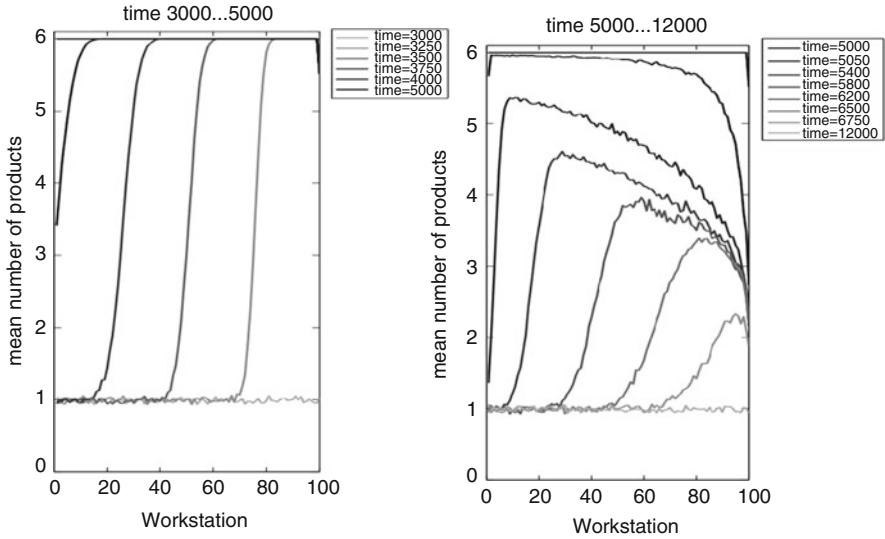


Fig. 5 Computational results for the discontinuous flux function from a χ -simulation for $\lambda(t) < \mu(x)$, cf. Fig. 3.6 in [15]. We observe the following phenomena from left to the right: shutdown of the last machine and congestion—release of production draining after the last machine has been repaired. The figure shows snapshots of averaged WIP profiles

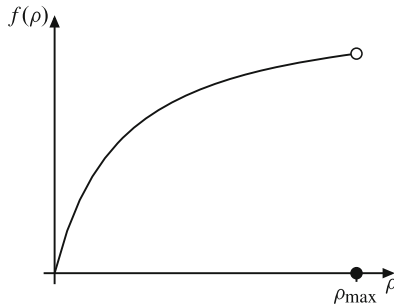


Fig. 6 Example of a clearing function given by Eq. (10)

all x and $k = 0.7$. We start with an empty factory at time $t = 0$ and a constant arrival rate $\lambda(t) < \mu(x)$ at $x = 0$.

The computed results cover the essential scenarios ramp up, blocking and release as described in Fig. 5. Apparently, the system behavior reproduced by a continuous model is obtained at lower computational costs compared to the DES. We want to stress that the model (10) is not derived in a rigorous way as done in Sect. 2.1, but is solely based on observations. It is unclear for now if a rigorous derivation is possible (Figs. 6 and 7).

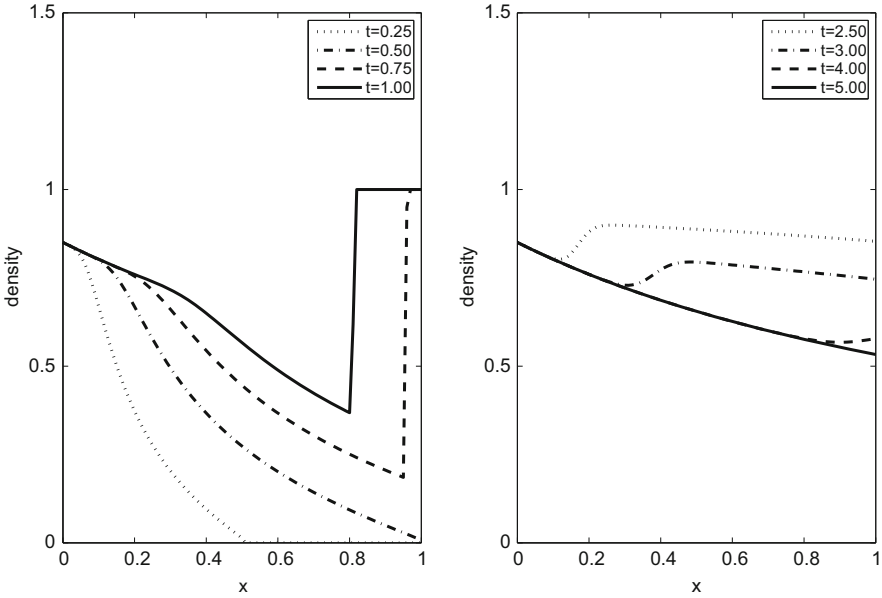


Fig. 7 Computational results for the discontinuous clearing function (10) used for a comparison with Fig. 6. The pictures are from reference [18]

2.3 Data-Fitted Simulated Clearing Functions

In this section we propose a general method to derive clearing functions based on a DES simulation using real factory data. As discussed previously non-stationary queuing theory predicts that there is no fixed functional relationship between product density and flux [26]. However, transient clearing functions have been proposed starting with the work of [1] and [28] to incorporate dynamic effects. We proceed in this spirit in order to obtain a coarse-grained model of transport type (1).

To exemplify we use data from a mid-size German manufacturing company. The available data are order and release data of the major single production step. The layout is precisely as in a theoretical queuing model, i.e., we have a buffer where parts arrive and a machine applying a manufacturing step. Available is production data for 1 year (2012). Mathematically, a probability distribution for the interarrival times is computed from the data. Further, a probability distribution for the production times is computed from the data. Here, we use as sample interval single days. The resulting probability distributions based only on the available data are depicted in Fig. 8. We observe a strong possibly exponential decay of the probability of high interarrival times. A similar observation is true for the production times. The discrete probability distribution is interpolated. This allows to have an arbitrary amount of data points available for later DES simulations. The sampled data are indicated by black dots in Fig. 9.

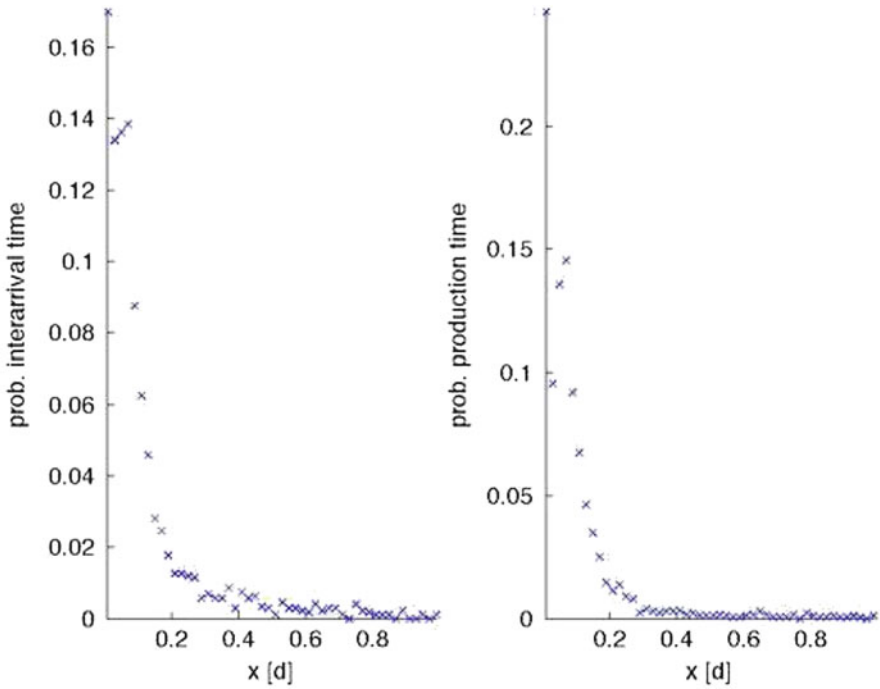


Fig. 8 Probability distribution of number of parts per day from a German manufacturing plant. *Left:* inter-arrival times, *right:* production times

Theoretically, now different approaches are possible. On the one hand we may fit an exponential probability distribution function $\Phi_r(x) = r \exp(-rx)H(x)$ of mean $\frac{1}{r}$ to each discrete resampled probability distribution. This leads to a Poisson distributed interarrival process of a certain (fitted) rate (called λ) and a Poisson distributed production process of a data fitted rate μ . Then, the setup is precisely as in an M/M/1 queuing model with the well-established relation between WIP W and flux $f \equiv \lambda$ as

$$f = \frac{\mu W}{1 + W}.$$

This relation is obtained also when simulating a DES with interarrival process given by Φ_λ and a production process described by Φ_μ . We have the advantage of deriving a single explicit formula closing Eq. (1). However, the data-fitting happens prior to simulating the dynamics.

In the second approach we reverse the procedure. We first apply a DES simulation sampling from the interpolated probability distributions. Then, we record the WIP and flux of several DES simulation. Note that for a DES simulation is not required to have exponentially distributed times. However, we do not expect a

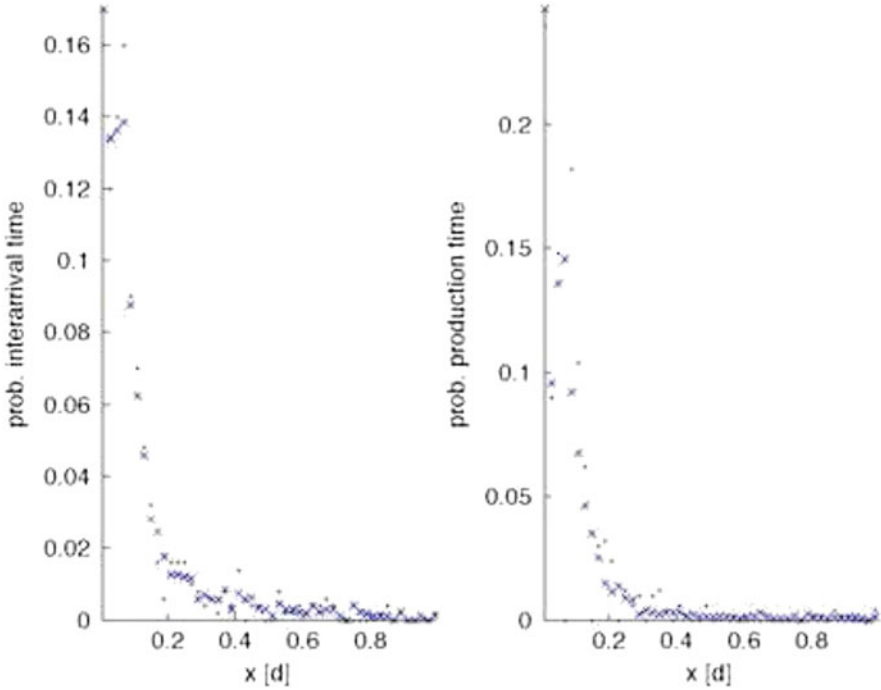


Fig. 9 Probability distribution of number of parts resampled from real data depicted in Fig. 8. In *black* are simulated points, in *blue* available data points. *Left*: interarrival times, *right*: production times

closed formula since the probability distributions are not exponentially distributed and therefore the process is not necessarily Poisson. The resulting WIP and flux values for 2000 simulations of the interpolated data is shown in Fig. 10. Clearly, we observe a spread of the data across the diagram related to the fact that the underlying interarrival and production probabilities are obtained from interpolated data. However, the data suggests an empirical clearing function $f = f(W)$. Several choices are possible. We depict in Fig. 11 a clearing function fitted to the mean of the data for any fixed WIP. This relation can not be expressed explicitly in a functional form. However, it also provides a closure relation for Eq. (1). In order to use this relation in a predictive model we would need to table the fitted clearing function. However, the computational effort is very small compared with a DES simulation. For example, here we require to table 50 pairs of WIP and flux in order to describe the closure relation. Within the second approach the averaging therefore happens after the DES simulation leading to a more detailed WIP flux relation. It is interesting to note that with the presented results the WIP flux relation would not be monotone any more. This allows therefore to also obtain a more complex flow pattern predicted by Eq. (1).

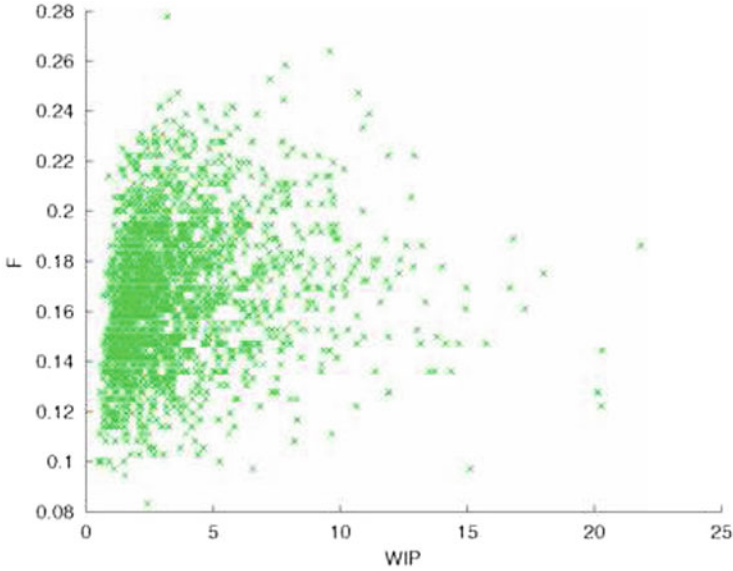


Fig. 10 WIP-flux relations for the DES simulated data. 2000 simulations are performed. The interarrival and production probabilities are interpolated from the existing data of the German company

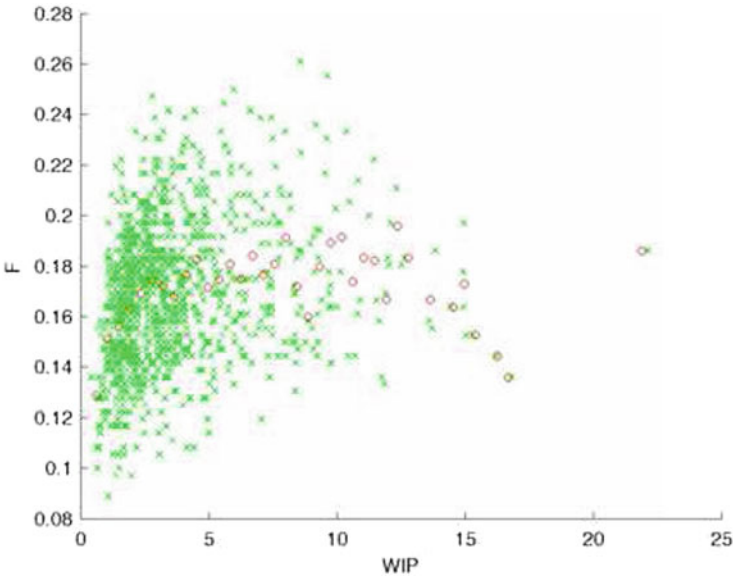


Fig. 11 WIP-flux relations for the DES simulated data. The final clearing function f is shown in red. In green color we show the underlying simulation results from 2000 simulations. The interarrival and production probabilities are interpolated from the existing data of the German company

We also mention a different approach presented in [14]. In Fig. 11 we observe that the simulation averages (depicted as red dots) WIP flux relation resembles for small values of the WIP a shape similar to

$$f(\mu, W) = \frac{\mu W}{1 + W}. \quad (11)$$

However, the value of μ is not necessarily equal to the value obtained in the first approach.

Furthermore, the simulation averages do not cover well the spread of the data. In [14] we propose to combine Eq. (1) with Eq. (11). This leads to an *additional unknown* $\mu = \mu(x, t)$ in the system. We need to prescribe a model for this quantity in order to close the system. To this end we note that μ resembles a production rate. This rate is supposedly known when parts arrive (stating a release date). However, this rate might change for the new parts. Hence, it is reasonable to assume that μ is a quantity that is moved with the product flow. The equation describing this observation is given by

$$\partial_t \mu(x, t) + v(x, t) \partial_x \mu(x, t) = 0. \quad (12)$$

Herein, v is the velocity of the moving parts of density ρ which is given by

$$\rho(x, t) v(x, t) = \frac{\mu(x, t) \rho(x, t)}{1 + \rho(x, t)} =: f(\mu(x, t), \rho(x, t)). \quad (13)$$

Summarizing, the full model proposed in [14] is given by Eqs. (1), (12) and (13). Among the properties of the system are hyperbolicity except at zero production density. The eigenvalues are at most $v(x, t)$. Therefore, there is only a finite speed of propagation of information bounded by the speed of the produced parts. This coincides with the expected behavior of a production line. The clearing functions of the extended model form a family of functions of the type (11) for a fixed value of μ . This allows to capture the spread in the data more efficiently.

Summarizing, several possibilities to extend classical M/M/1 queuing theory to time dynamic models of continuous type exist. Depending on the quality of the available data as well as the possible spread in the resulting DES simulation several approaches exit. We focus on the presentation of continuous models thereby neglecting detailed dynamics.

3 Outlook

We have presented recent approaches on modeling production flows using continuous partial differential equations. Compared with classical modeling approaches as DES, queuing theory or mixed-integer modeling, the differential equations allow

for a reduced computational complexity as well as efficient and structure preserving optimization and control approaches. However rigorous derivations of models based on differential equations are only possible for simplistic models of production scenarios. In case of more complex problems two other approaches have been presented. The approach based on the observed behavior has so far been able to capture the main effects of production lines with limited buffers. The approach based on available data of interarrival and production times has led to a second-order model. The theory of a rigorous justification based on the underlying product dynamics is still its infancy for both cases. Future work may include progress in the mean field limits, the extension of the models towards control and optimization problems as well as the extension towards large scale production networks. In all fields there are challenging mathematical as well as computational problems. The derived equations resemble to some extended fluid dynamical equations and one may adapt those methods here. However, hyperbolic transport properties are *fundamentally* different from fluid dynamics and require adapted and different theoretical and numerical treatment.

Acknowledgements This work has been supported by the Cluster of Excellence 'Integrative Production Technology for High-Wage Countries', the DFG grant GO 1920/3-1, the BMBF Project KinOpt and DAAD VRC.

References

1. Abate, J., Whitt, W.: Transient behavior of regulated Brownian motion, I: starting at the origin. *Adv. Appl. Probab.* **19**, 560–598 (1987)
2. Armbruster, D., Uzsoy, R.: Continuous dynamic models, clearing functions, and discrete-event simulation in aggregate production planning. In: Smith, J.C. (ed.) *New Directions in Informatics, Optimization, Logistics, and Production*. TutORials in Operations Research. INFORMS, Maryland (2012)
3. Armbruster, D., Marthaler, D., Ringhofer, C.: Kinetic and fluid model hierarchies for supply chains. *Oper. Res.* **2**, 43–61 (2003)
4. Armbruster, D., Marthaler, D., Ringhofer, C., Kempf, K., Jo, T.-C.: A continuum model for a re-entrant factory. *Oper. Res.* **54**, 933–950 (2006)
5. Armbruster, D., Degond, P., Ringhofer, C.: A model for the dynamics of large queuing networks and supply chains. *SIAM J. Appl. Math.* **66**, 896–920 (2006)
6. Armbruster, D., Göttlich, S., Herty, M.: A scalar conservation law with discontinuous flux for supply chains with finite buffers. *SIAM J. Appl. Math.* **71**, 1070–1087 (2011)
7. Asmundsson, J., Rardin, R.L., Uzsoy, R.: Tractable nonlinear production planning: Models for semiconductor wafer fabrication facilities. *IEEE Trans. Semicond. Wafer Fabr. Facil.* **19**, 95–111 (2006)
8. Asmundsson, J., Rardin, R.L., Turkseven, C.H., Uzsoy, R.: Production planning with resources subject to congestion. *Nav. Res. Logist.* **56**, 142–157 (2009)
9. Banks, J., Carson, J.S., Nelson, B.: *Discrete-Event System Simulation*. Prentice Hall International Series in Industrial and Systems Engineering. Prentice Hall, New Jersey (1984)
10. Bressan, A., Canic, S., Garavello, M., Herty, M., Piccoli, B.: Flow on networks: recent results and perspectives. *Eur. Math. Soc. Surv. Math. Sci.* **1**, 47–111 (2014)

11. D'Apice, C., Manzo, R.: A fluid dynamic model for supply chains. *Netw. Heterog. Media* **3**, 379–398 (2006)
12. D'Apice, C., Manzo, R., Piccoli, B.: Modelling supply networks with partial differential equations. *Q. Appl. Math.* **67**, 419–440 (2009)
13. D'Apice, C., Göttlich, S., Herty, M., Piccoli, B.: *Modeling, Simulation, and Optimization of Supply Chains: A Continuous Approach*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2010)
14. Forestier-Coste, L., Göttlich, S., Herty, M.: Data-fitted second-order macroscopic production models. *SIAM J. Appl. Math.* **75**, 999–1014 (2015)
15. Goossens, P.: Modeling of manufacturing systems with finite buffer sizes using PDEs. Masters Thesis, TU Eindhoven, Department of Mechanical Engineering, SE 420523 (2007)
16. Göttlich, S., Herty, M.: Dynamic models for simulation and optimization of supply networks. In: *Strategies and Tactics in Supply Chain Event Management*, pp. 249–265. Springer, Berlin (2008)
17. Göttlich, S., Herty, M., Klar, A.: Network models for supply chains. *Commun. Math. Sci.* **3**, 545–559 (2005)
18. Göttlich, S., Klar, A., Schindler, P.: Discontinuous conservation laws for production networks with finite buffers. *SIAM J. Appl. Math.* **73**, 1117–1138 (2013)
19. Graves, S.C.: A tactical planning model for a job shop. *Oper. Res.* **34**, 522–533 (1986)
20. Guillemin, F., Boyer, J.: Analysis of $M/M/1$ queue with processor sharing via spectral theory. *Queueing Syst.* **39**, 377–397 (2001/2002)
21. Herty, M., Klar, A., Piccoli, B.: Existence of solutions for supply chain models based on partial differential equations. *SIAM J. Math. Anal.* **39**, 160–173 (2007)
22. Herty, M., Jörres, C., Piccoli, B.: Existence of solution to supply chain models based on partial differential equation with discontinuous flux function. *J. Math. Anal. Appl.* **401**, 510–517 (2013)
23. Karmarkar, U.S.: Capacity loading and release planning with work-in-progress (WIP) and lead-times. *J. Manuf. Oper. Manage.* **2**, 105–123 (1989)
24. Missbauer, H.: Order release and sequence-dependent setup times. *Int. J. Prod. Econ.* **49**, 131–143 (1997)
25. Missbauer, H.: Aggregate order release planning for time varying demand. *Int. J. Prod. Res.* **40**, 699–718 (2002)
26. Missbauer, H.: Models of the transient behaviour of production units to optimize the aggregate material flow. *Int. J. Prod. Econ.* **118**, 387–397 (2009)
27. Newell, A., Rosenbloom, P.S.: *Mechanisms of skill acquisition and the law of practice. Cognitive Skills and Their Acquisition*, vol. 1. Erlbaum, Hillsdale (1981)
28. Odoni, A.R., Roth, E.: An empirical investigation of the transient behavior of stationary queueing systems. *Oper. Res.* **31**, 432–455 (1983)

Max-Plus-Linear Systems for Modeling and Control of Manufacturing Problems

Xavier David-Henriet, Laurent Hardouin, and Jörg Raisch

Abstract In this chapter, the dynamics of manufacturing systems is characterized through the occurrence of events such as parts entering or leaving machines. Furthermore, we assume that the relations between events are expressed by synchronizations (i.e., conditions of the form: for all $k \geq l$, occurrence k of event e_2 is at least τ units of time after occurrence $k - l$ of event e_1). Note that this assumption often holds when the considered manufacturing system is functioning under a predefined schedule. First, we discuss the modeling of such systems by linear state-space models in the $(\max, +)$ -algebra (due to this property, such systems are often called $(\max, +)$ -linear systems). Second, standard open-loop and closed-loop control structures for $(\max, +)$ -linear systems are recalled. These control structures lead to a trade-off between the rapidity of systems and their internal buffer sizes. Some techniques to influence this trade-off are presented.

X. David-Henriet (✉)

Université d'Angers, LARIS, ISTIA, 62 Av. Notre-Dame du Lac, 49000 Angers, France

Technische Universität Berlin, Fachgebiet Regelungssysteme, Einsteinufer 17, 10587 Berlin, Germany

Max-Planck-Institut für Dynamik komplexer technischer Systeme, Fachgruppe System- und Regelungstheorie, Sandtorstr. 1, 39106 Magdeburg, Germany

e-mail: david-henriet@control.tu-berlin.de; xavier.davidhenriet@gmail.com

L. Hardouin (✉)

Université d'Angers, LARIS, ISTIA, 62 Av. Notre-Dame du Lac, 49000 Angers, France

e-mail: laurent.hardouin@univ-angers.fr

J. Raisch (✉)

Technische Universität Berlin, Fachgebiet Regelungssysteme, Einsteinufer 17, 10587 Berlin, Germany

Max-Planck-Institut für Dynamik komplexer technischer Systeme, Fachgruppe System- und Regelungstheorie, Sandtorstr. 1, 39106 Magdeburg, Germany

e-mail: raisch@control.tu-berlin.de

1 Introduction

A discrete event system (e.g., [1]) is a dynamical system driven by the instantaneous occurrences of events. In a discrete event system, two basic elements are distinguished: the event set and the rule describing the behavior of the system. By considering events such as parts entering or leaving machines, discrete event systems offer an interesting framework to model manufacturing systems at a high level of abstraction. Many formal approaches such as finite-state automata (e.g., [2]) and Petri nets (e.g., [3]) have been investigated to express the rule describing the behavior of the system. In the following, we focus on discrete event systems where this rule is only composed of synchronizations (i.e., conditions of the form: for all $k \geq l$, occurrence k of event e_2 is at least τ units of time after occurrence $k-l$ of event e_1 with $\tau \in \mathbb{N}_0$ and $l \in \mathbb{N}_0$). The behavior of manufacturing systems functioning under a predefined schedule can often be adequately modeled by synchronizations (see Example 1).

Discrete event systems where the rule describing the behavior is only composed of synchronizations are called $(\max, +)$ -linear systems. This terminology is due to the fact that a specific behavior, namely the behavior under the earliest functioning rule, is described by linear equations in particular algebraic structures such as the $(\max, +)$ -algebra. In the literature, only this specific behavior is usually considered. For $(\max, +)$ -linear systems, it is possible to partition the set of events into input, internal, and output events and, based on this partition, to derive a $(\max, +)$ -linear state-space model of the system. Therefore, much effort has been made during the last decades to adapt key concepts from standard control theory to $(\max, +)$ -linear systems. Transfer function matrices have been introduced for $(\max, +)$ -linear systems by using formal power series [4]. Furthermore, some standard control approaches such as optimal feedforward control [5], model reference control [6–8], and model predictive control [9] have been extended to $(\max, +)$ -linear systems. For manufacturing systems, model reference control is particularly interesting, as it offers techniques to both reduce the size of internal buffers and take into account unexpected disturbances.

We emphasize that the purpose of this contribution is not to compare different modelling and control approaches for manufacturing systems. On the contrary, we concentrate on a specific class of manufacturing systems exclusively governed by synchronization and delay phenomena. As pointed out above, models for this class of systems are linear in certain algebraic structures. For this reason, many methods for designing control can be adapted from standard linear systems theory to be applicable to the discussed class of manufacturing systems. A key advantage of this approach is that the desired control policy, i.e., the way control reacts to external inputs and measured outputs, can be computed analytically and offline. Hence, the required computational online effort is negligible.

The rule describing the behavior of $(\max, +)$ -linear systems can also be expressed by specific timed Petri nets called timed event graphs (TEGs). A TEG is a directed bipartite graph, where the set of nodes is partitioned into a set of places and a set of transitions, and arcs are either from places to transitions or from transitions to places. Moreover, in a TEG, each place has precisely one incoming and one outgoing arc. Each place is equipped with a holding time. Places may contain tokens, and transitions are associated with events. A transition can “fire” (i.e., the associated event can occur) if and only if each place from which an arc leads to the transition (“upstream place”) has at least one token residing in the respective place for at least the corresponding holding time. If the transition “fires” (i.e., the associated event occurs), all upstream places lose one token and all downstream places (places to which there is an arc from the considered transition) gain one token. Places and transitions are graphically represented by circles and bars, and the holding times, if nonzero, are indicated by adding numbers to places. In the following, we focus on $(\max, +)$ -linear representations to formally manipulate systems, but use timed event graphs to graphically represent systems.

This chapter is structured as follows. In the next section, necessary mathematical tools are recalled. The modeling of the considered class of discrete event systems in the $(\max, +)$ -algebra and in the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ is presented in Sect. 3. Finally, Sect. 4 focuses on control for $(\max, +)$ -linear systems. Throughout this chapter, the simple manufacturing system introduced in Example 1 is used to illustrate and clarify the presented concepts. We emphasize that illustration and clarification is the *sole* purpose of this example. However, methods based on $(\max, +)$ -linear systems are also suitable for industrially relevant systems: for example, in [10], this approach is used to model and control high-throughput screening systems (i.e., systems to rapidly test thousands of biochemical substances) with over one hundred events and dozens of activities and resources.

Example 1 A simple manufacturing system composed of three machines, denoted M_1 , M_2 , and M_3 , is considered. Machine M_1 consumes workpieces of type 1 and releases workpieces of type 3. Machine M_2 consumes workpieces of type 2 and releases workpieces of type 4. Machine M_3 pairwise assembles workpieces of type 3 and 4 and delivers workpieces of type 5. The production of a new workpiece of type 5 from workpieces of type 1 and 2 starts after the receiving of an order from a customer. Orders and workpieces of type 1 and 2 correspond to the inputs of the manufacturing system (i.e., external influences either from suppliers or from customers) and workpieces of type 5 correspond to the output of the manufacturing system. Each machine has a capacity of one. The processing time associated with machine M_1 , denoted τ_1 , is four units of time and the processing time associated with machine M_2 (resp. M_3), denoted τ_2 (resp. τ_3), is two units of time. Furthermore, a machine M_i with $1 \leq i \leq 3$ can start processing the next workpiece as soon as it finishes processing the current workpiece. The buffers have an infinite capacity.

To formally describe the dynamics of this manufacturing system, we define the following events:

- event u_i (with $i = 1, 2$) a workpiece of type i enters the system
- event s_i (with $1 \leq i \leq 3$) machine M_i starts to process a (pair of) workpiece(s)
- event f_i (with $1 \leq i \leq 3$) machine M_i delivers a processed workpiece
- event o an order is received
- event y a workpiece of type 5 leaves the system

The behavior of the considered manufacturing system is completely expressed by synchronizations of the events defined above. Two synchronizations are needed to express the dynamics of each machine M_i with $1 \leq i \leq 3$. The first synchronization models the process associated with machine M_i : for all $k \geq 0$, occurrence k of event f_i is at least τ_i units of time after occurrence k of event s_i . The second synchronization models the capacity constraint: for all $k \geq 1$, occurrence k of event s_i is at least zero units of time after occurrence $k - 1$ of event f_i . Furthermore, to model the flow of workpieces outside the machines some additional synchronizations are needed. The supply of workpieces of type i with $i = 1, 2$ is modeled by “for all $k \geq 0$, occurrence k of event s_i is at least zero units of time after occurrence k of event u_i ” with $i = 1, 2$. The supply for machine M_3 of workpieces processed by machine M_i with $i = 1, 2$ is expressed by “for all $k \geq 0$, occurrence k of event s_3 is at least zero units of time after occurrence k of event f_i ” with $i = 1, 2$. The release of workpieces of type 5 is modeled “for all $k \geq 0$, occurrence k of event y is at least zero units of time after occurrence k of event f_3 ”. Finally, orders are taken into account by “for all $k \geq 0$, occurrence k of event s_i is at least zero units of time after occurrence k of event o ” with $i = 1, 2$.

The timed event graph associated with this manufacturing system is shown in Fig. 1, where holding times (if nonzero) are indicated by numbers attached to places.

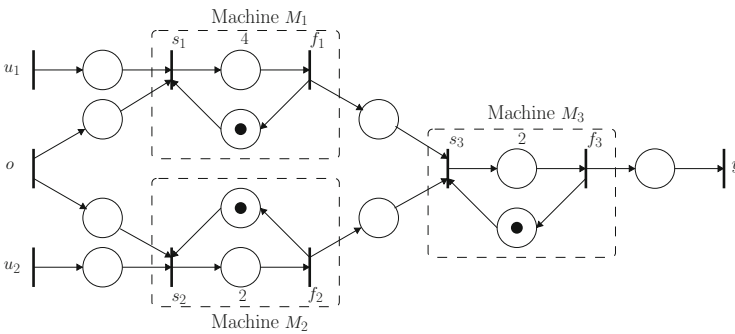


Fig. 1 A simple manufacturing system

2 Mathematical Preliminaries

In this section, necessary elements of dioid theory and residuation theory are recalled. A complete survey on these topics is available in [4] and [11], respectively.

2.1 Dioid Theory

Dioids (or idempotent semirings) are algebraic structures which play a major role in the modeling of (max, +)-linear systems.

Definition 1 (Dioid) A dioid is a set \mathcal{D} endowed with two binary operations, denoted \oplus and \otimes , such that:

- \oplus is associative, commutative, idempotent ($\forall a \in \mathcal{D}, a \oplus a = a$), and admits a neutral element ε .
- \otimes is associative and admits a neutral element e .
- \otimes is distributive with respect to \oplus from both sides:

$$\forall a, b, c \in \mathcal{D}, \quad \begin{cases} a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) \\ (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \end{cases}$$

- ε is absorbing for \otimes , i.e., $\forall a \in \mathcal{D}, a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.

If \mathcal{D} is closed for infinite sums and distributivity is extended to infinite sums, then dioid \mathcal{D} is said to be complete.

Formally, the operations \oplus and \otimes are very similar to the standard operations $+$ and \times . Therefore, these operations are respectively called addition and multiplication. Then, ε is called the zero element of the dioid \mathcal{D} and e is its unit element. As in classical algebra, \otimes is often omitted and the product is simply denoted by juxtaposition (i.e., ab corresponds to $a \otimes b$). As \oplus is associative, commutative, and idempotent, it induces a partial order \leq on \mathcal{D} defined by $a \leq b \Leftrightarrow a \oplus b = b$. Hence, a dioid is a partially ordered set.

By analogy with standard linear algebra, the operations \oplus and \otimes are extended to matrices with entries in a dioid \mathcal{D} .

$$\forall A, B \in \mathcal{D}^{n \times p}, \quad (A \oplus B)_{ij} = A_{ij} \oplus B_{ij}$$

$$\forall A \in \mathcal{D}^{n \times p}, \forall B \in \mathcal{D}^{p \times q}, \quad (A \otimes B)_{ij} = \bigoplus_{k=1}^p A_{ik} B_{kj}$$

The operation \oplus also provides a partial order \leq over $\mathcal{D}^{n \times p}$. Formally, for $A, B \in \mathcal{D}^{n \times p}$, $A \succeq B \Leftrightarrow A = A \oplus B$. The next proposition gives the algebraic structure of

the set of square matrices with entries in a dioid endowed with the operations \oplus and \otimes defined above.

Proposition 1 ([4]) *Let \mathcal{D} be a dioid. The set $\mathcal{D}^{n \times n}$ endowed with the operations \oplus and \otimes defined above is a dioid. Besides, if \mathcal{D} is complete, then $\mathcal{D}^{n \times n}$ is complete.*

The next theorem plays an essential role in the following to solve implicit inequalities of the form $X \succeq AX \oplus B$ where A , X , and B are matrices with entries in a complete dioid.

Theorem 1 (Kleene Star Theorem [4]) *Let \mathcal{D} be a complete dioid and $A \in \mathcal{D}^{n \times n}$, $B \in \mathcal{D}^{n \times p}$. Denote the unit element of $\mathcal{D}^{n \times n}$ by e . Then, the inequality $X \succeq AX \oplus B$ admits A^*B as least solution, where the Kleene star of A , denoted A^* , is defined by*

$$A^* = \bigoplus_{k=0}^{+\infty} A^k \text{ with } A^k = \begin{cases} e & \text{if } k = 0 \\ A \otimes A^{k-1} & \text{otherwise} \end{cases}$$

In Sect. 3, modeling of $(\max, +)$ -linear systems in the $(\max, +)$ -algebra and in the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ will be discussed. Next we briefly describe these two dioids.

2.1.1 The $(\max, +)$ -Algebra

The $(\max, +)$ -algebra, denoted $\overline{\mathbb{N}}_{\max}$, is defined as the set $\mathbb{N}_0 \cup \{-\infty, +\infty\}$ endowed with the operations \max and $+$. This corresponds to a complete dioid with \max as addition \oplus and $+$ as multiplication \otimes . The zero element ε is equal to $-\infty$ and the unit element e is equal to 0. The order \leq induced by the operation \oplus corresponds to the standard order, as

$$a \leq b \Leftrightarrow a \oplus b = b \Leftrightarrow b = \max(a, b) \Leftrightarrow a \leq b$$

Example 2 In the following, some simple calculations in $\overline{\mathbb{N}}_{\max}$ are described. In the scalar case,

$$5 \oplus 3 = \max(5, 3) = 5 \text{ and } 5 \otimes 3 = 5 + 3 = 8$$

In the matrix case,

$$\begin{pmatrix} 5 & 3 & +\infty \\ \varepsilon & 4 & \varepsilon \\ e & \varepsilon & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} 2 & \varepsilon & 2 \\ 3 & e & 4 \\ e & \varepsilon & +\infty \end{pmatrix} = \begin{pmatrix} 5 & 3 & +\infty \\ 3 & 4 & 4 \\ e & \varepsilon & +\infty \end{pmatrix}$$

$$\begin{pmatrix} 5 & 3 & +\infty \\ \varepsilon & 4 & \varepsilon \\ e & \varepsilon & \varepsilon \end{pmatrix} \otimes \begin{pmatrix} 2 & \varepsilon & 2 \\ 3 & e & 4 \\ e & \varepsilon & +\infty \end{pmatrix} = \begin{pmatrix} +\infty & 3 & +\infty \\ 7 & 4 & 8 \\ 2 & \varepsilon & 2 \end{pmatrix}$$

2.1.2 The Dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

In the following, a brief introduction to the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ is given. This dioid is especially convenient for modeling and control of $(\max, +)$ -linear systems. For a formal definition of this dioid, the reader is invited to consult [4]. A C++-library dedicated to computation in the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ is described in [12]. First, the concepts of daters and operators are recalled.

Definition 2 (Dater) A dater is a non-decreasing mapping from \mathbb{Z} to $\overline{\mathbb{N}}_{\max}$ equal to ϵ over $\{n \in \mathbb{Z} | n < 0\}$. The set of daters is denoted D .

In the following sections, daters will be used to describe the occurrence times of events. Then, for a dater d associated with a particular event, $d(k), k \geq 0$, will denote the time when the event occurs for the k th time. Note that it is customary to start enumeration of event occurrences by 0 (instead of 1).

Of particular interest are the daters ϵ_D and e_D defined by

$$\forall k \in \mathbb{Z}, \quad \epsilon_D(k) = \epsilon \text{ and } e_D(k) = \begin{cases} \epsilon & \text{if } k < 0 \\ e & \text{if } k \geq 0 \end{cases}$$

The set of daters is endowed with an operation, denoted \oplus , derived from the operation \oplus over $\overline{\mathbb{N}}_{\max}$. Formally,

$$\forall d_1, d_2 \in D, \forall k \in \mathbb{Z}, \quad (d_1 \oplus d_2)(k) = d_1(k) \oplus d_2(k)$$

Definition 3 (Operator) An operator is a mapping from D to D . The set of operators is denoted \mathcal{O} .

Using the operation \oplus over D , a matrix of operators is defined as a mapping between vectors of daters. Matrix $O \in \mathcal{O}^{n \times p}$ corresponds to the mapping from D^p to D^n defined by

$$\forall d \in D^p, \quad O(d)_i = \bigoplus_{j=1}^p O_{ij}(d_j)$$

Of particular interest are the operators $\epsilon_{\mathcal{O}}, e_{\mathcal{O}}, \gamma$, and δ defined by

$$\forall d \in D, \quad \epsilon_{\mathcal{O}}(d) = \epsilon_D \text{ and } e_{\mathcal{O}}(d) = d$$

$$\forall d \in D, \forall k \in \mathbb{Z}, \quad \gamma(d)(k) = d(k-1) \text{ and } \delta(d)(k) = 1d(k)$$

The set of operators is endowed with an operation, denoted \oplus , derived from the operation \oplus defined over D . Formally,

$$\forall o_1, o_2 \in \mathcal{O}, \forall d \in D \quad (o_1 \oplus o_2)(d) = o_1(d) \oplus o_2(d)$$

Furthermore, an operation \otimes over \mathcal{O} is defined as the composition of mappings: for all $o_1, o_2 \in \mathcal{O}$, $o_1 \otimes o_2 = o_1 \circ o_2$. Under some conditions, the set of operators \mathcal{O} endowed with the operations \oplus and \otimes defined above is a complete dioid. Then, the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ is defined to be the complete dioid spanned by $\{\varepsilon_{\mathcal{O}}, e_{\mathcal{O}}, \gamma, \delta\}$. Let $\nu \in \mathbb{N}_0$ and $\tau \in \mathbb{N}_0$. The operator $\gamma^\nu \delta^\tau$ belongs to $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ and corresponds to

$$\forall d \in D, \forall k \in \mathbb{Z}, \quad (\gamma^\nu \delta^\tau)(d)(k) = \tau d(k - \nu)$$

By construction, calculation rules are available to simplify expressions in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Operators γ and δ commute:

$$\begin{aligned} \forall d \in D, \forall k \in \mathbb{Z}, \quad (\gamma\delta)(d)(k) &= \delta(d)(k - 1) \\ &= 1d(k - 1) \\ &= 1\gamma(d)(k) \\ &= (\delta\gamma)(d)(k) \end{aligned}$$

Furthermore, let l_1, l_2 in \mathbb{N}_0 . For all $d \in D$ and $k \in \mathbb{Z}$,

$$\begin{aligned} (\delta^{l_1} \oplus \delta^{l_2})(d)(k) &= l_1 d(k) \oplus l_2 d(k) \\ &= (l_1 \oplus l_2) d(k) \\ &= \delta^{\max(l_1, l_2)}(d)(k) \\ (\gamma^{l_1} \oplus \gamma^{l_2})(d)(k) &= d(k - l_1) \oplus d(k - l_2) \\ &= d(k - \min(l_1, l_2)) \text{ as dater } d \text{ is non-decreasing} \\ &= \gamma^{\min(l_1, l_2)}(d)(k) \end{aligned}$$

Hence, $\delta^{l_1} \oplus \delta^{l_2} = \delta^{\max(l_1, l_2)}$ and $\gamma^{l_1} \oplus \gamma^{l_2} = \gamma^{\min(l_1, l_2)}$.

Representing Daters in the Dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

The dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ offers a method to elegantly manipulate daters: a dater d is associated with the operator $\bigoplus_{k=0}^{+\infty} \gamma^k \delta^{d(k)}$ where $\delta^{-\infty}$ (resp. $\delta^{+\infty}$) stands for ε (resp. δ^*). Then, the operator o in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ associated with a dater d is the single operator in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ satisfying $o(e_D) = d$. Using calculation rules specific to $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, the expression of the operator associated with a dater is often much simpler than the expression of the dater itself. In the following, we do not distinguish between a dater and the associated operator in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

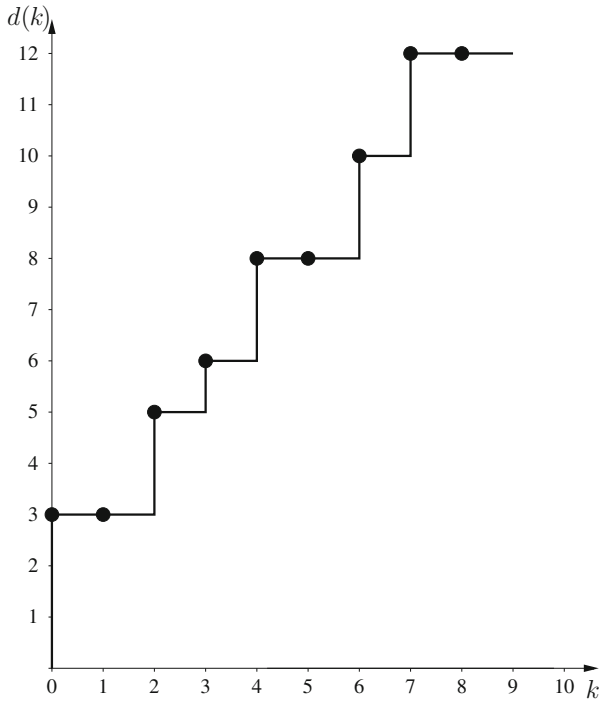


Fig. 2 Dater d

Example 3 Let us consider the dater d defined by

$$d(k) = \begin{cases} \varepsilon & \text{if } k < 0 \\ 3 & \text{if } k = 0, 1 \\ 5 & \text{if } k = 2 \\ 6 + 4j & \text{if } k = 3 + 3j \text{ with } j \in \mathbb{N}_0 \\ 8 + 4j & \text{if } k = 4 + 3j, 5 + 3j \text{ with } j \in \mathbb{N}_0 \end{cases}$$

The dater d is pictured in Fig. 2. In $\mathcal{M}_{in}^{ax}[\gamma, \delta]$,

$$d = \bigoplus_{k=0}^{+\infty} \gamma^k \delta^{d(k)} = \delta^3 \oplus \gamma \delta^3 \oplus \gamma^2 \delta^5 \oplus (\gamma^3 \delta^6 \oplus \gamma^4 \delta^8 \oplus \gamma^5 \delta^8) (\gamma^3 \delta^4)^*$$

Using calculation rules specific to $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, the expression of dater d is simplified:

$$d = \delta^3 \oplus \gamma^2 \delta^5 \oplus (\gamma^3 \delta^6 \oplus \gamma^4 \delta^8) (\gamma^3 \delta^4)^*$$

2.2 Residuation Theory

Residuation theory gives the theoretical foundation for the control of (max, +)-linear systems.

Definition 4 (Residuaded Mapping) Let $f : E \rightarrow F$ with E and F ordered sets. Mapping f is said to be residuated if f is non-decreasing and if, for all $y \in F$, the least upper bound of the subset $\{x \in E \mid f(x) \leq y\}$ exists and lies in this subset. This element in E is denoted $f^\sharp(y)$. Mapping f^\sharp from F to E is called the residual of f .

Let a be an element in a complete dioid \mathcal{D} . The mappings $L_a : x \mapsto a \otimes x$ (left-multiplication by a) and $R_a : x \mapsto x \otimes a$ (right-multiplication by a) over \mathcal{D} are residuated. The residuals are denoted by $L_a^\sharp(x) = a \backslash x$ (left-division by a) and $R_a^\sharp(x) = x \phi a$ (right-division by a). By definition, $a \backslash b$ (resp. $b \phi a$) denotes the greatest solution x of the inequality $a \otimes x \leq b$ (resp. $x \otimes a \leq b$).

The operations \backslash and ϕ are also extended to matrices. Hence, $A \backslash B$ (resp. $B \phi A$) corresponds to the greatest solution X of the inequality $AX \leq B$ (resp. $XA \leq B$).

Example 4 For a, b in $\overline{\mathbb{N}}_{\max}$,

$$a \backslash b = b \phi a = \begin{cases} +\infty & \text{if } a = \varepsilon \text{ or } b = +\infty \\ \varepsilon & \text{if } a \succeq b \text{ and } a \neq b \\ b - a & \text{if } b \succeq a \text{ and } a, b \in \mathbb{N}_0 \end{cases}$$

3 Modeling

After some preliminary remarks on the modeling assumptions, the modeling of (max, +)-linear systems is presented both in the (max, +)-algebra and in the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

3.1 Preliminaries

3.1.1 Input, Output, and Internal Events

The event set of a (max, +)-linear system is partitioned into

input events: these events are the source of synchronizations, but not subject to synchronizations. Input events correspond to external events affecting the system (e.g., external supplies of workpieces or orders from customers).

output events: these events are subject to synchronizations, but not the source of synchronizations. Output events correspond to events in the system which are directly seen by other systems (e.g., deliveries of finished products).

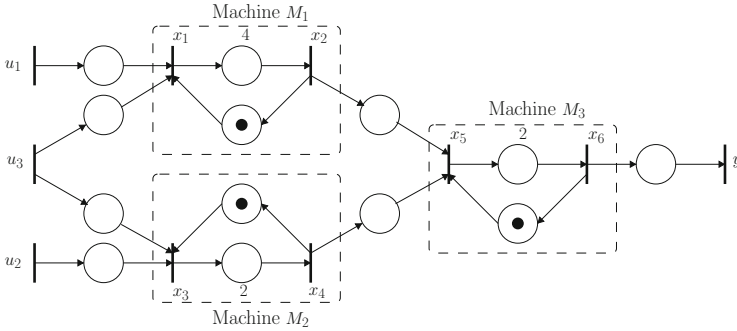


Fig. 3 A manufacturing system

internal events: these events are both subject to and the source of synchronizations. Internal events model the internal dynamics of the system.

Events which are neither subject to nor the source of synchronizations are neglected, as we focus on interactions between events. In the rest of this chapter, we consider (max, +)-linear systems, where:

- the sets of input, output, and internal events are not empty
- there exist no direct synchronizations of output events by input events

In practice, these assumptions either hold or can be made to hold by adding some fictitious internal events. Furthermore, the following convention for notation is used. The numbers of input, output, and internal events are respectively denoted by m , p , and n . Input, output, and internal events are respectively denoted by u , y , and x and integer subscripts are used to distinguish events of the same kind.

Example 5 In the considered example, the event set is partitioned into

- input events u_1 , u_2 , and u_3
- internal events s_1 , s_2 , s_3 , f_1 , f_2 , and f_3
- output event y

These events are relabeled according to the above notation (see Fig. 3). For this system, $m = 3$, $n = 6$, and $p = 1$.

3.1.2 Earliest Functioning Rule

Synchronizations (i.e., conditions of the form: for all $k \geq l$, occurrence k of event e_2 is at least τ units of time after occurrence $k - l$ of event e_1) only specify conditions enabling occurrences of events, but never force an event to occur. Therefore, a (max, +)-linear system is not univocally determined: a predefined timing pattern of the input events may lead to different timing patterns for internal and output

events. The only requirement is that these patterns are admissible with respect to the synchronizations required by the considered system.

In the following, we only consider a particular behavior for $(\max, +)$ -linear systems, namely the behavior under the earliest functioning rule. The earliest functioning rule requires that each internal or output event occurs as soon as possible. Under the earliest functioning rule, a $(\max, +)$ -linear system is univocally determined: a predefined timing pattern of the input events leads to a unique timing pattern for internal and output events. This fundamental property is a direct consequence of the model in the $(\max, +)$ -algebra presented later.

Example 6 In the considered example, the earliest functioning rule is suitable, as the aim is to meet the orders as soon as possible.

3.1.3 Modeling with Daters

To capture the timed dynamics of a discrete event system, a dater is associated with each event such that the dater gives the times of occurrences of the considered event. In the following, no distinction in the notation is made between an event and the associated dater. Hence, for an event d , $d(k)$ denotes the time of occurrence k of event d . This leads to the following interpretation for daters:

- $d(k) = \varepsilon$: occurrence k of event d is at $t = -\infty$. By convention, occurrence k , with $k < 0$, of an event is always at $t = -\infty$.
- $d(k) \in \mathbb{N}_0$: occurrence k of event d is at time $d(k)$.
- $d(k) = +\infty$: occurrence k of event d never happens.

The fact that daters are non-decreasing (i.e., for a dater d , $d(k+1) \succeq d(k)$ for all $k \in \mathbb{Z}$) is always satisfied as occurrence $k+1$ of event d is never before occurrence k of event d .

3.2 Modeling in the $(\max, +)$ -Algebra

Next, we show how to model $(\max, +)$ -linear systems by recursive equations in the $(\max, +)$ -algebra. Using daters, the synchronization “for all $k \geq l$, occurrence k of event e_2 is at least τ units of time after occurrence $k-l$ of event e_1 ” corresponds to

$$\forall k \in \mathbb{Z}, \quad e_2(k) \geq \tau + e_1(k-l)$$

in the standard algebra or to

$$\forall k \in \mathbb{Z}, \quad e_2(k) \succeq \tau e_1(k-l)$$

in the $(\max, +)$ -algebra. Furthermore, the effect of several synchronizations on a single event is also expressed by a single inequality. For example, the synchronizations “for all $k \geq l_1$, occurrence k of event e_2 is at least τ_1 units of time after occurrence $k - l_1$ of event $e_{1,1}$ ” and “for all $k \geq l_2$, occurrence k of event e_2 is at least τ_2 units of time after occurrence $k - l_2$ of event $e_{1,2}$ ” are both expressed by a single inequality either in the standard algebra

$$\forall k \in \mathbb{Z}, \quad e_2(k) \geq \max(\tau_1 + e_{1,1}(k - l_1), \tau_2 + e_{1,2}(k - l_2))$$

or in the $(\max, +)$ -algebra

$$\forall k \in \mathbb{Z}, \quad e_2(k) \geq \tau_1 e_{1,1}(k - l_1) \oplus \tau_2 e_{1,2}(k - l_2)$$

Hence, the rule describing the behavior of the system can be expressed by the following matrix inequalities in $\overline{\mathbb{N}}_{\max}$.

$$\begin{cases} x(k) \geq \bigoplus_{i=0}^L (A_i x(k - i) \oplus B_i u(k - i)) \\ y(k) \geq \bigoplus_{i=0}^L C_i x(k - i) \end{cases} \quad (1)$$

where x , u , and y respectively correspond to the vectors of daters associated with internal, input, and output events, and L denotes the greatest parameter l over all synchronizations. Furthermore, matrices A_i , B_i , and C_i belong respectively to $\overline{\mathbb{N}}_{\max}^{n \times n}$, $\overline{\mathbb{N}}_{\max}^{n \times m}$, and $\overline{\mathbb{N}}_{\max}^{p \times n}$. The entries of these matrices are given by the parameters of the synchronizations.

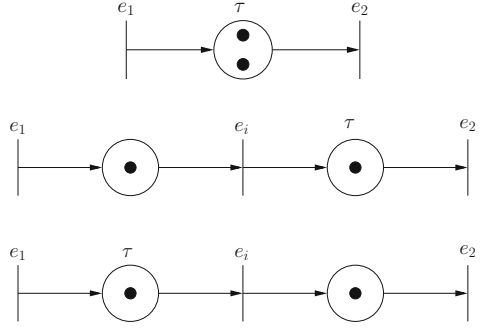
To simplify (1), the event set of the considered $(\max, +)$ -linear system is extended by additional internal events. The resulting extended set of internal events is referred to as the set of state events. The daters of all state events are collected in a single vector, which, slightly abusing notation, is again called x . This allows us to convert (1) to a first-order recursion. The resulting inequalities are given in (2). The validity of this step results from the equivalence between the different synchronization relations between events e_1 and e_2 pictured in Fig. 4.

$$\begin{cases} x(k) \geq A_0 x(k) \oplus A_1 x(k - 1) \oplus B_0 u(k) \\ y(k) \geq C_0 x(k) \end{cases} \quad (2)$$

By convention, $x(k)$ and $y(k)$ have all entries equal to ε for $k < 0$. This choice is valid according to (2). As the behavior under the earliest functioning rule is considered, the time of occurrence $k \geq 0$ of state and output events is given by the least solution for $x(k)$ and $y(k)$ in (2). Considering that x is composed of daters (i.e., $x(k) \geq x(k - 1)$ for all $k \in \mathbb{Z}$), we have

$$\begin{aligned} x(k) &\geq A_0 x(k) \oplus A_1 x(k - 1) \oplus B_0 u(k) \\ \Leftrightarrow x(k) &\geq A_0 x(k) \oplus (A_1 \oplus e) x(k - 1) \oplus B_0 u(k) \end{aligned}$$

Fig. 4 Equivalent synchronizations if no other synchronizations affect event e_i



Hence, using Theorem 1, the following (max, +)-linear state-space model is obtained:

$$\begin{cases} x(k) = Ax(k-1) \oplus Bu(k) \\ y(k) = Cx(k) \end{cases} \quad (3)$$

where $A = A_0^*(A_1 \oplus e)$, $B = A_0^*B_0$, and $C = C_0$. Hence, (max, +)-linear systems are deterministic and, as expected, (max, +)-linear (i.e., a (max, +)-linear combination of inputs induces the corresponding (max, +)-linear combination of outputs).

Example 7 The synchronizations in the considered example are represented by the following matrix inequalities in $\overline{\mathbb{N}}_{\max}$.

$$\begin{cases} x(k) \succeq \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon \end{pmatrix} x(k) \oplus \begin{pmatrix} \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} x(k-1) \oplus \begin{pmatrix} e & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & e \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} u(k) \\ y(k) \succeq (\varepsilon \ \varepsilon \ \varepsilon \ \varepsilon \ \varepsilon \ e) x(k) \end{cases}$$

This leads to the following (max, +)-linear state-space model:

$$\begin{cases} x(k) = \begin{pmatrix} e & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 4 & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 2 & 2 & \varepsilon & \varepsilon \\ 4 & 4 & 2 & 2 & e & e \\ 6 & 6 & 4 & 4 & 2 & 2 \end{pmatrix} x(k-1) \oplus \begin{pmatrix} e & \varepsilon & e \\ 4 & \varepsilon & 4 \\ \varepsilon & e & e \\ \varepsilon & 2 & 2 \\ 4 & 2 & 4 \\ 6 & 4 & 6 \end{pmatrix} u(k) \\ y(k) = (\varepsilon \ \varepsilon \ \varepsilon \ \varepsilon \ \varepsilon \ e) x(k) \end{cases}$$

Let us consider the input corresponding to a supply of five workpieces of type 1 and type 2 at time 0 and an order of five workpieces of type 5 at time 0. Hence the k th, $0 \leq k \leq 4$, occurrence of event u_1 (“a workpiece of type 1 enters the system”), u_2 (“a workpiece of type 2 enters the system”) and $u_3 = o$ (“an order is received”) is at time 0. The associated daters are

$$u_1(k) = u_2(k) = u_3(k) = \begin{cases} \varepsilon & \text{if } k < 0 \\ e & \text{if } 0 \leq k < 5 \\ +\infty & \text{if } k \geq 5 \end{cases}$$

The induced output can be easily calculated from the linear difference equation (3):

$$y(k) = \begin{cases} \varepsilon & \text{if } k < 0 \\ 6 \otimes 4^k & \text{if } 0 \leq k < 5 \\ +\infty & \text{if } k \geq 5 \end{cases}$$

Hence, a workpiece of type 5 is delivered at time 6, 10, 14, 18, and 22.

3.3 Modeling in the Dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

Next, we show how to model (max, +)-linear systems in the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Let us consider the synchronization “for all $k \geq l$, occurrence k of event e_2 is at least τ units of time after occurrence $k - l$ of event e_1 ”. As mentioned before, this corresponds to the following inequality in $\overline{\mathbb{N}}_{\max}$:

$$\forall k \in \mathbb{Z}, \quad e_2(k) \succeq \tau e_1(k - l)$$

Rewriting this relation with the operators γ and δ leads to the following inequality over daters: $e_2 \succeq (\delta^\tau \gamma^l)(e_1)$. Furthermore, the combination of several synchronizations on the same event can be expressed in a single inequality by using the operation \oplus over daters. For example, synchronizations “for all $k \geq l_1$, occurrence k of event e_2 is at least τ_1 units of time after occurrence $k - l_1$ of event $e_{1,1}$ ” and “for all $k \geq l_2$, occurrence k of event e_2 is at least τ_2 units of time after occurrence $k - l_2$ of event $e_{1,2}$ ” are both expressed by a single inequality:

$$e_2 \succeq (\delta^{\tau_1} \gamma^{l_1})(e_{1,1}) \oplus (\delta^{\tau_2} \gamma^{l_2})(e_{1,2})$$

Hence, the rule describing the behavior of the system can be expressed by the following matrix inequalities.

$$\begin{cases} x \succeq A(x) \oplus B(u) \\ y \succeq C(x) \end{cases} \quad (4)$$

where x , u , and y respectively correspond to the vectors of daters associated with internal, input, and output events and matrices A , B , and C respectively belong to $\mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$, $\mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times m}$, and $\mathcal{M}_{in}^{ax}[\gamma, \delta]^{p \times n}$. Furthermore, as daters can be represented by elements in the dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, the vectors of daters x , u , and y appearing in (4) can be replaced by vectors with entries in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. This leads to the following matrix inequalities in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

$$\begin{cases} x \succeq Ax \oplus Bu \\ y \succeq Cx \end{cases} \quad (5)$$

Under the earliest functioning rule, $y = Cx$ and, using Theorem 1, $x = A^*Bu$. This leads to a transfer function matrix $H = CA^*B$. Hence, the output y induced by input u is given by $y = Hu$.

Example 8 The synchronizations in the considered example are represented by the following matrix inequalities in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$.

$$\begin{cases} x \succeq \begin{pmatrix} \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \delta^4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \gamma & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & e & \varepsilon & \gamma \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon \end{pmatrix} x \oplus \begin{pmatrix} e & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & e \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} u \\ y \succeq (\varepsilon \ \varepsilon \ \varepsilon \ \varepsilon \ \varepsilon \ e)x \end{cases}$$

Hence, using [12], the transfer function matrix H is given by

$$H = (\delta^6 (\gamma\delta^4)^* \delta^4 (\gamma\delta^2)^* \delta^6 (\gamma\delta^4)^*)$$

As before, let us consider the input corresponding to a supply of five workpieces of type 1 and type 2 at time 0 and to an order of five workpieces at time 0. The associated operators in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ are

$$u_1 = u_2 = u_3 = e \oplus \gamma^5 \delta^{+\infty}$$

The induced output is given by

$$y = \delta^6 \oplus \gamma\delta^{10} \oplus \gamma^2\delta^{14} \oplus \gamma^3\delta^{18} \oplus \gamma^4\delta^{22} \oplus \gamma^5\delta^{+\infty}$$

This result is of course coherent with the one obtained by modeling in the $(\max, +)$ -algebra.

4 Control

In this section, we focus on control methods modifying the internal dynamics of the system by adding a (max, +)-linear prefilter P (see Fig. 5a) or a (max, +)-linear output feedback F (see Fig. 5b). As in standard control theory, a prefilter is a dynamical system that processes an external input v as, e.g., a reference signal, and provides a suitable input $u = Pv$ to the system to be controlled. The notion of output feedback refers to a scenario where the system output y is fed back via a dynamical system F to generate the input $u = Fy \oplus v$ to the system to be controlled. Both control structures aim at modifying the given system dynamics to make it react in an appropriate way to *any* external input. In a manufacturing context, where external inputs are often non-controllable (e.g., orders from customers or parts delivered by suppliers), this is clearly an appropriate strategy. Note that other control methods such as optimal feedforward control [5] and model predictive control [9] are available to directly manipulate the inputs when this is possible.

The main purpose of the control approach discussed in this section is to reduce the size of internal buffers (and the number of workpieces in the production process at a given time instant) by adequately delaying the occurrences of input events. This effect can be easily quantified using second order theory for (max, +)-linear systems [13] (i.e., least upper bounds for the number of tokens in places are computed). However, the main drawback of this control approach is a possible slowing down of the system. Hence, choosing a prefilter or a feedback amounts to finding a trade-off between rapidity of the system and sizes of the internal buffers. In the following, we review some techniques to address this trade-off. The principle is to reduce as much as possible the internal buffers while satisfying some requirements on the rapidity of the system. Two typical requirements are: preservation of the transfer function matrix or preservation of the throughput.

Example 9 In the considered example, the internal buffers B_1 between machine M_1 and machine M_3 and B_2 between machine M_2 and machine M_3 are of interest. In the uncontrolled case, $u = v$. In this case, the sizes of the buffers B_1 and B_2 are both equal to $+\infty$, as the number of tokens between the transitions labelled x_2 (resp. x_4) and x_5 in Fig. 3 is unbounded. On the other hand, not controlling the system lets the system evolve maximally fast, as no synchronizations are added by a prefilter P or an output feedback F . Clearly, in practice, buffers always have restricted size, and it is therefore vital to introduce control.

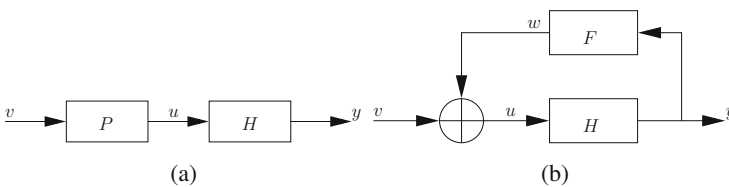


Fig. 5 Different control architectures

4.1 Model Reference Control

In model reference control [6–8], the requirement with respect to the rapidity of the system is expressed by a reference model G . The transfer function matrix of the controlled system, denoted H_c , must satisfy the condition $H_c \preceq G$. Hence, the reference model G is an upper bound for the transfer function matrix of the controlled system: the dynamics of the controlled system is required to be at least as fast as the one specified by the reference model G . In the following, model reference control is only considered for the case $G = H$ (i.e., the controlled system must be at least as fast as the uncontrolled one or, in other words, control is not allowed to “slow down” the output of the system). However, under some assumptions, the following discussion can be generalized to any reference model G . Next, model reference control by using either a prefilter or an output feedback is investigated.

4.1.1 Prefilter

Applying a prefilter P leads to the transfer function matrix HP for the controlled system. Hence, a prefilter P such that $HP \preceq H$ or, equivalently, such that $P \preceq H \setminus H$ is valid for model reference control. Under this restriction, we want to delay as much as possible the occurrences of input events, i.e., select the optimal (i.e., greatest) prefilter P such that $P \preceq H \setminus H$. Therefore, $H \setminus H$ seems to be the optimal prefilter. However, it is not always possible to implement this prefilter, as it may be non-causal (i.e., at time t this prefilter may need information available at time $t + 1$ or later). This problem is solved by using a specific mapping called causal projection and denoted Pr_+ (see [10, 14] for a formal discussion on the causal projection). Hence, the optimal prefilter, denoted P_H , is given by

$$P_H = \text{Pr}_+(H \setminus H)$$

By construction, $P_H \succeq e$ and $HP_H \preceq H$. Hence, $HP_H = H$. Thus, the prefilter P_H does not modify the transfer function matrix of the system.

Example 10 The prefilter P_H associated with the considered example is given by

$$P_H = \text{Pr}_+(H \setminus H) = \begin{pmatrix} (\gamma\delta^4)^* & \varepsilon & (\gamma\delta^4)^* \\ \delta^2 (\gamma\delta^4)^* & (\gamma\delta^2)^* & \delta^2 (\gamma\delta^4)^* \\ (\gamma\delta^4)^* & \varepsilon & (\gamma\delta^4)^* \end{pmatrix}$$

As expected, the prefilter P_H does not modify the transfer function matrix of the system:

$$HP_H = H = (\delta^6 (\gamma\delta^4)^* \delta^4 (\gamma\delta^2)^* \delta^6 (\gamma\delta^4)^*)$$

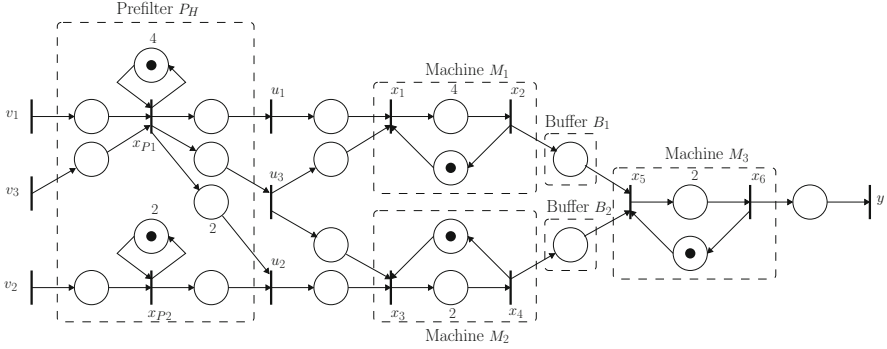


Fig. 6 Model reference control with prefilter

A state-space system realizing the transfer function matrix P_H is:

$$\begin{cases} x_P = \begin{pmatrix} \gamma\delta^4 & \varepsilon \\ \varepsilon & \gamma\delta^2 \end{pmatrix} x_P \oplus \begin{pmatrix} e & \varepsilon & e \\ \varepsilon & e & \varepsilon \end{pmatrix} v \\ u = \begin{pmatrix} e & \varepsilon \\ \delta^2 & e \\ e & \varepsilon \end{pmatrix} x_P \end{cases}$$

An implementation of this system in terms of a TEG is shown in Fig. 6. In the controlled system, the size of the internal buffer B_2 is equal to 0: as soon as a workpiece of type 4 is produced by machine M_2 , this workpiece is immediately used by machine M_3 . However, it can be easily seen that the size of the internal buffer B_1 is still equal to $+\infty$. Hence, in this example, using a prefilter that does not modify the system transfer function matrix will not allow to upper-bound all internal buffers.

4.1.2 Output Feedback

To understand the need for feedback, we have to consider perturbations in the model. In the following, we only consider additive state perturbations. This leads to a modified version of the model in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:

$$\begin{cases} x \geq Ax \oplus Bu \oplus q \\ y \geq Cx \end{cases} \quad (6)$$

where vector $q \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^n$ represents state perturbations. Note that, for manufacturing systems, additive state perturbations are sufficient to model a large class of uncertainties and failures such as machine breakdowns or changes in processing

times of machines. Considering perturbations leads to an additional transfer function matrix from q to y . Indeed,

$$y = Hu \oplus CA^*q$$

Perturbations do also affect the sizes of internal buffers. In many cases, the existence of perturbations strongly reduces the advantages induced by prefilters, as, by construction, prefilters cannot take into account perturbations.

Example 11 Taking into account perturbations annihilates the gain induced by the optimal prefilter P_H in the considered example. With the optimal prefilter P_H , the sizes of internal buffers B_1 and B_2 remain equal to $+\infty$ when perturbations are considered. Indeed, a breakdown of machine M_3 , such as machine M_3 is broken from the start (i.e., $q_4 = \delta^{+\infty}$ and $q_i = \varepsilon$ for $i \neq 4$), could lead to an infinite accumulation of workpieces in buffers B_1 and B_2 .

The previous discussion illustrates the need for control structures taking into account perturbations. In the following, we focus on output feedback, i.e., $u = Fy \oplus v$. The transfer function matrix of the controlled system is obtained as follows.

$$\begin{aligned} y &= Hu \oplus CA^*q \\ &= HFy \oplus Hv \oplus CA^*q \\ &= (HF)^*Hv \oplus (HF)^*CA^*q \end{aligned}$$

where the last equality follows from Theorem 1. Hence, if we choose the reference model $G = H$, i.e., we require feedback to not slow down the output of the system, we seek a feedback F such that $(HF)^*H \preceq H$. To delay the occurrences of input events as much as possible, we select the greatest causal feedback F such that $(HF)^*H \preceq H$. This feedback, denoted F_H , is given by

$$F_H = \text{Pr}_+(H \setminus H \phi H)$$

For the proof, the reader is invited to consult [6, 14]. As $(HF_H)^* \geq e$, $(HF_H)^*H \geq H$. Furthermore, by construction, $(HF_H)^*H \preceq H$. Hence, $(HF_H)^*H = H$. Thus, the feedback F_H does not modify the transfer function matrix of the system.

Example 12 The feedback F_H associated with the considered example is given by

$$F_H = \text{Pr}_+(H \setminus H \phi H) = \begin{pmatrix} \varepsilon \\ \gamma^2 (\gamma \delta^2)^* \\ \varepsilon \end{pmatrix}$$

As expected, the feedback F_H does not modify the transfer function matrix of the system:

$$(HF_H)^*H = H = (\delta^6 (\gamma \delta^4)^* \delta^4 (\gamma \delta^2)^* \delta^6 (\gamma \delta^4)^*)$$

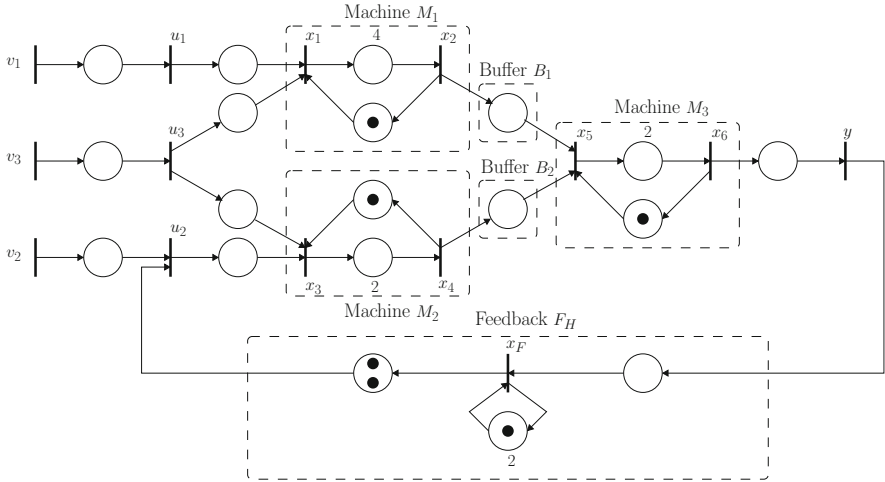


Fig. 7 Model reference control with output feedback

A state-space system realizing the transfer function matrix F_H is given by:

$$\begin{cases} x_F = \gamma \delta^2 x_F \oplus y \\ w = \begin{pmatrix} \varepsilon \\ \gamma^2 \\ \varepsilon \end{pmatrix} x_P \end{cases}$$

An implementation of this system in terms of a TEG is shown in Fig. 7. The size of the internal buffer B_2 is now equal to 2, whereas in the uncontrolled case it was equal to $+\infty$, i.e., by using an output feedback, we indeed succeed in reducing the size of internal buffer B_2 . However, the size of the internal buffer B_1 is still equal to $+\infty$. Hence, for this example, using an output feedback that does not modify the system transfer function matrix will not allow to upper-bound all internal buffers. In other words, the specification of not altering the system transfer function matrix is too strict. For this reason, we will now describe control for a less restrictive control specification.

4.2 Preserving the Throughput

The aim of this approach is to preserve the throughput (i.e., the maximal average production rate) of the system. Clearly, preserving the transfer function matrix, as done in model reference control, implies preserving the throughput. Hence, the latter is less restrictive (in terms of requirements on the rapidity of the system) than the former, and we expect smaller internal buffers, if all events are delayed as much

as possible subject to the respective requirement. In general, the optimal control preserving the throughput will slow down the system in the sense of providing a greater transfer function matrix. In the literature, this approach has only been investigated for feedback [15, 16]. As shown in [4], the greatest output feedback preserving the throughput leads to internal buffers of finite size.

Example 13 The throughput associated with the considered example amounts to one workpiece every four units of time. The greatest feedback F_σ preserving the throughput is

$$F_\sigma = \begin{pmatrix} \gamma^2 \delta^2 (\gamma \delta^4)^* \\ \gamma (\gamma \delta^4)^* \\ \gamma^2 \delta^2 (\gamma \delta^4)^* \end{pmatrix}$$

The resulting closed-loop transfer function matrix is

$$(HF_\sigma)^* H = (\delta^6 (\gamma \delta^4)^* \delta^4 (\gamma \delta^4)^* \delta^6 (\gamma \delta^4)^*)$$

while the open-loop transfer function matrix is

$$H = (\delta^6 (\gamma \delta^4)^* \delta^4 (\gamma \delta^2)^* \delta^6 (\gamma \delta^4)^*)$$

Hence, the transfer function matrix of the controlled system is strictly greater than the one of the uncontrolled one, i.e., the controlled system is slower than the uncontrolled one. However, as expected, the throughput of the controlled system and of the uncontrolled system are both equal to one workpiece every four units of time.

A state-space system realizing the transfer function matrix F_σ is given by:

$$\begin{cases} x_F = \gamma \delta^4 x_F \oplus y \\ w = \begin{pmatrix} \gamma^2 \delta^2 \\ \gamma \\ \gamma^2 \delta^2 \end{pmatrix} x_P \end{cases}$$

An implementation of this system in terms of a TEG is shown in Fig. 8. The size of internal buffer B_1 is equal to two, and the size of the buffer B_2 is equal to one. Hence, by appropriately slowing down the system, the suggested feedback has indeed succeeded in strongly reducing internal buffers B_1 and B_2 (in the uncontrolled case, the sizes of internal buffers B_1 and B_2 are both equal to $+\infty$). A behavior affected by the suggested feedback is provided by the input

$$v_1 = v_3 = e \oplus \gamma^5 \delta^{+\infty} \text{ and } v_2 = \delta^{20} \oplus \gamma^5 \delta^{+\infty}$$

This corresponds to an order of five workpieces and an arrival of five workpieces of type 1 at time $t = 0$, and an arrival of five workpieces of type 2 at time $t = 20$. In

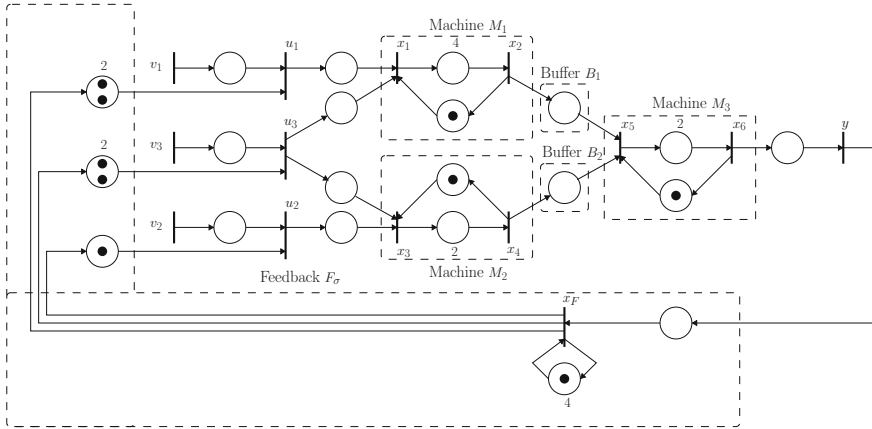


Fig. 8 Output feedback preserving the throughput

the uncontrolled system, workpieces of type 5 are delivered at time 24, 26, 28, 30, and 32. With feedback F_σ , workpieces of type 5 are delivered at time 24, 28, 32, 36, and 40. Hence, the feedback F_σ slowed down the system

5 Conclusion

In this chapter, we have explained how to use $(\max, +)$ -linear systems to model manufacturing problems characterized by synchronizations (i.e., conditions of the form: for all $k \geq l$, occurrence k of event e_2 is at least τ units of time after occurrence $k - l$ of event e_1). Furthermore, we have also presented some methods to address the trade-off between rapidity of the system and sizes of internal buffers. In particular, we have discussed two techniques preserving the transfer function matrix (i.e., the input-output behavior) and preserving the throughput (i.e., the maximal average production rate). Many other techniques have been investigated, e.g., preserving the response to a specific input [17] or preserving both the input-output behavior and the perturbation-output behavior [18].

References

1. Cassandras, C.G., Lafortune, S.: Introduction to Discrete Event Systems. Springer, New York (2006)
2. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley Longman Publishing, Boston (2006)
3. Murata, T.: Petri nets: properties, analysis and applications. Proc. IEEE 77, 541–580 (1989)

4. Baccelli, F., Cohen, G., Olsder, G.J., Quadrat, J.-P.: Synchronization and Linearity: An Algebra for Discrete Event Systems. Wiley, New York (1992)
5. Menguy, E., Boimond, J.-L., Hardouin, L., Ferrier J.-L.: Just-in-time control of timed event graphs: update of reference input, presence of uncontrollable input. *IEEE Trans. Autom. Control* **45**, 2155–2159 (2000)
6. Cottenceau, B., Hardouin, L., Boimond, J.-L., Ferrier, J.-L.: Model reference control for timed event graphs in dioids. *Automatica* **37**, 1451–1458 (2001)
7. Maia, C.A., Hardouin, L., Santos-Mendes, R., Cottenceau, B.: Optimal closed-loop control of timed event graphs in dioids. *IEEE Trans. Autom. Control* **48**, 2284–2287 (2003)
8. Hardouin, L.: Sur la Commande Linéaire des Systèmes à Événements Discrets dans l'Algèbre $(\max, +)$. Habilitation à Diriger des Recherches, Université d'Angers (2004)
9. De Schutter, B., van den Boom, T.J.J.: Model predictive control for max-plus-linear discrete event systems. *Automatica* **37**, 1049–1056 (2001)
10. Brunsch, T., Raisch, J.: Modeling and control of high-throughput screening systems in a max-plus algebraic setting. *Eng. Appl. Artif. Intell.* **25**, 720–727 (2012)
11. Blyth, T.S.: Lattices and Ordered Algebraic Structures. Springer, London (2005)
12. Hardouin, L., Cottenceau, B., Lhommeau, M.: Software tools for manipulating periodic series. In: Proceedings of the 6th International Workshop on Discrete Event Systems (2002). Available at www.istia.univ-angers.fr/~hardouin/outils.html
13. Cohen, G., Gaubert, S., Nikoukhah, R., Quadrat, J.-P.: Second order theory of min-linear systems and its application to discrete event systems. In: Proceedings of the 30th IEEE Conference on Decision and Control, vol. 2, pp. 1511–1516 (1991)
14. Cottenceau, B., Hardouin, L., Boimond, J.-L., Ferrier, J.-L.: Synthesis of greatest linear feedback for timed event graphs in dioid. *IEEE Trans. Autom. Control* **44**, 1258–1262 (1999)
15. Gaubert, S.: Resource optimization and min-plus spectral theory. *IEEE Trans. Autom. Control* **40**, 1931–1934 (1995)
16. Cottenceau, B., Lhommeau, M., Hardouin, L., Boimond, J.-L.: On timed event graphs stabilization by output feedback in dioid. *Kybernetika* **39**, 165–176 (2003)
17. David-Henriet, X., Brunsch, T., Raisch, J., Hardouin, L.: Stock reduction for timed event graphs based on output feedback. In: Proceedings of the 11th International Workshop on Discrete Event Systems (2012)
18. Lhommeau, M., Hardouin, L., Cottenceau, B.: Disturbance decoupling of timed event graphs by output feedback controller. In: Proceedings of the 6th International Workshop on Discrete Event Systems (2002)

Stochastic Optimal Sizing of a Warehouse

Luca Ghezzi

Abstract The problem is considered of determining how many pieces to stock in a warehouse, for a multitude of stockable goods and accounting for random market demand and supply lead time. Classical reorder point theory is revisited, the underlying model is no longer linearized and the involved stochastic variables need not be normally distributed but, rather, are empirically deduced from historical data. Uncertainty propagation is carried out either by Monte Carlo method or by a Polynomial Chaos Expansion. A Quadratic Programming procedure is proposed to regularize data by filtering rare events out. Performance vs. cost curves are obtained and the global problem of choosing optimal points over them, subject to a global cost budget constraint, is set as a combinatorial, constrained optimization. The solution of a simplified version is attained by Linear Programming, while the full problem is addressed by Mixed Integer Linear Programming.

1 Introduction

Storage of physical goods is a practically unavoidable need in industry and commerce, common to most market sectors. As well-known, inventories allow decoupling raw materials and sub-components supply chain from production, as well as the latter from the commercial distribution of final products. Moreover, large factories employ internal sub-component warehouses (i.e., the *make-to-stock* manufacturing policy) in order to ease industrial operations by decoupling successive, serial phases, as well as to allow, if applicable, the *assemble-to-order* manufacturing policy at the very last segment of the productive chain. Similarly, commercial as well as service organizations are usually hierarchically structured, with tiers decoupled by warehouses. On the one side, stocked volumes allow for strategic, massive purchase deals, simpler factory management policies as well as high service levels in deliveries to final customers. Drawbacks include the high cost of stocked capital, the risk for stocked product obsolescence, as well as the operative cost for running the inventory.

L. Ghezzi (✉)

ABB Italy S.p.A., Mathematical Modeling & Numerical Simulations, Vittuone, MI, Italy

e-mail: luca.ghezzi@it.abb.com

The impact of inventories over operations or overall company's costs is clearly business-dependent; nonetheless, strong cost reduction initiatives pertaining to suitably sizing the inventories have been observed in last decades. A first trend is toward reducing stocks and increasing their rotations. A second trend is forecasting expected needs, to be scheduled so to have materials available only when necessary, and using stocks to cover deviations from forecasts. A third trend is shifting inventories upstream, possibly to suppliers (a fact clearly not without impact when negotiating supply conditions). A complex and multi-disciplinary scenario emerges, encompassing sales, planning, production, and logistic, where the functional relationship between warehouse benefits and costs is emphasized in order to quantitatively evaluate alternatives and draw conclusions. Clearly, such a central and transversal topic in factory dynamics may strongly benefit from the use of Mathematics to model its behavior and allow its suitable dimensioning.

1.1 Goals and Results

Modeling and predictive approaches thus find a sound economic motivation. Particularly, it is here of interest to address the problems of:

- (A) Devising a performance vs. cost relationship for any stockable good, depending on the quantity being stocked;
- (B) Determining the optimal quantities to stock for all stockable goods, given an overall budget constraint and an overall weighted average performance indicator, possibly lower bounded (dubbed *optimal budget expenditure problem* in what follows).

The kind of results obtained by solving problem (A) is exemplified by Fig. 6, showing the performance vs. cost curve for a sample stockable good, as obtained by the proposed approach and opposed to the one deduced by traditional approaches. The kind of results obtained by solving problem (B) is reported in Fig. 8, showing the fundamental Pareto front for the inventory, that is, the overall performance vs. cost curve. Thanks to the latter, the decision maker may define the global working point. Mathematics connect the latter to relevant working points for all stockable goods, that is, the positioning along individual performance vs. cost curves like Fig. 6. The combination of the two problems is relevant to global warehouse design as well as local warehouse revisions, and needs being solved periodically (say, quarterly) to keep the warehouse up to date with the current market scenario.

Inventories are a crucial ingredient of the “digital factory”, because only their appropriate sizing allows a profitable and timely production flow. This means that, on the one hand, oversized stocks are completely inadmissible in a modern, lean production system and, on the other side, the large number of products, most of which with low volumes, make the traditional, deterministic or oversimplified stochastic approaches no longer adequate. It is precisely the stochastic nature of the problem which demands for a finer mathematical treatment than traditional variance propagation approaches.

1.2 A Historical Perspective

The basic theory dates back to 1913, with the first simple models to connect production to inventories; see [8]. The *reorder point* (ROP) is a rule-based inventory control method consisting of issuing a supply reorder up to the feeder factory (more generally, upstream) any time the stock level falls below a predetermined threshold, the ROP itself. The threshold must be high enough to prevent from stock-out while the supply order is being worked out and products are asked for from the market (more generally, downstream). Assuming constant and predictable market demand and supply lead time, a deterministic contribution is first found. Then, a *safety stock* margin is supplemented, to cope with stochastic factors. The safety stock is computed by means of uncertainty propagation: the ROP formula is linearized, variance is propagated over the linear approximation and finally a normal distribution is assumed to introduce a confidence level in safety assessment. Such procedure ultimately leads to Hadley-Whitin formula (1963), very popular among logistic professionals; see [7]. The higher the quantity being ordered, the longer the time in between two consecutive reorders, the lower the number of reorders to be issued in a given time horizon, and the higher the average stock level, market demand and all other features being the same. Accounting for fixed reorder costs (monotonically decreasing with the reordered quantity) and stock costs (monotonically increasing with the reordered quantity), the *Economic Order Quantity* (EOQ) is determined as the global cost minimizer. As an alternative to ROP-EOQ, the *reorder time* (ROT) method compares the stock level with a reference level every fixed period of time, issuing an upstream supply order to fill the possible gap.

Most of later literature is about inventory dynamics and control. Standard approaches are known for inventory management, addressing the problem of when to issue replenishing orders and how large such orders need to be, for a warehouse with prescribed sizing; see, e.g., [18] and references therein for a comprehensive exposition of the state of the art.

Most notably, in 1964 Orlicky introduced *Material Requirement Planning* (MRP) as an inventory management strategy, coordinated with production planning, that quickly became an indispensable tool in industry; see [13]. With reference to purchase orders received from the market and estimated, and the level of all involved inventories, the MRP takes care of scheduling production, back-propagating suitable production or purchase orders upstream. O. Wright introduced in 1983 the *Manufacturing Resource Planning* (MRP II), significantly improving the MRP by, among other things, accounting for rough-cut production capacity; see [17]. Latest trends attempt to include market demand prediction by time series modeling. See [14] for a modern, retrospective review, along with the significant improvements and advanced features added in recent times and collected under the denomination of *Demand Driven MRP* (DDMRP).

Since the 1980s, the *pull* logic (i.e., supply orders are issued downstream to upstream, only when supply is needed) has been widely adopted, whenever

applicable for the specific business. In 1981, Kimura et al. modeled a celebrated, self-regulated implementation with the so-called *Just in Time* (JIT) paradigm; see [9]. Particularly, very small production lots are handled at all productive levels, with production order cards (termed *kanban*, in Japanese language) traveling upstream, whenever the representative lot is exhausted, and back, together with a new lot to be stocked. Frequent and small supply actions, resembling a continuous flux, characterize this scenario. Still, the need arises to determine the required stock levels, that is, equivalently, the number of lots, or of *kanban* cards.

MRP, or DDMRP, or also JIT, allow remarkably reducing the stock levels needed to run the business. As a limit condition, goods should simply transit in the suitable quantities and at the suitable times throughout logistic hubs (for the sake of transport and distribution chain optimization), and without being stocked for longer times than those strictly needed for their handling. Anyway, some residual inventory quota is still needed, along with classic theories like ROP or ROT modeling their dynamics. This is basically to ensure a short term market coverage, so to cope with: Discrepancies between forecast and current demand; Client orders' withdrawal, or postponement, or expediting; Deviations from planned delivery dates along the productive chain (late, missed or erroneous deliveries, unforeseen production downs or transport problems, and the like); Fluctuations in production capacity, discrepancy between reference (usually estimated considering runner codes) and actual (on a code-by-code basis) production capacity, when rough-cut over infinite capacity is carried out; Discrepancy between estimated and actual defective product rates, and other possible mishaps.

Dynamics based approaches could be useful to address problems (A) and (B), by checking a posteriori, typically by means of *Discrete Event Simulation* (DES), the adequacy of a given, prescribed inventory sizing and then moving towards a (usually pseudo-) optimal sizing or, more frequently, simply towards a better-than-actual sizing by means of: Either scenario analysis (that is, by partial enumeration of possible alternatives); Or some improving strategy, consisting of either down-hill moves (derivative free optimization, surrogate modeling, etc.) or some heuristics (genetic algorithms, particle swarms, etc.). Drawbacks of such approaches include being time consuming and being strongly dependent on a multitude of system describing input data of difficult availability and sometimes questionable reliability. Moreover, accounting for the stochastic nature of the problem is possible but emphasizes the above mentioned limits; see, e.g., [11] for a comparison between DES and classic inventory modeling.

In order to avoid the mentioned drawbacks, we propose, in problem (A), to prescind from simulating the inventory evolution by means of some time-marching scheme but, rather, to focus onto the stochastic variable concept. Any artificial requirement of normally distributed probabilities is dropped. As an output, we get the performance vs. cost curves for all stockable goods. Secondly, by means of *Mathematical Programming* (MP) approaches, we deal with problem (B) as the combinatorial, constrained, optimization problem of finding the optimal positioning along the above mentioned curves for all stockable goods, possibly including the Boolean decision variable whether or not to stock some goods.

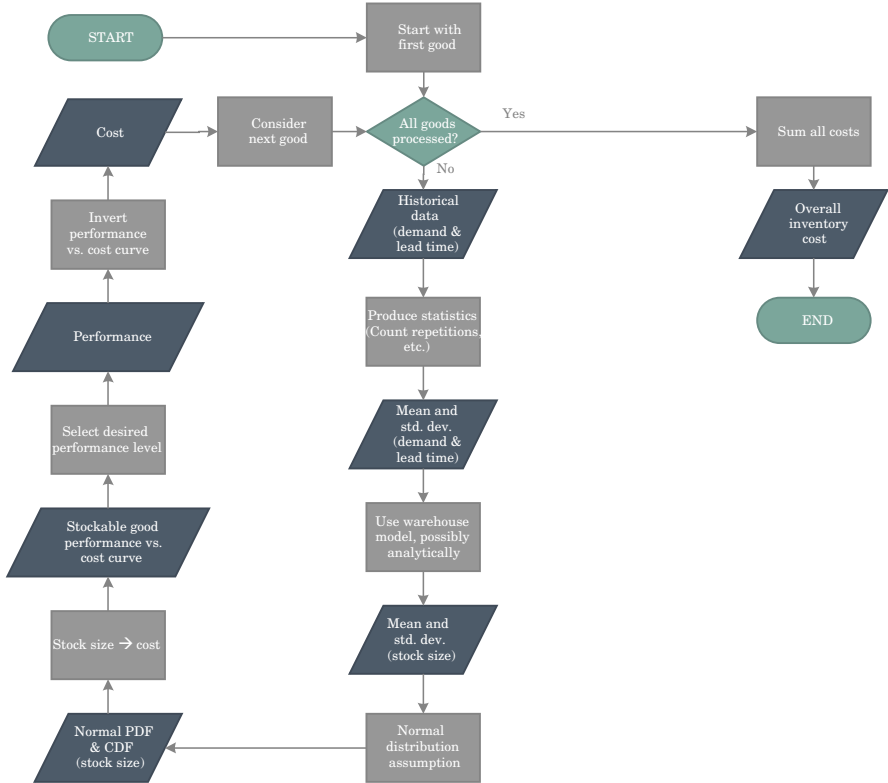


Fig. 1 Traditional warehouse sizing approach workflow

1.3 Traditional vs. Proposed Approach and Workflow

For the sake of a better comprehension, we provide a comparison between the traditional (see Fig. 1) and the proposed approach (see Fig. 2) to stock sizing. In both approaches, for each stockable good the required input data are the historical market demand and supply lead time time-series; see Sect. 2.2.

In the traditional approach to stock sizing, historical data are processed according to well-known statistical techniques, in order to obtain their mean value and standard deviation; see Sect. 2.2. Based on any applicable warehouse model (see, e.g., Sect. 2.1, or consider other models), the mean value and standard deviation for the required stock size are deduced, possibly analytically and usually with an underlying, implicit model linearization; see Sect. 2.3. Then, it is implicitly assumed that the required stock size be normally distributed, which needs not being the case, so to have available a probability density function (PDF), as well as a cumulated probability function (CDF). The latter represents the relationship between the stock size and the probability not to stock out. By converting the stock size into the

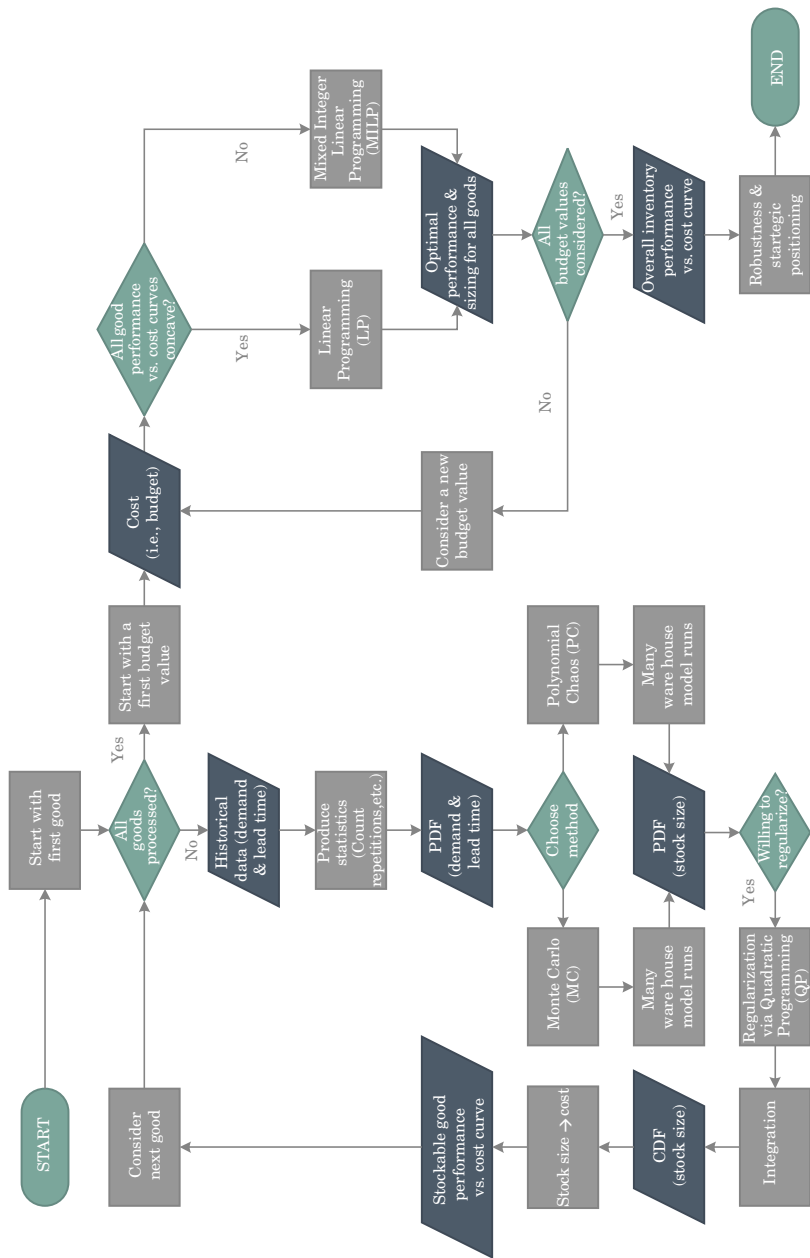


Fig. 2 Proposed warehouse sizing approach workflow

corresponding cost, a performance vs. cost curve is obtained for the stockable good at hand; see Sect. 4.1. Then, after a prescribed performance level has been selected, either arbitrarily or according to some suitable argument, from inverting the performance vs. cost curve the corresponding cost is found. Finally, the global inventory cost is obtained from the summation over all stockable goods.

The normality assumption is proven false by the finer approach proposed hereafter. This introduces both a theoretical and practical bias in the performance vs. cost curves, as well as in the computations following downstream. Moreover, there is no guarantee, a priori, that the global inventory cost be less than the available budget. In case not, the performance levels need being lowered, usually according to some trial-and-error scheme. Similarly, no guarantee is provided for optimal budget expenditure, meaning that if the global inventory cost is lower than the available budget, then the prescribed performance levels have to be increased (and also in this case there is no direct indication on how much) so to spend the additional available resources.

In the proposed approach to stock sizing, historical data are processed according to well-known statistical techniques, in order to obtain their probability density function (PDF), not just their mean value and standard deviation; see Sect. 2.2. Then, either with Monte Carlo (MC) method (see Sect. 2.4) or with Polynomial Chaos (PC) expansion method (see Sect. 2.5), the probability density function for the required stock size is deduced numerically, in both cases by means of a number of runs of a suitable warehouse model (see, e.g., Sect. 2.1, or consider other models). Then, a Quadratic Programming (QP) based, mathematical regularization technique is strongly advised, in order to clean the PDF from rare events and outliers; see Sect. 3.2. By PDF integration, the relevant CDF is obtained for the stock size. The regularization attempts to (and usually manages to) produce a concave CDF. The latter represents the relationship between the stock size and the probability not to stock out. By converting the stock size into the corresponding cost, a performance vs. cost curve is obtained for the stockable good at hand; see Sect. 4.1.

We stress that, contrarily to the traditional approach, the performance vs. cost curve is not the outcome of a forced, and usually false, normality distribution assumption, neither for input data (market demand and supply lead time) nor for the output data (required stock size). Rather, all PDFs and CDFs are now either empirically deduced or produced throughout a mathematical model that attempts to describe the real course of events. The finer the model, the more reliable the probability distributions. The fallacies introduced by the traditional approach may be observed in Fig. 6, in the case of the same warehouse model (ROP).

Also the problem of optimal resource allocation, that is, optimal budget expenditure, is now addressed in a radically different and mathematically structured manner. After all stockable goods have been dealt with as above, a first global inventory budget is considered, for instance the actually available one. Depending on whether all stockable good performance vs. cost curves be concave or not, the optimal budget expenditure problem is addressed by either Linear Programming (LP; see Sect. 4.2) or Mixed Integer Linear Programming (MILP; see Sect. 4.3), respectively.

In both cases, the optimal cost and service level for each stockable goods is obtained, along with the overall weighted inventory performance and overall cost. Differently from the traditional approach, now the overall inventory cost is guaranteed to exactly equal the available budget (in case of feasible problems, for, otherwise, side constraints need being relaxed, as shown in Sect. 4).

A yet further advantage of the proposed approach is that the optimal budget expenditure problem is suggested to be solved for other, different budget values, in such a way to deduce, pointwise, an overall performance vs. cost curve for the whole inventory. This allows gaining a measure of the robustness of the problem (see whether or not little budget perturbations result into little performance perturbations). Robustness analysis then plays a key role in the strategic choice of global inventory budget allocation, because the collocation in a sharp bending point of the performance vs. cost curve may be very alluring.

It is important to notice that the particular choice of the warehouse model is largely immaterial for the proposed approach to stock sizing, meaning that the latter is general and may embed any other model with the same output. Also the case of different input data (number and kind) may be handled with straightforward modifications of inessential aspects. We mentioned the two simplest models historically and commonly used in engineering practice, developed the exemplification with reference to the one currently adopted in the industrial ABB case at hand, and finally proposed a simple extension that may apply to both, in order to address the MRP management policy. This has been done both for the sake of simplicity and in order to more strictly adhere to the traditional approach in all inessential aspects.

1.4 Chapter Outline

The approach proposed in this chapter may be outlined as follows. In Sect. 2 we address problem (A). First, in Sect. 2.1, basic ROP theory is recalled and given a formal dress compatible with standard probability theory. An additional contribution is considered that accounts for time discrete inventory control or, which is equivalent, for large market requests. Then, in Sect. 2.2, input data probability distributions are estimated, mainly based on historical data. Three alternative approaches are discussed to propagate uncertainty through ROP model and deduce the probability not to stock-out, namely: The traditional variance propagation approach, in Sect. 2.3; *Monte Carlo* (MC) method, in Sect. 2.4; *Polynomial Chaos Expansion* (PCE, or PC), in Sect. 2.5. In Sect. 3, the output from problem (A), that is, performance vs. cost curves, are improved before being used as input to problem (B). A suitable, *Quadratic Programming* (QP) based regularization technique is introduced, aimed at enforcing concavity in the performance vs. cost curves portion of interest. In Sect. 4 we address problem (B) as a combinatorial, constrained optimization problem. Two alternative approaches are discussed, namely: A simplified, and computationally very slim, *Linear Programming* (LP)

based optimization, in Sect. 4.2; A more refined and versatile, but computationally more challenging, *Mixed Integer Linear Programming* (LP) based optimization, in Sect. 4.3. The application of the procedure to the industrial case of ABB, the results obtained, the possible extension of the very same machinery to other contexts of industrial operations, including but not limited to production management, as well as possibilities for further research are finally discussed, in Sect. 5.

2 The Local Stochastic Problem

In this section we deduce, for each single good, the stochastic relationship, i.e., a probability density function (PDF), relating the probability not to stock-out with the amount of items to be stocked; see Sect. 1, problem (A). Item amounts are large when mass produced goods are at hand, allowing for real numbers to describe what would rigorously require integer numbers.

2.1 Reorder Point and Other Models

A classical inventory management policy for *make-to-stock* (MTS) goods is the so-called *reorder point* method, modeling stock level dynamics according to a saw-tooth behavior; see Fig. 3. Starting from an arbitrary, high value, the stock level for a given item is progressively lowered by *market demand* D , until a *reorder point*

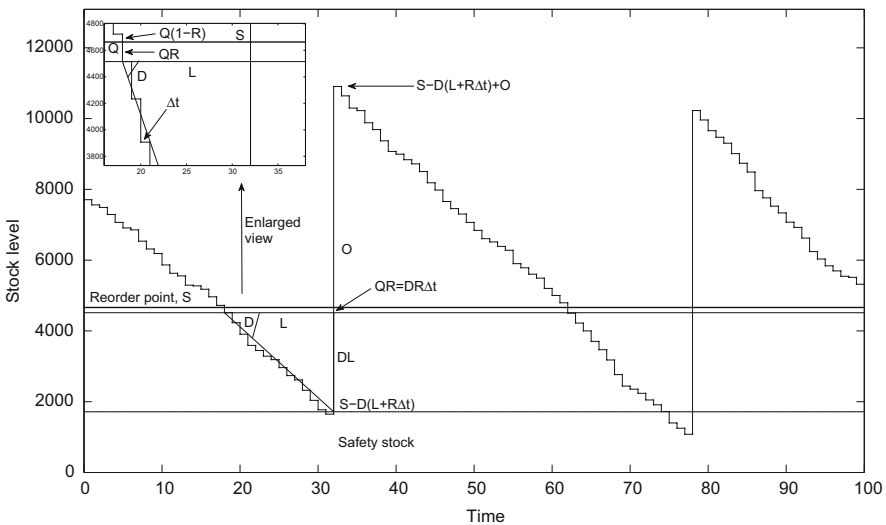


Fig. 3 Typical saw-tooth dynamics of a reorder point ruled inventory

threshold S is reached. A supply order is then issued upstream, and ordered goods eventually enter the warehouse after a *supply lead time* L during which time the stock level continues to go down below the reorder point, due to the last market requests received. Stock levels and reorders are dealt with at discrete time, i.e., they are spaced by a prescribed period of time Δt (a day, or an hour, etc., depending on the inventory type and dynamics). If Δt is non negligible compared to L , or equivalently, if market orders size may be large compared to S , then the last ordered quantity $Q = D\Delta t$ may result into a non negligible portion RQ below the safety stock, where $R \in [0, 1]$ is a proportionality coefficient. Elementary geometrical considerations (see Fig. 3, where the *safety stock* has to be initially neglected; as the name suggests, this additional quota is later introduced in order to raise the reorder point and cope with the stochastic nature of the problem) relate the reorder point to its contributors, reading

$$S = DL + RQ = D(L + R\Delta t). \quad (1)$$

In a deterministic frame, inventory dynamics would be known and repeatable, leading to the determination of the “exact”, sharp reorder point S directly by plugging sharp D , L and R values into (1). Actually, all involved quantities are beyond our full control and may be considered stochastic variables. Market demand may be associated (written $D \sim \phi^D(\xi^D)$) to a *probability density function* (PDF) $\phi^D(\xi^D)$, where ξ^D is any admissible value for D , while $\phi^D(\xi^D)$ is the probability density associated with such value. Let $\Phi^D(\xi^D) = \int_{-\infty}^{\xi^D} \phi(\zeta) d\zeta$ be the probability to draw values of D up to ξ^D , so that the relevant function is the *cumulated density function* (CDF). Similarly, $L \sim \phi^L(\xi^L)$, $R \sim \phi^R(\xi^R)$ and $S \sim \phi^S(\xi^S)$, with relevant CDFs. As a final result, the quantity S to be stocked is to be estimated as the value ξ^S such that

$$\Phi^S(\xi^S) = \int_{-\infty}^{\xi^S} \phi^S(\zeta) d\zeta = 1 - \alpha, \quad (2)$$

that is, such that the reorder point S will be sufficient to cope with inventory dynamics with confidence level $1 - \alpha$. Equivalently, the probability to stock-out will be no higher than $\alpha \in [0, 1]$ (typically, α is a number of the order of 5–10%, or even less). In what follows, a mathematical method is proposed to derive the actual expression of (2), in the case above as well as in the similar conditions introduced hereafter.

Generalizations or other, different models are also available. It is similarly deduced that, for goods that follow the ROT policy,

$$S = D(L + T + R\Delta t), \quad (3)$$

where T is the (prescribed) *reorder time*, i.e., the fixed period of time after which the current stock level is compared with the reference value S and a reorder quantity is possibly requested upstream, so to restore level S . Most goods are nowadays

handled by MRP. To estimate the residual safety inventory quota that is practically still needed, we propose to use the model

$$S = \max\{0, (D - E[D])(L + R\Delta t)\} \quad (4)$$

or

$$S = \max\{0, (D - E[D])(L + T + R\Delta t)\} \quad (5)$$

instead of (1) or (3), resp., with market demand netted of its expected value $E[D]$ (i.e., its mean value). Netting demand D with $E[D]$, rather than with a different quantity, is an arbitrary but reasonable choice in absence of more precise information and in order to express any deviation from the reference scenario, for which nothing more than average values is accessible in practice. Since negative reorder point values would make no sense, the max operator is used as a lower cut-off with zero. For the sake of simplicity, we shall develop the following with reference to (1). It remains understood that methods and conclusions are to be rewritten, with suitable modifications, with reference to (3), (4) or (5) in relevant cases.

2.2 Input Probability Densities

Stock sizing is generally accounted for in a stochastic frame, to a different extent of mathematical subtlety and reliability, as later discussed. In any case, knowledge is needed about input variables probability distributions. The estimation of PDFs is generally not easy and requires care. As for R , owing to the random time sequencing of market purchase orders as long as the reorder point threshold is randomly crossed, there is no conceptual reason to privilege some values compared to others and one may assume a uniform distribution for ϕ^R over the interval $[0, 1]$.

Supply lead time PDF ϕ^L could be estimated either by means of some suitable model describing its internal dynamics and policies (not covered here), or phenomenologically by counting the frequencies of supply within the terms of multiples of a suitably defined time unit like, e.g., days (or half days, or even less, depending on the reactivity of the supply structure). For instance, if the time unit is fixed to be the single day, then one may consider a sufficiently long historical series, covering, say, 6 months or a whole year, and then count how many times the supply lead time did not exceed 1 day, how many times it was in between 1 and 2 days, how many times in between 2 and 3 days, and the like.

In absence of more refined information, market demand PDF ϕ^D may be deduced from a rich enough historical data base of purchase orders received. A possible, proposed approach consists of considering three different time scales. First, we define a long enough period of time compared to inventory dynamics time constant like, e.g., 1 year or more (*long time scale*). It could be not advisable to consider longer durations, because markets are mutable and one wants to focus on present

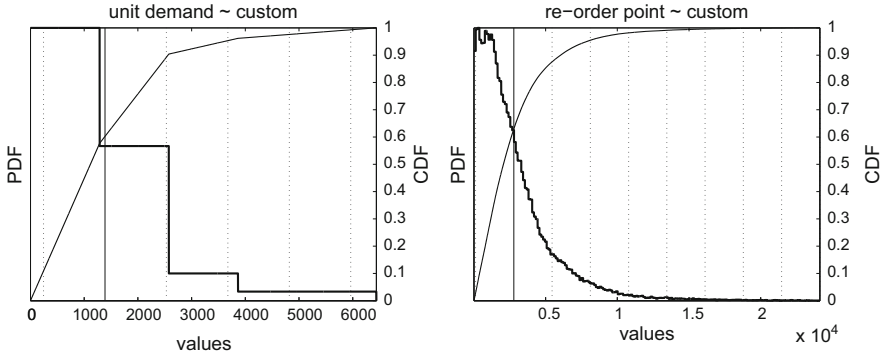


Fig. 4 Probability distributions for a sample product code in ABB inventory test case. *Left* Demand D , PDF ϕ^D (*bold*) and CDF Φ^D (*lite*), as empirically deduced from historical data records. *Right* Reorder point S , PDF ϕ^S (*bold*) and CDF Φ^S (*lite*), as computed by means of PC and fast MC on polynomial surrogate model. A remarkable difference is observed from the traditionally adopted Gaussian assumption: the actual PDF is single tailed and starts high-valued on the *left side*

trends. Second, the long period is subdivided into time buckets like, e.g., months, or weeks, or the like, so that their number is large enough to allow for a reasonable PDF resolution, as later discussed, and so that each is not too short to contain just a few orders (*middle time scale*). Finally, the third time scale is the time constant of the inventory dynamics like, e.g., the day or the like (*short time scale*). The latter can be the time scale with time constant Δt , and must be definitely smaller than the middle time scale. Market demand PDF ϕ^D requires cumulating all orders received over any single time bucket of the middle time scale and then counting the recurrence frequencies of ordered quantities, i.e., how many times the whole market required a quantity of pieces lying in any bin. See Fig. 4 (left) for the illustration of a sample product code in later discussed ABB inventory test case.

In all cases, histograms are obtained and PDFs are readily produced by normalization. Standard techniques must be applied in bin sizing, possibly considering variable and adaptive bin sizing, in order to guarantee statistical soundness. The approach above is admittedly poor, and more refined statistical methods could be conceived to enhance PDFs' quality. It must be anyway remembered that many other uncertainties and approximations affect the problem, so that extreme accuracy could be unmotivated and practically unattainable.

Large purchase orders, like, e.g., periodic and contract negotiated wholesalers' refills, need being detected and translated into the stemming multitude of smaller order actions spread over a long time horizon, as usually dealt with by logistic planning. Actually, since the latter actions are planned and thus known in advance, they should be handled in a deterministic way and removed from the stochastic sizing of the safety stock.

Some items may have such a rare and/or volatile demand, that the concept itself of probability density in the above sense would not properly hold. Such

and similar cases should not be considered in this modeling frame, nor stocked according to the logics of recurrent and abundant production and resell, i.e., they should not enter inventories. Rather, such special items should be more profitably supplied to the market according to suitably defined policies, and with terms and conditions negotiated with clients, like, e.g., *assemble-to-order* (ATO) or *make-to-order* (MTO).

2.3 Output Probabilities: Traditional Approach

The traditional approach for safety stock stochastic estimation is based on the linearization $S = (E[L] + \Delta t/2)D + E[D]L + E[D]R\Delta t - E[D](E[L] + \Delta t/2)$ of (1), obtained after straightforward simplifications from the relevant Taylor series expansion truncated to degree 1 terms and centered on point $(E[D], E[L], E[R] = 1/2)$, corresponding to the mean values of D , L and R . After the linearization, the mean value follows with a standard change of variables and reads

$$E[S] = \int_{\mathbb{R}} \xi^S \phi^S(\xi^S) d\xi^S = E[D](E[L] + \Delta t/2). \quad (6)$$

Variance follows similarly, yielding the well-known variance propagation formula that in the specific case reads

$$\begin{aligned} \sigma_S^2 &= \int_{\mathbb{R}} (\xi^S - E[S])^2 \phi^S(\xi^S) d\xi^S \\ &= (E[L] + \Delta t/2)^2 \sigma_D^2 + E[D]^2 \sigma_L^2 + (E[D]\Delta t)^2 \sigma_R^2, \end{aligned} \quad (7)$$

where D , L and R have been assumed to be mutually uncorrelated and σ_D , σ_L and $\sigma_R = 1/\sqrt{12}$, resp., are their standard deviations. When $\Delta t \rightarrow 0$ (i.e., Δt is small compared to L), the term $RD\Delta t$ is frequently neglected compared to the term DL in (1). Under such assumption, the mean value (6) reduces to $E[S] = E[D]E[L]$ while variance reduces to $\sigma_S^2 = E[L]^2 \sigma_D^2 + E[D]^2 \sigma_L^2$. This latter expression is to be compared with $\sigma_S^2 = E[L] \sigma_D^2 + E[D]^2 \sigma_L^2$, as deduced by Hadley-Whitin [7]. When supply from the feeder factory is governed by reliable policies such as prescribed and easily attainable delivery dates, then supply lead time is considered deterministic and known equal to a fixed value L (here intended as a deterministic variable, no longer as a stochastic variable), furtherly simplifying the mean to $E[S] = LE[D]$ and variance to $\sigma_S^2 = L^2 \sigma_D^2$, or to $\sigma_S^2 = L \sigma_D^2$ in the Hadley-Whitin form.

In the traditional approaches discussed above, the reorder point is estimated as

$$S = E[S] + c(\alpha)\sigma_S, \quad (8)$$

where $c(\alpha)$ is a suitable multiplier of standard deviation σ_S , corresponding to confidence level $1 - \alpha$. The $E[S]$ contribution may be considered as a sort of theoretical deterministic prediction, conservatively supplemented by the $c(\alpha)\sigma_S$ contribution that accounts for stochastic uncertainties. The latter contribution is frequently termed *safety stock*.

The actual probability distribution ϕ^S of S is never introduced nor used. If, on the one hand, this makes the mathematical developments extremely simple, on the other hand $c(\alpha)$ is questionably computed by implicitly assuming that S be normally distributed. There is no theoretical reason leading to Gaussian ϕ^S (actually, theory clearly tells this cannot be the case, because ϕ^S must be boundedly supported on the left, i.e., it cannot have a left tail, in order to exclude logically inconsistent negative reorder point values). Evidence from the application of the more reliable and logically consistent methods discussed in next sections clearly shows that S is not normally distributed, and that the actual PDF may be remarkably different; see also Fig. 6 for a visual comparison.

2.4 Output Probabilities: Monte Carlo

We are interested in adopting (2) instead of (8) in order to estimate the quantity to be stocked. Consequently, ϕ^S is required. One way is the well-known *Monte Carlo* (MC) method. MC basically amounts to repeatedly evaluating the deterministic model (1), with input values D, L, R randomly extracted from relevant PDFs. The correspondingly many random output S values are then used as samples to produce ϕ^S . On the one hand, MC method is conceptually simple, easy to implement and non-intrusive, in the sense that the deterministic model (1) is used as a black-box. An additional benefit, by far the most important, is that the convergence rate does not depend on the number of stochastic variables (which does *not* mean that the number of necessary runs does not depend on the number—and type—of stochastic variables).

On the other hand, the convergence rate is extremely slow, so that the number of deterministic model runs is usually very high, thus requiring long computational times, unless the deterministic model is extremely fast. Additionally, the successive moments of the sought for PDF converge progressively slower, let alone particular cases enjoying special properties, such as odd/even parity, and the like. As a consequence, the correct shape of ϕ^S may be very slow to converge.

2.5 Output Probabilities: Polynomial Chaos

An alternative way to obtain ϕ^S is based on the so-called (non-intrusive) *Polynomial Chaos* (PC) method, for which we need a digression into *orthogonal polynomials*. It is well-known that a general probability density distribution $\phi^x(\xi)$ associated with a

stochastic variable x satisfies the hypotheses of a measure. Thus it naturally induces the inner product $(\cdot, \cdot)_x : L^2(\mathbb{R}) \times L^2(\mathbb{R}) \rightarrow \mathbb{R}$ such that

$$(u(\xi), v(\xi))_x := \int_{\mathbb{R}} u(\xi)v(\xi)\phi^x(\xi) d\xi. \quad (9)$$

Scalar product (9) naturally induces the norm $\|\cdot\|_x : L^2(\mathbb{R}) \rightarrow \mathbb{R}$, such that $\|u(\xi)\|_x^2 = (u(\xi), u(\xi))_x$. Let $L^2(\mathbb{R}; \phi^x)$ be the set of bounded functions with reference to norm $\|\cdot\|_x$. One can use scalar product (9) in the frame of Gram-Schmidt process applied to the sequence $\{\xi^k\}_{k=0}^{+\infty} = \{1, \xi, \xi^2, \dots, \xi^k, \dots\}$. The outcome is an *orthogonal polynomial sequence* (OPS), that is, a sequence $\{\psi_k^x(\xi)\}_{k=0}^{+\infty}$ of polynomials which are mutually orthogonal with reference to (9). Therefore, by construction, $(\psi_j^x, \psi_k^x)_x = \|\psi_k^x\|_x^2 \delta_{j,k}$, $\forall j, k \in \mathbb{Z}_0$, where $\delta_{j,k}$ is Kronecker symbol. If desired, the OPS could be made *orthonormal*, to simplify notation and without modifying the essence of the problem. The above mentioned polynomials are readily seen to be linearly independent by a standard degree argument, leading to the conclusion that they form a basis for $L^2(\mathbb{R}; \phi^x)$; see, e.g., [3] for the easy details.

Superscript x in ψ_k^x explicitly shows the descendance of the OPS from stochastic variable x , to which it is linked and specific. It can be shown that some of the most common PDFs, continuous or discrete, induce celebrated OPSs; see, e.g., [10, p. 37]. For instance, uniform PDF ϕ^R induces correspondingly an OPS which coincides with the family of Legendre polynomials. Considering reorder point theory, three stochastic input variables are involved, each leading to its own OPS. Another OPS is obtained from the three above mentioned OPS by tensorization, as follows. A vector $\xi := (\xi^D, \xi^L, \xi^R)$ is defined to collect 3-ples of values from the three input variables. A multi-index $\mathbf{k} := (k^D, k^L, k^R)$ is introduced to collect the degrees of polynomials from the three OPS. Then, the OPS $\{\psi_{\mathbf{k}}(\xi)\}_{|\mathbf{k}|=0}^{+\infty}$ is defined such that

$$\psi_{\mathbf{k}}(\xi) := \psi_{k^D}^D(\xi^D)\psi_{k^L}^L(\xi^L)\psi_{k^R}^R(\xi^R) = \prod_x \psi_{k^x}^x(\xi^x), \quad (10)$$

where $x \in \{D, L, R\}$ in the products here and below and where $|\mathbf{k}| := k^D + k^L + k^R$ is the degree of each polynomial resulting from the tensorization. A new inner product $(\cdot, \cdot) : L^2(\mathbb{R}^3) \times L^2(\mathbb{R}^3) \rightarrow \mathbb{R}$ is naturally induced which, in turn, naturally induces the norm $\|\cdot\| : L^2(\mathbb{R}^3) \rightarrow \mathbb{R}$, such that $\|u(\xi)\|^2 = (u(\xi), u(\xi))$. Orthogonality is inherited from orthogonality of the three OPS, so that

$$(\psi_{\mathbf{j}}, \psi_{\mathbf{k}}) = \int_{\mathbb{R}^3} \prod_x [\psi_{j^x}^x(\xi^x)\psi_{k^x}^x(\xi^x)\phi^x(\xi^x)d\xi^x] = \|\psi_{\mathbf{k}}\|^2 \delta_{\mathbf{j},\mathbf{k}}, \quad (11)$$

where $\delta_{\mathbf{j},\mathbf{k}}$ generalizes Kronecker symbol to the present context, differing from 0 iff multi-indices \mathbf{j} and \mathbf{k} are equal in all of their portions. In (11) and in similar expressions, $\prod_x \phi^x(\xi^x) =: \phi(\xi)$ represents the joint probability density distribution of the three stochastic variables, in the hypothesis of independence. The extension

to the general, dependent case follows on the formal side by simply replacing the multiplicative distribution with the true joint PDF. Needless to say, practical difficulties would arise in both the theoretical and the empirical determination of the latter distribution.

A basis for $L^2(\mathbb{R}; \phi^D) \otimes L^2(\mathbb{R}; \phi^L) \otimes L^2(\mathbb{R}; \phi^R)$ is now available. We look for the closest approximation in this function space to function (1) expressing the dependance of stochastic variable S from stochastic variables D, L, R . Then, one may expand S in a (generalized) Fourier series according to the basis of polynomials (10) as the *polynomial chaos expansion* (PCE)

$$S(\xi) = \sum_{|\mathbf{k}|=0}^{+\infty} s_{\mathbf{k}} \psi_{\mathbf{k}}(\xi) \simeq \sum_{|\mathbf{k}|=0}^p s_{\mathbf{k}} \psi_{\mathbf{k}}(\xi), \quad (12)$$

where $p \geq 0$ determines a practically unavoidable truncation. The choice of p may be deduced a priori in simple cases like the one at hand (viz., $p = 2$ to get the exact expansion) or induced experimentally in more difficult ones, for instance by progressively increasing p until the marginal accuracy in the final results becomes negligible. Care must be taken, because some important modeling feature may require some high order term after a possibly long sequence of null lower order terms. A sound theoretical analysis of the kind of expected functional dependence is very important in order to deal with complex cases, and comparison with MC method in a simple and inexpensive test case may help solve the question.

After the expansion, the information content expressed by S resides in the sequence of coefficients $s_{\mathbf{k}}$, which, owing to orthogonality (11) of the basis, may be individually computed by orthogonal projection of S over each basis element. Precisely, scalarly multiplying both sides of (12) by $\psi_{\mathbf{k}}$ and recalling (11), one gets

$$s_{\mathbf{k}} = \frac{(S, \psi_{\mathbf{k}})}{\|\psi_{\mathbf{k}}\|^2}. \quad (13)$$

The scalar product in (13) requires numerically evaluating a multiple integral, a task that can be accomplished by means of a suitably defined Gauss quadrature formula; see [16]. One gets

$$(S, \psi_{\mathbf{k}}) = \int_{\mathbb{R}^3} S(\xi) \psi_{\mathbf{k}}(\xi) \phi(\xi) d\xi = \sum_{j=1}^N w_j S(\xi_j) \psi_{\mathbf{k}}(\xi_j) \phi(\xi_j), \quad (14)$$

where N is the number of Gauss points while w_j and ξ_j are, resp., the j th Gauss point and weight, $j \in \{1, \dots, N\}$. In simple cases like the one at hand, the necessary number of Gauss points allowing exact integration may be found a priori. In more difficult cases one may try heuristically, experimenting by progressively increasing N until the marginal accuracy increment becomes negligible. Each Gauss point requires a function evaluation

$$S(\xi_j) = S(\xi_j^D, \xi_j^L, \xi_j^R) = \xi_j^D (\xi_j^L + \xi_j^R \Delta t). \quad (15)$$

Owing to the tensor nature of the approach, the number of model evaluations grows rapidly with the number of input stochastic variables. This problem prevents the method from being applied in cases with large numbers of input stochastic variables. Reduced schemes could be pursued, alternative to the full factorial scheme discussed above, that help mitigating the computational burden without excessively sacrificing accuracy. Still, the issue remains for large enough number of variables.

As soon as coefficients $s_{\mathbf{k}}$ have been computed, a basic, standard change of variable and orthogonality (11) easily lead (see, e.g., [10, p. 39]) to the analytical deduction of the expected value (i.e., the mean)

$$E[S] = \int_{\mathbb{R}} \xi^S \phi^S(\xi^S) d\xi^S = s_{\mathbf{0}} \|\psi_{\mathbf{0}}\|^2 \quad (16)$$

and of the variance

$$\sigma_S^2 = \int_{\mathbb{R}} (\xi^S - E[S])^2 \phi^S(\xi^S) d\xi^S = \sum_{|\mathbf{k}|=1}^{+\infty} s_{\mathbf{k}} \|\psi_{\mathbf{k}}\|^2 \simeq \sum_{|\mathbf{k}|=1}^p s_{\mathbf{k}} \|\psi_{\mathbf{k}}\|^2. \quad (17)$$

Notice that the actual expression of $\phi^S(\xi^S)$ is not required in order to compute mean and variance: (16) and (17) immediately allow removing the linearization drawback from traditional approaches to ROP; see Sect. 2.3. More elaborated expressions may be deduced for higher order moments. If, like in the case at hand, $\phi^S(\xi^S)$ becomes necessary for the following developments, it can be computed by, e.g., “fast” MC simulation adopting the last term of (12) as a surrogate model for the real $S(\xi)$ functional dependence (1). The CDF (2) is finally computed in a straightforward manner. See Fig. 4 (right) for the illustration of a sample product code in later discussed ABB inventory test case.

Clearly, the basic reorder point model (1) at hand, of natively polynomial kind, is so simple that, on the one hand, well suits the explanatory goal but, on the other hand, PCE (eventually requiring a “fast” MC) is not advantageous over a direct MC. Nonetheless, as soon as additional features are introduced into the model, thus complicating its analytical expression and possibly requiring the solution of ODE, or PDE, or even possibly resulting into a very complicated black box, like in the case of DES or other dynamics based inventory models, then PCE may outscore direct MC in terms of required model runs, with a remarkable computational gain.

3 Local Data Regularization via QP

The CDF (2) is potentially affected by artifacts deriving from the empirical sampling of the input data that lead to its deduction. Typically, spurious spikes in the originating PDFs may appear, due to rare events, i.e., with a recurrence period greater than the long time scale, but that nonetheless happened to manifest therein

by mere chance. If not suitably dealt with, that is, if data are kept as they are after sampling (resp., if they are removed), then rare event recurrence frequency would result artificially and misleadingly increased (resp., decreased). Owing to a reasonable underlying regularity assumption, the presence of such anomalies is easily spotted in PDFs by inspection.

Since spikes in PDFs result into concavity changes in CDFs, a mathematical regularization technique based on concavity healing is here proposed to remove or reduce the above mentioned artifacts, thus improving the quality of (2). We shall see, when dealing with the final global optimization problem, that the proposed regularization is also beneficial in reducing the computational effort; see Sect. 4. For this reason we privilege this approach compared to other possible ways of filtering. See Fig. 5 for an applicative example based on later discussed ABB inventory analysis.

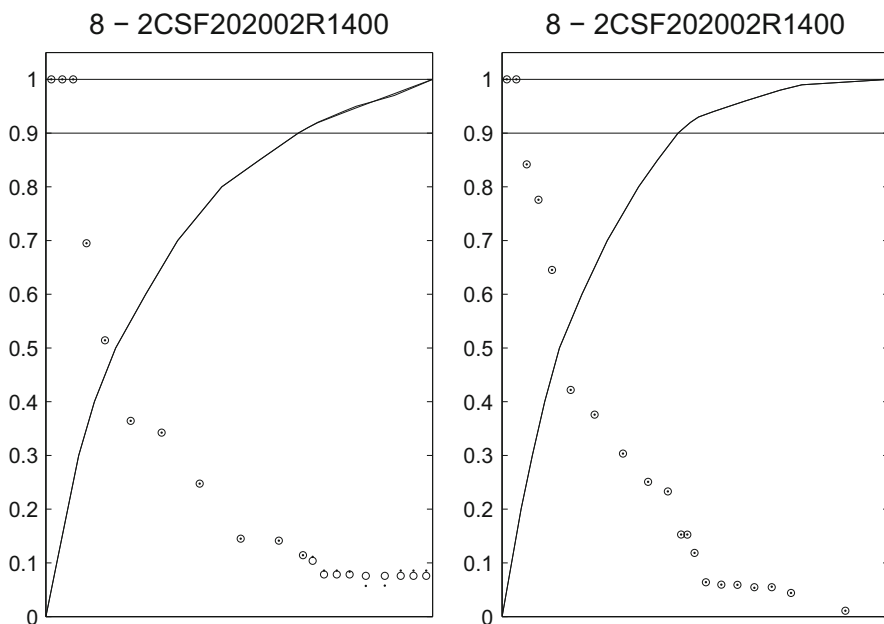


Fig. 5 Effects of QP regularization procedure over one product code in ABB inventory test case (downsampled curves are shown, for clarity). *Left* If PDF ϕ^S and CDF Φ^S are generated from a 10^2 runs MC analysis (extremely poor and inadequate) some differences can be noted between original (dots) and regularized (circles) PDFs, while the effect over CDFs (solid) is barely noticeable. *Right* With 10^3 runs MC originated curves, differences become barely noticeable also in PDFs. CDFs are already concave in the region of interest with 10^4 runs MC or higher, so that no regularization is needed. In ABB inventory test case, 10^5 runs MC are used

3.1 Concave CDFs

A general real function $y = f(x)$ of real variable x is convex over a (possibly unbounded) connex interval $U \subseteq D$, where D is its domain, iff $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$, $\forall x_1, x_2 \in U, \forall t \in [0, 1]$, i.e., if the graph of function f over the interval between any two points x_1 and x_2 in U does not lie “above” the line segment from $(x_1, f(x_1))$ to $(x_2, f(x_2))$. A function f is concave over U if $-f$ is convex over U , i.e., if the graph does not lie “below” such segment. Linear and affine functions are immediately seen to be both convex and concave. A straightforward link with calculus shows that, in case of piece-wise twice differentiable functions, concavity coincides with non-positive second derivative. In other words, the first derivative needs being monotonically non-increasing over U in order f to be concave over U . See, e.g., [15] for in-depth discussion. As a consequence, when $f = \Phi^x$ is a CDF, concavity over U means that the relevant PDF $\phi^x = (\Phi^x)'$ be non-increasing over U .

In the context of ROP estimation, for a given stockable item the relevant PDF $\phi^S(\xi^S)$ is typically single tailed on its “right” portion, witnessing the increasing unlikeliness of extreme events. Along the tail, ϕ^S is non-increasing, by the definition of tail, so that Φ^S needs being concave over a suitable set U contained in the tail. Nonetheless, spurious and unphysical spikes may propagate through the computations above from empirical PDFs ϕ^D and ϕ^L to computed PDF ϕ^S . Spikes alter the natural non-increasing nature of ϕ^S , thus spoiling the concavity of Φ^S in its rightmost portion. In most cases, one could always envisage rare enough events that require a correspondingly longer historical data series than what experimentally available. As a consequence, spikes should be deemed a negative noise affecting data, to be removed for the sake of accuracy, and data regularization must be a due diligence before attacking the problem.

3.2 Concavity Enforcement

In the general case of a PDF $\phi(\xi)$ which is expected to be right-tailed, an a posteriori method is proposed to remove possible spikes by finding the right-tailed, spikeless, non-increasing PDF $\phi'(\xi)$ preserving mean value and minimizing a natural distance from the original one. After this found, the relevant CDF is immediately found by integration as $\Phi'(\xi) = \int_{-\infty}^{\xi} \phi'(\zeta) d\zeta$. Application to ϕ^S and Φ^S will constitute a special case. According to the preceding developments, we may assume without loss of generality to deal with a general, compactly supported, *piece-wise constant* (PWC) function reading

$$\phi(\xi) = \sum_{j=1}^{n-1} \phi_j \chi_{[\xi_j, \xi_{j+1})}(\xi), \quad (18)$$

where the convex support U has been partitioned into $n - 1$ intervals $[\xi_j, \xi_{j+1}]$, for $j \in \{1, \dots, n - 1\}$, so that $\chi_{[\xi_j, \xi_{j+1}]}(\xi)$ is the characteristic function of $[\xi_j, \xi_{j+1}]$, equal to 1 if ξ belongs to such interval and null otherwise, while ϕ_j is the value constantly taken by $\phi(\xi)$ over such interval. An analogous expression holds for $\phi'(\xi)$.

As a result of integration, the CDF $\Phi(\xi) = \int_{-\infty}^{\xi} \phi(\zeta) d\zeta$ is continuous and *piece-wise linear* (PWL). As such, it is characterized by the sequence of n vertices $\{(\xi_j, \Phi_j)\}_{j=1}^n$, where $\Phi_j = \Phi(\xi_j)$, so that $\phi_j = (\Phi_{j+1} - \Phi_j) / (\xi_{j+1} - \xi_j) \geq 0$, the latter sign constraint holding because PDFs are non-negative by definition. Let us collect known values ϕ_j , for $j \in \{1, \dots, n - 1\}$, into vector $\boldsymbol{\phi} \in \mathbb{R}_+^{n-1}$, where \mathbb{R}_+ denotes the non-negative real half line. Similarly, let us collect the sought for values ϕ'_j into vector $\boldsymbol{\phi}' \in \mathbb{R}_+^{n-1}$.

For the application at hand, only the final tail of the PDF is potentially affected by spikes. Therefore, an initial CDF part, up to a given number of points $m \leq n$, may be preserved as is. For any given m , the unknown vector $\boldsymbol{\phi}'$ is found by solving the *quadratic programming* (QP) problem

$$QP_m : \left\{ \begin{array}{l} \min_{\boldsymbol{\phi}' \in \mathbb{R}_+^{n-1}} g(\boldsymbol{\phi}') = \sum_{j=1}^{n-1} \phi_j'^2 - 2 \sum_{j=1}^{n-1} \phi_j \phi_j' \\ \text{subject to} \\ \sum_{j=1}^{n-1} \phi_j' (\xi_{j+1} - \xi_j) = 1 \\ \sum_{j=1}^{n-1} \phi_j' (\xi_{j+1}^2 - \xi_j^2) = \sum_{j=1}^{n-1} \phi_j (\xi_{j+1}^2 - \xi_j^2) \\ \phi_j' = \phi_j, \quad \forall j \in \{1, \dots, m - 1\} \\ \phi_j' \leq \phi_{j-1}', \quad \forall j \in \{m, \dots, n - 1\}, \end{array} \right. \quad (19)$$

now described in detail.

One obvious goal is not to excessively perturbate the basic PDF structure, that is, to aim at minimizing some reasonably defined distance between the original and regularized PDFs. A straightforward goal function is thus (the square of) the Euclidean distance $f(\boldsymbol{\phi}') := \|\boldsymbol{\phi} - \boldsymbol{\phi}'\|_2^2 = \sum_{j=1}^{n-1} (\phi_j - \phi_j')^2$, to be minimized. The minimization problem is equivalently set with reference to goal function

$$g(\boldsymbol{\phi}') = f(\boldsymbol{\phi}') - \sum_{j=1}^{n-1} \phi_j^2 = \sum_{j=1}^{n-1} \phi_j'^2 - 2 \sum_{j=1}^{n-1} \phi_j \phi_j' = \boldsymbol{\phi}'^T \mathbf{I} \boldsymbol{\phi}' - 2 \boldsymbol{\phi}^T \boldsymbol{\phi}', \quad (20)$$

differing from $f(\phi')$ by an inessential additive constant and being homogeneous in ϕ' . The quadratic form (20) is convex because the identity matrix \mathbf{I} is positive definite. As a PDF mandatory requirement, ϕ'_j must be non-negative, $j \in \{1, \dots, n-1\}$. This is explicitly enforced by searching for a minimizer ϕ' in \mathbb{R}_+^{n-1} , i.e., the space obtained by the tensorization of $n-1$ copies of the non-negative half line.

The 0th moment (i.e., the area) of the sought for PWC PDF ϕ' reads

$$\int_{\mathbb{R}} \sum_{j=1}^{n-1} \phi'_j \chi_{[\xi_j, \xi_{j+1})}(\xi) d\xi = \sum_{j=1}^{n-1} \phi'_j \int_{\xi_j}^{\xi_{j+1}} d\xi = \sum_{j=1}^{n-1} \phi'_j (\xi_{j+1} - \xi_j). \quad (21)$$

Therefore, the first constraint in QP_m (19) traduces a linear (and thus convex) PDF normalization condition.

The first moment (i.e., the mean value) of the sought for PWC PDF ϕ' reads

$$\int_{\mathbb{R}} \xi \sum_{j=1}^{n-1} \phi'_j \chi_{[\xi_j, \xi_{j+1})}(\xi) d\xi = \sum_{j=1}^{n-1} \phi'_j \int_{\xi_j}^{\xi_{j+1}} \xi d\xi = \sum_{j=1}^{n-1} \phi'_j \frac{\xi_{j+1}^2 - \xi_j^2}{2}. \quad (22)$$

Doing similarly with known PWC PDF ϕ and neglecting an inessential factor $1/2$ in both terms, the second constraint in QP_m (19) is obtained, traducing a linear (and thus convex) mean preserving condition.

Initial PDF portion preservation is easily enforced into problem QP_m as $\phi'_j = \phi_j$, $j \in \{1, \dots, m-1\}$, meaning that $m-1$ variables are immediately resolved. If desired, such equalities may be used to a priori resolve up to $m-1$ variables and later deal with a reduced problem. The last part of the sought for PDF ϕ' is required to be monotonically non-increasing, which implies that CDF Φ' be concave over the range of interest. The requirement is immediately enforced in problem QP_m (19) as the linear inequality constraints $\phi'_j \leq \phi'_{j-1}$, $\forall j \in \{m, \dots, n-1\}$. Clearly, other ranges than $j \in \{m, \dots, n-1\}$ could be considered in different contexts, depending on the specific nature of the application at hand, without requiring conceptual modifications to the proposed technique. This completes the deduction of (19).

The constrained optimization problem QP_m (19) may be attacked as follows. As noted above, the goal function and all constraints are convex. It is well-known that a convex minimization problem either has a unique minimum (not necessarily a unique minimizer), or it is infeasible (empty admissible region). An initial guess $m = m_0$ value is chosen small enough so to cover a correspondingly large enough interval U . Even though, for a given m , there is no mathematical guarantee that the admissible region be non-empty, nonetheless it is clear that a non-increasing PDF with suitable mean value exists if $m = 0$, i.e., if any ϕ'_j is free to take a different value from the corresponding ϕ_j . Therefore, starting from $m = m_0$, if m is progressively decreased, so that a progressively larger number of degrees of freedom is introduced into the problem, then a feasible QP_m is eventually attained. The solution to problem QP_m corresponding to the first (i.e., largest) m for which this happens is retained as the final solution to the problem. In the unlikely event that, in order to find a feasible

QP_m , m needs being reduced excessively (say, below a given and reasonably defined threshold m'), then the decision may be drawn to abandon the iterative procedure earlier and to keep a non completely concave CDF. Under this latter respect, large possibilities exist for heuristic criteria.

A pseudo-code description of the proposed, iterative method reads

```
m=m0; while (QPm is not feasible AND m>m'), m=m-1; end
if (QPm is feasible) then
  solve QPm; substitute original PDF and CDF;
else
  keep original PDF and CDF;
end
```

where both feasibility detection and QP solution are addressed by standard, state of the art Mathematical Programming techniques; see [1, 4–6].

4 Global Warehouse Optimal Sizing via MP

Given a collection of curves expressing the relationship between ROP and confidence level not to stock-out, obtained according to the above developments, the final step consists in assigning suitable and reasonably different ROPs to all involved goods in stock (typically hundreds of goods, considering single product families individually), and consequently choosing suitable service levels for all items, so that the global (and suitably weighted) service level be maximized and under the constraint that the global cost comply with a global budget; see Sect. 1, problem (B).

4.1 Pareto Fronts

Let us consider the i th stockable item. A *key performance indicator* (KPI), or *service level*, is naturally expressed by the probability not to stock-out

$$P_i(\xi_i^S) = \Phi_i^S(\xi_i^S) = \int_{-\infty}^{\xi_i^S} \phi_i^S(\zeta) d\zeta. \quad (23)$$

The *cost* of stock may be estimated based on the average stock level over time; see Fig. 3. Within the assumptions of the model adopted, as soon as the reordered quantity $O_i \geq E[D_i](E[L_i] + E[R]\Delta t)$ enters the stock, a level $S_i - D_i(L_i + R\Delta t) + O_i$ is attained, eventually decreasing linearly down to a level $S_i - D_i(L_i + R\Delta t)$ just before receiving the next reordered quantity. Such saw-tooth behavior is repeated indefinitely. Adopting average values for other variables than the reorder point S_i

and recalling that $E[R] = 1/2$, the average stock level¹ for the i th good is related to ξ_i^S as

$$\bar{S}_i = \xi_i^S + \frac{O_i}{2} - E[D_i] \left(E[L_i] + \frac{\Delta t}{2} \right). \quad (24)$$

Stock costs can then be expressed as

$$C_i(\xi_i^S) = C_i^a + C_i^u \bar{S}_i = C_i^f + C_i^u \xi_i^S, \quad (25)$$

where $C_i^a \geq 0$ is an *activation cost*, i.e., independent of the stocked quantity, while $C_i^u > 0$ is a *unit cost*, so that $C_i^u \bar{S}_i$ is a cost proportional to the average stock level \bar{S}_i and, finally, $C_i^f = C_i^a + C_i^u(O_i/2 - E[D_i](E[L_i] + \Delta t/2))$ is a *fixed cost* with reference to variable ξ_i^S . The unit cost C_i^u is intrinsically non-null, for otherwise the stock problem for the relevant good would be trivially solved by assigning an arbitrarily large reorder point, a meaningless case in the applicative context. Indeed, $C_i^u \neq 0$ implies that map (25) be bijective and thus invertible. Moreover, the inverse map C_i^{-1} is still affine and such that $\xi_i^S = (C_i - C_i^f)/C_i^u$.

According to the commutative diagram

$$\begin{array}{ccc} C_i \in \mathbb{R} & \xrightarrow{C_i^{-1}} & \mathbb{R} \ni \xi_i^S \\ & \searrow & \downarrow \Phi_i^S \\ & & [0, 1] \ni P_i \end{array}$$

$F_i := (C_i^{-1})^*(\Phi_i^S) = \Phi_i^S \circ C_i^{-1}$

it is possible to translate (23) and (25) into a parametric description of the $P_i = F_i(C_i)$ curve (*performance vs. cost curve*; see, e.g., Fig. 6), where $F_i := (C_i^{-1})^*(\Phi_i^S) = \Phi_i^S \circ C_i^{-1}$ is the *pullback* of Φ_i^S . Notice that concavity (as well as convexity) is invariant under composition with affine maps, that is, if $g = f \circ l$, where f is concave and l is affine, then

$$\begin{aligned} (f \circ l)(tx_1 + (1-t)x_2) &= f(l(tx_1 + (1-t)x_2)) \\ &= f(tl(x_1) + (1-t)l(x_2)) \\ &\geq tf(l(x_1)) + (1-t)f(l(x_2)) \\ &= t(f \circ l)(x_1) + (1-t)(f \circ l)(x_2), \end{aligned}$$

so that also g is concave. Therefore, since both C_i and its inverse C_i^{-1} are affine, concavity of F_i is equivalent to concavity of Φ_i^S . Concavity in case of performance vs. cost curves is reasonably justified empirically: Saturation frequently occurs in

¹In case of models (3)–(5), similar considerations lead to $\bar{S}_i = \xi_i^S - E[D_i](E[L_i] + \Delta t/2 + T/2)$, $\bar{S}_i = \xi_i^S + O_i/2$ and $\bar{S}_i = \xi_i^S$, resp.

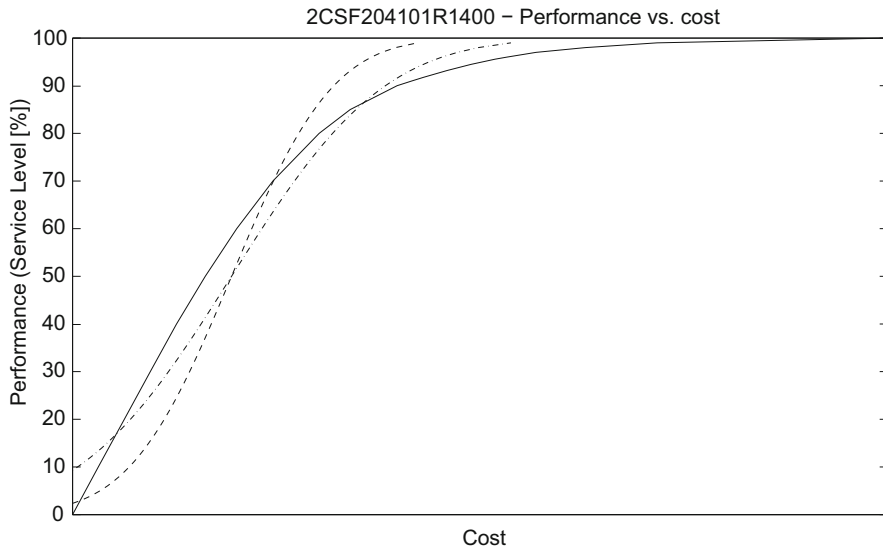


Fig. 6 Performance vs. cost curve for a sample product code, computed according to the proposed method, i.e., by means of a nonlinear model and without any normality assumption (*solid*). Also shown are the curves computed according to the traditional normality assumption and with standard deviation computed by means of variance propagation, i.e., with model linearization, (*dash-dot*) or with Hadley-Whitin formula (*dashed*); non negligible deviations from the correct behavior are observed in the latter two curves

many economical systems, so that the same marginal cost increment results into progressively smaller marginal performance increments as the performance level increases. The performance vs. cost curve for a sample product code is shown in Fig. 6, where the discrepancy between proposed and traditional approach can be appreciated.

Performance vs. cost curves satisfy *Pareto front* requirements. Precisely, any given point (C_i, P_i) on the curve is optimal in the sense that with a given cost C_i it is not possible to obtain a better performance than $P_i = F_i(C_i)$, whereas lower performances than P_i are unjustified, since the system has the capability to behave better. Dually, to attain a given performance P_i it is not possible to spend less than C_i , whereas to spend more is economically unjustified. In conclusion, points above the front are infeasible and those below are uneconomical; see Fig. 8, where the same considerations are illustrated for the whole inventory Pareto front, to be later deduced.

We consider now the problem of choosing a suitable positioning somewhere along the front. This amounts to solving a global optimization problem over a collection $\{(G_i, w_i)\}_{i=1}^N$ of N stockable goods G_i , each one with a relevant performance vs. cost curve and additionally provided with a global weight $w_i \in [0, 1]$, characterized

in that $\sum_{i=1}^N w_i = 1$ and expressing the good's relevance in the frame of the global weighted average performance

$$\bar{P} = \sum_{i=1}^N w_i P_i. \tag{26}$$

Weights are possibly chosen equal to $1/N$ for all goods, whenever no distinction has to be made.

4.2 Problem Formulation: LP

Let us first assume that all $P_i = F_i(C_i)$ curves are concave, $i \in \{1, \dots, N\}$. Local data regularization via QP may be used to force non concave cases to the assumption at hand; see Sect. 3. Let us collect costs C_i and performances P_i into vectors \mathbf{c} and \mathbf{p} , resp. Then, the unknown vectors \mathbf{c} and \mathbf{p} are found by solving the *linear programming* (LP) problem

$$LP : \left\{ \begin{array}{l} \max_{\mathbf{c}, \mathbf{p} \in \mathbb{R}_+^N} \sum_{i=1}^N w_i P_i \\ \text{subject to} \\ \sum_{i=1}^N w_i P_i \geq P^l \\ \sum_{i=1}^N C_i \leq B \\ a_{i,j}^c C_i + a_{i,j}^p P_i \leq b_{i,j}, \quad \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, n-1\} \\ C_i^l \leq C_i \leq C_i^u, \quad \forall i \in \{1, \dots, N\} \\ P_i^l \leq P_i \leq P_i^u, \quad \forall i \in \{1, \dots, N\}, \end{array} \right. \tag{27}$$

now described in details.

The linear goal function is the weighted average performance (26), to be maximized. A (possibly null) minimal admissible global performance P^l is enforced by the first constraint in problem LP (27). A global budget constraint is expressed by the second constraint in problem LP (27), where B is a given global cost budget.

Notice that, owing to the joint constraining action of the two inequalities above, the problem is either infeasible or a performance non worse than P^l is obtained. In the former case, a problem revision is needed in order to make the problem feasible, and either the minimal performance P^l is reduced or the maximal budget B is increased.

The actual shapes of performance vs. cost curves have to be enforced. Since we deal with PWL curves, the information relevant to the i th such curve is expressed by the collection of n vertices $\{(C_{i,j}, P_{i,j})\}_{j=1}^n$ the Pareto front consists of (notice that index i runs over goods while index j runs over the curve vertices for the i th good). The equation of the line through any two consecutive vertices $(C_{i,j}, P_{i,j})$ and $(C_{i,j+1}, P_{i,j+1})$ along the i th front is of the kind

$$a_{i,j}^C C_i + a_{i,j}^P P_i = b_{i,j}, \quad j \in \{1, \dots, n - 1\}, \tag{28}$$

where $a_{i,j}^C := P_{i,j} - P_{i,j+1}$, $a_{i,j}^P := C_{i,j+1} - C_{i,j}$ and $b_{i,j} := C_{i,j+1}P_{i,j} - C_{i,j}P_{i,j+1}$. Since the i th front is concave by assumption, the plane portion “below” the front is the locus of points (C_i, P_i) “below” the plurality of all lines (28), as expressed by the third collection of inequality constraints found in problem LP (27); see Fig. 7. The mathematical characterization of the Pareto front, as discussed in Sect. 4.1, will force (C_i, P_i) to adhere to the front for, otherwise, a more economical solution and with same performance could be found (or, dually, a better performing solution and with the same cost). Finally, lower bounds C_i^l (resp., P_i^l) and upper bounds C_i^u (resp.,

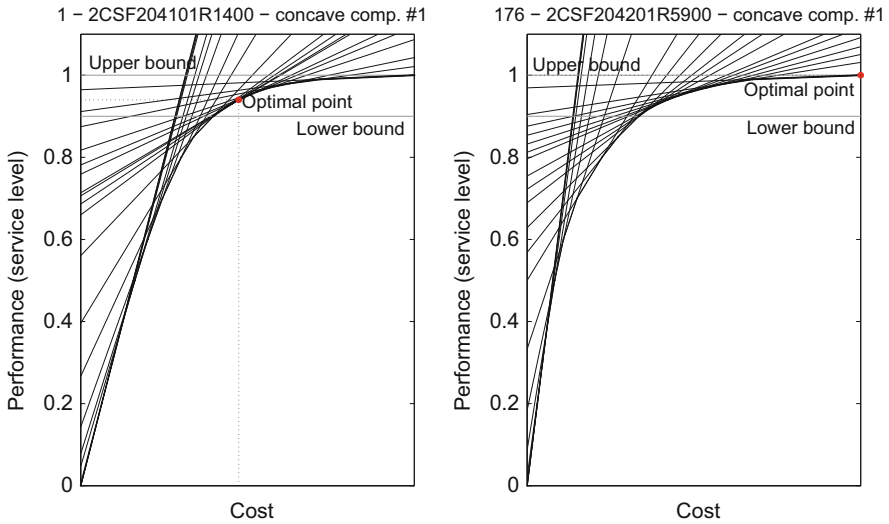


Fig. 7 PWL performance vs. cost curves (pareto fronts) for two sample stockable goods (first and last moving in the inventory), as rendered by means of a system of linear inequalities in LP problem (27); minimally and maximally acceptable service levels are also shown as *horizontal lines* defining the front portion of interest (*bold*) along which the optimal point is restrained to reside

P_i^u) could be imposed onto C_i (resp., P_i), as expressed by the remaining constraints in problem LP (27).

Different inventory management policies may modify the restraint set, retaining only part of the above. The LP problem is solved with standard, state of the art Mathematical Programming techniques, such as the well-known and efficient *simplex method*; see, e.g., [2].

4.3 Problem Formulation: MILP

As an extension of Sect. 4.2, let us now address the general case in which some or all performance vs. cost curves are non concave. Unfortunately, this case is mathematically way harder than the concave case. To understand why, one must recall that each pair (C_i, P_i) of variables in LP problem (27) belongs to the region below the corresponding concave performance vs. cost curve, including the latter curve as part of the boundary. (In the optimal solution, the variables do belong, pairwise, to such boundary curves.) Now, since the performance vs. cost curve is concave, the feasible region for each pair of variables is convex. Additionally, as detailed above, the feasible region is described by a collection of linear inequalities.

Both of these advantages are lost in the general, non-concave case. Since the intersection of convex regions is convex and since half planes delimited by lines are convex regions, if one should now try to describe the region below a non-concave performance vs. cost curve by means of linear inequalities, a convex region would be obtained, which is not the case by assumption. Nonetheless, the attack strategy here proposed consists of partitioning each generally non-convex feasible region into convex parts. Then, the optimal solution must reside in one, and only one, of such parts. This fact may be handled by introducing a potential pair of variables for each part, along with Boolean, disjunctive variables. The latter are used to express the constraint that one, and only one, of such potential pairs of variables be active, all the others being “phantoms”. The contributions of phantoms must be suitably removed, since such fictitious points are only introduced for the sake of convenience, but they are not “real”. It is easily understood that the disjunctive nature of the variables and constraint make the problem combinatorial. The mathematical tool of integer programming is then required, with a remarkable increase of both the theoretical complexity and, above all, of the computational burden.

We now show the proposed approach in details. Due to its PWL nature, it is always possible, without loss of generality, to assume that the generic i th performance vs. cost curve consist of K_i concave portions. If such curve is originally concave, then $K_i = 1$. As an extreme case, $K_i = n - 1$ and concave portions coincide with the linear spans of the PWL function. With reference to the i th performance vs. cost curve, let us introduce a collection $\{(C_{ik}, P_{ik})\}_{k=1}^{K_i}$ of the sought for points, the k th of which belonging to the k th concave portion. Vectors \mathbf{c} and \mathbf{p} need being suitably enlarged so to host all of the so introduced unknown variables for all curves, for a total of $K = \sum_{i=1}^N K_i$ entries. Let us also assume that the k th portion of the i th

curve extend over the $[C_{ik}^l, C_{ik}^u]$ interval, so that the “right”-most extreme coincide with the “left”-most extreme of the next interval.

Additionally, Boolean decision variables are introduced, so that $z_{ik} \in \mathbb{Z}_2 \cong \{0, 1\}$ be associated with the k th concave portion of i th stockable item performance vs. cost curve. The z_{ik} 's are collected into vector $\mathbf{z} \in \mathbb{Z}_2^K$, where the latter space \mathbb{Z}_2^K is the lattice of multidimensional points obtained by tensorization of K copies of \mathbb{Z}_2 . The basic idea is that, if $z_{ik} = 1$, then the point on the i th Pareto front lies on its k th concave portion (the *active portion*), and $z_{ik} = 0$ for all other k 's. For the sake of convenience, the points (C_{ik}, P_{ik}) relevant to unused concave portions are still formally present, and they are conventionally located at the “left”-most point of such portion, i.e., $(C_{ik}^l, F_i(C_{ik}^l))$. Suitable terms will consequently arise in the mathematical description, so to compensate the presence of such “phantom” points.

The unknown vectors \mathbf{c} , \mathbf{p} and \mathbf{z} are found by solving the *mixed integer linear programming* (MILP) problem

$$\begin{aligned}
 & \max_{\mathbf{c}, \mathbf{p} \in \mathbb{R}_+^K, \mathbf{z} \in \mathbb{Z}_2^K} \sum_{i=1}^N \sum_{k=1}^{K_i} w_i P_{ik} - \sum_{i=1}^N \sum_{k=1}^{K_i} w_i F_i(C_{ik}^l)(1 - z_{ik}) \\
 & \text{subject to} \\
 & C_{ik} - (C_{ik}^u - C_{ik}^l)z_{ik} \leq C_{ik}^l, \\
 & \quad \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K_i\} \\
 & \sum_{i=1}^N \sum_{k=1}^{K_i} w_i P_{ik} - \sum_{i=1}^N \sum_{k=1}^{K_i} w_i F_i(C_{ik}^l)(1 - z_{ik}) \geq P^l \\
 MP : & \left\{ \begin{aligned} & \sum_{i=1}^N \sum_{k=1}^{K_i} C_{ik} - \sum_{i=1}^N \sum_{k=1}^{K_i} (1 - z_{ik}) C_{ik}^l \leq B \\ & a_{ik,j}^C C_{ik} + a_{ik,j}^P P_{ik} \leq b_{ik,j}, \\ & \quad \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, n-1\}, \forall k \in \{1, \dots, K_i\} \\ & \sum_{k=1}^{K_i} z_{ik} = 1, \quad \forall i \in \mathcal{I}_{mh} \\ & \sum_{k=1}^{K_i} z_{ik} \leq 1, \quad \forall i \notin \mathcal{I}_{mh} \\ & C_{ik}^l \leq C_{ik} \leq C_{ik}^u, \quad \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K_i\} \\ & P_i^l \leq P_{ik} \leq P_i^u, \quad \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K_i\}, \end{aligned} \right. \quad (29)
 \end{aligned}$$

now described in details.

The basic structure of problem MP (29) is clearly inherited from problem LP (27). Notice that the maximizer is sought for in a multidimensional space, obtained by tensorization, where \mathbf{c} and \mathbf{p} are both in \mathbb{R}_+^K , and thus the relevant entries must be non-negative, while \mathbf{z} is in \mathbb{Z}_2^K , and thus the relevant entries are only allowed to take either value 1 or 0.

The penultimate collection of constraints forces C_{ik} to belong to the relevant concave portion. Let us consider the first collection of constraints. In case the k th is the i th good active portion, then $z_{ik} = 1$ and the constraint reduces to $C_{ik} \leq C_{ik}^u$, that is, a redundancy. Otherwise, $z_{ik} = 0$ and the constraint reduces to $C_{ik} \leq C_{ik}^l$. The only possibility is thus $C_{ik} = C_{ik}^l$, so that phantom points are forced to coincide with the “left”-most point of the relevant concave portion. The goal function is still the weighted average global performance (26), with the additional summation compensating phantom points. As a matter of fact, the active portions do not contribute to such additional summation ($1 - z_{ik} = 0$), whereas all other portions remove a $w_i F_i(C_{ik}^l)$ contribution from the global performance ($P_{ik} = F_i(C_{ik}^l)$ for phantom points).

The same reasoning applies to the minimal performance P^l constraint and to the maximal budget B constraint ($C_{ik} = C_{ik}^l$ for phantom points). As a matter of fact, also in this problem a (possibly null) minimal admissible global performance P^l is enforced by the second constraint in problem MP (29), and a global budget constraint is expressed by the third constraint in problem MP (29), where B is a given global cost budget. Notice that, exactly like in problem LP (27) and owing to the joint constraining action of the two inequalities above, the problem is either infeasible or a performance non worse than P^l is obtained. In the former case, a problem revision is needed in order to make the problem feasible, and either the minimal performance P^l is reduced or the maximal budget B is increased.

Since discrete decision variables have to be introduced in order to handle the non-concave case, it is worth taking advantage of their modeling power and introduce the possibility to decide whether to stock or not some goods, based on global optimality considerations. Precisely, let us introduce an index set \mathcal{I}_{mh} of *must-have* goods, necessarily to be stocked for any relevant strategic reason. Obviously, the subset \mathcal{I}_{mh} can possibly coincide with the universe of stockable goods, if desired. All other goods are subject to possibly being excluded from the MTS inventory management policy, depending on the solution of problem MP (29). Must-have goods ($i \in \mathcal{I}_{mh}$) are characterized in that *exactly one* concave portion is active (i.e., exactly one $z_{ik} = 1$, all others being null), while all other goods ($i \notin \mathcal{I}_{mh}$) are characterized in that *at most one* concave portion is active (i.e., possibly one $z_{ik} = 1$ and not more, all others—possibly all—being null). Such conditions are mathematically enforced by means of the fourth last and third last constraints, resp.

The resulting problem, especially after the data regularization procedure discussed in Sect. 3, is usually lean enough to be readily handled with standard branch-and-bound techniques on standard computers; see, e.g., [12].

5 Conclusions, Extensions and Future Research

The proposed approach to stochastic optimal sizing of inventories is the result of activities that the present author has undertaken in order to support the rationalization of ABB Low Voltage Products Division warehouse located in Vercelli, Italy, and serving the local national market from ABB factories located nationwide and abroad. Global optimization according to Sect. 4 leads to a global Pareto front, at inventory level, obtained pointwise by varying the global cost budget B in (27) or (29); see Fig. 8.

As for problem (A), i.e., reorder point based performance vs. cost curve determination, the proposed approach goes beyond ABB’s state of the art (i.e., Hadley-Whitin formula) in that the stochastic problem is not forcedly linearized and in that the assumption that involved probability density functions be Gaussian is relaxed to handling generic, experimental distributions, deduced on a sampling

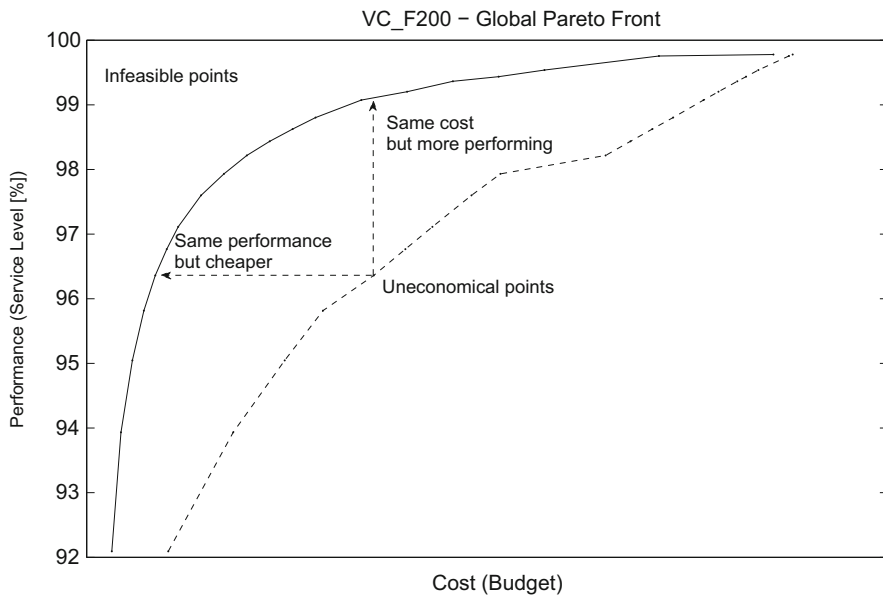


Fig. 8 Global Pareto front (solid) for a portion of ABB Italy—Low Voltage Products Division warehouse, relating global costs (sum over 176 different products) and overall weighted service level; points above the front are infeasible, while points below are uneconomical, because others may be found on the front with same performance and cheaper or, dually, with same cost and better performing. In case no global optimization is carried out but, rather, the same prescribed service level is assigned to all goods, then, by varying the prescribed service level, a remarkably lower curve is obtained (dashed), inside the uneconomical region. As expected, the optimal and non-optimal curve converge at the lowest admissible (90%) and highest possible (100%) service level, where exactly the same quantities are stocked for all goods, while inoptimality is greater elsewhere

basis. A posteriori, it is found that involved distributions are actually very far away from being Gaussian.

As for the global optimization problem (B) of locating working points for all involved goods along relevant performance vs. cost curves, a very fast approach is here proposed, based on forced regularization of non concave curves and LP, along with an alternative, more refined, but also potentially more cumbersome approach based on MILP. ABB priorly followed a rule-based approach following an ABC categorization of goods, intrinsically “local” and with no guarantee for global optimality.

The whole mathematical machinery has been condensed into a Matlab package currently available to ABB professionals operating in Logistics. This application reads suitably formatted data extracted from the Company’s ERP (basically, the last 12 months movements, on a rolling basis), processes them, and finally delivers suggested reorder points, along with graphical indication of performance vs. cost curves, on a good-by-good basis.

The proposed methodology for the global optimization of systems consisting of a collection of performance vs. cost curves (see Sect. 4) could be extended to other systems than just inventories, by abstracting from what “performance” and “cost” actually represent. Examples in the world of industrial operations include finding the optimal strategy for running a multi-commodity productive system, where the performance may be the volumes of each commodity produced, and the cost may be some suitable measure of allocated resources, such as, e.g., manpower, worked hours, externally outsourced work, and the like, each one being usually restrained to a globally available budget. Many other interesting applicative fields could be easily spotted, with different interpretation of involved quantities but with the same or similar mathematical structure.

Performance vs. cost curves may also have a different origin than the one discussed above for inventories, including empirical curves. Concavity driven data regularization (see Sect. 3), may still be adopted in order to simplify the global optimization problem.

Applications may be envisaged requiring large and thus challenging instances of the MILP problem to be solved. Therefore, devising highly efficient and robust approaches to problem MP (29) could find sound motivation. Future research may go beyond general purpose branch-and-bound and seek for an attacking strategy possibly exploiting the specific mathematical structure of the problem. Cumulating a plurality of additional constraints than those examined here may also be an interesting and motivated future research effort. Last, but not least, a deeper statistical analysis and handling of input data could be highly beneficial. The method hereby proposed could easily be adapted in order to be compatible with other than PWC PDFs.

References

1. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: *Nonlinear Programming – Theory and Algorithms*, 3rd edn. Wiley, New York (2006)
2. Bazaraa, M.S., Jarvis, J.J., Sherali, H.D.: *Linear Programming and Network Flows*, 4th edn. Wiley, New York (2010)
3. Chihara, T.S.: *An Introduction to Orthogonal Polynomials*. Dover, New York (2011)
4. Coleman, T.F., Li, Y.: A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM J. Optim.* **6**, 1040–1058 (1996)
5. Gill, P.E., Murray, W., Wright, M.H.: *Practical Optimization*. Academic, London (1981)
6. Gould, N., Toint, P.L.: Preprocessing for quadratic programming. *Math. Program. Ser. B* **100**, 95–132 (2004)
7. Hadley, G., Whitin, T.M.: *Analysis of Inventory Systems*. Prentice-Hall, Englewood Cliffs, NJ (1963)
8. Harris, F.H.: How many parts to make at once. *Fact. Mag. Manag.* **10**, 135–136 (1913)
9. Kimura, O., Terada, H.: Design and analysis of pull system, a method of multistage production control. *Int. J. Prod. Res.* **19**, 241–253 (1981)
10. Le Maître, O.P., Knio, O.M.: *Spectral Methods for Uncertainty Quantification*. Springer, Berlin (2010)
11. Murphy, E.: A comparison of inventory optimization and discrete-event simulation for supply chain analysis. *Simulation Conference, 2007 Winter*, IEEE, New York (2007)
12. Nemhauser, G.L., Wolsey, L.A.: *Integer and Combinatorial Optimization*. Wiley, New York (1999)
13. Orlicky, J.: *Material Requirements Planning – The New Way of Life in Production and Inventory Management*. McGraw-Hill, New York (1975)
14. Ptak, C., Smith, C.: *Orlicky’s Material Requirements Planning*. McGraw-Hill, New York (2011)
15. Rockafellar, R.T.: *Convex Analysis*. Princeton University Press, Princeton, NJ (1970)
16. Szegő, G.: *Orthogonal Polynomials*. American Mathematical Society Colloquium Publications, vol. XXIII, 4th edn. American Mathematical Society, Providence, RI (1975)
17. Wright, O.: *Manufacturing Resource Planning – MRP II*, Revised edn. Wiley, New York (1996)
18. Zipkin, P.: *Foundations of Inventory Management*. McGraw-Hill, New York (2000)

Challenges of Mechatronical Engineering of Production Systems: An Automation System Engineering View

Arndt Lüder and Nicole Schmidt

Abstract The importance of quality and efficiency of engineering process for production system is continuously increasing. Engineering sciences are encouraged to improve its tool and method sets to face this challenge. But in several cases engineers are not the real specialists for improving the toolbox of engineering. Here mathematical science can assist engineering sciences.

Within this paper open research issues for mathematical sciences are derived from the current state of the art in mechatronical engineering of production system intending to encourage joined research activities of mathematical and engineering science.

1 Introduction

The increasing global competition between companies from different global regions with completely different economical conditions forces European companies on the one hand to increase product variety, often until complete individualization to meet customer needs. In parallel, on the other hand, these companies are encouraged to increase production system flexibility regarding resource capabilities and quantities as well as regarding used production system technologies. Finally, they shall reduce both the product life cycle as well as the plant life cycle. But this results in an increased production system complexity which has to be handled within the entire production system life cycle adequately.

One of the key initiatives dealing with this challenge is the German Industrie 4.0 initiative focusing on increasing flexibility of production systems and improving vertical and horizontal integration of production system components, and striving to nothing else than the 4th industrial revolution. Key elements of this initiative are (among others)

- the Industrie 4.0 component, a self-aware and self-adaptable production system component,

A. Lüder (✉) • N. Schmidt

Faculty Mechanical Engineering, Otto-von-Guericke University, Magdeburg, Germany

e-mail: arndt.lueder@ovgu.de; nicole.schmidt@ovgu.de

- the intelligent networking of Industrie 4.0 components to provide flexibility on system level using adaptation capabilities and plug-and-work capabilities of Industrie 4.0 components, and
- the integrated exchange of Industrie 4.0 component information related to engineering and runtime phases along the production system life cycle.

Comprehensive technological developments during the last centuries are the foundation of this discussion enabling new technical possibilities within the design and use of production systems today [1]. It can be observed, that the wide-ranging capabilities of information processing systems from the consumer market has found their way into production systems realizing the vision of Computer Integrated Manufacturing (CIM) in a new fashion.

As the Industrie 4.0 component is a controlled part of a production system including manufacturing physics as well as control intelligence the Industrie 4.0 component shall be considered as a cyber physical system [2, 3] and shall be considered in the triangle of products, production processes, and resources. Each product requires for its production the processes defined in its Bill of Operation. These processes will be processed on a production resource. Each production resource will process sets of products and will be able to execute processes. Finally, each process is used for the production of products and can be executed by a resource [3]. Thus, the Industrie 4.0 component shall act as a resource providing production processes useable to produce products by exploiting its production physics and controlled by its internal control intelligence.

As proposed in [4] the life cycles of production systems and products are interlinked as presented in Fig. 1. The use of Industrie 4.0 components within these life cycles is mostly related to the plant and process development, the production system engineering, the commissioning, the use for production, the maintenance planning, and the maintenance life cycle phases (given in dark blue in Fig. 1).

Nevertheless, within these phases several engineering disciplines are involved in the development and use of the Industrie 4.0 components. Thereby, each phase consists of several engineering activities often related to necessary design decisions within one of the involved engineering disciplines. Process planning, mechanical engineering, electrical engineering, control and robot programming, and virtual commissioning are the most relevant disciplines [5]. As visualized in Fig. 2 the different engineering activities depend on each other (require engineering results of prior engineering activities) and exploit different engineering tools. In most cases, these tools are tailored to an efficient execution of the necessary work engineering activities (the optimal execution of design decisions and creation of required engineering artefacts) [6]. They are based on their own model type and their own data structure optimised to the tool use and software structure. But following the chain of engineering activities it is hard to enable a consistent and lossless exchange of engineering data (digital engineering artefacts or parts of them) between the engineering tools [7].

One mean to address the problem of consistent engineering of production systems integrating different engineering disciplines (covering the data exchange

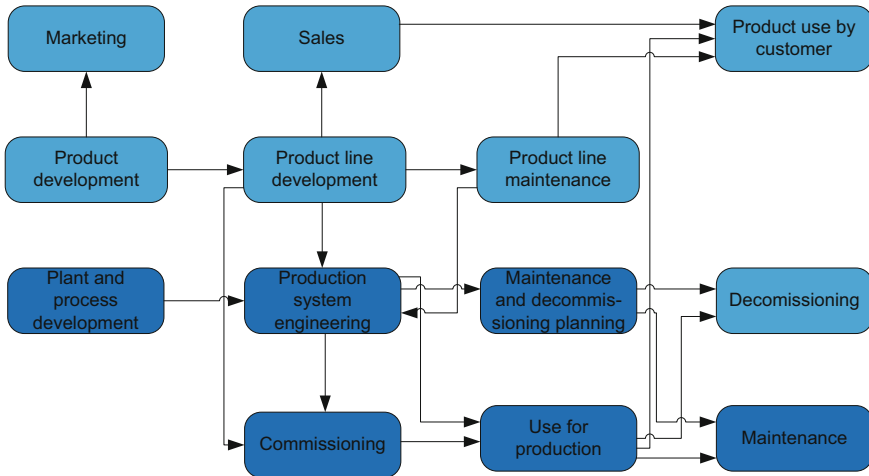


Fig. 1 Value chain oriented view on the product and production system life cycles

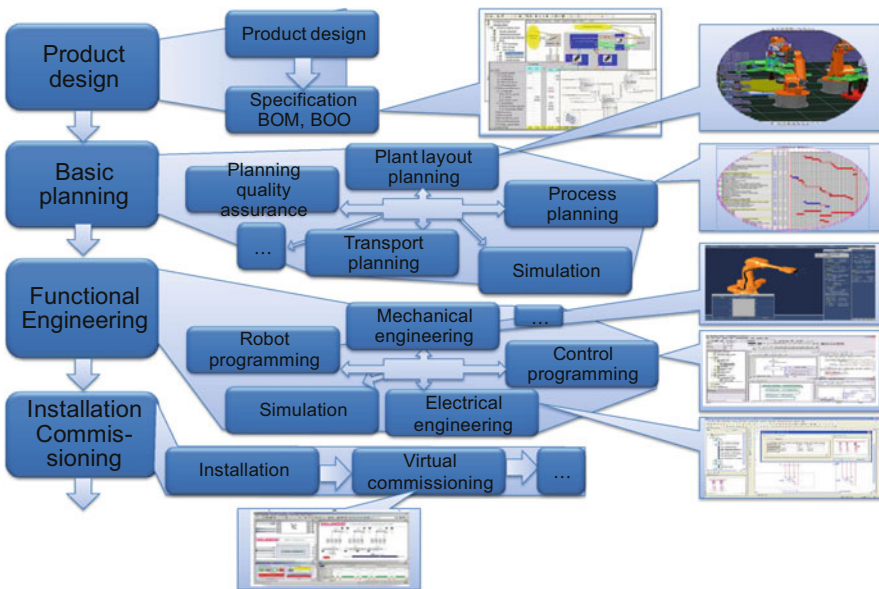


Fig. 2 Hierarchical structure of engineering process of production systems

problem, the consistency problem, etc.) is mechatronic engineering. Mechatronic thinking and mechatronic engineering, based on it, are common in product engineering and design since the seventies and the eighties of the last century.

Initially, mechatronic has been considered as supporting guideline in product design where the meaningful combination of different engineering disciplines has

provided an added value for the product properties and functionalities [8, 9]. Here systems like CD players or antilock braking system have been developed. Over time, this combination had been proven useful for engineering of production systems since production systems are also product (even with the special nature of a single piece of its own) [10–12].

Mechatronic thinking within production system engineering is resulting in specific engineering processes as well as in specific production system architectures [6, 13]. Both have an essential impact on the work of engineers.

Within this paper the main concepts of mechatronic engineering within production system engineering will be considered. The terms of a mechatronic unit and a mechatronically oriented control architecture are described. Based on them the mechatronic engineering process is described as well as its main steps going beyond classical engineering approaches within machine and plant engineering. With this background the paper will analyse benefits, challenges and limitations of the mechatronic engineering process from the viewpoint of the engineering of production systems and the automation systems used within. It will draw conclusions for open research questions possibly answered by mathematical research.

2 Mechatronic Engineering

Mechatronic thinking and engineering had evolved from similar developments within the industrial countries in the late seventies and early eighties of the last century. In Germany, to give an example, the so-called “Feinwerktechnik” (precision engineering) has emerged covering the combination of mechanical and electrical engineering. The term “Mechatronic” originating from Japan has been internationally adopted for the advantageous combination of mechanical, electrical, and electronic engineering quickly. Within the following years more engineering disciplines have been integrated like optics and information sciences [14, 15].

Initially, mechatronic was focused on the design and engineering of products where the meaningful combination of different engineering disciplines can provide an additional value for the functionality, stability, etc. of intended products [8, 16]. But mechatronic has been proven to be also helpful for the structuring, design, and engineering of production systems and beyond [11, 12, 17].

2.1 Mechatronic Units and Systems in Production Systems

In recent years a broad agreement about the definition of the term Mechatronic has been established. Following this agreement it holds:

A mechatronic unit is a closed system providing a dedicated (mostly physical) behaviour within a production system utilizing sensors, actuators, and intelligent

control devices in a closed loop control structure. Thereby, the mechatronics unit combines on the one hand software (for control program development) and hardware (mechanics, electronics, . . .) and on the other hand different engineering disciplines to achieve an optimal functionality.

A mechatronics system is established by the systematic combination/interlinking of mechatronics units and/or mechatronics systems within a hierarchical structure. Thereby, each mechatronics system will contain its own information processing used for optimal control of the functionality and the interaction of the different interlinked mechatronics units and mechatronics systems of the lower hierarchy layers.

The distinction between mechatronics units and mechatronics systems results from the consideration of the hierarchy of mechatronics units and mechatronics systems. Usually the leaves of this hierarchy, i.e. the ends of paths, are regarded as mechatronics units while all other objects in the layers above are regarded as mechatronics systems. But most important, the mechatronics units have direct access and control of the underlying physics of the production system. It depends on the system of interest whether a drive is seen as the mechatronics unit or a power train including drive, gearbox, and frequency converter, or the complete conveyor with lifting table.

The structure and interlinking of mechatronics units and mechatronics systems covering only two layers (as simplification) is depicted in Fig. 3 to give a hierarchy example.

Following [12, 17] the complete structure of a mechatronics oriented production system can be represented by a six layer hierarchy. The lowest of these six layers is formed by mechanical and electrical parts like metal stiffeners, electrical wires, and screws. They are arranged in sub-function groups which, in combination with other sub-function groups, will provide basic functionalities of the production

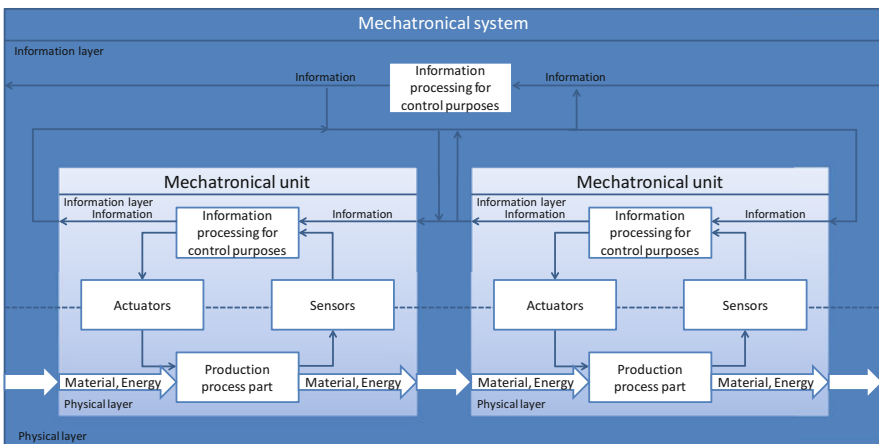


Fig. 3 Mechatronics structure consisting of two layers

system. Thus sub-function groups are grouped to function groups. For example, single clamping fixtures are combined to clamping fixture groups providing the production function “fixing material” which is required in a robot based welding cell or combining a drive, a gearbox, a frequency converter, and some shafts within a power train to provide the function “motion”. Thus, function groups provide more complex functionalities which are of importance for the execution of production steps by providing essential parts of production steps. Complete production steps which are usually part of the bill of operation of a product will be provided by main groups. Main groups integrate a set of function groups as it is the case for clamping fixture groups, power trains and other function groups within a milling machine. Together, they can execute a milling function on a work piece. Main function groups can be combined to manufacturing cells able to execute sets of manufacturing steps. For example a milling machine can be combined with a robot for material handling and a storage for different milling tools in a milling cell. Finally, a set of cells can be combined to a site as a set of milling cells can be combined to an engine production site of a car manufacturer.

Usually sites, cells, main groups and function groups can be considered as mechatronical systems while cells, main groups, function groups, and sub-function groups can be regarded as mechatronical units. Here, the relevant viewpoint is essential for the definition of the lowest level of consideration which will constitute the mechatronical units. The hierarchical structure is depicted in Fig. 4.

Within the mechatronical engineering the mechatronical unit or system shall be represented by an engineering artefact covering the information sets of all relevant engineering disciplines, a kind of digital representation or digital shadow called mechatronical information object.

A mechatronical information object is an engineering artefact combining the modelling of mechatronical units of a manufacturing system with its different characteristics like signals, electrical drawings, function blocks or devices in one information object. It is the information representation of a mechatronical unit

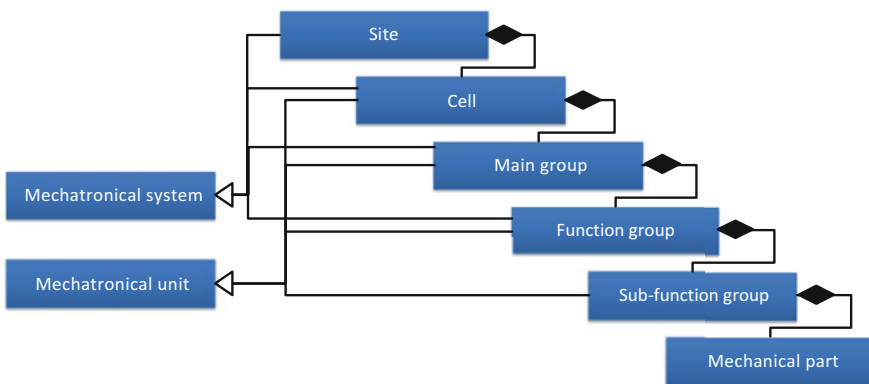


Fig. 4 Hierarchical structure of a Mechatronical oriented production system

within a mechatronic engineering process. Thus, it has to cover at least the following information sets.

- Topology data including the hierarchy of sub-elements (other mechatronic units and/or devices),
- Mechanical data including mechanical constructions with geometry and kinematics (especially mechanical drawings/MCAD),
- Electrical, pneumatic, and hydraulic data including electrical construction as wirings of the different types and their plugs,
- Function describing data like functional models of controlled and uncontrolled behaviour,
- Process control data like control code of any kind, and
- Generic data summarizing further organizational, technical, economical, and other data like order information or handbooks and guidelines.

These information sets are depicted in Fig. 5.

The relations between mechatronic engineering, mechatronic units, mechatronic systems and mechatronic information objects are given in Fig. 6.

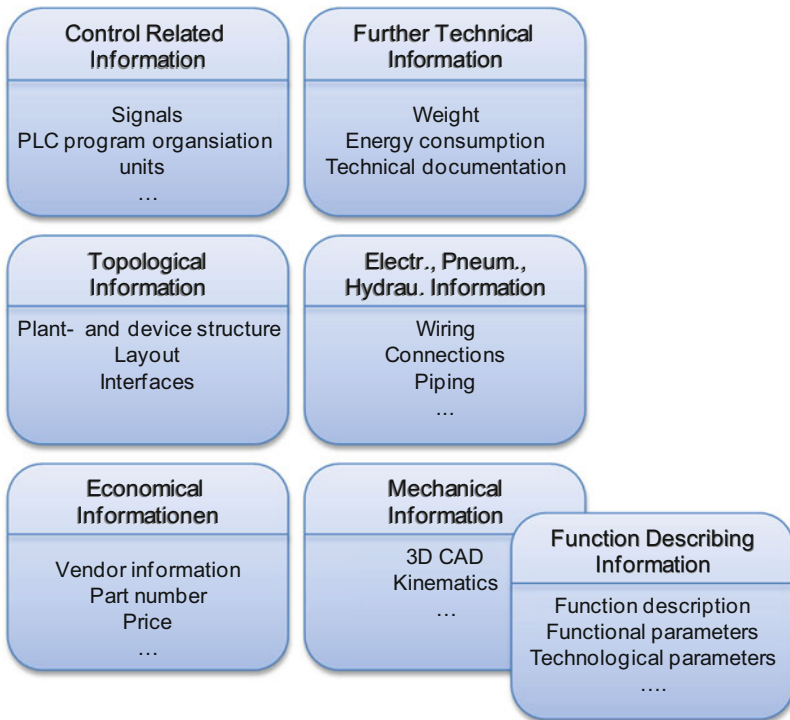


Fig. 5 Information sets of a mechatronic information object

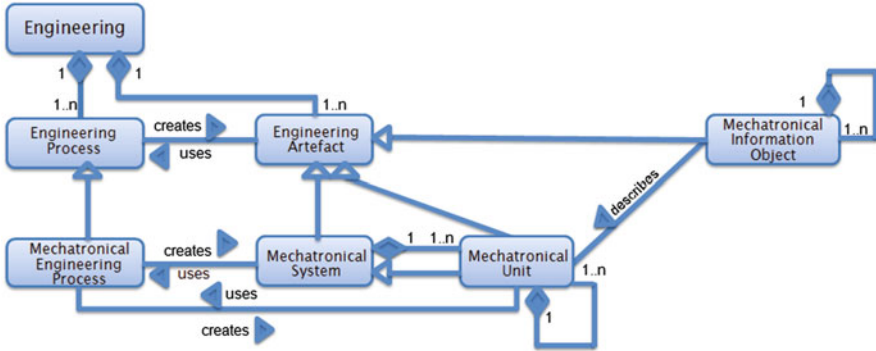


Fig. 6 Relations between mechatronic related terms defined

Within mechatronically structured production systems the control applications for production system automation are distributed among the different information processing components of the different mechatronical units and mechatronical systems. This distribution can either be a physical distribution on different control devices or a virtual distribution on the same hardware but with different execution contexts. Thereby, the control decisions executed on the different information processing units are oriented on the automation pyramid layers they belong to. Hence, on site level enterprise resource planning (ERP) decisions and control functions are executed, on cell and main function level manufacturing execution control (MES) decisions and functions are relevant, and on the different function layers field control decisions are made [18].

For the design and engineering of the automation and control applications (as well as for the complete production system design and engineering) it is useful to specify a stable interface structure for the information process units as depicted in Fig. 7 [12].

Mechatrical systems of higher layers of the production system hierarchy can access lower layered mechatrical systems using their own device interface and the execution interface of the lower layered mechatrical systems. By their execution interface lower layered mechatrical systems will provide access points to their provided production functions (or parts of it) which can be accessed by other mechatrical systems. Higher layered mechatrical systems know the required lower layered production function and can access and parameterize them appropriately by their device interface. Thus, a distributed but clear control decision hierarchy can be established.

This hierarchy is especially applied at field and MES layers of the control pyramid. Here the control application is split into components related to physical properties of production functions as depicted in Fig. 8 [19]. At the lowest level the function blocks are related to the direct physics control similar to the drivers within PC operating systems. They are responsible for operating control devices, i.e. they are relevant on the sub-function group layer of Fig. 4. Above them there

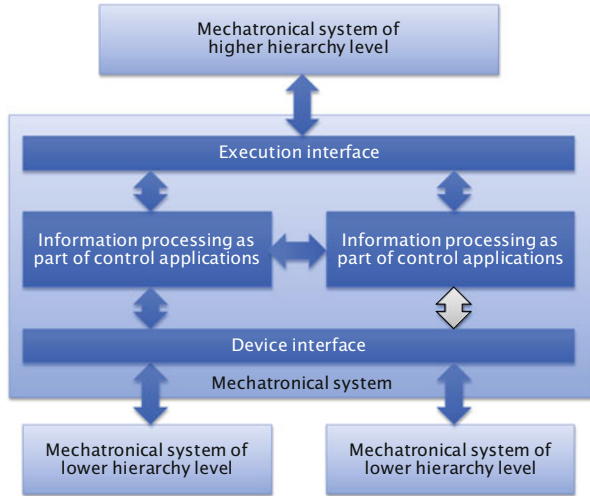


Fig. 7 Interface structure of mechatronic systems [12]

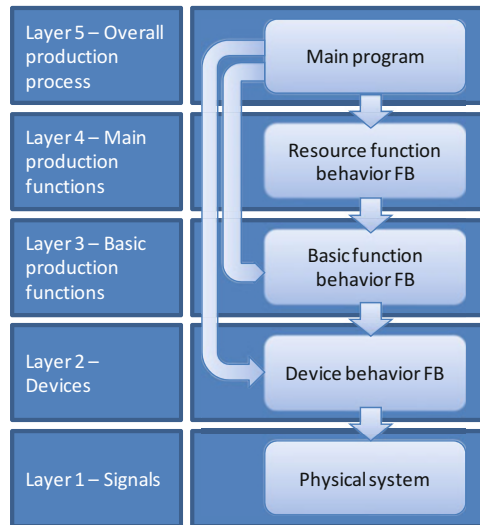


Fig. 8 Possible control hierarchy at field layer

are layers combining control devices to basic functions of the controlled system like moving a work piece until a sensor indicates its presence or move a robot into a certain pose. They belong to the function layer of Fig. 4. These function blocks are again aggregated to production system functions applicable to execute a certain production step of a product like make a welding point at position X or make a set

of welding points. Thus, these function blocks belong to the main group or the cell layer of Fig. 4.

2.2 *Mechatronical Engineering of Production Systems*

The engineering of mechatronically structured production systems is executed with direct application of mechatronical units¹ in the structure as described above [8, 20, 21].

It can be observed, that there are two main processes to be distinguished (see Fig. 11). The first process is focused on the design, engineering, installation and commissioning of a production system intended for a special production purpose (i.e. able to produce a special product portfolio) and can be considered as project dependent engineering creating a solution for a special customer. In the course of this process mechatronical units and mechatronical systems (or parts of them) are exploited as starting points taken from a library of reusable mechatronical units.

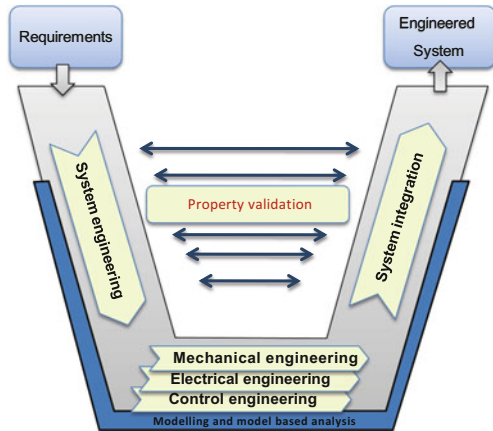
The second engineering process serves the design, engineering, and test of reusable mechatronical units and their integration in the named library (it should be reminded that this library is not a single entity but a distributed one exploiting different storing and management technologies) and can be considered as project independent engineering creating reusable engineering artefacts independent from customer orders. These mechatronical units can be exploited within the project dependent engineering process. The design, engineering, and test of reusable mechatronical units is based on the abstraction of engineering results of the project dependent engineering process under inclusion of expert knowledge about the industrial domain the intended production systems should belong to (see [22]).

Thus, the project dependent engineering process is assigned to the engineering, implementation and use of production systems. There are structure guidelines developed for this engineering process by research projects or applied in practice (see for example [23–27]). All of them have more or less the same background of systems engineering as applied in the SysMod methodology [28]. If these processes are applied to production systems, they follow a structure presented for mechatronical engineering in [8].

This engineering process starts with the collection of requirements of the production system to be engineered. These requirements emerge on the one hand from the product portfolio to be produced, i.e. the bill of operations to be executed on the bill of material of the intended products. On the other hand there are several requirements coming from legal entities like human and environmental safety or from economical considerations (increase of earnings). Based on these requirements the overall production system is engineered in a top down decomposition approach

¹In the following the term mechatronical unit will be used also as representative for mechatronical systems.

Fig. 9 Engineering process following [8]



finally resulting in a component structure to be applied in the system engineering phase. If the overall system structure is defined the different involved engineering disciplines (at least mechanical, electrical, and control engineering) have to execute the detailed engineering resulting in a detailed description of the system components. Afterwards the in detail developed system components are composed in the system integration phase and their properties are validated with respect to the initial requirements resulting in the final engineered production system. The system and detailed engineering as well as the system integration and property validation are usually accompanied by activities of modelling and model analysis to assist the engineering. This V-Model like process is depicted in Fig. 9.

Important for the mechatronical engineering of production systems are the system engineering and the system integration phases. Within these phases, at first the production function to be executed is decomposed to a function hierarchy. Therefore, the production steps to be executed are analysed and decomposed to sub-steps following the idea of main functions, functions and sub-functions executable by mechatronical units of the production system hierarchy given in Fig. 4. If necessary the identified production process related functions are assisted by auxiliary functions required to enable the production functions. In each decomposition step it is analysed if there are possible realisations of the functions of interest within the mechatronical library, i.e. are there solution elements for the required functions. If so these solution elements are assigned to the functions. Thereby, in parallel to the function hierarchy also a solution structures is developed [29, 30].

An example of this decomposition is given in Fig. 10. Here a welding cell for car body manufacturing is considered. This welding cell is dedicated to execute a set of spot welding steps following the assembly sequence of the car body. Thus, for welding the welding function is required but also clamping functions and transportation functions within the welding cell. To have the material to be welded in the welding cell it has to be inserted into the cell in an insertion area and to be transported to the insertion area. All these necessary functions are given by

dark blue boxes in Fig. 10. For each function applicable technical realizations are available given in light blue in Fig. 10. For example the welding can be executed by a welding gun either mounted on an industrial robot or on a static welding station. In case of a robot mounted gun the material has to be fixed on a geo station while in case of a static welding station the material shall be clamped and moved by a robot to and within the welding station. There are several human based or human free realization possibilities for material insertion and material provision as well.

The system integration phase can start if at least one possible solution element is identified for all leaves of the function hierarchy by either selecting it from the mechatronics library or by developing it from scratch within the detailed engineering. Then, the different system components are combined, connected, validated, implemented, and commissioned.

As named above, within this activity mechatronics units are used as input from a mechatronics library. On the one hand they are an input to the system function decomposition and solution element identification. In addition, they are applied and sometimes adapted in the detailed engineering, implementation, and commissioning of the production system providing necessary engineering artefacts for these phases.

If a project dependent engineering process for a production system is finalized the engineering results can be considered for identification of reusable system elements (i.e. mechatronics units). The system elements of the developed system are evaluated against customer and market requirements as well as technological progress expectations. Thereby, mechatronics units are identified, separated, completely engineered, possibly realized and tested, and finally added to the mechatronics unit library (Fig. 11).

As an example for this process the engineering of the control architecture within a production system shall be considered in detail.

Within the system engineering phase a hierarchy of mechatronics units is identified realizing the necessary production process. Following the control architecture of a mechatronics unit depicted in Fig. 7, the interaction of mechatronics units depicted in Fig. 3, and the hierarchy of mechatronics units given in Fig. 8, to each

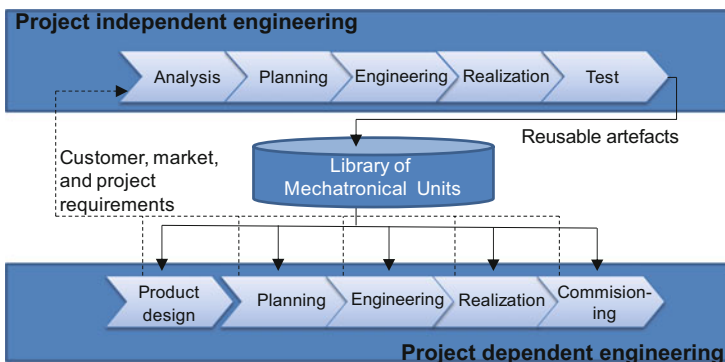


Fig. 11 Two processes within mechatronics engineering following [20]

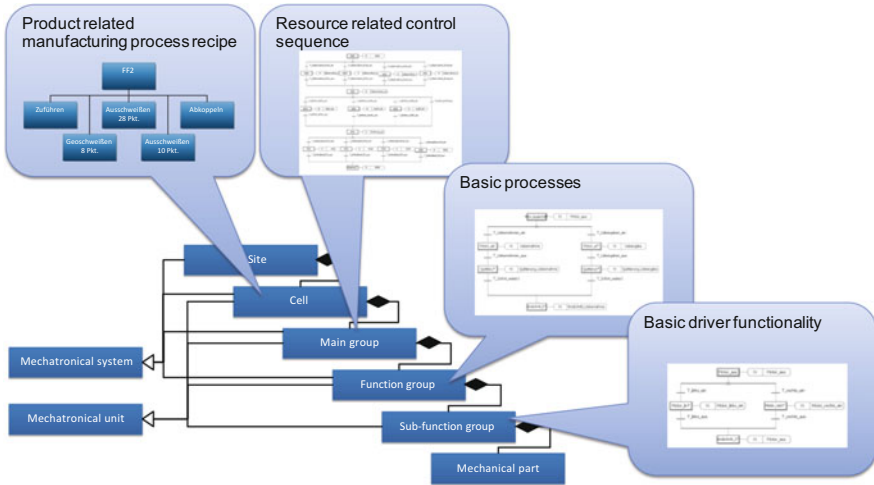


Fig. 12 Hierarchy of control application

of the layers of the identified hierarchy of mechatrical units a dedicated control application is assigned to.

At the cell layer this part represents the sequence of production process steps to be established in the cell. In the welding cell example these are the material transport and the welding steps. The cell layer exploits the resource related control sequences of the different main groups (e.g. robots, conveyers, etc.) creating the necessary sequences of the main groups (e.g. move robot to a position, transport material to a position, close clamping fixture). The main groups exploit the function groups and their control in the same way and finally the function groups exploit the sub-function groups representing the lowest layer of control accessing the sensor and actuator devices of the production system (e.g. start motion of a drive, read state of inductive proximity switch). This control code hierarchy is depicted in Fig. 12.

Having the necessary mechatrical units with their control code for the defined control applications within the mechatronic library the control code design step within the detailed engineering as well as the validation of the integrated system behaviour becomes easier. Here model based engineering actions as described in [26, 31–33] (for control engineering) and [34–36] (for control system validation) can be applied to name only some examples.

Based on several successful engineering processes the provider of production system components of the different layers of the mechatrical hierarchy can identify design pattern for component control. Usually, similar applications are grouped to classes of application as identified for drive applications in [37, 38]. The 12 identified application types are presented in Fig. 13.



Fig. 13 Different application types as identified by Lenze [38]

3 Existing Benefits of Mechatronical Engineering

A first benefit of the mechatronical engineering is obvious, the reuse of existing engineering artefacts as envisioned by [22] or [24]. After executing a project dependent engineering process appropriate project parts and/or system components can be identified and treated for reuse and integrated in the library of mechatronical units/systems in a project independent engineering process. Within the next project dependent engineering process requiring similar solutions these mechatronical units can be applied. Thereby, engineering effort in detailed engineering, validation, installation, and commissioning can be reduced providing a shorter engineering process with better tested components and less errors, and, finally, a better project quality.

The second benefit is related to the problem of cooperation of several engineering disciplines within the engineering process. As envisioned in VDI 3695 [20] a common architecture for the work of the different engineers involved in an engineering process is improving the process quality and efficiency. It will provide a kind of a common dictionary containing common system element types (with different discipline dependent views), a common system structure (plant hierarchy), and, finally, the ability of identification of common object entities in the different disciplines.

Beyond the common vocabulary avoiding misunderstandings between the involved engineers the data management within the engineering process can be improved drastically. Mechatronical engineering can be exploited related to data base based data exchange within the engineering process with common object semantics as discussed in [39] and related to exchange data format based engineering chain setup as discussed in [6]. This will result in improved tool chains (all disciplines crossing data exchange) providing the ability of lossless and consistent data exchange among involved engineering tools.

The fourth benefit is less visible as it is not related to engineering artefacts of the engineering chain. Within the system engineering phase the production system is initially considered from a function oriented point of view independent from the different possible technical realisations of the production system. The function hierarchy is developed. To each relevant function the set of possible realisations is assigned. Using this assignment, theoretically, the complete set of possible plant structures can be developed and the optimal one is selected. Despite the fact, that this optimization problem is not solvable realistically, it is possible to identify a set of meaningful candidates for final realization and discuss benefits and drawbacks of the realizations based on a more abstract level. This approach has been successfully applied within graduation activities of the research institute of the authors in the fields of welding cells in car body manufacturing, cutter systems in roller mills, stone mills, punching systems for metal sheet processing and robot gripper design to name only some examples.

4 Existing Challenges

To realise the named benefits mechatronical engineering provides some challenges to be solved to enable the successful application of this engineering methodology.

At the beginning of the engineering process (within the system engineering phase) appropriate mechatronical units/ systems have to be identified applicable as technical realisation of required functions of the production system. In the case of a body welding shop for car manufacturing for example, the engineering process is given as a set welding points to be made on special steel geometries to weld them together. Here libraries of mechatronical units can assist engineers by providing best practice system components. In the welding case the welding point structure and the steel part geometry for example can call for a special welding gun which will be provided as mechatronical unit or a complete welding robot consisting of a 6-axis robot with welding gun and cable-hose assembly. But these mechatronical units have to be modelled appropriately and should be automatically selectable based on relevant differentiating factors (in the car welding case for example the welding current and the gun size). The selection process requires a comparison of the required function and the functional capabilities of the mechatronical unit/system (like welding with welding parameters like temperature and material types). In addition, the general conditions of the usability of the mechatronical unit have to

be evaluated. It is an ongoing discussion how the required and provided capabilities of mechatronic units can be modelled. There are first existing ideas based on the generic description of manufacturing functions as described in [29, 40–42].

Theoretically, if a set of possible mechatronic units has been selected for the different elements of the function hierarchy the optimal set of mechatronic units/systems can be calculated. The optimization methodologies required for this problem could be a kind of a linear program integrating the capabilities of the mechatronic units, their mutual exclusion and dependencies, and the economical effect the application can have (costs, throughput, etc.). Similar ideas for a manufacturing process flexibility based optimization in the field of scheduling have been presented in [43]. It needs to be regarded that this optimization problem may suffer from the theoretical size of a production system and its hierarchies. In [44] a nine layer hierarchy of production system components has been proposed applicable for the modelling of a production system in the automotive industry. On each of these layers mechatronic units can be found leading to a capability of reusing mechatronic units on these layers. Up to now there is no stringent mathematical modelling available covering this optimization problem (Fig. 14).

Having a system structure defined, the involved engineers of the different engineering disciplines (including the control engineer) have to execute a detailed engineering providing the detailed description of the production system to be built. During the process flow of the engineering the made descriptions get more and more detailed exploiting different types of descriptions ranging from high level/abstract models over more detailed models down to implementable code and detailed drawings. In the case of a welding shop for car body manufacturing the set







	Layer	Example	
9	Production network	VW cooperation	
8	Factory	Golf 7 production system	
7	Production Line	body shop line	
6	Production Line Segment	vehicle body line	
5	Work Unit	vehicle body plant	
4	Work Station	welding cell	
3	Function Group	welding group (robot, controls, welding equipment)	
2	Component	welding gun	
1	Construction Element	welding cap	

Fig. 14 Production system hierarchy following [44]

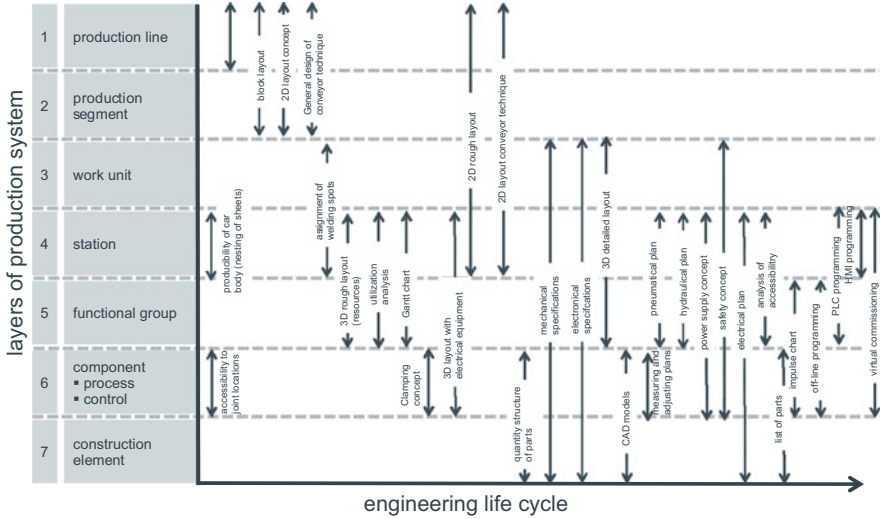


Fig. 15 Engineering artefacts required within welding shop engineering for automotive industry (selection)

of engineering artefacts is larger than 30. Figure 15 gives an overview how they are assigned to production system layers.

All created artefacts (models, drawings, etc.) represent the same system and shall be consistent to each other crossing engineering disciplines and levels of detail. For example, the mechanical, electrical, pneumatical and hydraulic models need to be in line with the 3D layouts and the part lists. In addition some of its characteristic properties will depend on each other. If, for example, a drive defined in the mechanical engineering has a special electrical interface this interface need to be connected in the electrical engineering were the connections enable the necessary ampere and volt of the energy flow to the drive. An example of first approaches addressing this problem is given in [39].

Needless to repeat, the availability of models within the engineering process opens up the complete box of the Pandora for model driven engineering. A survey of model driven engineering for distributed control systems is given in [45] as an example.

If the overall system is engineered in detail the correctness of the engineering can be validated based on virtual commissioning approaches. Therefore, the different models of different engineering disciplines need to be combined and a joint system simulation has to be executed [46]. Thereby, an adequate combination of models of production system physics and production system control at different layers are required. To come back to the welding shop example the physical stability of steel parts and its twist based on gravity forces within the material holders may have an effect on the necessary positions of material fixtures as well as the positioning of the welding gun. This information is only available if material physics is also considered

in the simulation model. As the different modelled objects are of different nature (continuous vs. discrete event, abstract vs. detailed, physical vs. logical, etc.) the models to be combined are very heterogeneous and their simulation requires various simulation strategies and tools.

5 Open Research Questions

Looking on the challenges of mechatronical engineering (which are far not new) still a set of open issues for mathematical research can be identified improving the applicability of mechatronical engineering.

As the engineering of production systems gets more and more model driven, there is an increasing need for methodologies following the model driven thinking. Models have to be created, processed, applied for generation of other (often more detailed) models or other descriptions and finally be executed (for example in controllers). In addition these models follow a multi-model approach with different usually overlapping models of the different involved engineering disciplines.

Research Question 1 Mathematical research can assist production system engineering by improving the capabilities for model generation, model transformation, model integration and model consistency management. Especially the crossover between models of different model nature (discrete event, continuous, hybrid ...) as well as models of different disciplines is of interest which has to be based on a common meta modelling approach for production systems.

One essential part in mechatronical engineering of production system is the automatic selection of potentially mechatronical units implementing special production system functions. Therefore, models of manufacturing functions are required applicable for automatic comparison of provided and requested functions. First approaches like [47–51] need to be extended and enriched with respect to expressiveness to be applicable in optimization methods.

Research Question 2 Mathematical research can assist production system engineering by enhancing production function models towards applicability in comparison and optimization methodologies.

At the end of the engineering of production systems more and more virtual commissioning methodologies are applied enabling a validation/verification of production system properties. Therefore, the created system of engineering artefacts (models) need to be combined appropriately and executed in simulation systems. Currently the simulation is only possible for limited sets of models over limited model sizes.

Research Question 3 Mathematical research can assist production system engineering by enhancing model combination and model simulation/co-simulation strategies improving the applicability of virtual commissioning to larger and more complex systems.

Facing the named challenges can be a task for a joined effort of mathematical and engineering science research. This paper will explicitly appeal interested researchers to cooperate under the roof of the Industrie 4.0 approach.

References

1. Bauernhansl, T., ten Hompel, M., Vogel-Heuser, B. (Hrsg.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer (2014)
2. Kagermann, H., Wahlster, W., Helbig, J. (Editoren): Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 – Deutschlands Zukunft als Industriestandort sichern, Forschungsunion Wirtschaft und Wissenschaft, Arbeitskreis Industrie 4.0, http://www.plattform-i40.de/sites/default/files/Umsetzungsempfehlungen%20Industrie4.0_0.pdf
3. Verein Deutscher Ingenieure e.V. - VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA) Fachausschuss “Industrie 4.0”: Industrie 4.0 - Gegenstände, Entitäten, Komponenten, Status report, April 2014, <http://www.vdi.de/technik/fachthemen/mess-und-automatisierungstechnik/industrie-40/>, last access Feb 2015
4. VDI/VDE – GMA Fachausschuss 7.21 “Industrie 4.0”: VDI-Statusreport Industrie 4.0 – Wertschöpfungsketten, VDI, Frankfurt/main, http://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/VDI_Industrie_4.0_Wertschoepfungsketten_2014.pdf
5. Lüder, A., Foehr, M., Hundt, L., Hoffmann, M., Langer, Y., Frank, St.: Aggregation of engineering processes regarding the mechatronic approach. In: 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011), Toulouse, France, September 2011, Proceedings-CD
6. Hundt, L., Lüder, A.: Development of a method for the implementation of interoperable tool chains applying mechatronical thinking – use case engineering of logic control. In: 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012), Krakow, Poland, September 2012, Proceedings-CD
7. Drath, R., Fay, A., Barth, M.: Interoperabilität von Engineering-Werkzeugen. *Automatisierungstechnik*. **59**(7), 451–460 (2011)
8. VDI: Design methodology for mechatronic systems (VDI 2206). VDI, VDI-Gesellschaft Produkt- und Prozessgestaltung (2004)
9. Czichos, H.: *Mechatronik: Grundlagen und Anwendungen technischer Systeme*. Vieweg (2006)
10. Kiefer, J.: *Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau*. Schriftenreihe Produktionstechnik, Band 43 Universität des Saarlandes (2007)
11. Thramboulidis, K.: Challenges in the development of mechatronic systems: the mechatronic component. In: IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 624–631 (2008)
12. Lüder, A., Foehr, L., Wagner, T., Zaddach, J.-J., Holm, T.: Manufacturing system engineering with mechatronical units. In: IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–8 (2010)
13. Foehr, M., Leitao, P., Wagner, T., Jäger, T., Lüder, A.: Integrating mechatronic thinking and multi-agent approaches. In: 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012), Krakow, Poland, September 2012, Proceedings-CD
14. Harashima, F., Tomizuka, M., Fukuda, T.: Mechatronics - what is it, why, and how? An editorial. *IEEE/ASME Trans. Mechatron.* **1**, 1–4 (1996)
15. Tomizuka, M.: Mechatronics: from the 20th to 21st century. *Control Eng. Pract.* **10**(8), 877–886 (2004)

16. Panich, S.: *Mechatronic Systems: Foundations and Applications*. LAP Lambert Academic Publishing (2013)
17. Kiefer, J., Baer, T., Bley, H.: Mechatronic-oriented engineering of manufacturing systems taking the example of the body shop. In: 13th CIRP International Conference on Life Cycle Engineering, Leuven, Belgium, June 2006, Proceedings, <http://www.mech.kuleuven.be/lce2006/064.pdf> (2006)
18. Ferrarini, L., Lüder, A. (eds.): *Agent-Based Technology Manufacturing Control Systems*. ISA Publisher (2011)
19. Lüder, A., Schmidt, N., Rosendahl, R.: Behavior validation of production systems within different phases of the engineering process. In: 39th Annual Conference of the IEEE Industrial Electronics Society (IECON' 2013), November 2013, Vienna, Austria, Proceedings
20. VDI (Verein Deutscher Ingenieure): *VDI Richtlinie 3695 – Engineering von Anlagen – Evaluieren und Optimieren des Engineerings*. VDI Publisher, Düsseldorf, May 2009 (in German)
21. Lüder, A., Foehr, M., Hundt, L., Hoffmann, M., Langer, Y., Frank, St.: Aggregation of engineering processes regarding the mechatronic approach. In: 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011), Toulouse, France, Proceedings-CD (2011)
22. Maga, C., Jazdi, N., Göhner, P., Ehben, T., Tetzner, T., Löwen, U.: Mehr Systematik für den Anlagenbau und das industrielle Lösungsgeschäft – Gesteigerte Effizienz durch Domain Engineering. *Automatisierungstechnik*. **9**, 524–532 (2010) (in German)
23. Wagner, T., Haußner, C., Elger, J., Löwen, U., Lüder, A.: Engineering Processes for Decentralized Factory Automation Systems, *Factory Automation 22*, In-Tech Publ., ISBN 978-953-7619-42-8 (2010)
24. aquimo project consortium: aquimo – Ein Leitfaden für Maschinen- und Anlagenbauer. VDMA Verlag (2010)
25. MEDEIA consortium: MEDEIA – Model-Driven Embedded Systems Design Environment for the Industrial Automation Sector, www.medeia.eu (2008)
26. Pfrommer, J., Stogl, D., Aleksandrov, K., Schubert, V., Hein, B.: Modelling and orchestration of service based manufacturing systems via skills. In: 19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2014), Barcelona, Spain, September 2014, Proceedings-CD
27. Fava project consortium: Fava project homepage, <http://ifatwww.et.uni-magdeburg.de/FAVA/>, last access Feb 2015
28. Weilkens, T.: *Systems Engineering with SysML/UML*. MK/OMG Press (2008)
29. Gehrke, M.: *Entwurf mechatronischer Systeme auf Basis von Funktionshierarchien und Systemstrukturen*. PhD Thesis, University Paderborn (2005)
30. Winzer, P.: *Generic Systems Engineering*. Springer Vieweg, Springer-Verlag, Berlin, Heidelberg (2013)
31. Barbieri, G., Fantuzzi, C., Borsari, R.: A model-based design methodology for the development of mechatronic systems. *Mechatronics*, 2014, http://www.academia.edu/7728404/A_model-based_design_methodology_for_the_development_of_mechatronic_systems
32. Vogel-Heuser, B., Schütz, D., Frank, T., Legat, C.: Model-driven engineering of manufacturing automation software projects - a SysML-based approach. *Mechatronics*. **24**(7), 883–897 (2014)
33. Yang, C., Vyatkin, V., Pang, C.: Model-driven development of control software for distributed automation: a survey and an approach. *IEEE Trans. Syst. Man Cybern. Syst.* **44**(3), 292–305
34. Vyatkin, V., Hanisch, H.: Application of Visual Specifications for Verification of Distributed Controllers, E-systems and E-man for Cybernetics in Cyberspace, vol. 1, pp. 646–651. IEEE, Piscataway, NJ
35. Puntel-Schmidt, P., Fay, A., Riediger, W., Schulte, T., Köslin, F., Diehl, S.: Validierung von Steuerungscode mit Hilfe automatisch generierter Simulationsmodelle. *Automatisierungstechnik*. **63**(2), 111–120 (2015)

36. Soliman, D., Frey, G.: Verification and validation of safety applications based on PLCopen safety function blocks. *Control Eng. Pract.* **19**(9), 929–946 (2011)
37. Kiehl, E. (ed.): *Antriebslösungen - Mechatronik für Produktion und Logistik*. Springer, Berlin (2007)
38. Götz, O.: Engineering-Effizienz für Antriebs- und Automatisierungslösungen, Within: AutomationML – Fachexperten erklären das Format, Whitepaper, https://www.automationml.org/o.red/uploads/dateien/1391503893-SPS-Magazin_Whitepaper_AutomationML.pdf, last access Feb 2015
39. Moser, T., Biffel, S.: Semantic integration of software and systems engineering environments. *IEEE Trans. Syst. Man Cybern. Part SMC-C (Application and Reviews)*, Special issue on Semantics-enabled Software Engineering **42**(1), 38–50 (2012), IEEE, ISSN: 1094-6977
40. Mertens, M.: Verwaltung und Verarbeitung merkmalsbasierter Informationen: vom Metamodell zur technologischen Realisierung. PhD Thesis, RWTH Aachen (2011)
41. Hadlich, T., Diedrich, C.: Verwendung von Merkmalen für die funktionale Modellierung. *Automation Congress 2014*, Baden-Baden- Germany, June 2014, Proceedings
42. Deter, S.: Plug-and-Participate for Limited Devices in the Field of Industrial Automation. PhD Thesis, University Marburg, Germany (2003)
43. Kis, T.: Planning and Scheduling in the Digital Factory, KOMSO Challenge Workshop -Math for the Digital Factory, Berlin, Germany, May 2014, Proceedings
44. Zawisza, J., Hell, K., Röpke, H., Lüder, A., Schmidt, N.: Generische Strukturierung von Produktionssystemen der Fertigungsindustrie, 16. Branchentreff der Mess- und Automatisierungstechnik (Automation 2016), 07. und 08. Juni 2016, Baden-Baden. - Düsseldorf: VDI-Verl. (in German)
45. Yang, C.H., Vyatkin, V., Pang, C.: Model-driven development of control software for distributed automation: a survey and an approach. *IEEE Trans. Syst. Man Cybern.*, <http://www.vyatkin.org/publ/2013smcaYVP.pdf> (2013)
46. Lee, C., Park, S.: Survey on the virtual commissioning of manufacturing systems. *J. Comput. Des. Eng.* **1**(3), 213–222 (2014)
47. Kluge, S.: Methodik zur fähigkeitsbasierten Planung modularer Montagesysteme. PhD Thesis, University Stuttgart, Germany (2011)
48. Pfrommer, J., Schleipen, M., Beyerer, J.: PPRS: production skills and their relation to product, process, and re-source. In: 18th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2013), Cagliari, Italy, September 2013, Proceedings
49. ISA-95.com: ISA 95 technology description, <http://isa-95.com/technology-isa95/>, last access Feb 2015
50. VDI (Verein Deutscher Ingenieure): VDI/VDE Richtlinie 3682 - Formalisierte Prozessbeschreibungen. Beuth Publisher, Parts 1 & 2 (2014)
51. NIST (National Institute for Standards and Technology): Process Specification Language, <http://www.mel.nist.gov/psl/>, last access Feb 2015

Part II
Optimization of Production Lines

Physics-Based Simulation for Energy Consumption Optimization of Automated Assembly Systems in the Automotive Industry

Felix Damrath, Anton Strahilov, Thomas Bär, and Michael Vielhaber

Abstract In this chapter a simulation based approach for optimizing energy consumption of automated assembly systems in the automotive industry from a production planning perspective is presented. Employing innovative simulation capabilities, originating from the computer gaming industry, automated assembly system’s energy consumption is prognosticated and visualized in virtual validation procedures, based on its corresponding digital models. Potential energy efficiency improvement measures (EEIMs) gathered from different fields of application are identified and exemplarily tailored to specific automated assembly system’s requirements. Considerations for suitable EEIM implementation to create energy-efficient system designs are proposed. Ultimately, a case study for improving energy-efficiency of automated assembly systems including preliminary results is presented.

1 Energy Considerations of Automated Assembly Systems in the Automotive Industry

At first energy-efficient production is motivated with respect to automotive manufacturing in industrial countries. The generic structure of automated assembly systems in automotive industries and corresponding energy consumption are further outlined.

F. Damrath (✉) • T. Bär
Daimler AG, Wilhelm-Runge Straße 11, 89081 Ulm, Germany
e-mail: felix.damrath@daimler.com; thomas.baer@daimler.com

A. Strahilov (✉)
EKS InTec, Heinrich-Hertz-Straße 6, 88250 Weingarten, Germany
e-mail: anton.strahilov@eks-intec.de

M. Vielhaber
Saarland University, Campus Building A5 1, 66123 Saarbrücken, Germany
e-mail: m.vielhaber@mx.uni-saarland.de

1.1 Motivation and Introduction

Energy-efficient manufacturing becomes a topic of primary importance in the field of industrial production. Over the last decade cost for electrical energy procurement and electricity prices for industrial customers in Germany have significantly risen [1]. The European Union targets a reduction of primary energy consumption of 20% below 2007 levels by 2020 while Kyoto-II will further restrict CO₂-emissions [2]. Additionally, growing environmental awareness among customers boosts the demand of environmentally sound manufactured goods. Energy-efficient production system design has therefore significant impact on production costs and total CO₂-emission, respectively. As a consequence, streamlining industrial production in terms of energy consumption is worthwhile and economically beneficial.

Highly automated automotive manufacturing in industrial countries accounts for substantial amounts of energy consumption [3]. In order to achieve challenging cost saving targets and to fulfill political restrictions, many original equipment manufacturers (OEMs) have implemented multiple energy efficiency improvement measures on the shop-floor, e.g. [4, 5]. Nevertheless, energy efficiency is insufficiently considered in production system's planning phase due to a lack of appropriate features in digital planning and validation tools [6, 7]. Capabilities for energy consumption prognosis of the entire production system based on digital prototypes would enable production planner to ensure an energy-efficient production system layout in early system development and thus significantly reduce energy consumption, production costs, and CO₂-emission.

1.2 Automated Assembly Systems in the Automotive Industry

Automated production systems for automotive manufacturing are distinguished by field of application according to the automotive manufacturing stages, i.e. press-shop, body-shop, paint-shop, and assembly. Press-shop and paint-shop feature special structures and tools for metal forming and applying different layers of paint, respectively. In body-shop and assembly systems similar operations can be identified (joining, clamping, positioning, etc.), but automated assembly systems must handle and process multiple product variants. In particular, automated assembly systems are characterized by less standardization in terms of system design, layout, and implemented components in comparison to body-shop systems. A simplified overview of an automated assembly system structure is exemplified in Fig. 1.

From a systems engineering perspective automated assembly systems can be perceived as hierarchically structured mechatronic systems representing the bottom layers of the conventional automation pyramid. Centralized factory process control manages multiple signals including several Programmable Logic Controllers (PLCs) that control individual assembly cells [8]. A single assembly system features different (sub-)assemblies (i.e. turntables, robots, etc.) that are composed of individual components (e.g. sensors, actuators) exchanging signals with the

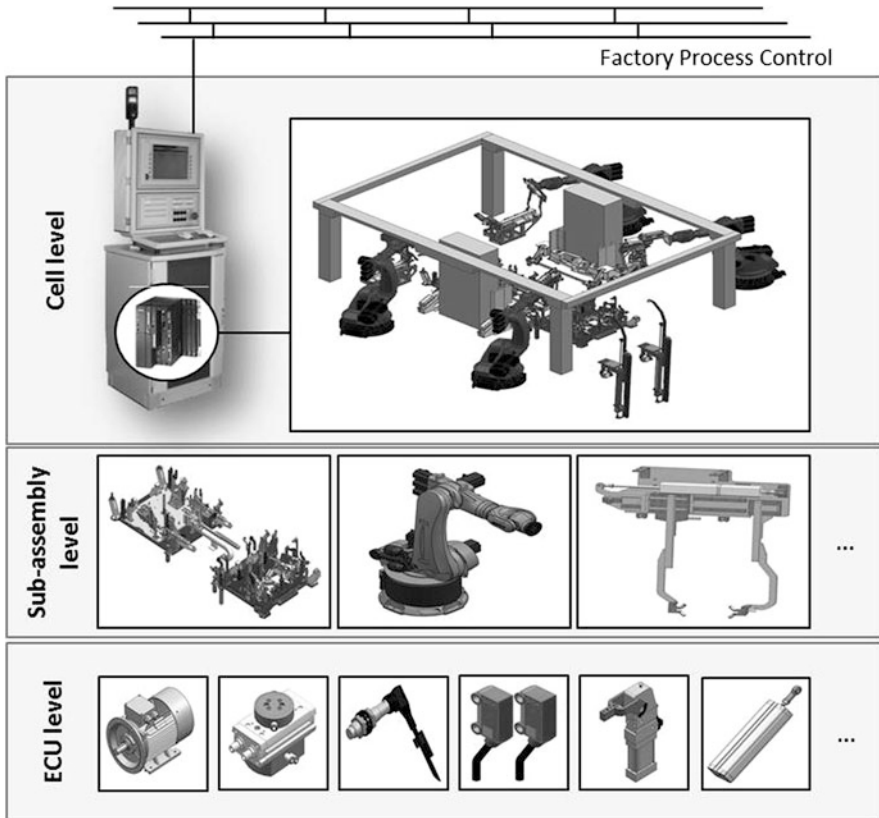


Fig. 1 Structural classification of automated assembly systems

PLC [9]. Those components are highly standardized but their arrangement and operation mode must be individually designed within the development process based on product requirements and specific assembly operations.

1.3 Fundamentals of Energy Consumption of Automated Assembly Systems

The entire energy value sequence comprises energy conversion from primary commercial, secondary, and final commercial to useful energy. Industrial end users are billed for final commercial energy consumption stipulated by complex contracts with energy suppliers. Final commercial energy accounts for all transformed products and sources, e.g. coal, gas, fuel oil, and electricity, whereas useful energy corresponds to the energy effectively made available through end user equipment, e.g. process-heat, mechanical work or lighting [10]. Despite this economic view,

in physics mechanical, thermal, electromagnetic and other forms of energy are differentiated [11]. State variable energy (E) is further linked with process variable work (W) that accurately specifies the energetic difference required for a system's change of states. Power (P) relates to work done in a certain period of time [12].

$$W = \int_t P dt \quad (1)$$

$$P = \frac{\Delta W}{\Delta t} \quad (2)$$

With respect to the automotive manufacturing stages, assembly accounts for almost one quarter of the entire energy consumption of an automotive production plant [13]. In automotive assembly mainly electrical energy (i.e. electrical work [Wh]) and pneumatic energy (i.e. compressed air volume [m³]) are the forms of useful energy required to execute value-adding assembly operations. Other forms of energy consumed in automotive assembly factory buildings (heat or gas) do not have significant impact on production system's energy consumption. Major consumers of electrical energy in automotive assembly are production equipment, lighting, and radiation system, whereupon the production equipment accounts for 23% of the entire electrical energy consumption [14]. Essential processes in automotive assembly such as transporting, positioning, fixing and joining of semi-finished parts are frequently recurring operations. Those operations are realized via electrical and pneumatic actuators such as electrical engines or pneumatic drives. Electrical and pneumatic components require energy to fulfill multiple assembly operations (clamping, driving, etc.) and are consequently designated as Energy Consumption Units (ECUs).

2 Virtual Validation of Automated Assembly Systems

Automated production systems in automotive manufacturing are virtually validated based on digital models prior to commissioning and ramp-up. Based on a generic assembly system development process, Daimler's virtual validation procedures Virtual Engineering and Virtual Commissioning are introduced. Furthermore, attempts of physics-based simulation of production systems with regard to Virtual Commissioning are presented.

2.1 Virtual Engineering and Virtual Commissioning

The development process of complex automated assembly systems for automotive manufacturing comprises several design phases, quality gates, and project

milestones. Reviewing the literature multiple schematic development processes of automated assembly systems can be found varying primarily in terms of design task allocation to different phases [15, 16]. Substantial similarities with the development of body-shop production systems can be identified [17]. The development of automated assembly systems presented in this paper arose out of several interviews with engineers and designers of four plant manufacturers [18].

At project kick-off the plant manufacturer (PM) starts conceptual design, i.e. systems engineering (Fig. 2) [19, 20]. Based on rough system layout set up by the OEM’s production planning department, integrative conditions and functions, e.g. external material flow, are determined and attested within *Conceptual Design Approval*. In mechanical design internal material flow, 3D geometric component modeling, and detailed process sequences are designed. In parallel with mechanical design, electrical design is initiated: all electric components are integrated in electric layouts specifying interconnections in electric diagrams. Software development for PLC and control programs and procurement are triggered after successful *Mechanical Design Approval*. Occasional system installation at PM’s site (Fig. 2A) is followed by *Installation Kick-off* at the OEM’s shop floor (Fig. 2B). Commissioning (C) and ramp-up (D) precedes *Start of Production* and final *System Release* [21]. Concurrently with system design virtual validation procedures are conducted in close collaboration with OEM’s production planning departments and PM’s

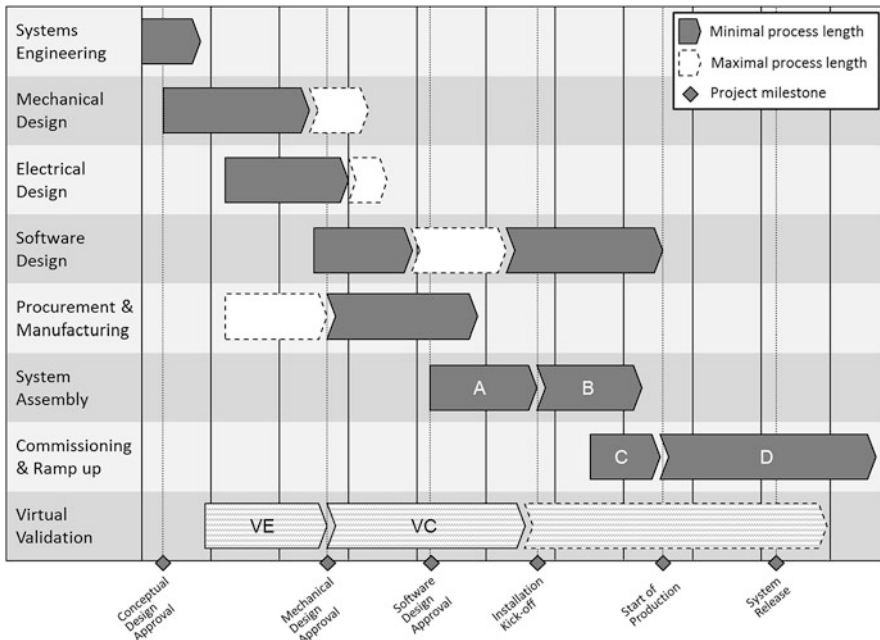


Fig. 2 Development process of automated assembly systems (based on [18])

engineers. Tools of the Digital Factory are used to ensure correct system behavior based on virtual 3D geometric system modeling and digital signal processing [22, 23].

At Daimler two virtual validation methodologies *Virtual Engineering* and *Virtual Commissioning* are well established for body-shop systems and are increasingly deployed for automated assembly systems. The methodology Virtual Engineering (VE) aims at visualizing and validating system designs in terms of process sequence, cycle time, and collision avoidance for different product variants by means of an extended *3D geometric model*. The methodology Virtual Commissioning (VC) utilizes *mechatronic system model* in order to validate PLC and control programs implementing hardware-in-the-loop. Both methodologies enable early evaluation, optimization, and validation of the entire production system based on its digital representations.

2.2 *Physics-Based Simulation of Production Systems*

Both validation methods Virtual Engineering and Virtual Commissioning require 3D geometric representations of the production system. Nowadays, several CAD/CAE tools from the Digital Factory are widely applied by practitioners in industry for virtual validation such as Delmia or Process Simulate [24]. However, process sequence design for complex production systems with those tools requires substantial engineering efforts, e.g. modeling simple transportation operations. Interactions between product and production system must be predominantly modeled individually. Furthermore, the majority of state of the art tools provide solely pure kinematic multi-body simulation features, thus not representing correct dynamic behavior of the production system with respect to relevant parameters such as mass, inertia, etc.

In order to reproduce multi-body dynamics in simulation environments with respect to real-time restrictions of virtual validation procedures, several research efforts focus on physics-based simulation, e.g. [25–27]. Physics-based simulation capabilities based on game engine technology feature physics-engines that approximate physical phenomena of rigid body dynamics, soft body dynamics and fluid dynamics (particles). Commercial or open-source physics-engines such as ODE, Bullet or PhysX are precompiled software libraries that iteratively solve differential equations based on Newtonian mechanics resulting in realistic object movement and interaction. Physics-based simulation of production systems for Virtual Commissioning has been subject of several research projects and is widely discussed among researchers and practitioners. Further attempts to improve and streamline physics-based Virtual Commissioning for the case of automated assembly systems in the automotive industry are currently undertaken in the ITEA-project AVANTI (Test methodology for virtual commissioning based on behavior simulation of production systems) [28].

3 Energy Consumption Optimization of Automated Assembly Systems

In recent years many efforts have been directed to reduce energy consumption of automated production systems resulting in measures to realize energy-efficient automotive manufacturing, e.g. [4, 5]. The majority of those energy efficiency improvement measures (EEIMs) are implemented on the shop floor during the production system's operating phase. Nevertheless, considering product development theory, exerting influence on design decisions in early system design leverages enforceable design adjustments and is financially worthwhile [29]. Consequently, implementing EEIMs already in early system design is continuously becoming more popular [30]. This contribution focuses on EEIMs that may potentially be implemented in early design phases of automated assembly system development. In particular, these EEIMs that can be implemented via virtual validation methods from a production planning perspective are given special attention. Prior to EEIM analysis a brief introduction to energy efficiency for manufacturing systems is given.

3.1 *Energy Efficiency of Automated Production Systems*

With regard to the energy value sequence (Sect. 1.3) this contribution focuses on optimizing useful energy consumption for efficient execution of assembly operations. There is no common definition for energy efficiency of automated production systems. The term energy efficiency is generic, it highly relies on considered system boundaries, and has different meanings depending on context, scale, and conceptual background, i.e. indicator for macroeconomic performance, energy conversion or energy demand [31, 32]. General definitions concern energy efficiency as a ratio of output performance goods, service or energy to total energy input. Furthermore, energy efficiency as quantifying indicator can solely be applied appropriately in comparison of two similar systems with identical required output specifications. As a consequence, two strategies to achieve higher energy efficiency can be identified, i.e. reduce energy input while keeping constant output, or increasing output while marginal energy input increase.

Energy efficiency labels have been established in recent years for several product families (e.g. automobiles, white goods, consumer electronics), categorizing products in energy efficiency classes based on significant product properties (e.g. size or operating volume) [33]. Those product families' benefit can be quantified easily, however, this is impossible for complex production systems in automotive industry. Especially for highly customized automated assembly systems, energy efficiency must be assessed individually due to limited number of comparable systems, insufficient standardization and high variety in system composition. With respect to the energy efficiency of manufacturing systems a physical-thermodynamic definition is

widely applied, considering energy efficiency as ratio of production output quantity (with designated quality) to the system’s total energy input [34, 35].

$$Energy\ Efficiency_{Manufacturing\ System} = \frac{Production\ Output}{Total\ Energy\ Input} \quad (3)$$

3.2 Potential Energy Efficiency Improvement Measures for Automated Assembly Systems

Based on the structural analysis of an automated assembly system (Sect. 1.2), elements from sub-assembly and ECU level that consume useful energy to execute assembly operations are considered for optimization. Here three types of sub-assemblies and components will be further analyzed: *Electrical motors*, *Pneumatic drives*, and *Robots* (Fig. 3). Additionally, component interactions within the assembly process need to be considered which results in the item *Process*. By means of literature review multiple EEIMs from different research fields could be identified and analyzed with respect to potential utilization for automated assembly system components. All EEIMs are listed without claiming completeness and assigned to different categories according to their specific field of action (Fig. 3).

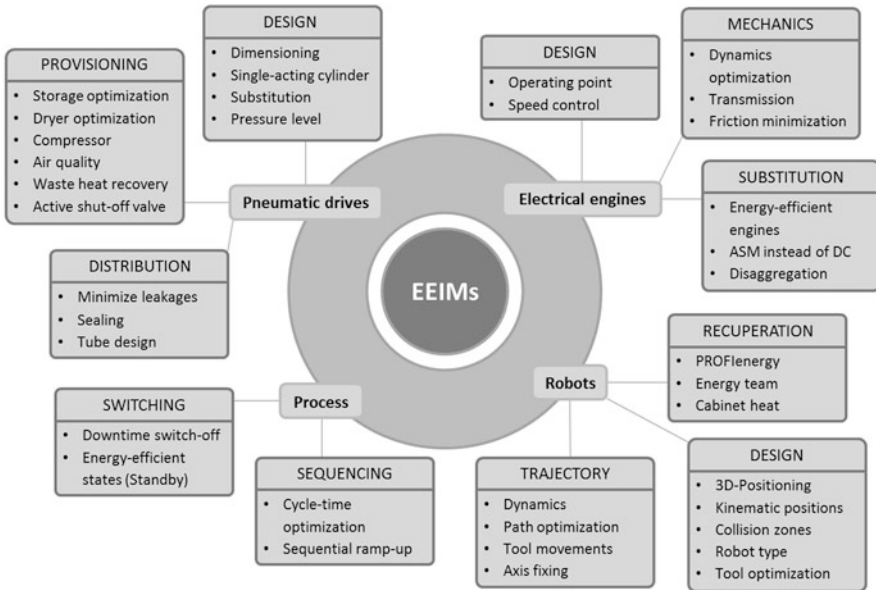


Fig. 3 Potential assembly system-oriented energy efficiency improvement measures (EEIMs)

Pneumatic drives (cylinders, clampers, etc.) offer multiple leverages of energy efficiency improvements. The category *Provision* encompasses all EEIMs along the entire process chain of compressed air provisioning on the shop-floor (creation, conditioning, distribution, and usage). In particular adequate compressor, dryer or storage system design contributes to energy savings in terms of reduced electrical energy consumption. Avoiding leakages and ensure proper sealing through sound and frequent maintenance exemplifies the category *Distribution*. The category *Design* is of paramount importance, e.g. ECU design decisions in terms of dimensioning and layout (size, pressure level, cylinder type, etc.) massively impact pneumatic energy consumption. Furthermore, tube design needs to be carefully considered for proper air flow in order to ensure ECU functionality.

Process refers to the potential optimization of ECUs due to timing conditions. *Switching* comprises the opportunities to switch robots and electrical engines to energy-efficient standby modes for longer standstill periods such as production breaks, weekends or idle time within the assembly process. The category *Sequencing* implies optimized timing of assembly operations (within cycle time) considering energy consumption, e.g. sequential ramp-up of originally parallel starting motors avoids expensive peaks in electrical energy consumption.

Regarding electrical engines several categories could be identified as well. Traditional physical measures are treated in the category *Mechanics*: decreasing moving masses, lower velocity or less friction reduce engine's electrical energy consumption. Via *Substitution* saving potentials could be realized by implementing energy-efficient motors or substituting DC machines by ASM (Asynchronous Motor). In terms of motor engineering and design with respect to energy efficiency it is very beneficial to run the engine in optimal operating point, thus avoiding idle time and partial payloads. This corresponds with the category *Design* by means of employing frequency converters or pole-changing engines to realize running in optimal operating point. Energy *Recuperation* could be realized via additional hardware (intermediate circuit) or generating and storing electrical energy by running the motor in generator mode. Further energy savings can be achieved by implementing reactive power compensation: energy suppliers design infrastructural equipment (transformers, cables) based on required apparent power to guarantee effective power, thus charging extra fees for over-engineered energy provision.

Energy-efficient use of industrial robotics is a broad research field where multiple EEIMs could be identified. The category *Design* aims at ensuring kinematic advantageous robot 3D positioning and realizing time saving potentials by reducing unnecessary robot work spaces overlap. Correct robot dimensioning considers adequate robot type and robot tool choice for the designated assembly operation. Many measures could be identified with respect to *Trajectory* optimization, e.g. optimized path generation as well as dynamics variations lead to significant energy savings. Energy *Recuperation* and synchronized energy exchange between multiple robots can be realized installing additional hardware components.

4 Implementation Strategy for Energy Consumption Optimization in Virtual Validation of Automated Assembly Systems

Building upon suitable EEIMs, the conceptual approach for energy consumption optimization is introduced and its preliminary implementation strategy is presented. A more detailed allocation of EEIMs to the respected validation procedures is suggested.

4.1 Conceptual Approach for Energy Consumption Optimization

Current virtual validation procedures of automated assembly systems do not feature physics-based simulation capabilities, thus providing solely kinematic multi-body simulation results (Sect. 2.2). Fundamental steps towards physics-based virtual validation of automated assembly systems and initial implementations to prove conceptual feasibility have been carried out [15, 27]. In addition, energy efficiency receives often no consideration within virtual validation procedures of automated assembly systems. Physics-based simulation capabilities enable simulation of mechanic energy consumption within physics-based virtual validation based on digital production system models [36]. Mechanical energy consumption on the one hand can be mapped to electrical energy consumption for electrical ECUs and on the other hand to pneumatic energy consumption in terms of pneumatic ECUs. Consequently, physics-based simulation in virtual validation delivers an energy signature based on digital models early in the development process of automated assembly systems. Conducting individual analysis of the respected assembly system, EEIMs shall be implemented in the digital model by minimizing required energy input while maintaining production output and important production parameters such as output rate, output quality or cycle time.

4.2 Implementation Strategy

The strategy for process implementation of the above mentioned conceptual approach encompasses three phases (Fig. 4). The first phase establishes an initial energy signature by means of physics-based simulation based on the original assembly system design. Energy signatures pinpoint energy consumption in terms of compressed air consumption and electrical energy consumption along the assembly process. *Physics-based 3D geometric model* and *mechatronic system model* of the entire assembly system must be created based on individual physics-based ECU

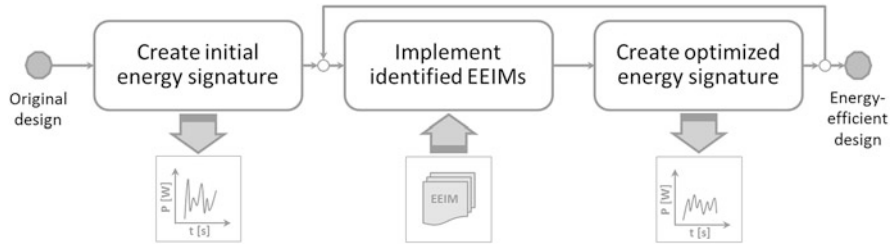


Fig. 4 Strategy to implement EEIMs into virtual validation procedures

modeling (Sect. 2.1). Individual component data is extracted from components' data sheets and sequence diagram is used for process modeling.

Within the second phase potential EEIMs are analyzed with respect to the assembly system's individual characteristics. As input a catalog of potential EEIMs is given and tailored for the application to the respected assembly system. Underpinning the initial energy signature, potentials for energetic optimization is determined, e.g. unexploited periods in cycle time, oversized components, unnecessary tool movements, etc. Those tailored EEIMs are implemented in the design of the assembly system and an optimized energy signature is created by means of physics-based simulation. Those optimization efforts must be implemented iteratively while continuously checking the updated system design for desired and undesired effects, e.g. reduced energy consumption shall be achieved while retaining invariant boundary conditions (cycle time, collision avoidance, etc.). The entire implementation strategy requires substantial manual engineering efforts and depends significantly on competences and experience of the system designer. The implementation strategy entails an energy efficient system design based on the original assembly system layout.

4.3 EEIM Allocation in Virtual Validation Procedures

PM's system designers and OEM's production planning department must be supported for the implementation of EEIMs into system design. Assigning EEIMs to the respected validation procedures and providing supplementary information required for EEIM's implementation are fundamental in order to enable engineers to create energy-efficient system designs.

Physics-based Virtual Engineering considers mechanical validation of the system design, i.e. sub-assembly motions are tested in terms of collision avoidance and operation time. All EEIMs targeting adjustments in mechanical design, dimensioning, process sequences, and layout shall be addressed in Virtual Engineering. Considering the example of optimal sizing of pneumatic drives, since many pneumatic drives are oversized due to excessive safety margins: the force required

to transfer a certain payload by means of a linear pneumatic cylinder is visualized and adequate dimensioning can be undertaken. The designer requires supplementary information about which smaller drive to choose and if the smaller drive's installation further affects the system's entire layout. With respect to robot repositioning in a layout to enable more energy-efficient axis-positions or motions entails switching its corresponding work space. The work space and adjusted trajectories shall be visualized to ensure proper improved system layout and avoid contrary energy consumption rebound effects. Layout adjustments and mechanical design changes must both be implemented in Virtual Engineering prior to *Mechanical Design Approval* (Sect. 2.1) that freezes mechanical design layouts. Subsequent changes in mechanical design layouts are not envisaged and cause substantial efforts in change management and lead to significant cost for modification.

Physics-based Virtual Commissioning examines correct logic behavior based on PLC and control programs, i.e. physics-based 3D geometric model is linked with individual logic control behavior models of individual system components and is connected to hardware PLC. All EEIMs not targeting mechanical designs and layout adjustments shall be addressed in Virtual Commissioning, e.g. optimized sequencing, code optimizations, etc. In the case of optimized sequencing several motors could be started sequentially to avoid expensive power consumption peaks. Although this approach can also be validated in Virtual Engineering, precise control signal triggering and exchange is required for its implementation that can only be realized within Virtual Commissioning.

5 Case Study: Opportunities and Preliminary Results

In order to demonstrate the feasibility of the presented approach, two case studies were conducted. Single EEIM implementation within Virtual Engineering and Virtual Commissioning is presented and preliminary results are quantitatively discussed.

5.1 Energy Consumption Optimization in Virtual Engineering

For demonstrating energy consumption optimization in Virtual Engineering an entire automated assembly system for front-axle positioning was modeled using physics-based simulation approach. The assembly process sequence consists of clamping, vertical lifting, turning, downshifting, and unclamping. For the energy optimization solely clamping is considered, which is realized via multiple pneumatic drives to fix the front-axle in a mechanical frame for smooth vertical lifting. Two linear cylinders are designated to move the frame's sub-assembly closer to the axle and are focused for optimization. Physics-based Virtual Engineering revealed that this positioning could also be executed by the use of downsized cylinders

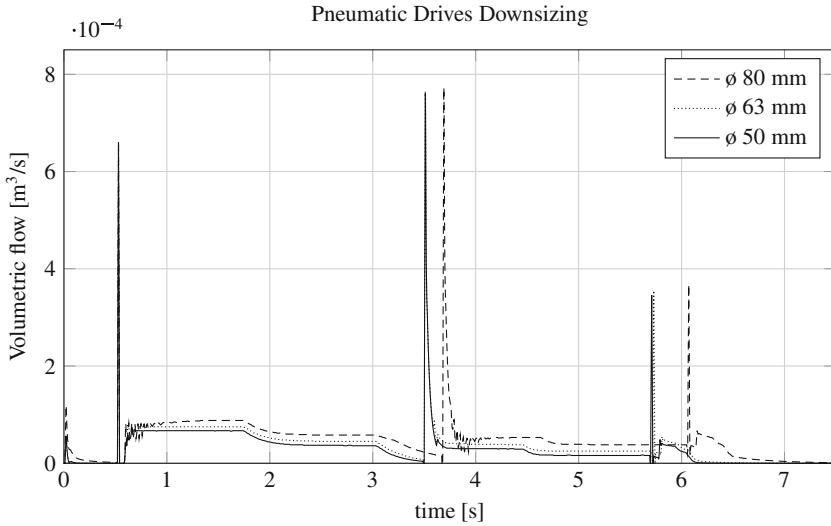


Fig. 5 Energy consumption optimization in virtual engineering

while maintaining pressure level of 10 bar. With respect to EEIMs classification, the measure *Dimensioning* of category *Design* is applied (Sect. 3.2). The diameter of both cylinders is successively downsized from originally 80 to 50 mm and corresponding compressed air consumption is visualized (Fig. 5).

Bullet physics-engine was used with accurate dynamics settings and simulation time-step was set to 10 ms. Assembly operation could be fulfilled with all diameter settings while deviations in compressed air consumption over time and total compressed air consumption could be determined. Implementing pneumatic drives with downsized diameter reduces compressed air consumption significantly and even shortens process duration. Total compressed air consumption of both pneumatic drives was reduced by approximately 37%.

5.2 Energy Consumption Optimization Virtual Commissioning

For exemplifying energy consumption optimization in Virtual Commissioning an assembly system with several motors was modeled employing physics-based simulation approach. For optimization only two asynchronous motors (ASMs) are considered that are executing independent assembly operations. In the initial system design both ASMs started concurrently, but assembly process analysis unfolds that ASMs can be started sequentially while still fulfilling assigned assembly operations. Referring to EEIM classification, the measure *Sequential ramp-up* of the category *Process* is implemented.

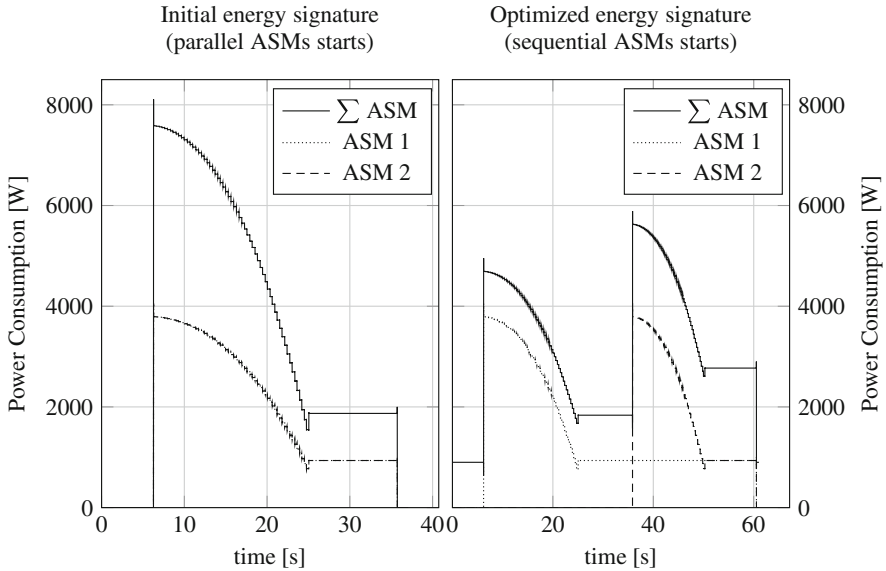


Fig. 6 Energy consumption optimization in virtual commissioning

Bullet physics-engine was used with accurate dynamics settings and simulation time-step was set to 10 ms. The initial design entails significant peak of approximately 8 kW of total power consumption due to parallel start of both ASMs (Fig. 6). Delaying one ASM's start prolongs total process duration but reduces power peak by 25% to approximately 6 kW. Since lower power peak was achieved by exploiting unused cycle time and avoid idling, energy saving potential was tapped.

6 Conclusions and Potential for Future Research

A brief summary of the contribution is given by resuming results and achievements. Furthermore, steps of ongoing and future research are outlined.

6.1 Summary and Achievements

The importance of energy consumption of automated production systems has significantly risen in recent years. Due to approaching shortages of fossil energy sources, pollution caused by energy generation, and increased environmental consciousness among customers many OEMs aim at reducing energy consumption of their production plants. While complex and highly automated production processes

in automotive manufacturing especially in industrial countries require substantial amounts of different energy sources, OEMs continuously strive to reduce investment and operating cost of their production systems even though an increased number of product variants must be processed at high quality standards within short cycle times.

Automated assembly systems can be considered as mechatronic systems implying several sub-assemblies and components that consume energy in order to execute assembly operations. Current approaches in the development of automated assembly systems in the automotive industry often neglect energy consumption as a design parameter due to the lack of digital tools to predict energy usage. Especially early phases of the development process bear significant potential to ameliorate the production system's design according to energy-efficient criteria. In particular virtual validation procedures, that proof the production system's functionality based on virtual models, offer several opportunities to modify important system parameters in order to create a more energy-efficient system design. Initial attempts to integrate physics-based simulation capabilities, originating from the gaming industry, in virtual validation procedures show promising results. Physics-based virtual validation procedures further enable energy consumption prognosis of automated assembly systems based on its corresponding virtual models.

Electric and pneumatic energy are predominantly consumed by components and sub-assemblies of automated assembly systems designated as ECUs. Energy efficiency of a manufacturing system can be described by relating total energy input to production output, baselining the initial energy signature for optimization. Based on a literature study several EEIMs could be identified from different fields of research and tailored to assembly system's characteristics. The presented implementation strategy transforms the initial system design to an energy-efficient system design by applying shortlisted EEIMs while iteratively validating system functions. Furthermore, EEIM allocation to the respected virtual validation procedure was suggested. Preliminary results were presented in two case studies implementing single EEIMs in Virtual Engineering and Virtual Commissioning, respectively. Significant energy savings could be achieved and quantitatively justified.

6.2 Outlook and Future Work

The contribution presents work in progress and thus ongoing research activities. Future efforts are directed to enable an exact energy consumption simulation of an entire automated assembly system thus establishing an encompassing energy signature via physics-based modeling. Modeling activities focus on adding new models and extending existing models for electrical engines. Quantitative and qualitative validation of electrical engine's power consumption calculation still remains. In addition, refining physics-based robot modeling with respect to proper dynamics and trajectory optimization is under development. Data acquisition for including constant power consumers (e.g. sensors) or joining technologies (e.g.

welding) into the energy signature is carried out. Challenges to be addressed in the future are the limitations of physics-based simulation approach for complex systems with respect to run-time and signaling or the integration of the suggested approach into OEM's tools and business processes.

Acknowledgements The authors gratefully acknowledge the support by the Federal Ministry of Education and Research within the project *AVANTI Test methodology for Virtual Commissioning based on behaviour simulation of production systems*.

References

1. VDA Verband der Automobilindustrie: Jahresbericht 2014, Berlin (2014)
2. Hesselbach, J.: Energie- und klimaeffiziente Produktion - Grundlagen, Leitlinien und Praxisbeispiele. Springer Vieweg, Berlin (2012)
3. VDI Zentrum Ressourceneffizienz: Analyse von Potenzialen der Material- und Energieeffizienz in ausgewählten Branchen der Metall verarbeitenden Industrie. VDI Zentrum Ressourceneffizienz, Berlin (2013)
4. TU Chemnitz: Spitzentechnologiecluster "Energieeffiziente Produkt- und Prozessinnovationen in der Produktionstechnik" (eniPROD). <http://www.eniproduct.tu-chemnitz.de/Cited20Jan2015>
5. Zschocke, C.: Energie-Effizienzcontrolling am Beispiel der Automobilindustrie. <http://www.eneffco.de/Cited20Jan2015>
6. Müller, E., Engemann, J., Löffler, T., Strauch, J.: Energieeffiziente Fabriken planen und betreiben. Springer, Berlin (2009)
7. Junge, M.: Simulationsgestützte Entwicklung und Optimierung einer energieeffizienten Produktionssteuerung. Produktion & Energie, Band 1. Kassel University Press, Kassel (2007)
8. Kropik, M.: Prozessleitsysteme in der Automobilfertigung. Springer, Berlin (2009)
9. Kiefer, J., Ollinger, L., Bergert, M.: Virtuelle Inbetriebnahme - Standardisierte Verhaltensmodellierung mechatronischer Betriebsmittel im automobilen Karosserierohbau. In: atp - Automatisierungstechnische Praxis, 07/2009, pp. 40–46. Deutscher Industrieverlag, München (2009)
10. Madureira, N.L.: Key Concepts in Energy. Springer, Heidelberg (2014)
11. VDI 4661:2003-09: Energiekenngrößen - Definitionen, Begriffe, Methodik. Beuth Verlag, Berlin (2003)
12. Stuart, H. A., Klages, G.: Kurzes Lehrbuch der Physik. Springer, Berlin (2010)
13. Müller, E., Löffler, T.: Improving energy efficiency in manufacturing plants - case studies and guidelines. In: Proceedings of the 16th CIRP International Conference on Life Cycle Engineering, pp. 465–471 (2009)
14. Imgrund, C.: Ganzheitliche Nachhaltigkeit in der Fahrzeugmontage. ATZ Produktion, Ausgabe 169, 44–48 (2011)
15. Strahilov, A.: Simulation des physikalischen Verhaltens bei der digitalen Absicherung von automatisierten Montageanlagen. VDE Verlag, Karlsruhe (2014)
16. Vielhaber, M.: Assembly Oriented Design - Zusammenbauorientiertes Konstruieren im Produktentstehungsprozess der Automobilindustrie am besonderen Beispiel des Karosserierohbaus. Schriftenreihe Produktionstechnik, Band 34, Saarbrücken (2005)
17. Kiefer, J.: Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau. Schriftenreihe Produktionstechnik, Band 43, Saarbrücken (2007)
18. Drescher, B., Stich, P., Kiefer, J., Strahilov, A., Bär, T., Reinhart, G.: Physikbasierte Simulation im Anlagenentstehungsprozess - Einsatzpotenziale bei der Entwicklung automatisierter Montageanlagen im Automobilbau. In: Dangelmaier, W. (Hrsg.) Simulation und Produktion und Logistik 2013, pp. 271–281. HNI Verlagsschriftenreihe, Paderborn (2013)

19. Brandis, R.: Systematik für die integrative Konzipierung der Montage auf Basis der Prinziplösung mechatronischer Systeme. Verlagshaus Monsenstein und Vannerdat OHG, Münster (2014)
20. Gausemeier, J., Lanza, G., Lindemann, U.: Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung. Carl Hanser Verlag, München (2012)
21. König, F., Betker, J.: Änderungsmanagement im Anlauf am Beispiel des Mercedes-Benz-Werks Bremen. In: Schuh, G., Stölzle, W., Straube, F. (Hrsg.) Anlaufmanagement in der Automobilindustrie erfolgreich umsetzen - ein Leitfaden für die Praxis, pp. 221–228. Springer, Berlin (2008)
22. Schak, R.-J.: Methodik zur bewertungsorientierten Skalierung der Digitalen Fabrik. Forschungsberichte IWB, Band 207. Herbert Utz Verlag, München (2007)
23. Bracht, U., Geckler, D., Wenzel, S.: Digitale Fabrik - Methoden und Praxisbeispiele. Springer, Berlin (2011)
24. Single, U.: Virtuelle Inbetriebnahme – Potential einer modularen Softwarearchitektur für eine effiziente virtuelle Inbetriebnahme. In: 9. Fachkongress Digitale Fabrik@Produktion, Berlin (2013)
25. Lacour, F.-F.: Modellbildung für die physikbasierte Virtuelle Inbetriebnahme materialflussintensiver Produktionsanlagen. Forschungsberichte IWB, Band 257. Herbert Utz Verlag, München (2012)
26. Spitzweg, M.: Methode und Konzept für den Einsatz eines physikalischen Modells in der Entwicklung von Produktionsanlagen. Forschungsbericht IWB, Band 233. Herbert Utz Verlag, München (2009)
27. Rossdeutscher, M.: Entwicklung eines Verfahrens zum Programmtest in der robotergestützten Montage. Shaker Verlag, Aachen (2011)
28. Bär, T.: Test methodology for virtual commissioning based on behaviour simulation of production systems. <http://avanti-project.de/>. Cited 20 Jan 2015
29. Pahl, G., Beitz, W.: Konstruktionslehre. Springer, Berlin (2007)
30. Bayerisches Landesamt für Umwelt: Energieeffizienz bei Planung und Betrieb von Anlagen. Bayerisches Landesamt für Umwelt, Augsburg (2012)
31. Peht, M.: Energieeffizienz - Ein Lehr- und Handbuch. Springer, Berlin (2010)
32. European Commission: Reference Document on Best Available Techniques for Energy Efficiency. EC Joint Research Centre, Seville (2009)
33. DENA Deutsche Energie-Agentur GmbH: Das EU-Energielabel - Entscheidungshilfe für Verbraucher. Bundesministerium für Wirtschaft und Energie, Berlin, (2013)
34. Thiede, S.: Energy Efficiency in Manufacturing Systems. Springer, Berlin (2012)
35. Engelmann, J.: Methoden und Werkzeuge zur Planung und Gestaltung energieeffizienter Fabriken. Wissenschaftliche Schriftenreihe des Instituts für Betriebswissenschaften und Fabrikssysteme, Band 71. Chemnitz (2009)
36. Damrath, F., Strahilov, A., Bär, T., Vielhaber, M.: Establishing Energy Efficiency as Criterion for Virtual Commissioning of Automated Assembly Systems. In: Proceedings of the 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems. Procedia CIRP, vol. 23, pp. 137–142, Dresden (2014)

Optimisation of Power Consumption for Robotic Lines in Automotive Industry

**Pavel Burget, Libor Bukata, Přemysl Šůcha, Martin Ron,
and Zdeněk Hanzálek**

Abstract A novel mathematical formulation of the energy optimisation problem for robotic lines is presented, which allows minimising the energy consumption in a robotic cell while keeping the required production cycle time. Different energy saving modes of the robots are utilised as well as the fact that the robot energy consumption during its movement depends on the movement duration. This dependency is modelled with a so-called energy function, which can be obtained by measurements, physical modelling of the robots or simulation. Each of these areas is covered by the presented work. The achieved results show there is a good potential to achieve energy savings at existing robotic cells and their series, and an even bigger potential if the presented approach is used during the design phase of new robotic cells.

1 Introduction

The utilisation of robots in production lines has become a very important aspect to increase the productivity, throughput and efficiency of the production. Especially car manufacturers have been investing big effort in obtaining reliable, precise and high-throughput robotic production lines. Hand in hand with devoting higher ratio of the production work to the robots and thus increasing the number of robots participating in the production, the amount of energy consumed by the robots increases as well. Therefore, it is of great importance to search for ways how to improve the energy efficiency of the robot operations. In this chapter the focus is put

P. Burget (✉) • P. Šůcha • M. Ron
Faculty of Electrical Engineering, Department of Control Engineering, Czech Technical University in Prague, Prague, Czech Republic
e-mail: burgetpa@fel.cvut.cz; suchap@fel.cvut.cz; ronmarti@fel.cvut.cz

L. Bukata • Z. Hanzálek
Faculty of Electrical Engineering, Department of Control Engineering, Czech Technical University in Prague, Prague, Czech Republic

Czech Institute of Informatics, Cybernetics and Robotics, Czech Technical University in Prague, Prague, Czech Republic
e-mail: bukatlib@fel.cvut.cz; hanzalek@fel.cvut.cz

on robotic welding lines, which already exist and which have been in production. Because of this reason, there are only limited possibilities in adding additional sensors or in performing changes in the robotic programs and in the programs of the superordinate controllers. However, such existing lines can still be improved in terms of their energy consumption. This contribution concentrates on methods how to optimise the robotic operations and how to get the necessary energy models of the robots that are needed for the optimisation. All this while keeping in mind the above stated requirements to utilise the existing production line infrastructure. Moreover, the underlying optimisation model together with the energy model of the robots are general enough to be used also during the design phase of new production lines.

The core of this contribution is a novel mathematical formulation of the energy optimisation problem for robotic lines. Contrary to the existing works the proposed solution considers different trajectories of robots, gravity and order of robot operations from the global point of view of the whole robotic cell. In fact, it may also be enhanced to a series of cells but it is out of the scope here. Moreover, the presented mathematical formulation takes into account the robots' power saving modes such as staying on brakes or "falling asleep", to which robot in a stationary position can switch to save even more energy. The optimal solution to the problem is the one which is both the most energy-efficient and meets the desired production cycle time.

The basic characteristics of the robots with respect to their energy consumption is their so-called energy function, which represents the dependency of the consumed electrical energy on the duration of given robotic operations. Such an energy function may be obtained using electrical power measurements on the real robots, by simulations or by analytical computations based on a physical model of the robots. Moreover the robotic operations must be clearly separated in order to get their boundaries to be entered into the optimisation model, as well as to be able to compare the energy function of the individual robots with their real behaviour.

The physical modelling of the robots is based on graphically-oriented computer-aided concept that exploits CAD software such as NX or Solidworks and simulation environment Matlab Simulink with SimMechanics and SimPowerSystems libraries. These software tools are used for the composition of a dynamical simulation model that represents both mechanical robot structure and robot drives during the robot motions. Thus the power needed for the robot movements can be obtained and the energy functions can be calculated.

The robotic operations may also be simulated in another environment, which is used to design and simulate complete robotic lines such as Process Simulate. Next to the design of the robot trajectories it is possible to simulate the robot controller itself if an appropriate Robot Controller Software (RCS) and Realistic Robot Simulation (RRS) modules are available. Recently, RCS and RRS modules that allow simulating the energy consumption have been provided by robot manufacturers.

1.1 Contribution

In this work the global optimisation of the robotic lines is devised with respect to the identified energy aspects which resulted from measurements and simulations. Compared with the existing works such as [22–25], the presented solution is more general by considering the robot power saving modes such as brakes, bus-power-off or hibernation, and different locations are taken into account where a robot operation can be performed. Moreover, the presented formulation enables the robotic line designers to specify path alternatives, i.e. selecting the best order in accordance with precedence relations. The achieved results have revealed that a significant energy saving is possible.

An important part of the optimisation model is the energy function of the robot movements, which allows choosing the optimal speed of the robotic operations and thus minimising the energy consumption of the robotic cell not only from a local point of view but also from the point of view of the cell or even series of cells. This work presents several ways how to obtain the energy function.

1.2 Related Works

The current research on energy optimisation of robotic lines can be categorised into two groups. The first one is the optimisation of individual robot trajectories with respect to physical limitations of robots and obstacles to be avoided. The second one, rarely occurring in the literature, is the optimisation of the robotic line as a whole using mathematical models. Both the groups are not necessarily disjunctive and there are a few papers dealing with both of the aspects.

As an example from the first group the following works are worth to be mentioned. In the work of Saramago et al. [16] both the time of the robot movement and mechanical energy consumed by actuators are taken into account. The multi-objective optimisation problem was solved by using the DOT program (Design Optimisation Tools Program) and tested on three and six degrees of freedom manipulator arms.

A real-time planning of energy efficient trajectories for the robot catching small flying objects was proposed by Lampariello [9]. The authors formulated the non-linear constrained optimisation problem, nevertheless, to be able to find good trajectories in a real-time the global planner was generalised using the learning methods, such as nearest neighbour, Support Vector Machines, and Gaussian process regression. The proposed approach has shown to be efficient on the ball-catching task.

Michna et al. [12] developed an algorithm for the generation of time optimal trajectories for wheeled robots. The trajectory is interpolated by the cubic Hermite spline curves and a speed profile is determined by the algorithm. To accelerate the calculation of collision-free trajectories the authors propose to use neural network. Nevertheless, the energy consumption is not considered and the approach was only tested on a hypothetical example.

The following works perceive the robotic lines as a whole to find globally good solutions. In the work of Mashaei and Lennartson [11] an energy model of the Pallet-Constrained Flow Shop problem was formulated to find an optimal switching control strategy leading to the desired throughput and minimal energy consumption. Idle states of machines were also taken into account to reduce energy consumption if the machine is not working. However, the model requires a line with special structure, i.e. closed-loop pallet system, and therefore it is not generally applicable to the robotic lines.

There are a few similar papers [22–25] focusing on both the local and global optimisation of the robotic lines. For example, in the work of Wigström et al. [25] a physical model of a robot with AC synchronous motors is created and optimal control problem, determining how to control the robot moving along the specified partial trajectory in an energy efficient way, was solved using Dynamic Programming. Nevertheless, the geometry path was fixed and the initial time optimal trajectory obtained from ABB Robot Studio was required. Afterwards, the locally optimised trajectories were used as an input for the global solver (Mixed Integer Non-Linear Programming) to find a solution that is energy efficient and satisfying demanded production cycle time. Although the model is the first model considering the global energy aspects, there are a few drawbacks limiting the possible energy saving—the robot power saving modes are not taken into account and different positions of robots during the work are not considered. The same authors provide more details about the formulation in [23]. As a verification of the energy-aware solution [24] suggests to use Hybrid Cost Automata.

Riazi et al. in [14] combine the optimisation of individual trajectories with ordering the robot operations. The decrease of energy consumption during movements is achieved by minimising the sum of square roots of accelerations over the trajectory. Worth mentioning is the fact that no model of the robot is needed because the acceleration vector is obtained by sampling the existing movements and the optimised trajectories are uploaded back into the robot afterwards.

As part of the movement optimisation in order to reduce the energy consumption it is necessary to calculate, measure or at least assess the energy consumption of the robot under different circumstances. Several methods exist that are based on mathematical analysis, i.e. on modelling the kinematics and dynamics of the robots such as in [3, 17] or [18] where a specific expression of the energy consumption equation in dependence on a given robot trajectory is presented. Papers such as [4, 5], or [21] focus on processing and analysing real data obtained from physical measurements.

The way how measurements can be done differs if the robot is in a laboratory environment or if the measurements must be done in a production environment on a robot that is usually part of a robotic cell. In such a case pattern matching or machine learning techniques are used to process the data and identify the robot operations. In [6] region-based segmentation stemming from frequency analysis of the original signal is used. Le et al. [10] relates to a state estimation and a corresponding energy audit of injection moulding machines and the focus is given to identifying the production state of the machine using a two-level neuron network to classify the

states. Paper [20] focuses on pattern recognition of 1-D signal in industrial batch dryer with a goal to slice the measured data of pressure into time windows of the periodic batch processing intervals using supervised learning of a Takagi-Sugeno fuzzy model. Ron et al. in [15] use 1-D pattern matching with correlation and feature extraction techniques.

1.3 Outline of the Chapter

The following sections provide the details about the individual parts of the energy optimisation problem. Specifically, Sect. 2 defines the problem formally and shows examples of a simple robotic line, schedule of operations and corresponding energy function for a robot movement.

Section 3 deals with different ways how to obtain data for modelling the energy function such as measurements of the power consumption on a robot in a laboratory, creation of a kinematic and dynamic model of a robot and its electrical drives to get an equation for the energy consumption, and how to simulate the robotic movements to get the energy function from simulation. A special attention is put to the identification of the robotic operations at a production line, which completes the whole picture in such a way that it is possible to evaluate the obtained energy functions of the robot movements with respect to their real energy consumption during production operations.

Section 4 shows how to use the energy functions to optimise the energy consumption. The mathematical model in terms of Integer Linear Programming is defined there and a way is proposed how to compute a lower bound using Lagrangian relaxation. Section 5 describes the results of the optimisation, which has been performed on generated problem instances as well as on an industrial use case from Škoda Auto car manufacturer. Section 6 summarises the results and concludes the chapter.

2 Problem Statement

The following aspects of energy saving at robots are crucial: (a) *selection of stationary positions* represented by different robot configurations, which take into account the robot energy consumption, (b) *power saving modes*, whose utilisation may result in significant energy demand decrease,¹ (c) *trajectory selection and alternatives* again with respect to their energy consumption, and (d) *speed of the movement*, which is dual to the duration of the movement. A detailed analysis supporting this statement is provided in Sect. 3.1.

¹The more energy-saving mode is the longer time is required to have the robot back in a ready-to-operate mode.

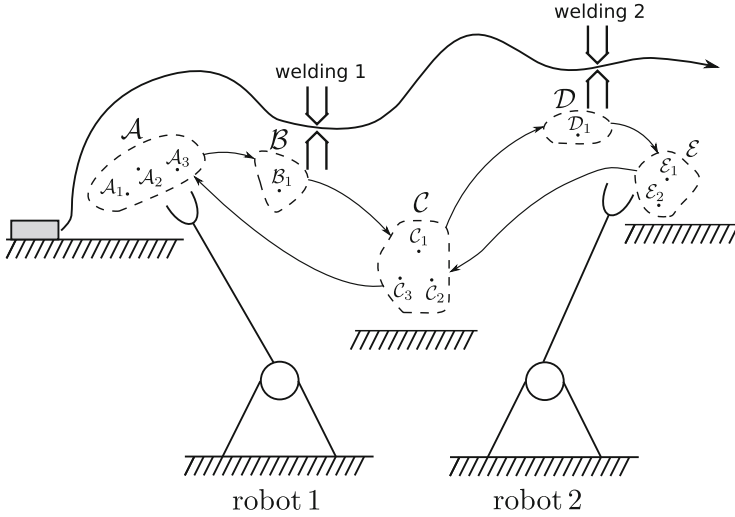


Fig. 1 Example of the line with two robots

Energy aspects are illustrated by a simple robotic line depicted in Fig. 1. In this example the first robot takes the weldment, performs a welding operation, and puts it on the bench where the second robot takes it, carries out a welding operation, and finally puts the weldment on the conveyor belt. In each subspace, denoted as a calligraphy letter without subscript (e.g. \mathcal{A} , \mathcal{B} , \mathcal{C} , ...), there are possible points (i.e. gun coordinates), in which the robot can conduct a task, e.g. welding, assembling, taking the workpiece, putting it on the bench, or handing it over to another robot. The points of a subspace are indexed, for example \mathcal{A}_1 , \mathcal{A}_2 , and \mathcal{A}_3 are points located in subspace \mathcal{A} . Between subspaces the robot can move in a direction indicated by the arrow that corresponds to the set of point-to-point movements. From the set no more than one movement is selected and the duration of motion and required energy is determined according to the energy function obtained either from the measurement or from the energy model of the robot. In Fig. 2 the measured movements corresponding to points were interpolated with function

$$f_E^{si}(d_i) = a_{-1}d_i^{-1} + a_0 + a_1d_i \quad (1)$$

where d_i is the duration of the movement, $f_E^{si}(d_i)$ consumed energy, and finally a_{-1} , a_0 , a_1 are coefficients calculated by e.g. the Gauss-Newton algorithm. The subscript letter E means energy and superscript letter s represents the fact that the function is parametrized by the trajectory the robot moves along. The function was empirically proposed using the following ideas and supported by the measurement results from Sect. 3.1. As the duration tends towards 0_+ the power consumption

Fig. 2 Relation between the duration of the movement and energy consumption

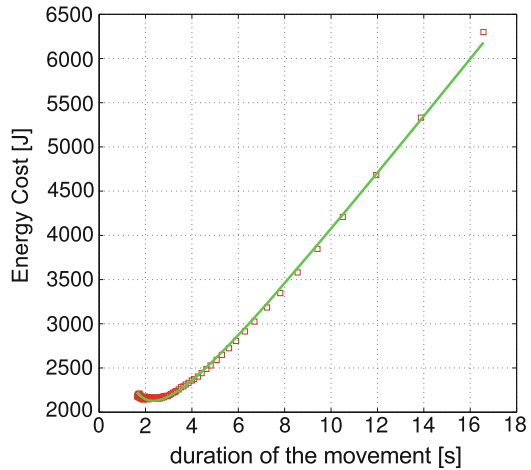
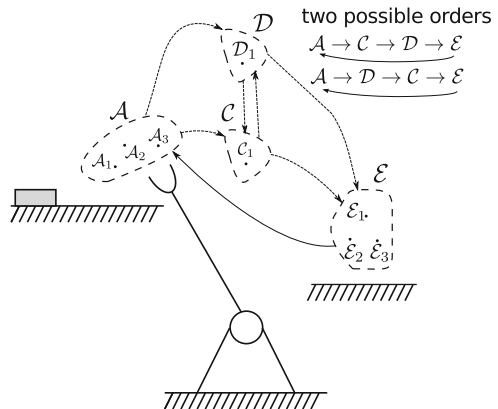


Fig. 3 Illustration of alternatives



increases to ∞ , i.e. due to the first term $a_{-1}d_i^{-1}$. On the other hand, if the duration is very lengthy, only the gravity part can be considered and the consumption increases linearly, which is represented by the third term a_1d_i . Finally, term a_0 is a constant offset of the function. The function is convex provided that coefficient $a_{-1} \geq 0$. It is also possible to use higher degrees of the approximation polynomial if it fits better the robot behaviour but it is subject to further evaluation in the particular case.

The alternatives are illustrated in Fig. 3. There are two possible paths that the robot can take and one of them is probably more energy efficient. However, both paths have a bit different timing (synchronisation between robots) and order of operations.

To demonstrate how the problem can be represented by a graph and how the final schedule looks, the robotic line from Fig. 1 was used for the example in Fig. 4, which shows the structure of the robotic line in terms of operations and movements. The dashed arrows are synchronisation edges \mathcal{L} , i.e. time lags, that ensure the correct handover of the weldment to the second robot using a turntable. The solid arrows

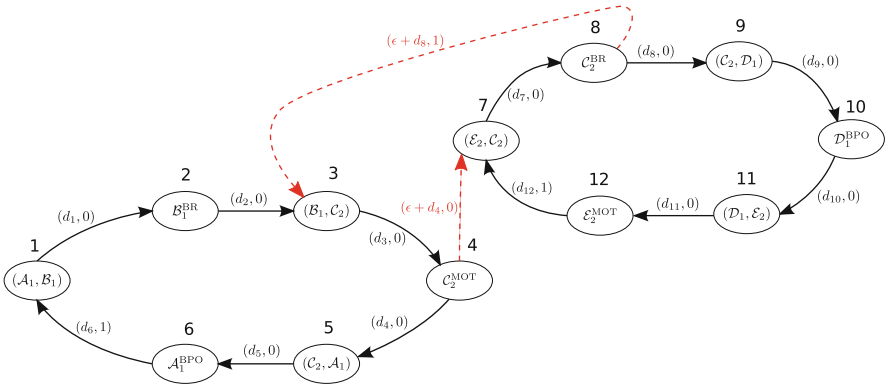


Fig. 4 Graph representation of the robotic line

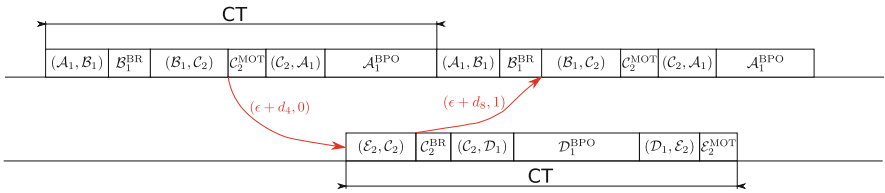


Fig. 5 An example of a schedule

guarantee that operations and movements are performed in a desired order (orders in case of alternatives). All edges are weighted by length $L(e_{i,j})$ and height $H(e_{i,j})$ where the length corresponds to the duration or time offset, and the height binds the previous or future cycles with the current one. In Fig. 4 we added a small time tolerance (i.e. constant ϵ) to the lengths of time lags to guarantee a safe handover of a weldment. For more information about time lags please refer to [7]. Finally, the selected robot positions, power saving modes (i.e. brakes—BR, motors—MOT, and bus-power-off—BPO), and movements are indicated in the graph nodes. One of the possible schedules is depicted in Fig. 5 where CT is the production cycle time.

As it can be seen the problem is similar to cyclic scheduling, however, there are a few differences. At fist, there is a synchronisation between robots, and as a consequence rotated schedules are not equally good as it is in cyclic scheduling. Secondly, durations are not fixed (energy functions) as it is the case of cyclic scheduling. If only one robot is taken into account, the problem is equivalent to the Travelling Salesman Problem with the exception of non-constant edge weights.

3 Energy Function of the Robot Movements

There are several ways how the energy function of the robots can be obtained. The following subsections propose three ways, i.e. measurement at a real robot, physical modelling of the robot kinematics and dynamics, and simulation based also on a software model of the robot controller. Section 3.4 deals with a way how to identify the individual robotic operations automatically from the power measurement data obtained at a production cell with multiple robots. This procedure allows evaluating the energy model of the robot against the robot behaviour in the cell.


3.1 Power Measurements

Detailed measurements were performed at industrial robot *KUKA KR5 arc* for different speeds of movements, trajectories and robot positions to find out the energy-saving potential. Such a measurement cannot be done in a production cell typically. However, it is presented here to support the hypothesis about using the energy function. A brief description of the robot, which has been used, can be found in Table 1.

In the experiment the *static consumption* and *dynamic consumption* were measured. The static consumption is perceived as an amount of energy consumed by a robot in a stationary position. A non-moving robot can also get to a power saving mode (brakes, bus-power-off, hibernate) to save even more energy. The dynamic consumption corresponds to energy consumed during the robot movement.

The measured profile of active power is shown in Fig. 6. The left part of the graph (up to 80 s) can be used to evaluate the static consumption for the robot being held on brakes or motors. The rest of the graph is designated for the measurement of the dynamic consumption. For each speed, denoted as ‘T2 : X%’ where X is a relative speed of the robot, the sequence of movements (peaks in the graph) $p_1—p_2—p_1 \searrow p_3 \nearrow p_1$ is executed. ‘ $p_i—p_j$ ’ is a movement between points p_i and p_j which are at the same height, ‘ $p_i \searrow p_j$ ’ is a descending movement, and finally ‘ $p_i \nearrow p_j$ ’ is ascending one. Before each sequence the robot is moved from the home position, i.e. an initial position of the robot determined by the robotic cell designer, to p_1 and after the sequence from p_1 to the home position.

Table 1 Basic parameters of KUKA KR 5 arc (KUKA Industrial Robots [8])

	Working range	1412 mm
	Maximal load	5 kg
	Weight	127 kg
	Idle power: held on brakes	cca 180 W
	Idle power: held on motors	cca 350 W
	Idle power: bus-power-off	cca 134 W

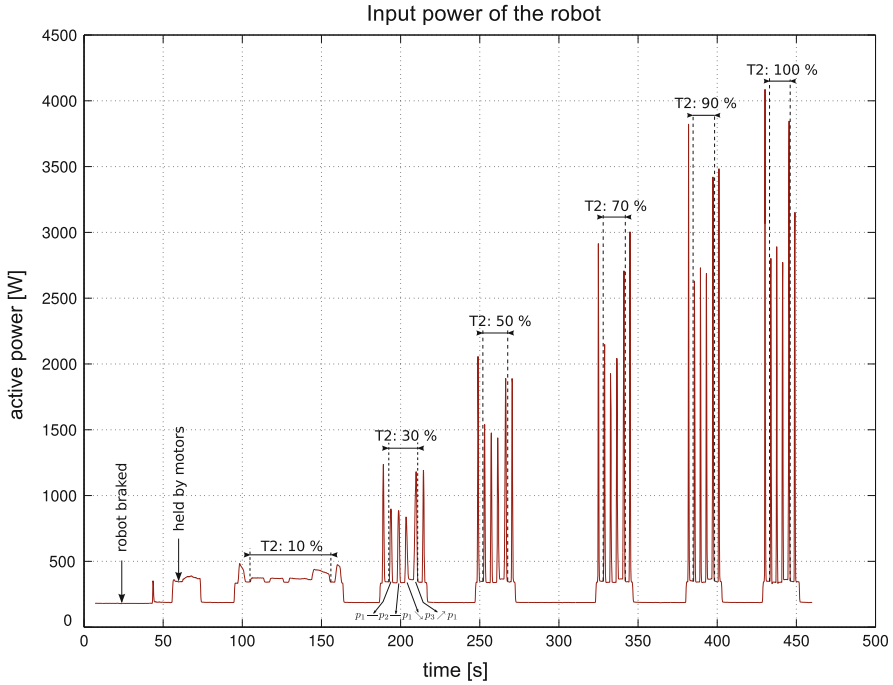


Fig. 6 Energy profile of the robot power consumption

The measured data (see Fig. 6) were analysed and the results are presented in Table 2. From the first three rows of the table it can be deduced that it is possible to save about 48% ($= 100 * (347.9 - 180.3)/347.9$) of energy if the robot is braked instead of being held in the position by motors. The difference could be even bigger if the robot was loaded or a less energy efficient configuration (i.e. a position of the robot) was selected. Another experiment not mentioned before, was related to the measurement of how the robot configuration influences the power consumption. It was found out that the robot vertically stretched out required 344 W compared to 366 W for the robot horizontally stretched out. A relatively small difference was caused by using a small industrial robot without load. Along with the experiment, it was measured that the robot consumes 134 and 30 W in the bus-power-off and hibernate modes, respectively.

To evaluate the effect of different movement speeds on the energy consumption the average input power and total consumption were calculated (last two columns in Table 2). From the results it is no surprise that the energy consumption was confirmed to be higher for $p_1 \nearrow p_3$ movement than for $p_3 \searrow p_1$ one. With respect to the speed of the robot it was shown that there is no need to consider too slow movements as the gravity part constitutes huge loss of energy. In a similar way, too fast movements increase the consumption dramatically while the duration of the

Table 2 Analysis of the measured data

Interval	t_1	t_2	Δt (s)	Input power (W)	Energy consumption (J)
Idle—brakes	25.0	30.0	5.0	180.3	901.3
Idle—motors	57.6	58.2	0.6	347.9	208.7
T2: 30% ($p_1 \rightarrow p_2$)	192.7	194.8	2.1	649.5	1364.0
T2: 30% ($p_2 \rightarrow p_1$)	197.7	199.8	2.1	641.4	1346.9
T2: 30% ($p_1 \searrow p_3$)	202.5	205.0	2.5	583.8	1459.6
T2: 30% ($p_3 \nearrow p_1$)	208.0	210.8	2.8	755.8	2116.2
T2: 50% ($p_1 \rightarrow p_2$)	252.2	253.8	1.6	858.9	1374.2
T2: 50% ($p_2 \rightarrow p_1$)	256.5	258.0	1.5	874.0	1310.9
T2: 50% ($p_1 \searrow p_3$)	260.7	262.3	1.6	860.0	1376.0
T2: 50% ($p_3 \nearrow p_1$)	265.2	267.2	2.0	1015.7	2031.3
T2: 70% ($p_1 \rightarrow p_2$)	328.2	329.4	1.2	1204.1	1444.9
T2: 70% ($p_2 \rightarrow p_1$)	332.0	333.4	1.4	1001.1	1401.5
T2: 70% ($p_1 \searrow p_3$)	336.0	337.4	1.4	1102.9	1544.0
T2: 70% ($p_3 \nearrow p_1$)	340.3	341.8	1.5	1420.6	2130.8
T2: 90% ($p_1 \rightarrow p_2$)	385.0	386.3	1.3	1198.1	1557.5
T2: 90% ($p_2 \rightarrow p_1$)	388.8	390.0	1.2	1233.9	1480.7
T2: 90% ($p_1 \searrow p_3$)	392.5	393.8	1.3	1220.1	1586.1
T2: 90% ($p_3 \nearrow p_1$)	396.5	398.0	1.5	1492.7	2239.0
T2: 100% ($p_1 \rightarrow p_2$)	433.3	434.5	1.2	1270.7	1524.8
T2: 100% ($p_2 \rightarrow p_1$)	437.0	438.2	1.2	1182.6	1419.2
T2: 100% ($p_1 \searrow p_3$)	440.7	441.9	1.2	1411.2	1693.5
T2: 100% ($p_3 \nearrow p_1$)	444.5	446.0	1.5	1494.5	2241.7

movement decreases only little. For instance, there is a fall of about 6.3% in energy for the $p_1 \rightarrow p_2 \rightarrow p_1$ movement if the relative speed is set to 70% instead of 90%.

3.2 Physical Model of the Robot

A complete modelling approach has been presented in [13], which deals with a description of the fully graphically-oriented computer-aided modelling and its mathematical analysis that is used for the determination of the robot energy consumption. The computer-aided modelling follows from a pure geometrical 3D model of the robot that is split by CAD software such as NX or Solidworks into particular robot components. They are supplemented with appropriate physical parameters like volumes, masses and moments of inertia. Such a component model, which represents physically a mechanical robot structure, can be converted into a simulation model operated in the Matlab/Simulink environment as shown in Fig. 7. The blocks in the figure have the following meaning. (A) is a World frame block, which represents the global reference frame; (B) is a Mechanism

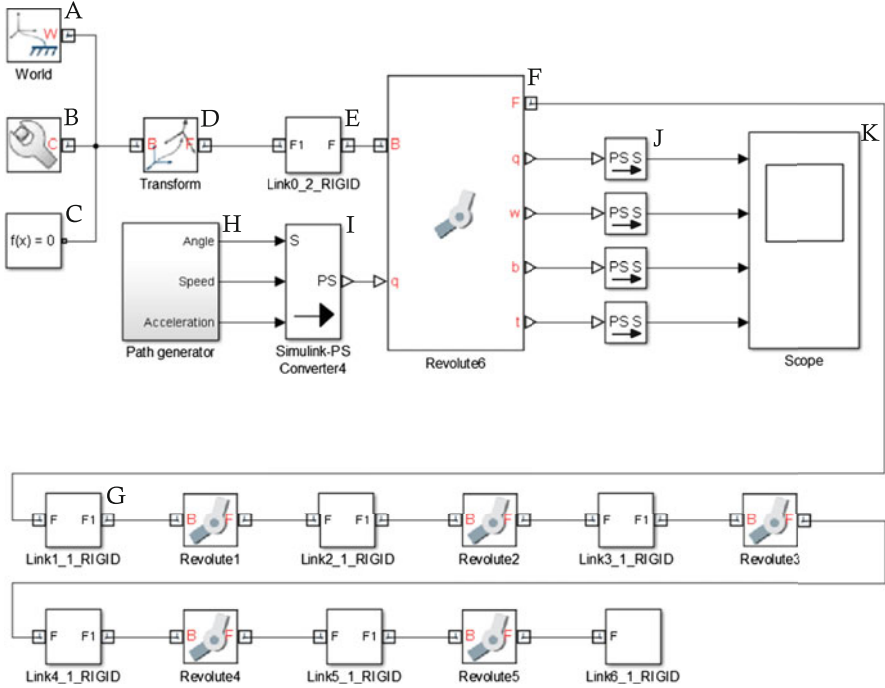


Fig. 7 Block diagram of robot structure in Simulink

configuration block of general parameters used in the simulation; (C) is a block for the configuration of the solver to configure the simulation; (D) is a Rigid transform block representing a transformation matrix that allows a following mechanical robot element to move with respect to the basic frame; (E) is a Link block, which represents the rigid body with its Denavit-Hartenberg frame and appropriate information about the body mass, moment of inertia related to its center of gravity; (F) is the Revolute joint block with one DOF, where the information about its angle, angular velocity, angular acceleration and actuating torque are obtained from a built-in joint sensor; and (G) is the next Link connected to the next revolute robot joint. (H) is the path generator that feeds the trajectory coordinates to block (I) that converts unit-less signals to physical quantities with appropriate units. The meaning of the other blocks is straightforward. The mechanical model from Fig. 7 is completed by blocks representing the robot drives, which allows getting Eq. (3) of torque equilibrium.

On the basis of the mathematical analysis (see [13]), Eq. (2) gives the input power for a single electrical motor with stator resistance R_{S_i} and current i_{q_i} for axis i in q -coordinate of the (d, q) coordinate system where the torque component of current is

aligned along the q axis.

$$P = \frac{3}{2} \sum_{i=1}^n R S_i i_{q_i}^2 + \sum_{i=1}^n \omega_{m_i} (\tau_i + B_i \omega_{m_i} + J_i \frac{d}{dt} \omega_{m_i}) \quad (2)$$

This equation is based on the equilibrium between electromagnetic torque and mechanical torque for the individual components, i.e. for the axes of the robot, as expressed in Eq. (3)

$$\tau_{e_i} = \tau_i + B_i \omega_{m_i} + J_i \frac{d}{dt} \omega_{m_i} \quad (3)$$

where J_i is the inertia and B_i is the friction of the motor and the load; ω_{m_i} is mechanical rotor speed related to electrical speed as $\omega_{m_i} = \omega_{e_i}/p$ considering p for a number of pole pairs; τ_i is a one load torque component of the torque vector τ from the dynamic model as expressed in Eq. (4); and τ_{e_i} is electromagnetic torque.

$$B(\vartheta) \ddot{\vartheta} + C(\vartheta, \dot{\vartheta}) \dot{\vartheta} + g(\vartheta) = \tau \quad (4)$$

where $B(\vartheta)$ is an inertia matrix, $C(\vartheta, \dot{\vartheta})$ is a coefficient matrix of Coriolis and centrifugal force effects, $g(\vartheta)$ is a vector of gravitational effects, $\vartheta = [\vartheta_1 \ \vartheta_2 \ \dots \ \vartheta_n]^T$ is a vector of joint angles and $\tau = [\tau_1 \ \tau_2 \ \dots \ \tau_n]^T$ is a vector of torques acting on appropriate joints as shown e.g. in [19].

By integrating the total input power over the interval corresponding to the considered motion trajectory the robot energy consumption is obtained in Eq. (5), where Γ is the duration of the robotic movement and the other variables come from Eqs. (2)–(4).

$$E = \int_0^\Gamma \left[\frac{3}{2} \sum_{i=1}^n R i_{q_i}^2 + \sum_{i=1}^n \omega_{m_i} (\tau_i + B_i \omega_{m_i} + J_i \frac{d}{dt} \omega_{m_i}) \right] dt \quad (5)$$

By evaluating the energy consumption for different durations of the motion it is possible to construct the energy function for the considered trajectory.

3.3 Simulated Energy Function

In some situations it is not possible to perform all necessary experiments and measurements with real equipment because of some physical or organizational limitations. As an example, a robotic cell being part of a regular production may be named. Thus, it may not be possible to change the robotic paths to perform any additional movements, which are not part of productive robotic operations, to

identify a set of parameters relating to the dependency of the power consumption on a specific robot trajectory, to the robots' dynamic parameters etc.

A simulation environment can be used such as Tecnomatix Process Simulate that contains robot controllers implemented according to the Robot Controller Simulation (RCS) specification, i.e. the robot controllers perform Realistic Robot Simulation (RRS). One of the features of the recent Process Simulate version is the possibility to simulate power consumption of the robot movements. By summation of the total energy used by a robot for one particular movement performed repeatedly with different speed settings energy function $f_E^v(x)$ can be constructed—see Eq. (1).

3.4 Robotic Operations

As mentioned above, knowing the robotic operations and the consumption of the energy for each of them helps evaluate the energy function for particular robot movements. Moreover, it is also used to group robotic operations together to form the activities that are used in the mathematical model to describe the behaviour of the robots (see Sect. 4). According to this model, it is necessary to differentiate operations, which represent movement trajectories, and operations representing work such as welding, holding a part in a specific position, etc. A set of subsequent e.g. welding operations that are executed in a given order and are next to each other, is typically represented as a single activity because it is not expected that changing the trajectories between the welding points would mean any significant savings. Figure 8 shows a sequence of operations of one robot in the welding cell. The horizontal bars with o_1, o_2 , etc. represent the operations, which may be grouped into activities that are later for the optimisation represented as single dynamic or static activities (see Sect. 4). The upper horizontal bar is the length of one production cycle, i.e. the time after which the robot repeats its set of operations for the next part. Interval T_R shows the time remaining between the last operation and the

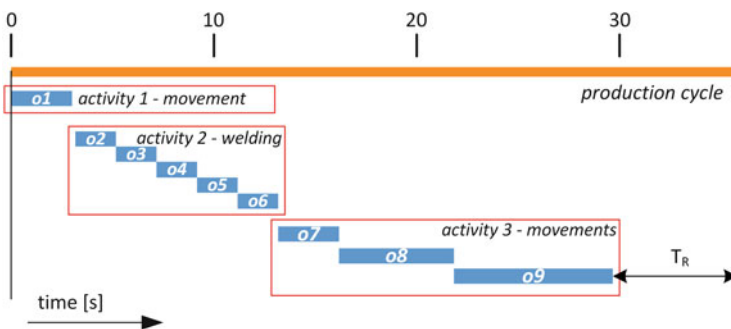


Fig. 8 Operations of one robot in the cell

beginning of the next production cycle. Thus, operation o_1 forms activity 1, which is dynamic, and represents a movement of the robot to its first position. Operations o_2 – o_6 represent a set of welding operations and short movements between the neighbouring welding spots and forms static activity 2. Subsequent operations o_7 – o_9 correspond to gripping the part and moving it to another position (such as a turn table) and all together form activity 3. This is to demonstrate that also several movements, i.e. movement operations, can be formed into a single activity, which in case of activity 3 is dynamic. Such a sequence of operations is specified for each robot in the cell. Finally, as described in Sect. 4, T_R is modelled as an additional static activity that allows the robot to wait for the start of the next production cycle.

In the following text, which relates to [15], the problem of identifying individual robotic operations from the actual power needed by the robots to execute the movements is described.

The basic idea is to label sections of the data of power measurements, which correspond to particular operations of each robot in the cell. This may be done by observing the robots working in the cell and placing marks in the data. After that patterns are marked in the labelled data as model patterns, i.e. one pattern as a representative of one robotic operation, that will be searched for in the whole set of the power consumption data.

The procedure of identifying the operations consists of (a) *data filtering* to suppress the noise and errors stemming from misplaced samples, (b) *evaluating the similarity* of all segments of the power consumption data, and finally (c) taking the *measure of similarity as a threshold* to search for local maxima, which correspond to the location of the model patterns in the data.

Data Filtering To get rid of the error (noise, misplaced samples) a median filter is used. Its filtering window length has been chosen to filter out random signal errors with a big amplitude difference such as unsynchronised neighbouring samples, where two neighbours are swapped. The filter still conserves high peaks that are used as classification feature during the detection phase.

Similarity of Segments To assess the similarity of two same-length vectors of one-dimensional data signals the analysis in frequency or in time domain may be done. According to the measurements, which had been performed at the robots in a production cell, the frequency analysis has proven not to be suitable because the signals are very similar in their spectra. There may be two instances of the same operation but from different time points during the day, whereas one time point is at the end of the working shift and the other is at the beginning of the next one, whose correlation is 83%. This value is too low contrary to the fact that both spectra correspond to the same operation. Moreover the frequency spectra of two different operations may be correlated up to 85%. Based on this observation the frequency analysis is not suitable.

The get the similarity measure in the time domain Pearson's correlation coefficient is computed for its simplicity and suitability for this intended case. It is defined as follows.

$$\rho_{\mathbf{x},\mathbf{y}} = \frac{cov(\mathbf{x}, \mathbf{y})}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \quad (6)$$

$$cov(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_{\mathbf{x}})(y_i - \mu_{\mathbf{y}}) \quad (7)$$

$$\sigma_{\mathbf{x}} = \sqrt{\frac{(\mathbf{x} - \mu_{\mathbf{x}})^T (\mathbf{x} - \mu_{\mathbf{x}})}{N}} \quad (8)$$

$$\sigma_{\mathbf{y}} = \sqrt{\frac{(\mathbf{y} - \mu_{\mathbf{y}})^T (\mathbf{y} - \mu_{\mathbf{y}})}{N}} \quad (9)$$

Coefficients $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ are the standard deviations of values of vectors \mathbf{x} and \mathbf{y} , respectively. The $cov(\mathbf{x}, \mathbf{y})$ is the cross-covariance of the vectors \mathbf{x} and \mathbf{y} . The $\mu_{\mathbf{x}}$ and $\mu_{\mathbf{y}}$ are the mean values of \mathbf{x} , \mathbf{y} , respectively, and N is the length of the model pattern vector. However, the mean values $\mu_{\mathbf{x}}$, $\mu_{\mathbf{y}}$ are not known exactly and thus their approximations by computing the mean values of \bar{x} , \bar{y} from the measured data are used.

Measure of Similarity After having established the measure of a good match of two segments for the power consumption data it must be considered, which segments to compare. A straightforward procedure is to compare every possible vector, i.e. a window in the data, which has the same length as the model pattern vector. Such a strategy guarantees a good precision of localization in time but is computationally demanding. Nevertheless, the procedure lies in picking a vector of length K of samples from the power consumption data where K is the same as the length of the model pattern vector. Then the Pearson's correlation coefficient is computed and stored in a vector of results \mathbf{r} , whose length can be expressed as

$$\dim(\mathbf{r}) = \dim(\mathbf{d}) - K + 1 \quad (10)$$

where \mathbf{d} is vector of energy consumption data. The bigger K is the less correlations are needed to be computed but the more multiplications must be performed to compute each of them. Roughly $3K$ multiplications need to be done for each window on the power consumption data vector.

To reduce the length of the model pattern and thus to lower the computational cost to compute the correlation vector, distinctive features can be extracted from the power consumption data and the correlation can be computed on them. Local maxima have been chosen as these features because they can be detected during one iteration over the power consumption data and they provide good measure to match the patterns. Undesirable peaks are avoided by applying a threshold to choose only

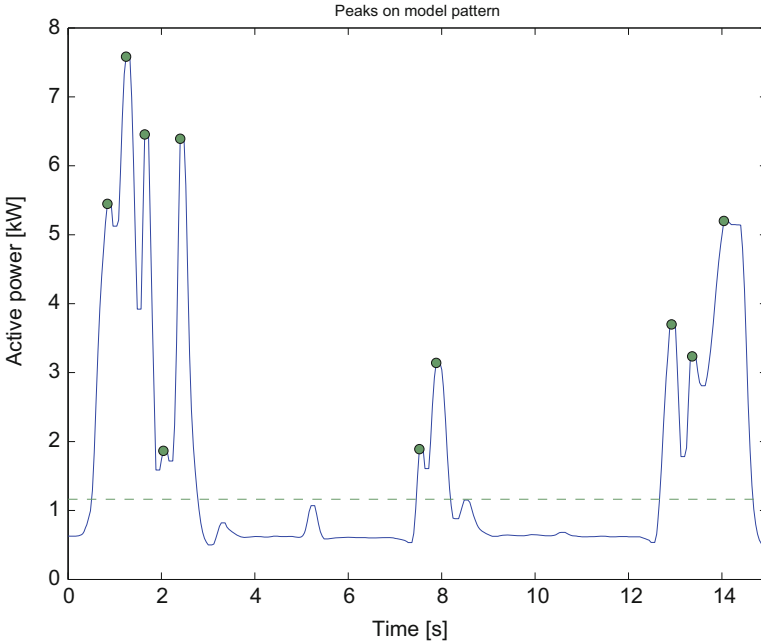


Fig. 9 Peaks extracted from the model pattern

dominant peaks, which contain enough information for classification. In case more peaks occur in the defined neighbourhood only the biggest one is picked and in case of same-valued peaks the first one is prioritised. Figure 9 shows the model pattern and peaks chosen according to the rules above. However, there are also peaks under the dashed threshold line that may get above the threshold during the peak extraction process because of the drifts in offset during the working shifts. This fact would cause that some patterns physically generated by the same type of operation would have more dominant peaks than others. Such a situation is solved as follows.

The model pattern vector and the vector, in which the matching is done, must have the same length to be able to be compared. In fact, by the extraction of the feature vector a new down-sampled data vector with adaptive sampling time is created. Basically three distinctive situations may occur if different peaks, i.e. not all peaks are correctly recognised, are extracted. There may be (1) a perfect match, (2) there is one peak missing, and (3) there is one additional peak in the power consumption data. To avoid decreasing the correlation coefficient because of the described situation the corresponding timestamps must be paired and the peaks relating to unpaired timestamps must be dropped. Thus, because of the fact that the duration of the model pattern is known and the patterns that are searched for should not deviate much from it, a window being at least as long as the model pattern is chosen. Then each sample of model pattern vector is assigned a sample from the data vector based on the shortest Euclidean distance of timestamps. This procedure

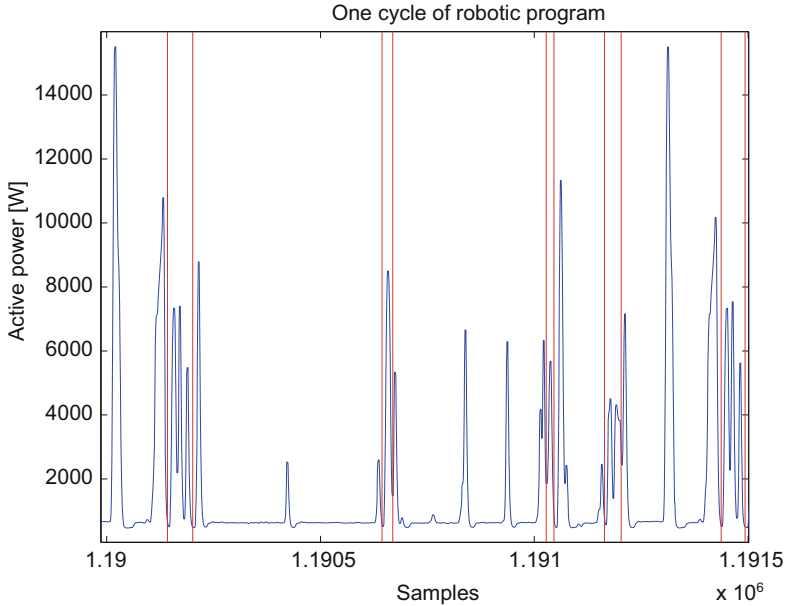


Fig. 10 Identified operations' boundaries

equalises the sizes of the compared vectors. Finally the correlation of the vectors of the aligned timestamps is evaluated and only such vectors that are correlated enough, are passed further for value correlation. Thanks to this procedure the correlation not only of the order of samples, but also of their position is considered. Moreover, situations when the robot operations are interrupted abruptly are also coped well with. The reason for interruptions, which are in fact unplanned pauses in the production, may mean a failure in the equipment, interruption of the material flow, etc. Figure 10 shows an example how the boundaries (i.e. the red vertical lines) of the operations of one robot are identified.

4 Optimisation

The following terms are used in the formulation of the optimisation problem as an Integer Linear Programming problem. The first one is the set of *static activities* $V_{\mathcal{S}}$ where static activity $i \in V_{\mathcal{S}}$ corresponds to a robot operation (i.e. subspace) such as e.g. welding or assembling. The set of *dynamic activities* $V_{\mathcal{D}}$ corresponds with all the movements where dynamic activity $i \in V_{\mathcal{D}}$ consists of all possible point-to-point movements between two subspaces, i.e. black arrows in Fig. 1 in Sect. 2. Both the static and dynamic activities are multi-mode activities; it means that there are different states that an activity can attain. Mode $t \in T_i$ of dynamic activity i

Table 3 Model variables

W_i	Required energy by activity i
s_i	Start time of activity i
d_i	Duration of activity i
x_i^p	True if robot position $p \in P_i$ for static activity i is selected, otherwise false
z_i^m	True if robot power saving mode $m \in M_i$ is selected in static activity i , otherwise false
y_i^t	True if movement $t \in T_i$ of dynamic activity i is selected, otherwise false
$h_{i,j,r}^*, w_{i,j}$	Decide the order of activities

selects one of the point-to-point movements between related subspaces. In case of static activity i it is the selected position $p \in P_i$ (six coordinates— x, y, z, rx, ry, rz) and power saving mode $m \in M_i$ of the robot. It is obvious that the set of activities $V = V_{\mathcal{S}} \cup V_{\mathcal{D}}$. Activities associated with robot $r \in R$ will be denoted by set V_r .

Static activities $V_{\mathcal{S}}$ can be further divided into three sets— V_{IN}, V_{OP}, V_{OUT} . In set V_{IN} there are *input activities*, i.e. activities related to taking a workpiece. In a similar way set V_{OUT} consists of *output activities* related to passing a workpiece to another robot or machine. And finally, the rest of robot operations, i.e. $V_{\mathcal{S}} \setminus \{V_{IN} \cup V_{OUT}\}$, are activities such as welding, assembling, disassembling, cutting, etc.

In activity set V there are mandatory and optional activities. *Mandatory activities* $V_{\mathcal{M}}$ have to be carried out in all cases, whereas *optional activities* $V_{\mathcal{O}}$ are not necessary to be performed. The optional activities were introduced by considering alternatives where different paths can be taken, and as a result there may exist dynamic activities (i.e. $V_{\mathcal{O}} \subseteq V_{\mathcal{D}}$) conditionally executed. Robot operations $V_{\mathcal{S}}$, however, have to be performed every time, and therefore $V_{\mathcal{S}} \subseteq V_{\mathcal{M}}$. It is evident that $V = V_{\mathcal{M}} \cup V_{\mathcal{O}}$.

4.1 Integer Linear Programming Model

The objective is to minimise the overall energy consumption of activities. Note that not all activities in $V_{\mathcal{O}}$ have to be performed. In that case their W_i and d_i are set to zero due to the criterion (Table 3).

Equations (11) and (12) bind activity duration d_i with its power consumption W_i . Both the equations can be enabled or disabled depending on selected activity modes where \bar{W} is an upper bound on energy. Equation (11) is proposed for static activities,² whose power demand $a_{i,p}^m$ depends on robot configuration p and power saving mode m . In case of dynamic activities, i.e. Eq. (12), the energy function was approximated by a set of linear functions with coefficients $a_{i,k}^t$ and $b_{i,k}^t$ where $k \in K$ is the k th segment of the energy function for movement t . The energy function has

²Each activity can be performed by only one assigned robot.

to be convex to ensure validity of the model.

$$\begin{aligned} & \text{minimise } \sum_{\forall i \in V} W_i \\ & \text{s.t. } a_{i,p}^m d_i - \overline{W} (2 - z_i^m - x_i^p) \leq W_i \end{aligned} \quad (11)$$

$$\forall i \in V_{\mathcal{G}}, \forall p \in P_i, \forall m \in M_i$$

$$a_{i,k}^t d_i + b_{i,k}^t - \overline{W} (1 - y_i^t) \leq W_i \quad (12)$$

$$\forall i \in V_{\mathcal{G}}, \forall t \in T_i, \forall k \in K$$

Equations (13) and (14) state that each static activity $i \in V_{\mathcal{G}}$ has the position and robot power saving mode assigned. In a similar way, Eq. (15) ensures that one of the movements is selected for each mandatory activity.

$$\sum_{\forall p \in P_i} x_i^p = 1 \quad \forall i \in V_{\mathcal{G}} \quad (13)$$

$$\sum_{\forall m \in M_i} z_i^m = 1 \quad \forall i \in V_{\mathcal{G}} \quad (14)$$

$$\sum_{\forall t \in T_i} y_i^t = 1 \quad \forall i \in V_{\mathcal{G}} \cap V_{\mathcal{M}} \quad (15)$$

Flow preservation constraints (16) and (17) mean that if the robot moves to position p it also has to move away from p . In other words, if a movement to position p is selected then a movement from p is selected as well. Inward and outward movements are found by enumerating predecessors and successors respectively.

$$\sum_{\forall j \in \text{pred}(i)} \sum_{\forall t \in T_j(p_{\text{from}}, p)} y_j^t = x_i^p \quad \forall i \in V_{\mathcal{G}}, \forall p \in P_i \quad (16)$$

$$\sum_{\forall j \in \text{suc}(i)} \sum_{\forall t \in T_j(p, p_{\text{to}})} y_j^t = x_i^p \quad \forall i \in V_{\mathcal{G}}, \forall p \in P_i \quad (17)$$

Equations from (18) to (23) are related to activity ordering. Equation (18) sets time relations for mandatory activities ($V_{\mathcal{M}} \subseteq V_{\mathcal{G}}$). Alternatives are taken into account in Eqs. (19)–(21) where $w_{i,j}$ is a decision variable determining whether dynamic³ activity $i \in V_{\mathcal{D}}$ with movements to static activity j will be selected or not. Binary variables $h_{i,j,r}^*$ decide which activity $i \in V_{\mathcal{D}}$ is closing (i.e. is the last one) the production cycle for each robot (see Eqs. (22) and (23)) as it was found out

³The dynamic activity has exactly one successor and one predecessor.

that rotated schedules have to be taken into consideration due to time lags.

$$s_j - s_i = d_i - \text{CT}h_{i,j,r}^* \quad (18)$$

$$\forall r \in R, \forall i \in V_r \cap V_{\mathcal{M}}, \forall j \in \text{suc}(i)$$

$$\sum_{\forall t \in T_i} y_i^t = w_{i,\text{suc}(i)} \quad \forall i \in V_{\emptyset} \cap V_{\mathcal{D}} \quad (19)$$

$$s_j - s_i + (1 - w_{i,j}) \text{CT} \geq d_i - \text{CT}h_{i,j,r}^* \quad (20)$$

$$\forall r \in R, \forall i \in V_{\emptyset} \cap V_r \cap V_{\mathcal{D}}, \forall j \in \text{suc}(i)$$

$$s_j - s_i - (1 - w_{i,j}) \text{CT} \leq d_i - \text{CT}h_{i,j,r}^* \quad (21)$$

$$\forall r \in R, \forall i \in V_{\emptyset} \cap V_r \cap V_{\mathcal{D}}, \forall j \in \text{suc}(i)$$

$$\sum_{\forall i,j} h_{i,j,r}^* = 1 \quad \forall r \in R \quad (22)$$

$$h_{i,j,r}^* = 0 \quad \forall r \in R, \forall i \in V, \forall j \notin V_{\text{IN}} \quad (23)$$

The duration of the activity is bound in Eqs. (24) and (25). Minimal time of staying in a static activity \underline{d}_i^m is determined by the selected robot power saving mode. Maximal duration \bar{d}_i can be limited by a robot operation, e.g. immersion of a workpiece in paint to get a protective coating. The duration of dynamic activity i is influenced by selected trajectory t lasting from \underline{d}_i^t to \bar{d}_i^t .

$$\underline{d}_i^m z_i^m \leq d_i \leq \bar{d}_i \quad \forall i \in V_{\mathcal{D}}, \forall m \in M_i \quad (24)$$

$$\underline{d}_i^t y_i^t \leq d_i \leq \bar{d}_i + \text{CT}(1 - y_i^t) \quad \forall i \in V_{\mathcal{D}}, \forall t \in T_i \quad (25)$$

Finally, the last two Eqs. (26) and (27) ensure the correct synchronisation between robots. Equation (26) guarantees time constraints, e.g. the workpiece is taken away after it has been put on the bench, whereas Eq. (27) warrants proper handovers in terms of robot configurations. For each position p of activity $i \in V_{\text{OUT}}$ one of compatible positions $p' \in \text{CP}(i, p)$ of activity $j \in V_{\text{IN}}$ can be selected. Both the equations can be perceived as the global ones because they link robots to each other.

$$s_j - s_i \geq l_{i,j} - \text{CT}h_{i,j} \quad \forall (l_{i,j}, h_{i,j}) \in \mathcal{L} \quad (26)$$

$$x_i^p \leq \sum_{\forall p' \in \text{CP}(i,p)} x_j^{p'} \quad \forall i, j \subseteq V_{\text{OUT}} \times V_{\text{IN}} \quad (27)$$

$$W_i, s_i, d_i \in \mathbb{R}_0^+ \quad x_i^p, z_i^m, y_i^t, h_{i,j,r}^*, w_{i,j} \in \{0, 1\}$$

Table 4 Statistics of the energy consumption for feasible instances

	Time limit 100 s (J)	Time limit 7200 s (J)
min UB	28,038.8	27,656.2
avg UB	33,796.7	32,681.6
max UB	43,043.0	40,849.9

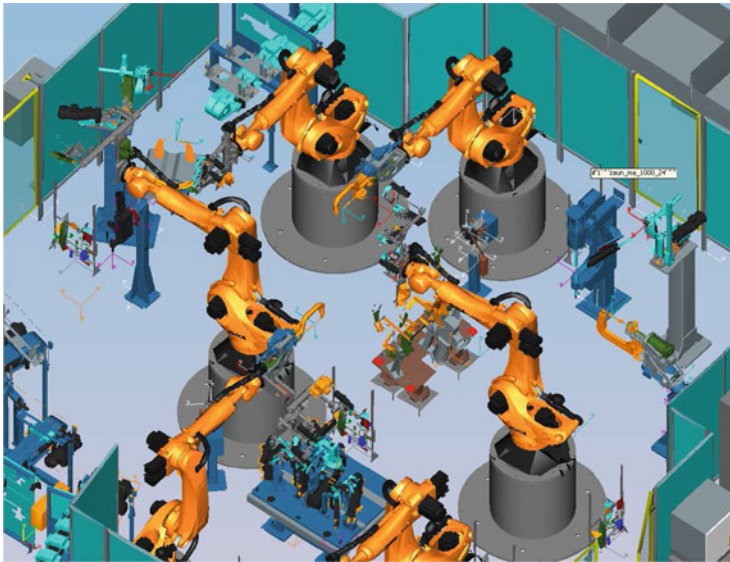


Fig. 11 Structure of the welding cell

average gap, which is a relative distance between the best found upper bound (the best solution) and the lower bound, was 27.5% for the 2-h limit. The size of the model was roughly about 10,000 constraints and 1000 variables.

The Lagrangian relaxation was tested on four selected instances as it had been shown that other feasible problems are too computationally expensive. Nonetheless, it took a few hours to find a high quality lower bounds by using the subgradient method and ILP solver since solving even one robot to optimality usually takes more than a minute even though all 12 CPU cores are utilised. The subgradient algorithm stops if more than 200 consecutive deteriorations were reached. Results reveal that the Lagrangian relaxation provides much tighter lower bounds than Cplex solver since average gap was 3.5% in comparison with 16.7% gap proved by Cplex.

An industrial use case has also been used as an instance for the optimisation problem. This use case represents a robotic cell with six robots and other pieces of equipment such as turn tables, conveyors, gluing machine and welding guns. A general structure of the welding cell is depicted in Fig. 11. The behaviour of the cell can be described shortly as follows. The basic part together with two smaller parts are put onto the turn table by the operator. This turn table cannot be seen in Fig. 11 as it is hidden below its bottom. The turn table turns and the first robot—R1 (which is visible only partially in the figure) performs the welding to mount

the small parts to the basic one. After this welding is finished the robot, which besides of the servo gun possesses also a gripper, takes the part and moves it to the following static table. Here, the next robot—R2 places another parts onto the basic one and performs the welding afterwards. Then, the part is taken by the next robot—R3 from the table and is moved to the next table. In the meanwhile, the next robot—R4 prepares another part and has a glue put on it by the gluing machine. This part is placed onto the basic part, which has been moved already, and the welding is performed. Subsequently, the part is taken by the next robot—R5 and brought to the static welding gun that performs the next welding operations. Finally, the last robot—R6 takes the part, performs additional welding operations at the last static welding gun and puts the resulting part onto the outgoing conveyor, which conveys the part out of the cell. The timing of the operations was retrieved from the robot programs and each trajectory's energy function f_E has been interpolated from the points, obtained from simulations in Tecnomatix Process Simulate. Welding, glueing, and assembling operations have not been changed by the optimisation to ensure repeatability of the production process. Only the robot speeds and power saving modes (at home position) were addressed in the optimization since the minimal intervention is desirable for existing robotic cells. The results show that the original energy consumption 500 kJ (maximal speeds, without power saving modes) can be decreased to 391 kJ (reduced speeds, power saving modes), which makes up roughly 20% of energy saving.

6 Conclusion and Future Work

In this work a general mathematical model was proposed to optimise the energy consumption of robotic lines that allows taking into account robot power saving modes and different locations of operations. The optimisation problem was formulated as an Integer Linear Programming problem and solved using Cplex solver. The achieved results confirm the correctness of the model and show that a significant reduction of the energy consumption can be achieved. In addition to the mathematical model the Lagrangian relaxation was used to devise a very tight lower bound.

Each of the presented methods to calculate the energy function has its pros and cons and their usage depends on specific conditions for a given robotic cell to be optimised. The measurement of the energy consumption of the robots moving on different trajectories with various speeds is usually not possible at an existing robotic cell, which participates in the production. Mathematical modelling depends on having the right 3D models not only of the robots but also of the parts that the robots carry. Realistic Robotic Simulation depends on the precision of the simulation model whereas an exact or even an approximate value is usually not known. Thus in a typical situation a combination of more approaches must be used to obtain the energy function of all the robots.

The outcomes of the industrial use-case optimisation show a significant potential to reduce energy consumption of existing robotic cells and even more can be expected for planned robotic cells as the full potential of the optimisation algorithm can be exploited.

The future work will thus concentrate mainly on making the mathematical model of the robots more precise and on decreasing the uncertainties in the models by providing further information e.g. from measurements. The robot models will probably always contain uncertainties because of the lack of publicly available information. Therefore, measurements must be performed to supplement the missing information and to complete the mathematical models. Last but not least more stress is going to be put on evaluating the optimisation results with industrial use cases to prove their viability.

Acknowledgements This work has been conducted in cooperation with Skoda Auto within contract 830-8301343/13135, with support of the Grant Agency of the Czech Technical University in Prague, grant No. SGS13/209/OHK3/3T/13 and the Grant Agency of the Czech Republic under the Project GACR P103-16-23509S.

Glossary

CT	Production Cycle Time
d_i	Duration of activity i
$E(d_i)$	Energy function linking the consumption with time of movement
$h_{i,j}$	The height of the edge in cyclic scheduling
$H(e_{i,j})$	See $h_{i,j}$
$l_{i,j}$	The length of the edge in cyclic scheduling
\mathcal{L}	Time lags
$L(e_{i,j})$	See $l_{i,j}$
M_i	Set of the robot power saving modes that can be used in activity $i \in V_{\mathcal{S}}$
$\text{pred}(i)$	Predecessors of activity i
P_i	Set of possible robot configurations for activity $i \in V_{\mathcal{S}}$
s_i	Start time of activity i
$\text{suc}(i)$	Successors of activity i
T_i	Set of possible movements of activity $i \in V_{\mathcal{D}}$
$V_{\mathcal{S}}$	Set of static activities, i.e. robot operations
$V_{\mathcal{D}}$	Set of dynamic activities, i.e. robot movements
$V_{\mathcal{A}}$	Set of activities that have to be executed
$V_{\mathcal{O}}$	Set of activities ($\subseteq V_{\mathcal{D}}$) that can optionally be executed
W_i	Energy consumed by activity i
x_i^p	Binary variable set to true iff the robot configuration p was selected for activity $i \in V_{\mathcal{S}}$
y_i^t	Binary variable set to true iff movement t was selected for activity $i \in V_{\mathcal{D}}$
z_i^m	Binary variable set to true iff the robot power saving mode m was selected for activity $i \in V_{\mathcal{S}}$

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Upper Saddle River, NJ (1993)
2. Bard, J.F., Purnomo, H.W.: Cyclic preference scheduling of nurses using a Lagrangian-based heuristic. *J. Sched.* **10**(1), 5–23 (2007)
3. Bjorkenstam, S., Gleeson, D., Bohlin, R., Carlson, J., Lennartson, B.: Energy efficient and collision free motion of industrial robots using optimal control. In: *IEEE International Conference on Automation Science and Engineering*, pp. 510–515 (2013)
4. Bryan, C., Grenwalt, M., Stienecker, A.: Energy consumption reduction in industrial robots. In: *Proceedings of ASEE North Conference*, pp. 1–4 (2010)
5. Chemnitz, M., Schreck, G., Kruger, J.: Analyzing energy consumption of industrial robots. In: *Proceedings of IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–4 (2011)
6. Gavrovska, A.M., Paskaš, M.P., Dujković, D., Reljin, I.S.: Region-based phonocardiogram event segmentation in spectrogram image. In: *10th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2010 - Proceedings*, pp. 69–72 (2010). doi:10.1109/NEUREL.2010.5644108
7. Hanen, C., Munier, A.: A study of the cyclic scheduling problem on parallel processors. *Discret. Appl. Math.* **57**(2–3), 167–192 (1995). *Combinatorial optimization 1992*
8. KUKA: KUKA Industrial Robots (2014). Available at <http://www.kuka-robotics.com/en/>. Accessed 3 June 2014
9. Lampariello, R., Nguyen-Tuong, D., Castellini, C., Hirzinger, G., Peters, J.: Trajectory planning for optimal robot catching in real-time. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3719–3726 (2011)
10. Le, C.V., Pang, C.K., Gan, O.P., Chee, X.M., Zhang, D.H., Luo, M., Chan, H.L., Lewis, F.L.: Classification of energy consumption patterns for energy audit and machine scheduling in industrial manufacturing systems. *Trans. Inst. Meas. Control.* **35**(5), 583–592 (2012). doi:10.1177/0142331212460883
11. Mashaei, M., Lennartson, B.: Energy reduction in a pallet-constrained flow shop through on-off control of idle machines. *IEEE Trans. Autom. Sci. Eng.* **10**(1), 45–56 (2013)
12. Michna, V., Wagner, P., Cernohorsky, J.: Constrained optimization of robot trajectory and obstacle avoidance. In: *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–4 (2010)
13. Othman, A., Belda, K., Burget, P.: Physical modelling of energy consumption of industrial articulated robots. In: *15th ICCAS International Conference on Control, Automation and Systems* (2015)
14. Riazzi, S., Bengtsson, K., Wigström, O., Vidarsson, E., Lennartson, B.: Energy optimization of multi-robot systems. In: *20th IEEE CASE Conference on Automation Science and Engineering*, pp. 1345–1350 (2015)
15. Ron, M., Burget, P., Fiala, O.: Identification of operations at robotic welding lines. In: *20th IEEE CASE Conference on Automation Science and Engineering*, pp. 470–476 (2015)
16. Saramago Jr., S., Steffen, V.: Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system. *Mech. Mach. Theory* **33**(7), 883–894 (1998)
17. Saravanan, R., Ramabalan, S., Balamurugan, C.: Evolutionary optimal trajectory planning for industrial robot with payload constraints. *Int. J. Adv. Manuf. Technol.* **38**(11–12), 1213–1226 (2008)
18. Sharma, G.: Optimization of energy in robotic arm using genetic algorithm. *Int. J. Comput. Sci. Technol.* **2**(2), 315–317 (2011)
19. Siciliano, B., Sciavicco, L., et al.: *Robotics - Modelling, Planning and Control*. Springer, Berlin (2009)

20. Simon, L., Hungerbuehler, K.: Real time takagi-sugeno fuzzy model based pattern recognition in the batch chemical industry. In: IEEE International Conference on Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence), pp. 779–782 (2008). doi:10.1109/FUZZY.2008.4630459
21. Smetanová, A.: Optimization of energy by robot movement. *Mod. Mach. Sci. J.* **3**(1), 172–176 (2010)
22. Vergnano, A., Thorstensson, C., Lennartson, B., Falkman, P., Pellicciari, M., Leali, F., Biller, S.: Modeling and optimization of energy consumption in cooperative multi-robot systems. *IEEE Trans. Autom. Sci. Eng.* **9**(2), 423–428 (2012)
23. Wigstrom, O., Lennartson, B.: Integrated OR/CP optimization for discrete event systems with nonlinear cost. In: 2013 IEEE 52nd Annual Conference on Decision and Control (CDC), pp. 7627–7633 (2013)
24. Wigstrom, O., Sundstrom, N., Lennartson, B.: Optimization of hybrid systems with known paths. In: 4th IFAC Conference on Analysis and Design of Hybrid Systems, 2012, pp. 39–45 (2012). doi:10.3182/20120606-3-NL-3011.00007
25. Wigstrom, O., Lennartson, B., Vergnano, A., Breitholtz, C.: High-level scheduling of energy optimal trajectories. *IEEE Trans. Autom. Sci. Eng.* **10**(1), 57–64 (2013)

Production Line Optimization with Model Based Methods

T. Hajba, Z. Horváth, C. Kiss-Tóth, and J. Jósvai

Abstract In this paper we deal with different models of production lines of factories and consider the makespan optimization problem based on these models. We consider state-of-the-art and novel mathematical optimizers including exact and heuristic methods. We apply these optimizers to both standard academic and industrial data sets. We see that in a large rate of the considered cases the novel exact optimizers converged to the optimum fast which is surprising being the problems NP-hard and the problem sizes big. This shows the importance of exploiting the structure present in the industrial data using standardized industrial data sets for testing mathematical algorithms devoted to solve industrial problems and that some provided exact mathematical optimizers are fast and perform accurately on the considered industrial problems.

1 Introduction

One of the most important tasks at modern factories is the optimal scheduling of the order of jobs on the production lines of the factory since this affects highly the makespan of the set of jobs to be processed per day and thus determines the total number of jobs that can be processed per day. Though experienced managers can provide satisfactory schedules, with the components of the realization of the Digital Factory concept schedules of higher quality can be achieved. Namely, the Digital Factory concept (see [21]) can be regarded as mapping of the real physical processes of the factory to tools of the information technology. The Digital Factory methodology includes production simulation tools using real manufacturing data (bill of materials, production plan, operation sequence, makespan, capacity usage, lateness, etc.). In everyday environment where the production scheduling tasks are situation driven - because of unreliable information, production line fall out, material delivery failures, etc. - simulation based scheduling is asked. A production system has many influential parameters, to optimize a set of jobs in a system where there are conflicting goals, so the mathematical solving method can be very

T. Hajba • Z. Horváth (✉) • C. Kiss-Tóth • J. Jósvai
Széchenyi István University, Győr, Hungary
e-mail: hajbat@sze.hu; horvathz@sze.hu; ktchris@sze.hu; josvai@sze.hu

complex. Therefore a combination of simulation tools and mathematical methods can be an effective answer for this problem. Especially when we consider the Industry4.0 vertical and horizontal integration process within the production system. This integration and distributed decision making methodology need sophisticated modeling, simulation and mathematical optimization methods and tools. These aspects let us think about our research work, which combines modeling, simulation and mathematical optimization of production lines, gives important results for the next steps in the field of smart and networked digital factory. This allows automatic construction of models and/or simulations for the examined production flow, in our case the work on the production line, and poses the corresponding optimization problems at model/simulation level. Then the inverse mapping of the optima of the model or simulation based optimization problems gives a suggestion to the production line managers for the actual scheduling.

We note that often a combination of models and simulations is advantageous. Indeed, simulations validated at high accuracy are typically time consuming and thus simulation based optimization takes often too long for using it in a daily routine of a company. On the other hand, one objective evaluation at model based optimization is typically much faster than that with simulation but the accuracy of fast models is lower than that with an enhanced simulation. Thus a fast and reliable optimization can consist of a model based hierarchic optimization that includes the evaluation of the design elements with accurate simulation to check whether the actually used model is accurate enough.

In this paper we deal with optimal scheduling of the production lines using mathematical models and their optimization methods. Namely, we provide a review of the models based on mathematical optimization methods and test results on academic and industrial data. In Sect. 2 we define the considered models to the production line: first the basic model, the Permutation Flow Shop Problem (PFSP), which is studied thoroughly in the literature and then the recently introduced and studied models that handle more features of real industrial production lines than PFSP, the Permutation with Repetition Flow Shop Problem (R-PFSP) and the Permutation with Repetition Flow Shop Problem with Buffers and Palettes (PB-R-PFSP) (see [3]). Here repetition refers to the fact that in industrial situations there exist several types of jobs and each schedule contains many jobs of the same type. Then in Sect. 3 we define and discuss some frequently used heuristic optimization methods for the PFSP models. In Sect. 4 first we present mixed integer linear programming (MILP) formulations for the PFSP, which has been studied in several papers, see [8, 9, 12, 16–19, 22, 23] and three new MILP models of [2]. According to numerical experiments in papers [18, 19] the MILP formulations of the PFSP models combined with exact solvers (e.g. those based on branch and bound) are capable to solve only small sized PFSPs. However, applying our new MILP models to R-PFSP and PB-R-PFSP we see in Sect. 5 that large scale industrial problems arising from the automotive industry (see [5]) and their analogues become exactly solvable. We close the chapter with drawing conclusions.

2 Production Line Models: The PFSP and the PB-R-PFSP

In the regular permutation flow shop problem we are given a production system of M machines and a set of N jobs. Every job has to be processed on every machine in the same order, i.e. every job has to be processed first on the first machine then on the second machine and so on. The processing times of the jobs at the machines are known in advance and deterministic. The regular PFSP entails the following assumptions which form the constraints of the optimization problem:

- Machines are continuously available from time 0.
- Every job is available for processing at time 0.
- Each job can be processed only on one machine at a time.
- Each machine can process at most one job at a time.
- The jobs must be processed on the machines without preemption.
- Setup times are included in the processing times, or ignored.
- Any number of jobs can wait between consecutive machines.

In this paper the objective of the optimization problem is the minimization of the makespan, i.e. the minimization of the completion time of the last job of the order on the last machine.

Many manufacturing problems have special features which are not included in the regular PFSP. One such property is that in real-life problems we often have a lot of jobs that have the same processing times on every machines (see [2]). It is said in this case that these jobs have the same type. Taking this property into consideration can drastically reduce the number of different permutations, i.e. the design space for the PFSP. Namely, if there are T types and n_t jobs of type t then the number of permutation reduces from $N!$ to $\frac{N!}{(n_1)! \cdot (n_2)! \cdot \dots \cdot (n_T)!}$.

Another property of industrial situations is the presence of palette usage at lines. Namely, jobs are often carried on palettes on the line and the number of palettes is bounded from above, typically less than the number of jobs. This means that only a limited number of jobs can be on the line at a time.

Moreover, since the palette size and the space between consecutive machines are given, only a limited number of palettes (hence limited number of jobs) can wait between consecutive machines which means that there are limited buffer sizes between the consecutive machines.

These properties are summarized as follows:

- the number of different types of the jobs is less than the total number of the jobs;
- only a given number of palettes can be used to carry the jobs;
- the buffer sizes between consecutive machines are finite and given in advance;

We call a PFSP which contains repeated jobs, limited buffer sizes between consecutive machines and bounded number of palettes *Permutation with Repetition Flow Shop Problem with Palettes and Buffer* (PB-R-PFSP).

3 Heuristic Optimization Methods for the PFSP Model

During the last three decades many kind of heuristic approaches were published to solve the PFSP problem. In this section we are going to give a review about some of these algorithms we tested on our problem sets.

3.1 NEH Heuristic

One of the most famous constructive solution for the FPSP was proposed by Nawaz, Ensore and Ham in 1983 [10]. Their algorithm is based on the assumption that jobs with higher total processing time make bigger effect on the objective function, this means we should give higher priority to them. NEH algorithm constructs a solution by inserting the jobs into an empty permutation one by one. We are going to describe the details below.

First of all we calculate the total processing times for all jobs, and after that we choose two jobs with the highest value. We consider the two possible partial schedules, and choose the better one. During the rest of the algorithm the relative positions of these two jobs are fixed.

In the next step we pick up the job with the third highest total processing time, and put it into the permutation to the proper place found by an exhaustive search procedure. This means that we place the job to the first, the second and to the third (last) place of the partial sequence, and keep the best solution we get. This process is repeated until all the jobs are placed into the permutation. The pseudo-code of the algorithm can be presented as follows (Fig. 1).

The major advantage of the NEH algorithm is that we are able to get a good solution in a short time, since the total number of iterations (makespan evaluations of partial solutions) is $\frac{N(N+1)}{2}$. Although many algorithms can be found in the literature, NEH is still a state of the art algorithm for the PFSP optimization and also used for creating initial solutions for many other heuristic methods with higher computational complexity such as in the following two heuristics described in the next sections.

- 1: For each job i calculate $T_i = \sum_{r=1}^M P_{ri}$ where P_{ri} is the processing time of job i on machine r .
- 2: Arrange the job indices i into a list L in descending order of T_i .
- 3: Pick the first two jobs of the list L , and find the best permutation π_2 of these two jobs by calculating the makespan for the two possible partial solutions. Fix the relative positions of these two jobs for the remaining steps. Set $i = 3$.
- 4: Pick the job from the i th position of the list L and find the best sequence by inserting it at all possible positions of π_{i-1} (without changing the relative positions of the previous jobs).
- 5: If $N = i$ then STOP, otherwise set $i = i + 1$, and go to Step 4.

Fig. 1 Pseudo-code for the NEH algorithm

3.2 Ant-Colony Algorithms

The name ant-colony algorithms (ACO) group a family of techniques to solve combinatorial optimization problems. In the framework of these techniques the motivation is to imitate the pheromone trail in the nature used by real ants for communication and feedback. Basically, these techniques are population-based, cooperative search procedures. During the optimization ant-colony algorithms use simple agents (called ants) that iteratively construct permutations, and this solution construction is guided by these artificial pheromone trails. The details how we calculate these artificial trails is a problem-specific heuristic information. This information has to be tuned for the problem we would like to solve.

To understand this optimization method we have to define the meaning of solution components. Ants iteratively construct possible permutations from these components, and leave pheromone. In this context pheromones indicate the intensity of ant-trails with respect to solution components. These values are determined from the influence of each solution component to the objective function. The trails also form a kind of adaptive memory in this search procedure: we update the intensities at the end of each iteration, the effect of the permutation created for the last is the biggest. In this framework π_{ij} denotes the trail intensity of setting job i in position j of a sequence. Since every job can be placed at every place we have to store N^2 intensity values.

In every iteration one single ant constructs a complete solution starting with an empty permutation and iteratively adding components until a complete solution is constructed. After the construction each ant gives feedback on the solution by leaving pheromone (updating trail intensity) on each solution component.

After updating the intensities we apply a local search scheme for the created permutation to find possibly the best solution in the neighborhood. Summarizingly the general structure of ACO algorithms can be described in Fig. 2.

Since many kind of ACO algorithms were published for the PFSP problem, we chose one, named PACO published by Rajendran and Ziegler in 2004 [13], supposed to perform the best on the Talliard problem set. In the rest of this section we will present the details of the PACO algorithm.

Fig. 2 General pseudo-code for ACO algorithms

- 1: Initialize the pheromone trails and parameters.
- 2: While termination condition is not met do the following:
 - ▷ construct a solution;
 - ▷ improve the solution by local search;
 - ▷ update the pheromone trail intensities.
- 3: Return the best solution found.

3.2.1 Initialization

For initializing the trail intensities we need an initial solution. Usually we can use the NEH heuristic to find an initial solution, in PACO we improve this by applying the job-index-based local search three times. We denote the makespan of this solution with Z_{best} . After this the initialization of the pheromone trails looks as follows:

$$\pi_{ik} = \begin{cases} \frac{1}{Z_{best}} & \text{if } |\text{position of job } i \text{ in the seed sequence} - k| + 1 \leq \frac{n}{4} \\ \frac{1}{2Z_{best}} & \text{if } \frac{n}{4} < |\text{position of job } i \text{ in the seed sequence} - k| + 1 \leq \frac{n}{2} \\ \frac{1}{4Z_{best}} & \text{otherwise.} \end{cases}$$

The idea behind this initialization is that we think the initial solution is good enough, so putting a job on a place which is closer to his place in the initial solution probably gives better result.

3.2.2 Construction of a New Solution

Every iteration of the PACO algorithm creates a new permutation iteratively. We place an unscheduled job i for place k using the following scheme:

- $T_{ik} = \sum_{q=1}^k \pi_{iq}$ and sample a uniform random number u in range $[0, 1]$.
- If $u \in [0, 0.4]$: choose the best unscheduled job in the best sequence obtained so far.
- If $u \in (0.4, 0.8]$: among the first five unscheduled jobs in the best sequence choose the job with the maximum T_{ik} value.
- If $u \in (0.8, 1]$: among the set of the first five unscheduled jobs in the best sequence select job i with the probability of $\frac{T_{ik}}{\sum_l T_{lk}}$.

If the number of the unscheduled jobs is less then five, then consider all of them. After this procedure apply the job-index-based local search procedure three times, and denote the makespan of this solution with $Z_{current}$.

3.2.3 Update of the Pheromone Intensities

Let h denote the position of job i in the resultant sequence. If the number of jobs is less or equal to 40:

$$\pi_{ik}^{new} = \begin{cases} \rho \cdot \pi_{ik}^{old} + \frac{1}{diff \cdot Z_{current}} & \text{if } |h - k| \leq 1 \\ \rho \cdot \pi_{ik}^{old} & \text{otherwise.} \end{cases}$$

If the number of jobs is greater than 40:

$$\pi_{ik}^{new} = \begin{cases} \varrho \cdot \pi_{ik}^{old} + \frac{1}{diff \cdot Z_{current}} & \text{if } |h - k| \leq 2 \\ \varrho \cdot \pi_{ik}^{old} & \text{otherwise.} \end{cases}$$

where $diff = (|\text{position of job } i \text{ in the best sequence obtained so far} - k| + 1)^{\frac{1}{2}}$ and ϱ is a constant, fixed as 0.75 during the algorithm.

We ran the PACO algorithm for 300 iterations.

3.3 Tabu Search Approaches

Tabu search (TS) is a general framework which can be used to find near-optimal solutions of hard combinatorial optimization problems. Using a tabu search algorithm first the neighborhood of a solution has to be defined. Then starting from an initial solution, at each iteration the algorithm examines the neighborhood of the actual solution and one of the neighbors (usually the best) is chosen to be the actual solution in the next iteration. To avoid returning back to a previously used solution tabu search algorithms use a so-called tabu list containing elements of forbidden moves. The algorithm stops if a stopping criterion (for example the number of total iterations) is reached.

Next we describe in more detail the TS heuristic for the PFSP of Nowicki and Smutnicki [11].

3.3.1 Moves and Neighborhood

In the PFSP a solution is represented as a permutation. To be able to define the neighborhood of a permutation we first introduce the concept of moves. Let π be a permutation and a and b two positions. Removing the job from position a of the permutation and putting it into position b of the permutation is called a *move* and denoted with (a, b) . The *neighborhood* of π is the set of permutations that can be reached from π with one move. Each job we remove can be placed to $N - 1$ places, since the move (a, a) does not change the permutation. Furthermore the moves $(a - 1, a)$ and $(a, a - 1)$ yield the same permutation hence every permutation has $(N - 1)^2$ neighbors. Since searching in such a large neighborhood would be time-consuming we will reduce the number of potential neighbors. In order to be able to do this we will introduce the definition of blocks and critical path.

It is known, that the makespan of a permutation π can be written in the following form:

$$C_{\max}(\pi) = \max_{1=j_0 \leq j_1 \leq \dots \leq j_{M-1} \leq j_M=N} \sum_{i=1}^M \sum_{j=j_{i-1}}^{j_i} P_{i\pi(j)}. \tag{1}$$

Let us associate with each permutation π a directed grid graph $G(\pi) = (V, E)$ with node weights:

$$V = \{1, \dots, M\} \times \{1, \dots, N\},$$

$$E = \bigcup_{i=1}^M \bigcup_{j=1}^{N-1} \{(i, j), (i, j+1)\} \cup \bigcup_{i=1}^{M-1} \bigcup_{j=1}^N \{(i, j), (i+1, j)\}.$$

where the node (i, j) represents the i th machine and the j th job of the permutation and the weight of node (i, j) is $P_{i\pi(j)}$. Then formula (1) means that the makespan of π is the weight of the critical (longest) path in this grid graph, from node $(1, 1)$ to (M, N) .

Let us suppose that the critical path in $G(\pi)$ is $(1, s_0), \dots, (1, s_1), (2, s_1), \dots, (2, s_2), \dots, (M, s_{M-1}), \dots, (M, s_M)$, where $1 = s_0 \leq s_1 \leq \dots \leq s_M = n$. Then this path consists of the vertical edges $((i, s_i), (i+1, s_i))$ for $i = 1, 2, \dots, M-1$ and horizontal subpaths $(i, s_{i-1}), \dots, (i, s_i)$ if $s_{i-1} < s_i$. If for machine i condition $s_{i-1} < s_i$ holds then the sequence of positions $s_{i-1}, s_{i-1} + 1, \dots, s_i$ is called a *block*. Each position can be contained in one or two blocks.

It can be proven that performing a move $v = (a, b)$ to permutation π for which a and b are inside the same block gives us a solution π' for which $C_{\max}(\pi') \geq C_{\max}(\pi)$, which means we do not have to analyze such neighbors. Based on their experiments Nowitzki and Smutnicki reduced further the neighborhood of a permutation.

A move (a, b) is called a right move, if $a < b$ else the move is called a left move. Tests showed that for position a if a is a beginning of the l th block or lies inside the l th block then it is enough to examine right moves (a, b) in which b lies in the first few positions of the $(l+1)$ th block (the next block). Similarly if a is an end of the l th block or lies inside the l th block then it is enough to examine left moves (a, b) in which b lies in the last few positions of the $(l-1)$ th block (the previous block). The algorithm uses the parameter ε which controls that for a exactly how many positions b from the next and previous blocks are considered for the right and left moves. The value of ε depends on the M and N , namely

$$\varepsilon = \begin{cases} 0 & \text{if } \frac{N}{M} > 3 \\ 0.5 & \text{if } 2 < \frac{N}{M} \leq 3 \\ 1 & \text{if } \frac{N}{M} \leq 2 \end{cases}$$

Denoting by $ZR_j(\pi, \varepsilon)$ the set of the above defined right moves of position j and by $ZL_j(\pi, \varepsilon)$ the set of the above defined left moves for position j the set of investigated moves:

$$Z(\pi, \varepsilon) = \bigcup_{j=1}^{N-1} ZR_j(\pi, \varepsilon) \cup \bigcup_{j=2}^N ZL_j(\pi, \varepsilon)$$

and the solutions we get after performing these moves are the neighbors of the solution π . It is worth mentioning that if $\varepsilon = 0$ then for every position j at most one left and one right move is examined hence in this case the neighborhood of a permutation contains at most $2n - 2$ elements. As we mentioned earlier for every permutation there are overall $(n - 1)^2$ different moves which means that the above procedure can drastically reduce the size of the neighborhood of a permutation making the algorithm to be faster. After defining the moves we describe the tabu list, the main idea of this heuristic approach.

3.3.2 Tabu List

Tabu list is a technique to prevent cycling during the search procedure. The TS algorithm of Nowicki and Smutnicki uses a tabu list with fixed length (*maxt*) (i.e. the list can contain *maxt* elements) which contains pair of jobs, initialized with elements $(0, 0)$ at the beginning. If during the algorithm a move $v = (a, b)$ is performed, then the first element of the list is deleted and the pair $(\pi(a), \pi(a + 1))$ if $a < b$ and $(\pi(a - 1), \pi(a))$ otherwise is added to the end of the list.

In the search procedure a move (a, b) from permutation β is „tabu” if at least one pair $(\beta(j), \beta(a)), j = a + 1, \dots, b$ is in the tabu list if $a < b$, and at least one pair $(\beta(a), \beta(j)), j = b, \dots, a - 1$ is in the tabu list otherwise.

After we defined all components of the algorithm, we describe the searching strategy.

3.3.3 Neighborhood Searching Strategy

At the first stage of the searching procedure from every set ZL_j and ZR_j we choose a representative with the smallest makespan. In this way we get a new set of neighbors containing $2(n - 1)$ elements independently from the value of ε denoted with \hat{Z} .

At the second stage we classify these solutions into three categories: unforbidden (*UF*), forbidden but profitable (*FP*) and forbidden and non-profitable (*FN*). A forbidden move from π is profitable, if it leads to a solution whose makespan is less then $F(C_{\max}(\pi))$ where F is an aspiration function.

Finally we decide to perform the best move from the set of the *UF* and the *FP* moves. If all moves are *FN* then we add $(0, 0)$ to the tabu list; this process is repeated until an *UF*-move can be chosen (Fig. 3).

- 1: Find sets *UF*-moves $X = \{v \in \hat{Z}(\pi, \varepsilon) : v \text{ is not tabu}\}$ and *FP*-moves $Y = \{v \in \hat{Z}(\pi, \varepsilon) : v \text{ is tabu, } C_{\max}(\pi_v) < F(C_{\max}(\pi))\}$.
- 2: If $X \cup Y \neq \emptyset$ then select $v' \in X \cup Y$ with the smallest makespan. Update the tabu list and *Exit*.
- 3: Add a zero element to the tabu list and go to **Step 2**.

Fig. 3 Pseudo-code for the neighborhood searching procedure

- 1: Set $\pi := \pi^*$, $C^* := C_{\max}(\pi)$, $iter := 0$, $ret := 0$, $T := \emptyset$ and $save = true$. Set $F(x) := \infty$ for all x non-negative integer.
- 2: Set $iter := iter + 1$, $ret := ret + 1$. Find $Z(\pi, \varepsilon)$ and $\hat{Z}(\pi, \varepsilon)$.
- 3: Find a move $v' \in \hat{Z}(\pi, \varepsilon)$, a neighbor $\pi' = \pi_{v'}$, a makespan $C' = C_{\max}(\pi')$ and modified T' using the NSP. Modify the aspiration function $F(C) := \min\{F(C), C'\}$, $F(C') := \min\{F(C'), C\}$. If $save = true$ then $Z^* := \hat{Z}(\pi, \varepsilon) \setminus \{v'\}$, $T^* = T$ and $save = false$. Set $\pi := \pi'$, $T := T'$ and $C := C'$.
- 4: If $C < C^*$ then set $\pi^* := \pi$, $C^* := C$, $ret := 0$, $save := true$ and go to **Step 2**.
- 5: If $ret < maxret$ go to **Step 2**.
- 6: If $Z^* \neq \emptyset$ and $iter < maxiter$ then set $\pi := \pi^*$, $\hat{Z}(\pi, \varepsilon) := Z^*$, $C := C^*$, $T := T^*$, $save := true$, $ret := 1$ and go to **Step 3**; otherwise *STOP*.

Fig. 4 Pseudo-code for the tabu search algorithm

After this we will show the tabu search algorithm proposed by Nowicki and Smutnicki.

The algorithm can be launched from an arbitrary initial solution, but we start our search procedure from the permutation π given by the NEH heuristics. We set $\pi^* = \pi$, and start with an empty tabu list. In each iteration we find the set of moves Z and the representatives \hat{Z} . Applying the neighborhood searching procedure we select a move $v' \in \hat{Z}$, create a new tabu list and modify the aspiration function. If the new solution is better than the best one found so far, we update C^* .

If C^* does not decrease during $maxret$ iterations then we jump back to π^* and continue searching with the stored representatives and tabu list Z^* and T^* . We have two stopping criteria: the maximum number of iterations ($maxiter$) performed, or Z^* is empty. We can summarize this algorithm into a pseudo-code in Fig. 4.

During the tests we used $maxiter = 30,000$, $maxt = 8$, $maxret = 500$.

4 Exact Optimization for the PFSP and PB-R-PFSP Models

One possible method to get the optimal solution of a PFSP is to formulate the PFSP as a mixed integer linear programming (MILP) problem and solve it by an appropriate software (such as CPLEX, GUROBI, and so on). The advantage of this approach compared to heuristic methods is that even if the software can not give optimal solution during the prescribed time limit for the running time it always gives a lower bound for the optimal value. This means that using MILP models we always know how far the given solution is from the optimal one. Since the PFSP is NP-hard for $M \geq 3$, the drawback of this approach is that only small or medium-sized problems can be solved optimally this way. However, taking into account the rapid growth of the performance of computers and softwares that can solve MILP models, one can expect that larger and larger problems become solvable this way.

4.1 MILP Models of the PFSP and R-PFSP

The MILP models of the PFSP [1, 8, 9, 12, 14–19, 22, 23] can be divided into two parts. The models of the Wagner family describe a permutation by giving for each position the job that is placed to this position. The models of the Manne family describe a permutation by giving for each pair a jobs i and j whether i precedes j in the permutation or not. Empirical studies showed [18, 19] that the models of the Wagner family are superior to the models of the Manne family with regard to the required solution times. Therefore in [2, 3] based on MILP models of the Wagner family for the PFSP 3 new MILP models for the R-PFSP and PB-R-PFSP were introduced. In this section we first describe the R-TS2 model for the PFSP and then the R-TS2 model for the R-PFSP and the PB-R-TS2 model for PB-R-PFSP are introduced.

4.1.1 The TS2 Model for the PFSP

The TS2 model was presented in [19]. Let us denote by $C_{r,j}$ the completion time of the j th job of the order on machine r (so $C_{r,j}$ is a continuous variable for all (r, j) pairs $(1 \leq r \leq M, 1 \leq j \leq N)$). Let Z_{ij} be a binary variable for all (i, j) indices $(1 \leq i \leq N, 1 \leq j \leq N)$. The value of Z_{ij} is equal to 1 if job i is placed to the j th place of the order, otherwise Z_{ij} is equal to 0. The constraints of the TS2-model imply that the following conditions are satisfied.

- Each job is assigned to exactly one place in the sequence.

$$\sum_{j=1}^N Z_{ij} = 1 \quad 1 \leq i \leq N \tag{2}$$

- Each position in the sequence is filled with exactly one job

$$\sum_{i=1}^N Z_{ij} = 1 \quad 1 \leq j \leq N \tag{3}$$

- The job in the $(j + 1)$ th position of the sequence can not finish on any machine until the job in the j th position of the sequence is finished on that machine and job in the $(j + 1)$ th position of the sequence is processed on that machine.

$$C_{rj} + \sum_{i=1}^N P_{ri}Z_{i,j+1} \leq C_{r,j+1}, \quad 1 \leq r \leq M, 1 \leq j \leq N - 1 \tag{4}$$

- A job can not be finished on machine $r + 1$ until its finished on machine r and processed on machine $r + 1$.

$$C_{rj} + \sum_{i=1}^N P_{r+1,i} Z_{ij} \leq C_{r+1,j}, \quad 1 \leq r \leq M - 1, 1 \leq j \leq N \quad (5)$$

- The first job of the order can not finish earlier on machine 1 than its duration time on machine 1.

$$\sum_{i=1}^N P_{1i} Z_{i1} \leq C_{11} \quad (6)$$

- The makespan is the completion time of the last job of the sequence on the last machine.

$$C_{max} = C_{MN} \quad (7)$$

Hence the TS2 model can be summarized as

Minimize (7) Subject to: (2)–(6).

4.1.2 The R-TS2 Model for the R-PFSP

The Permutation with Repetition Flow Shop Problem (R-PFSP) is a special PFSP which contains jobs that have equal processing times on every machines. We say that jobs i and j have the same *type* if they have equal processing times on each machine. To describe a permutation in an R-PFSP it is enough to give for each position j of the sequence the type of the job that is placed to that position. This means that the TS2 model for the PFSP can be simplified to model R-PFSP-s. The following R-version of the TS2model, named R-TS2 were introduced by Hajba and Horváth in [2].

We will denote by T the number of different types and by n_t the number of jobs of type t ($1 \leq t \leq T$). Let us denote the processing time of a job of type i on machine r by P'_{ri} and let Z_{ij} be a binary variable for all (i, j) indices ($1 \leq i \leq T, 1 \leq j \leq N$). The value of Z_{ij} is equal to 1 if a job of type i is placed to the j th place of the order, otherwise Z_{ij} is equal to 0. The constraints of the R-TS2-model imply that the following conditions are satisfied.

- There are n_i jobs in the sequence that are of type i .

$$\sum_{j=1}^N Z'_{ij} = 1 \quad 1 \leq i \leq T \quad (8)$$

- Each position in the sequence is filled with exactly one type of job.

$$\sum_{i=1}^T Z'_{ij} = 1 \quad 1 \leq j \leq N \tag{9}$$

- The job in the $(j + 1)$ th position of the sequence can not finish on any machine until the job in the j th position of the sequence is finished on that machine and job in the $(j + 1)$ th position of the sequence is processed on that machine.

$$C_{rj} + \sum_{i=1}^T P'_{ri} Z'_{i,j+1} \leq C_{r,j+1}, \quad 1 \leq r \leq M, 1 \leq j \leq N - 1 \tag{10}$$

- A job can not be finished on machine $r + 1$ until its finished on machine r and processed on machine $r + 1$.

$$C_{rj} + \sum_{i=1}^T P'_{r+1,i} Z'_{ij} \leq C_{r+1,j}, \quad 1 \leq r \leq M - 1, 1 \leq j \leq N \tag{11}$$

- The first job of the order can not finish earlier on machine 1 than its duration time on machine 1.

$$\sum_{i=1}^T P'_{1i} Z'_{i1} \leq C_{11} \tag{12}$$

- The makespan is the completion time of the last job of the sequence on the last machine.

$$C_{max} = C_{MN} \tag{13}$$

The R-TS2 can be formalized as follows below.

Minimize (13) subject to (8)–(12).

Size Complexity of the TS2 and R-TS2 Models

The size complexity of the TS2 and R-TS2 models are presented in Table 1. The main difference is that the R-TS2 model contains much fewer binary variable than the TS2 model. The reason for it is that in the R-TS2 model we only have to give for each position the type of the job that is placed to that position (instead of giving for each position the job that is placed to that position). Finally the R-TS2 model contains less constraints than the TS2 model.

Table 1 Size complexity of the models

Model	Binary variable	Real variable	Constraints
TS2	N^2	$MN + 1$	$2MN - M + N + 1$
R-TS2	NT	$MN + 1$	$2MN - M + T + 1$

Note that N = number of jobs, T = number of types, M = number of machines

4.1.3 The PB-R-TS2 Model of the PB-R-PFSP

A *Permutation with Repetition Flow Shop Problem with Palettes and Buffer* (PB-R-PFSP) is a special R-PFSP in which only a limited number of jobs can wait between consecutive machines and the jobs are carried on palettes through the line and the number of palettes is bounded from above. We will denote by K the number palettes and by b_i the buffer size between machines i and $i + 1$.

The following PB-R-TS2 model was introduced in [3]. The model is the extension of the R-TS2 model. The constraints of the PB-R-TS2 model of the PB-R-PFSP ensure that the following two extra conditions hold.

- The number of the palettes is equal to K which implies that the j th job of the sequence cannot start its processing on the first machine until the $(j - K)$ th job of the order is finished on the last machine.

$$C_{M,j-K} + \sum_{i=1}^T P'_{1,i} Z'_{ij} \leq C_{1j} \quad K + 1 \leq j \leq N \tag{14}$$

- At most b_r jobs can wait in the buffer between machines r and $r + 1$. This condition is formulated as

$$C_{r+1,j-b_r-1} - \sum_{i=1}^T P'_{r+1,i} Z'_{i,j-b_r-1} + \sum_{i=1}^T P'_{r,i} Z'_{ij} \leq C_{rj} \tag{15}$$

$$1 \leq r \leq M - 1, 2 + b_r \leq j \leq N$$

The PB-R-TS2 model can be formalized as follows below.

Minimize (13) subject to (8)–(12), (14) and (15).

4.2 Lower Bounds for the PFSP Model with Heuristic Methods

For the production line managers it is very helpful to get a guaranteed lower bound to the optimum since it can be used to estimate how far the solution is from the optimal solution. The MILP solvers always provide the user with a guaranteed lower

bound. However, for a high quality bound we need an excellent MILP solver which could not be available for the users for several reasons.

In this subsection we provide a pretty simple heuristic method to get a lower bound. To this aim we relax some of the assumptions of the problem such that the relaxed problem should be easily solvable. Then the optimal makespan of the relaxed problem is a valid lower bound for the original problem.

To get a lower bound we relax the constraints that every machine can process at most one job at a time. This idea was introduced by Lageweg in [7]. Suppose that M_k and M_l are different machines and relax the assumption that every other machine can process at most one job at a time (i.e. the other machine can process any number of jobs at a time). In the relaxed problem each job starts its processing on the first machine at time 0 and job i is ready for processing on machine M_k at time $\sum_{r=1}^{k-1} P_{ri}$. Similarly, when job i is finished on machine M_k then it takes $\sum_{r=k+1}^{l-1} P_{ri}$ time for it to be ready for processing on machine M_l and moreover after job i is finished on machine M_l then it takes $\sum_{r=l+1}^m P_{ri}$ time for it to process on machines $M_{l+1}, M_{l+2}, \dots, M_m$. This means that the resulting relaxed problem is a two-machine permutation flow shop problem with release dates, time lags and delivery times with objective function minimizing the makespan. This problem is denoted by $F2|r_j, l_j, q_j, prmu|C_{max}$ where for all job j the release date r_j is given by

$$r_j = \begin{cases} \sum_{i=1}^{k-1} P_{ij} & \text{if } k > 1 \\ 0 & \text{if } k = 1, \end{cases}$$

the time lag l_j is given by

$$l_j = \begin{cases} \sum_{i=k+1}^{l-1} P_{ij} & \text{if } k < l - 1 \\ 0 & \text{if } k = l - 1, \end{cases}$$

and the delivery time q_j is given by

$$q_j = \begin{cases} \sum_{i=l+1}^m P_{ij} & \text{if } l < m \\ 0 & \text{if } l = m. \end{cases}$$

Here the release date r_j of job j means that we can not start processing job j on the first machine earlier than r_j . The time lag l_j of job j means that after job j finishes on the first machine it has to wait at least l_j time before we can start its processing on the second machine. The delivery time q_j of job j can be thought as the time of post processing j after it finishes on the second machine. Finally $prmu$ states that only permutation schedules are allowed.

Since this is still a hard problem to solve we have to relax this problem too. Fixing the release dates and delivery times for every job j to $\min_{j \in J} r_j$ and $\min_{j \in J} q_j$ we get a two-machine PFSP with time lags $F2|l_j, prmu|C_{max}$. It was shown by Rinnoy Kan

[6] that this problem can be solved in polynomial time by applying the Johnson's rule [4]. Hence for machine pairs (M_k, M_l) a valid lower bound for the original PFSP is

$$LB_{k,l} = \min_{j \in J} r_{kj} + C_{\max}^{kl} + \min_{j \in J} q_{kj}$$

where C_{\max}^{kl} denotes the optimal makespan of the problem $F2|l_j, prmu|C_{\max}$. By running through all machine pairs a valid lower bound for the original PFSP is

$$LB = \max_{1 \leq k < l} LB_{kl}.$$

5 Numerical Experiments

5.1 Test Problems

To analyze the performance of the heuristics and the MILP models on academic and industrial problems we generated two sets of test problems. The problems in both sets can be described with three parameters: the number of jobs N , the number of types T and the number of machines M . The number of machines was fixed $M = 50$, and we generated problems for $N = 100$ and $N = 200$. The values of T were 5, 10, 20, 50, 100 for $N = 100$, and 5, 10, 20, 50, 100, 200 for $N = 200$. For every triple we generated 5 problems. This procedure gave us $(5 + 6) \times 5 = 55$ processing time matrices for both sets. The two sets differed in the distribution of the values of the processing times.

In the first set we tried to create industrial problems. Hence we took a real problem from the industry, containing 11 types and 50 machines, and for each machine r we calculated the mean m_r and the standard deviation σ_r of the processing times on machine r . After that for each machine r the processing times on machine r were random numbers drawn from the normal distribution with parameters m_r and σ_r .

In the second set we created academic type problems with the procedure introduced by Taillard [20]. This means that the processing times were random integers coming from a uniform distribution with range $[0, 99]$.

5.2 Results

Three heuristics (NEH, TABU SEARCH and PACO) and one MILP model (R-TS2 model) were applied to solve the test problems. The formulations of the TS2 model were written in GAMS modeling language and solved by using CPLEX 12.3 on an Intel Xeon E31225 3.1 GHz personal computer equipped with 4 GB RAM. The

CPLEX options employed were mixed integer programming, parallel mode with four threads and a time limit of 600 s.

The relative gap of a solution is calculated by the formula

$$relative\ gap = 100 \cdot \frac{C_{best} - LB}{C_{best}} \tag{16}$$

where C_{best} is the makespan of the solution, and if CPLEX optimally solved the TS2 model in 10 min then LB is equal to the optimal makespan of the problem else LB is the lower bound of the optimum calculated the way it was described in Sect. 4.2. The average relative gaps are presented in Tables 2 and 3.

Table 2 Average relative gaps (in percentage) in the industrial problems

N	T	NEH	Tabu	Paco	R-TS2
100	5	1.14	0.52	0.18	0
100	10	0.52	0.1	0	0
100	20	1.37	0.74	0.47	0.35
100	50	1.49	0.24	0.25	1.54
100	100	1.16	0.21	0.26	*
200	5	0.86	0.34	0.06	0.05
200	10	0.24	0.01	0.16	0.03
200	20	0.49	0.45	0.11	1
200	50	0.35	0.13	0.06	*
200	100	0.49	0.09	0.11	*
200	200	0.52	0.38	0.25	*

N = number of jobs, T = number of types, number of machines $M = 50$

*At least in one instance no solution were found in 10 min

Table 3 Average relative gaps (in percentage) in the Talliard problems

N	T	NEH	Tabu	Paco	R-TS2
100	5	7.59	5.88	3.55	8.83
100	10	13.80	8.74	6.81	*
100	20	12.79	8.68	8.78	*
100	50	15.88	12.26	12.78	*
100	100	18.05	14.94	15.30	*
200	5	3.62	2.68	0.92	6.07
200	10	7.62	3.95	1.68	*
200	20	8.73	4.74	4.00	*
200	50	11.64	8.55	8.07	*
200	100	12.90	10.21	9.85	*
200	200	13.96	11.86	11.85	*

N = number of jobs, T = number of types, number of machines $M = 50$

*At least in one instance no solution were found in 10 min

In the industrial problems for problem sizes ($N = 100, T = 5$) and ($N = 100, T = 10$) the MILP model R-TS2 found the optimal solution in all five instances while for problem sizes ($N = 200, T = 5$) and ($N = 200, T = 10$) the MILP model R-TS2 found the optimal solution in four of the five instances. For larger problems ($(N = 100, T = 100), (N=200, T=50,100,200)$) the R-TS2 model did not even find an initial solution in the 10-min time limit. Overall the Tabu search found the optimal solution in 18 of the 55 industrial problems while PACO found the optimal solution in 23 of the 55 instances.

In contrast to the industrial problems in the Talliard-like problems none of the four methods found the proven optimal solution in any of the 55 instances. It can be seen in Table 3 that for problem sizes with more than five types the R-TS2 MILP model did not even find an initial solution in any of the five instances. Furthermore the NEH, Tabu search and Paco heuristics have much larger relative gaps in the Taillard like problems than in the industrial like problems. The explanation of this phenomenon is that the algorithm described in Sect. 4.2 gives much stronger lower bounds in the industrial like problems than in the Taillard-like problems. Table 4 contains the lower bounds and optimal value of the 10 instances of industrial problems with problem size ($N = 100, T = 5$) and ($N = 100, T = 10$). It can be seen that in 4 of the 10 instances the lower bound calculated by the algorithm of Sect. 4.2 equals to the optimum of the problem.

5.2.1 Problems with Finite Buffer Sizes and Palettes

For the industrial problems with sizes ($N = 100; T = 5$) and ($N = 100; T = 10$) we solved the overall 10 instances with finite buffer sizes between the machines and finite number of palettes using the PB-R-TS2 model. The buffer sizes were chosen from real-world problems and the number of palettes was $K = 55$. We compared the optimal values of these PB-R-PFSPs with the makespan given by the Tabu search applied to the PFSP with infinite buffer sizes and infinite number of palettes (which can be considered as the relaxation of the PB-R-PFSP). It turned

Table 4 Comparing the lower bounds and the optimal values in the industrial problems

N	T	Lower bound	Optimum
100	5	48,721	48,721
		50,351	50,351
		52,527	52,814
		49,160	49,707
		50,621	50,789
100	10	49,484	49,607
		49,314	49,521
		47,632	47,632
		48,840	48,840
		49,556	49,627

out that in 9 of the 10 instances the solution given by Tabu search applied to the PFSP (with infinite buffer sizes and infinite number of palettes) was optimal for the PB-R-PFSP problem, too. In the remaining one case the Tabu search applied to the PFSP gave a solution to the PB-R-PFSP with relative gap 0.51.

6 Conclusions

We can summarize the lessons learnt with the observations as follows.

- Large-scale R-PFSPs in which the number of types is small and the machines are unbalanced can be solved efficiently by using MILP models and exact solvers.
- Both the Tabu Search and the PACO heuristics give good solutions (close to the optimum) for large-scale PFSPs containing unbalanced machines.
- For PFSPs containing unbalanced machines the two-machine relaxation of the problem gives lower bound close to the optimum.
- For industrial like PB-R-PFSPs (R-PFSPs with palettes and limited buffers) one may compute a good (initial) solution of the problem by omitting the palettes and the buffer sizes between the machines (i.e. setting the number of palettes and buffer sizes between consecutive machines to infinite) and solving the relaxed R-PFSP.

These scientific results can effectively be used in the digital factory environment. We consider a production line with planning and shop floor software tools of digital factory to collect, analyse and process real life manufacturing data. With integration of the solving methods investigated by our work into the planning and scheduling process of the digital factory, we can produce effective production schedule supporting the end-to-end digital integration goal of the digital factory. This integrated production schedule can be the one of the basics for the smart and networked Industry4.0 production environment.

References

1. Baker, K.R.: Introduction to Sequencing and Scheduling. Wiley, New York (1974)
2. Hajba, T., Horváth, Z.: New effective MILP models for PFSPs arising from real applications. *Cent. J. Oper. Res.* **21**, 729–744 (2012)
3. Hajba, T., Horváth, Z.: MILP models for the optimization of real production lines. *Cent. J. Oper. Res.* **23**, 899–912 (2015)
4. Johnson, S.M., Optimal two -and three stage production schedules with setup times included. *Nav. Res. Logist. Q.* **1**(1), 61–68 (1954)
5. Jósvali, J.: Production process modeling and planning with simulation method, mounting process optimisation. In: The International Conference on Modeling and Applied Simulation. Universidad de La Laguna, 23–25 September 2009, pp. 240–245 (2009)
6. Kan, A.H.G.R.: Machine Scheduling Problems: Classifications, Complexity and Computation. Nijhoff, The Hague (1976)

7. Lageweg, B.J., Lenstra, J.K., Kan, A.H.G.R.: A general bounding scheme for the permutation flow-shop problem. *Oper. Res.* **26**, 53–67 (1978)
8. Liao, C.L., You, C.T.: An improved formulation for the job-shop scheduling problem. *J. Oper. Res. Soc.* **43**, 1047–1054 (1992)
9. Manne, A.S.: On the job-shop scheduling problem. *Oper. Res.* **8**, 219–223 (1960)
10. Nawaz, M., Enscore, E.E., Ham, I.: A heuristic algorithm for the m -machine n -job flow-shop sequencing problem. *Omega* **11**, 91–95 (1983)
11. Nowicki, E., Smutnicki, C.: A fast tabu search algorithm for the permutation flow-shop problem. *Eur. J. Oper. Res.* **91**, 160–175 (1996)
12. Pan, C.H.: A study of integer programming formulations for scheduling problems. *Int. J. Sys. Sci.* **28**, 33–41 (1997)
13. Rajendran, C., Ziegler, H.: Ant-colony algorithms for permutation flowshop scheduling to minimize total makespan/total flowtime of jobs. *Eur. J. Oper. Res.* **155**, 426–438 (2004)
14. Stafford, E.F.: On the development of a mixed-integer linear programming model for the flowshop sequencing problem. *J. Oper. Res. Soc.* **39**, 1163–1174 (1988)
15. Stafford, E.F., Tseng, F.T.: On the Strikar-Gosh MILP model for the $N \times M$ SDST flowshop problem. *Int. J. Prod. Res.* **28**, 1817–1830 (1990)
16. Stafford, E.F., Tseng, F.T.: Two models for a family of flowshop sequencing problems. *Eur. J. Oper. Res.* **142**, 282–293 (2002)
17. Stafford, E.F., Tseng, F.T.: New MILP models for the permutation flowshop problem. *J. Oper. Res. Soc.* **59**, 1373–1386 (2008)
18. Stafford, E.F., Tseng, F.T., Gupta, N.D.: An empirical analysis of integer programming formulations for the permutation flowshop. *Omega* **32**, 285–293 (2004)
19. Stafford, E.F., Tseng, F.T., Gupta, N.D.: Comparative evaluation of the MILP Flowshop models. *J. Oper. Res. Soc.* **56**, 88–101 (2005)
20. Taillard, E.: Benchmarks for basic scheduling problems. *Eur. J. Oper. Res.* **64**, 278–285 (1993)
21. VDI: Digital Factory Fundamentals. VDI 4499 Guideline, Düsseldorf (2008)
22. Wagner, H.M.: An integer linear-programming model for machine scheduling. *Nav. Res. Log. Q.* **6**, 131–140 (1959)
23. Wilson, J.M.: Alternative formulations of a flow-shop scheduling problem. *J. Oper. Res. Soc.* **40**, 395–399 (1989)

Automatic Reconfiguration of Robotic Welding Cells

Dietmar Hömberg, Chantal Landry, Martin Skutella, and Wolfgang A. Welz

Abstract Robotic welding cells are at the core of many complex production systems, especially in automotive industry. In these cells, a certain number of robots perform spot welding tasks on a workpiece. The configuration of the cells can have a huge impact on the production rate. The smaller the cycle time is, the higher the production is. In this paper, we present a complete algorithm that automatically configures the welding cell such that the given cycle time of the production process is kept. This algorithm assigns tasks to the different robots, decides in which order the tasks are executed and computes the fastest collision-free trajectory of the robots between two consecutive tasks.

1 Introduction

Industrial manufacturing has by now reached a high degree of automation. Complex production lines have been created and consist of robots grouped together in workcells connected by conveyor belts and further devices for materials supply and temporary storage [32, 35]. Efficient production lines are essential to ensure the competitiveness of manufacturers. A major point of this quest for efficiency lies in the design of the workcells. These cells are usually configured by hand, the configuration lasting up to several months [31]. Our aim is to automate their design by using mathematical techniques. The resulting configuration will increase the production rate, whereas the time to reconfigure a cell will be drastically decreased.

D. Hömberg
Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany
e-mail: dietmar.hoemberg@wias-berlin.de

C. Landry (✉)
Zurich University of Applied Sciences, Winterthur, Switzerland
e-mail: chantal.landry@zhaw.ch

M. Skutella • W.A. Welz
Technische Universität Berlin, Berlin, Germany
e-mail: skutella@math.tu-berlin.de; welz@math.tu-berlin.de

Given the Computer Aided Design data of the workpiece, the location of the tasks and the number of robots, the aim is to assign tasks to the different robots and to decide in which order the tasks are executed such that the given cycle time of the production process is kept. In order to identify the overall processing times of these planned operations it is further necessary to automatically compute how the robots move to the next assigned task. We have called this problem, where spot welding tasks are performed, the *Welding Cell Problem (WCP)*. Up to now, the reconfiguration of such a cell is basically done by hand. The goal of this paper is to model and solve WCP.

It is worth noting that the task assignment, the sequencing of the tasks and the robot motion-planning cannot be handled separately. On one hand, the task assignment and the sequencing depend on the travel time between two tasks. This time is obtained by computing the fastest collision-free trajectory between these two given tasks. On the other hand, one needs to know between which tasks the robot motion-planning must be computed.

There is not much literature on the particular welding cell problem under consideration. A somewhat related laser welding problem has been discussed in [27]. For this particular problem the authors propose an approach to find makespan minimal tours for welding robots sharing limited laser sources that avoid conflicts with each other. The aspects of path and trajectory planning have not been considered in this context.

In [31, 32], Segeborn et al. optimize the weld load balancing in the whole production line. The tasks are assigned among all robots in the assembly line such that the cycle time is minimized. The authors do not include the computation of an optimal motion planning between the tasks. In [32], they spatially separate the robot weld workloads. In [31], the path planning instead of the optimal motion planning is used and performed by the IPS (Industrial Path Solutions) CAE tool. Björkenstam et al. in [5] include the optimal motion planning of robots between the tasks in their method to find an efficient sequencing of the welding tasks. They consider only one robot and the tasks are already assigned to the robot.

In [34], Spensieri et al. consider a similar problem to ours: to minimize the cycle time of a multirobot station. They develop an iterative method that first assigns tasks among the robot, then performs the sequencing and eventually coordinates the paths to avoid collisions. Unlike Spensieri et al., we solve the task assignment and the sequencing in one step and take into consideration optimal trajectories. Our algorithm assures to find a feasible solution of the WCP, whereas in [34] the method can stop after reaching the maximum number of iterations or maximum available time.

We have presented in [20] an algorithm that tries to efficiently combine continuous motion planning and discrete optimization without the need to calculate all the trajectories in advance. The proposed algorithm is an iterative method that couples discrete optimization with collision detection and optimal control problems. For this, the computationally less expensive discrete optimization process needs to be resolved in every iteration. Such a strategy has been developed in [28], but on a simpler case. The authors considered one robot that had to perform several tasks in a sequence to be optimally determined. This means, that no task assignment and no

collision avoidance were required. Only the sequencing of the tasks and the robot motion were addressed.

In this paper we now present a more advanced, combined approach that integrates the solving of the exact optimal control problem into the actual column generation approach of the underlying combinatorial assignment and sequencing problem. Thus, in Sect. 2 we present the actual Welding Cell Problem and then discuss the essential ideas and requirements for an algorithm combining the two parts. After that, our solution approach for the discrete part, namely the conflict free assigning and sequencing, is described in Sect. 3. In this section, we also explain how it is now possible, with only slight modifications, to integrate the trajectory calculations into the implemented column generation approach. The resulting algorithm solves the WCP by efficiently combining both parts. It manages to keep the trajectory calculations as low as possible while at the same time as much information as possible is reused from the current status of the discrete solution procedure. The collision-free motion planning as well as the detection of conflicts in the calculated trajectories are then discussed in Sects. 4 and 5. Numerical results are presented in Sect. 6. Our approach is applied to several two-dimensional workcells, that are composed of single body mobile robots.

2 Welding Cell Problem

A welding cell in a production line consists of a workpiece, a certain number of industrial robots and some obstacles such as the conveyor belt. On the workpiece, some welding tasks must be performed. The goal is to plan trajectories for each of the robots so that in the end all weld points are processed while the makespan of all trajectories is below the given cycle time of the production process.

Let J be the set of welding tasks. In fact, an element $j \in J$ represents at the same time a task and its location on the workpiece. Let R be the set of industrial robots in the workcell. Since each robot has its own characteristics (size, weight, welding tongue, etc.), a robot may not be able to perform all tasks. Therefore, each task $j \in J$ has a working set $W_j \subseteq R$ containing the robots that can process this job.

Let s^r be the initial and end position of robot $r \in R$ and let V be the set containing all task locations and initial positions, that is: $V := \{s^r : r \in R\} \cup J$. With these definitions, we can represent the welding cell problem as a complete directed graph $G = (V, A)$, where the node set is V and the arc set $A := V \times V$. An arc $a = (v_t, v_h)$ with the tail node $v_t \in V$ and the head node $v_h \in V$ corresponds to the fastest trajectory of a robot moving from v_t to v_h . This trajectory is collision-free with the obstacles present in the workcell. Let τ_a^r be the travel time needed by robot r to move from the tail to the head of the arc a . These travel times are the weights of the arcs and they are assumed to be at least zero. We note, that the

graph G is identical for all robots $r \in R$ only the weights differ. The travel times related to robot r are stored in the set \mathcal{D}^r , that is:

$$\mathcal{D}^r = \{\tau_a^r : r \text{ can move from } v_t \text{ to } v_h \text{ with } a = (v_t, v_h) \in A\}.$$

Moreover, we store all travel times in the set \mathcal{D} , i.e. $\mathcal{D} = \cup_{r \in R} \mathcal{D}^r$.

For each robot $r \in R$, we are looking for a tour $T^r = (a_1, \dots, a_{n^r})$, starting and ending in the position s^r , and a schedule I^r that assigns each arc $a \in T^r$ a left-closed time interval $I_a^r = [\underline{t}_a^r, \overline{t}_a^r)$, where \underline{t}_a^r corresponds to the departure time in the tail of a . This interval describes when the corresponding arc a is used by robot r . For formal reasons, we represent waiting and processing times in a node in T^r by a longer time interval of the arc used previously by robot r . It is however possible to start a tour later and thus effectively leave the start position s^r later.

No collision shall occur between robots moving with respect to their schedule. The information on colliding robots is stored in the set of conflicts \mathcal{C} . Let I_1 be the interval for which the arc a_1 is used and I_2 the corresponding interval for arc a_2 . The element $(r_1, a_1, I_1, r_2, a_2, I_2)$ is in \mathcal{C} if and only if at least one collision occurs between r_1 and r_2 , while r_1 (resp. r_2) moves from the tail to the head of a_1 (resp. a_2) during I_1 (resp. I_2).

Let t^{cycle} be the cycle time of the production process. The Welding Cell Problem can now be outlined as follows:

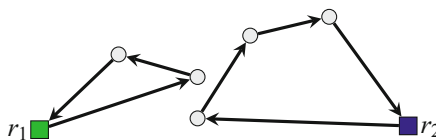
Find scheduled tours $T^r = (a_1, \dots, a_{n^r})$, $r \in R$, such that: (WCP)

1. Each tour T^r starts and ends in the position s^r , with $r \in R$.
2. Each job $j \in J$ is visited in exactly one tour T^r , with $r \in W_j$, and j is the tail of exactly one arc $a \in T^r$.
3. The travel times τ_a^r are minimized and the corresponding trajectory is collision-free with the obstacles present in the workcell.
4. For each robot $r \in R$, the schedule I^r must be feasible for tour T^r :

$$\begin{aligned} \tau_{a_i}^r &\leq \overline{t}_{a_i}^r - \underline{t}_{a_i}^r, & \forall i \in \{1, \dots, n^r\}, \\ \overline{t}_{a_i}^r &= \underline{t}_{a_{i+1}}^r, & \forall i \in \{1, \dots, n^r - 1\}. \end{aligned}$$

5. All robot moves are conflict-free with respect to their schedule.
6. The cycle time is kept: $\overline{t}_{a_{n^r}}^r \leq t^{\text{cycle}}$, $\forall r \in R$.

To further illustrate this definition, we give a simple example of scheduled tours. Consider the following solution for a WCP instance with two robots r_1 and r_2 . The tours T^{r_1} and T^{r_2} are given as follows:



A feasible schedule for these tours is for example given by:

$$I^1 = \{[0, 4), [4, 6), [6, 7)\},$$

$$I^2 = \{[0, 2), [2, 4), [4, 6), [6, 10)\} .$$

To solve (WCP), one needs to know the value of all travel times in \mathcal{D} and information for conflicting trajectories, which is given by \mathcal{C} . However, the travel times are the solution of an optimal control problem (see Sect. 4) and their computation is time consuming. Consequently, we first consider approximated travel times and compute the exact travel times only when needed.

The approximated times are the travel times of approximated trajectories. Such approximated trajectories can for example be obtained by putting a regular grid on the workcell, where the nodes of the grid which are located in or in the neighborhood of an obstacle are removed from the set of nodes. Now, a feasible trajectory between two given points corresponds to the shortest path—along that grid—connecting these two points. The shortest path is obtained by applying a Dijkstra-like algorithm in which the usage of nodes that are very close to the obstacles are penalized. The resulting trajectory is a sequence of segment lines which do not collide with the obstacles. These approximated travel times are much easier to obtain and are denoted by \mathcal{D}' .

As the conflict-information depends on the actual movements of the robots, they cannot be derived from the calculated line segments without marking segments conflicting that are actually feasible with respect to the exact trajectory. Thus, we initialize the corresponding conflict set \mathcal{C}' to the empty set, in order to avoid false positives.

The basic idea to solve (WCP) can be described as follows: Instead of solving (WCP) by finding scheduled tours with all times \mathcal{D} and conflict information \mathcal{C} , we look for tours according to different conflicts \mathcal{C}' and travel times \mathcal{D}' . The computed tours with respect to \mathcal{D}' and \mathcal{C}' will in general not be feasible for (WCP) as the travel times can differ largely and so far no conflict has been considered. However, they give us a promising assignment and sequencing. As a second step, we replace the approximated trajectories selected in the tours by computing the exact ones. Then, we check if a collision occurs between the robots moving according to their respective tours. Finally, we add the newly calculated conflicts to \mathcal{C}' and update the corresponding travel times in \mathcal{D}' . New promising scheduled tours can now be computed with respect to the updated sets \mathcal{D}' and \mathcal{C}' . This process is repeated iteratively until the returned solution no longer contains approximated line-segments and is conflict-free. This idea is formalized in Algorithm 1.

Thus, (WCP) consists of three major parts:

- **Conflict-free assignment and sequencing of welding tasks:** when the travel times between all weld points as well as conflict-information are known, finding a tour for each robot such that in the given cycle time all the tasks have been processed (lines 3 and 15 in Algorithm 1), is a variant of combinatorial optimization problems, such as the traveling salesman problem, vehicle routing

Algorithm 1: WCP algorithm

```

1 initialize travel times  $\mathcal{D}'$  w.r.t. the approximated trajectories;
2 initialize conflict set  $\mathcal{C}' = \emptyset$ ;
3 Find scheduled tours w.r.t.  $\mathcal{D}'$  and  $\mathcal{C}'$ ;
4 while feasible scheduled tours found do
5   foreach arc  $a$  in found tours do
6     if trajectory for  $a$  is estimated then
7       calculate exact trajectory for  $a$ ;
8       in  $\mathcal{D}'$  replace travel time of  $a$  with exact value;
9   if new scheduled tours with exact trajectories and travel times is still feasible then
10    check trajectories for conflicts;
11    if conflicts present then
12      add conflicting trajectories and usage-intervals to  $\mathcal{C}'$ ;
13    else
14      return found feasible scheduled tours;
15 Find scheduled tours w.r.t.  $\mathcal{D}'$  and  $\mathcal{C}'$ ;
16 return  $\emptyset$ ;
```

problems and scheduling [27]. This part is in the following called Combinatorial Problem (CP) and is presented in Sect. 3.

- **Motion planning of robots:** The computation of the fastest trajectory of a robot that avoids obstacles and observes the dynamic laws (line 7 in Algorithm 1) is called *kinodynamic motion planning* [11] and is a typical instance of optimal control problems. Details on this part are given in Sect. 4.
- **Detection of conflicting trajectories:** to detect collisions between robots moving along specified trajectories (line 10 in Algorithm 1), techniques from computational geometry must be used in order to efficiently approximate, or even compute, the distance between the robots. A description of this part is given in Sect. 5.

Therefore, we can observe that solving (WCP) requires an efficient interplay between discrete mathematics, nonlinear optimization and computational geometry.

3 Conflict-Free Assignment and Sequencing

The task of the discrete problem is to find feasible scheduled tours with respect to given travel times \mathcal{D} and conflict information \mathcal{C} . This problem represents one of the major parts of (WCP) (lines 3 and 15 in Algorithm 1) and is in the following denoted by CP. The definition of conflicts, however, slightly differs from the last section: For our solution approach of the CP it is crucial that the conflict set does not depend on the usage-intervals. We therefore convert the conflict set \mathcal{C} into a slightly different

set $\tilde{\mathcal{C}}$ containing only arc pairs. It must be guaranteed that all conflicts with respect to \mathcal{C} are also conflicts for $\tilde{\mathcal{C}}$, which leads to the following stricter definition:

$$\tilde{\mathcal{C}} := \{(r_1, a_1, r_2, a_2) : \exists I_1, I_2 \text{ with } (r_1, a_1, I_1, r_2, a_2, I_2) \in \mathcal{C}\} .$$

Using this definition, two arcs a_1 and a_2 are said to be in conflict with respect to the set $\tilde{\mathcal{C}}$, if they are in use at the same time by the corresponding robots: Let I_1 be the interval for which a_1 is used and I_2 the corresponding interval for arc a_2 , then the tours containing the arcs a_1 and a_2 are conflicting w.r.t. $\tilde{\mathcal{C}}$, if I_1 and I_2 intersect. This is a much stronger definition of conflicts than in the previous section. However, it is thus assured that any solution which is feasible w.r.t. $\tilde{\mathcal{C}}$ is also always feasible w.r.t. \mathcal{C} .

This converted set $\tilde{\mathcal{C}}$ is then used to solve the discrete problem $\text{CP}(\mathcal{D}, \tilde{\mathcal{C}})$. It can be formulated as a very general set partitioning problem that finds a $\tilde{\mathcal{C}}$ -conflict-free solution which visits each weld point exactly once within the given cycle time. The CP can be modeled as follows:

$$\min \sum_{T \in \Omega} c_T x_T \quad (\text{MP})$$

$$\text{s.t. } \sum_{T \in \Omega} \delta_{vT} x_T = 1 \quad \forall v \in V \quad (1a)$$

$$x_{T_1} + x_{T_2} \leq 1 \quad \text{for all } \tilde{\mathcal{C}}\text{-conflicting pairs of tours } (T_1, T_2) \quad (1b)$$

$$x_T \in \{0, 1\} \quad \forall T \in \Omega \quad (1c)$$

The model (MP) is a representation of $\text{CP}(\mathcal{D}, \tilde{\mathcal{C}})$ as an *Integer Linear Program (ILP)* that has a binary variable for every feasible scheduled tour. The set containing all the feasible tours is denoted by Ω . The set Ω only contains those tours that fulfill the requirements of the WCP, i.e. the travel time of a tour according to \mathcal{D} must be lower than t^{cycle} and they cannot contain the same weld point more than once.

An ILP is an optimization problem that has a linear objective function which is minimized with respect to several constraints. These constraints can either be linear (in)equalities or integrality constraints, that force some variables to be integers. A program only containing linear constraints is called a *Linear Program (LP)*.

In (MP) the variables x_T specify whether the corresponding scheduled tour denoted by T is chosen or not. All values of x_T that are not 0 or 1 do not represent a feasible tour assignment and must not be used in a solution. The cost c_T of tour T can in principle be chosen arbitrarily as we are only interested in a feasible solution. By setting c_T to the energy consumption of tour T , the identical approach could be used to find a feasible solution that also minimizes the total energy consumption of all operations. The coefficient δ_{vT} specifies whether node v is visited by the scheduled tour T or not. This allows us to formulate the condition that every node needs to be visited exactly once in $|V|$ linear constraints (1a). As the starting position s^r is also

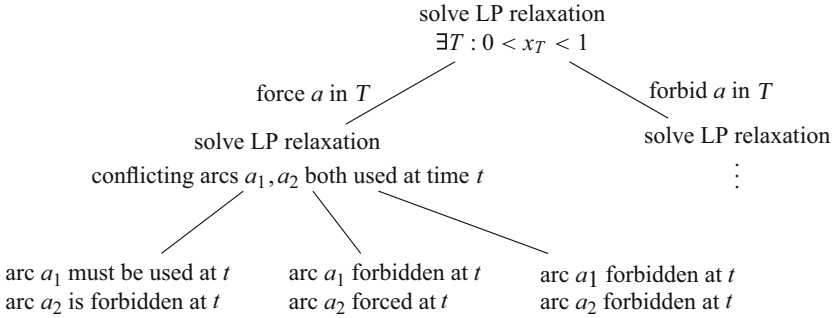


Fig. 1 Sketch of a tree resulting from the different branching rules for (MP): It was first branched on the fractional arc a and then on the conflicting arcs a_1 and a_2

contained in V , exactly one tour will be assigned to each robot $r \in R$. Conflicts are avoided in (1b) by explicitly listing all conflicting pairs of tours that contain arcs in $\tilde{\mathcal{C}}$ and forbidding that in each of these pairs more than one variable is set to 1.

In general, this would allow us to solve (MP)—and thus the CP—with any available ILP solver, such as SCIP¹ or CPLEX.² Unfortunately, due to the sheer number of variables—one for every existing tour with all its possibilities to wait—it is impossible to explicitly formulate and solve (MP) even for a handful of nodes. It is however possible to solve the problem for a much smaller subset of variables and then successively adding further variables from Ω in such a way that optimality of the final solution can be guaranteed, although not all possible variables have been considered. This process is called column generation and it is a solution approach which is used for many combinatorial optimization problems such as Vehicle Routing Problems, see e.g. [22]. While in general column generation can only be used for linear programs it can also be extended for ILPs by using the underlying linear program as a relaxation and then enforcing the integrality constraints by fixing them in different subproblems which in this context are also called branches. This technique is then called branch-and-price. To apply this technique it is necessary to integrate heavily problem dependent parts into the ILP solving process. An introduction to this technique can be found, e.g. in [10].

A similar difficulty occurs with the conflict constraints (1b), as even for a smaller number of paths there is a large number of inequalities to express mutually exclusion of tours. Numerical experiments showed that enforcing conflict-free tours using explicit branches usually helps to improve the solution time.

An approach that uses column generation as well as explicit conflict constraints for a very similar problem has also been described in [33]. For our problem, we implemented the same branching rules to enforce the constraints (1b) and (1c). A schematic diagram of this approach is shown in Fig. 1.

¹SCIP Optimization Suite: <http://scip.zib.de>.

²IBM ILOG CPLEX: www.ibm.com/software/products/en/ibmilogcpleoptistud.

- We need to assure that the variables are always either 0 or 1. If a fractional value occurs, the integrality can be guaranteed by dividing the problem into two subproblems. In the first subproblem arc a must be used in a tour, while in the second the arc is prohibited entirely and it cannot be part of any tour. As all arcs must either be used by exactly one tour or none at all, this process enforces the constraints (1c). The step of dividing the problem into several subproblems in such a way that the total search space decreases but no feasible solution is lost is called branching.

Since in the CP every node needs to be visited exactly once, this branching rule can be easily implemented by either removing arc a in G or by removing all other arcs leaving the tail of a .

- A similar approach is used to enforce the explicit constraints (1b): If two conflicting arcs are in use, this situation can be resolved by creating three branches in which either both of the arcs are forbidden for one of the conflicting time steps or one of them must be used while the other one cannot.

Each branch corresponds to a new problem very similar to (MP) where only constraints are added that enforce the properties of the subproblems stated above. All these branching decisions need to be taken into account for the optimization problem that identifies promising new tours, the so-called *pricing problem*. This results in a problem where the shortest scheduled tour with respect to the corresponding dual variables needs to be found. To correctly incorporate the branching decisions, such a scheduled tour must also respect the time windows introduced by the conflict branching. A more detailed explanation of the resulting pricing problem and different solution approaches can be found in [37]. For the performed computational experiments this branch-and-price algorithm has been implemented using the framework provided by SCIP [2].

3.1 Revisiting the WCP Algorithm

By using the CP solution approach described in the beginning of this section as a starting point, we can now devise a revised WCP algorithm. In the approach described in Sect. 2 every time after the computation of updated travel times the CP and thus the ILP described as (MP) needs to be solved again for the new sets \mathcal{D}' and \mathcal{C}' . An improved algorithm integrates the trajectory calculations directly into the solution process of (MP) and therefore leaves unchanged tours untouched. Fortunately, the described branch-and-price based approach is especially well suited for such a procedure. Explicit restarts are not necessary and the continuous part can be seamlessly integrated. This idea can be formalized very conveniently as a *Constraint Integer Program (CIP)*. This generalization of ILPs has been first proposed in [1] and it basically represents an ILP that can have additional arbitrary constraints which are enforced using branching. The framework SCIP itself is designed as a constraint integer program solver and it is therefore very straight

forward to implement the following problem using SCIP:

initialize \mathcal{D}' and \mathcal{C}' (CIP)

$$\min \sum_{T \in \Omega} c_T x_T$$

$$\text{s.t. } \sum_{T \in \Omega} \delta_{v_T} x_T = 1 \quad \forall v \in V \quad (2a)$$

$$x_T \in \{0, 1\} \quad \forall T \in \Omega \quad (2b)$$

$$\{T : T \in \Omega, x_T = 1\} \text{ is } \tilde{\mathcal{C}}' \text{-conflict-free} \quad (2c)$$

$$\{T : T \in \Omega, x_T = 1\} \text{ contains no estimated times} \quad (2d)$$

$$\{T : T \in \Omega, x_T = 1\} \text{ is conflict-free} \quad (2e)$$

The variables, objective function as well as the first constraints (2a) and (2b) are completely identical to the ILP described in Sect. 2, they just depend on potentially different travel times \mathcal{D}' . The constraint (2c) is a reformulation of (1b) using the conflict set \mathcal{C}' . However, the last two constraints (2d) and (2e) have been added and they corresponds to the checks performed in lines 6 and 11 of Algorithm 1. We assume that all constraints are checked and enforced from top to bottom. Therefore, if constraint (2d) is reached we are guaranteed to have an integral solution at hand that is also conflict-free with respect to the converted $\tilde{\mathcal{C}}'$ and thus also \mathcal{C}' . The actual constraint (2d) now has to check whether that solution only consists of exact trajectories or whether there exists an arc a for which \mathcal{D}' corresponds to the estimated value. If this is the case, the exact subproblem (described in Sect. 4) is called to calculate the correct trajectory segment for that arc. Now, we first remove all variables x_T from (CIP) representing tours that contain the updated arc a and the corresponding value in \mathcal{D}' is updated. The process of column generation then assures that if the tour T is still promising even after the update it will be re-added to the problem. As all other tours remain in the problem, this can be interpreted as a warm start of $\text{CP}(\mathcal{D}, \tilde{\mathcal{C}})$. When an integral solution containing only updated exact distances has been found, we check in constraint (2e) whether these tours have conflicts that are not yet contained in \mathcal{C}' . This step corresponds to line 10 of Algorithm 1 and its actual process is described in Sect. 5. If no such conflicts exist, the tours represent a feasible solution for the WCP. Otherwise, we add the conflicting segment pair to the set \mathcal{C}' , calculate the corresponding $\tilde{\mathcal{C}}'$ and then branch on that particular newly added conflict.

As the re-generation of updated tours is automatically handled by the column generation approach, these ideas can be incorporated very easily into the code for the (MP). Since SCIP is build as a solver for constraint integer programs, the integrated parts can very conveniently be implemented as custom *constraint handlers*. When the constraint handler is called to check the current solution for feasibility, we call the programs of the continuous part and update the distances and conflict information accordingly.

3.2 Analysis of the Combined Algorithm

The correctness of Algorithm 1 and (CIP) can be shown with the following observation. The only requirement is that the travel times \mathcal{D}' are always *underestimated*, i.e. that calculating the exact travel times never decreases the information currently present in \mathcal{D}' .

Lemma 1 *Let \mathcal{D} contain the exact travel times and let \mathcal{C} be the complete conflict set. Any feasible solution of the corresponding $\text{CP}(\mathcal{D}, \mathcal{C})$ is also feasible for $\text{CP}(\mathcal{D}', \mathcal{C}')$ for any \mathcal{D}' and \mathcal{C}' occurring in the solution process of (CIP).*

Proof The proof of this lemma can be found in [37].

As the number of jobs and thus also $|\tilde{\mathcal{C}}|$ is finite, the solution process of (CIP) will eventually terminate. However, usually only a small subset of the trajectory-segments needs to be computed. The computationally expensive part, however, remains in enforcing constraint (2d), where the initial distances are updated with the correct ones.

Lemma 1 can then directly be used to show that the algorithm actually never misses a feasible solution:

Theorem 1 *If (CIP) is solved using the initial sets $\mathcal{C}' \subseteq \mathcal{C}$ and \mathcal{D}' containing only underestimated distances, then it returns a feasible solution, if $\text{CP}(\mathcal{D}, \mathcal{C})$ is feasible.*

This shows that (CIP) can indeed be used to obtain a solution for the WCP. Its very convenient representation as a constraint integer program together with the column generation approach assure that we find a feasible solution of the WCP while at the same time much information from previous results of the intermediate CP solution processes is reused.

4 Time-Optimal Kinodynamic Motion Planning

In this section, we present a method to compute the motion of a robot $r \in R$ that moves between two given task locations, i.e. line 7 in Algorithm 1. Our goal is to compute the fastest trajectory of the robot that avoids fixed obstacles (kinematic constraints) and observes the dynamic laws and the bounds on the velocity or the acceleration (dynamic constraint). This type of problem is called *time-optimal kinodynamic motion planning* [11, 21].

It is worth repeating that the collision avoidance is between the robot and the existing obstacles in the workcell such as the conveyor belt. We do not consider the possible collision between the moving robots here, as this conflict-avoidance is handled as part of CP described in Sect. 3.

Several methods have been developed to compute the optimal trajectories of the robots [4, 6, 11, 14, 15]. The trajectory is in each case the solution of an optimal control problem. The methods differ in the description of the collision avoidance with the static environment and in the strategy to solve the optimal control problem.

Let us consider a robot composed of m links which are connected by revolute joints. Let $q = (q_1, \dots, q_m)$ denote the vector of joint angles at the joints of the robot. Moreover, let the vector $\dot{q} = (\dot{q}_1, \dots, \dot{q}_m)$ contain the joint angle velocities and let $u = (u_1, \dots, u_m)$ describe the torques applied at the center of gravity of each link. The robot is asked to move as fast as possible from a given position, denoted by $v_t \in V$, to a desire location, $v_h \in V$. Its motion is given in the Lagrangian form as follows

$$\begin{aligned} \frac{d}{dt}q(t) &= \dot{q}(t), \\ M(q(t)) \frac{d}{dt}\dot{q}(t) &= G(q(t), \dot{q}(t)) + F(q(t), u(t)), \end{aligned} \quad (3)$$

where $M(q)$ is the symmetric and positive definite mass matrix, $G(q, \dot{q})$ contains the generalized Coriolis forces and $F(q, u)$ is the vector of applied joint torques and gravity forces. The function F is linear in u .

The motion of robot r must follow (3), but also be collision-free with the obstacles present in the workcell. To establish the collision avoidance condition, robot r is approximated by a union of compact convex polyhedra, denoted by $\cup_{i=1}^{n_p} P_i$ with n_p denoting the number of polyhedra and P_i being a compact convex polyhedron. The robot must be included in the approximation, that is: $r \subseteq \cup_{i=1}^{n_p} P_i$.

Let us assume that the workcell contains n_q fixed obstacles, which are compact convex polyhedra and denoted by $Q_j, j = 1, \dots, n_q$. The approximation of robot r and the obstacles do not collide if and only if for each pair of polyhedra (P_i, Q_j) , with $i = 1, \dots, n_p$ and $j = 1, \dots, n_q$, the distance between the polyhedra is positive. To this end, we use the signed distance between two objects, which is negative if the objects intersect and non-negative otherwise. The signed distance between intersecting polyhedra is defined as follows [8, 18, 19]:

$$d(P_i, Q_j) = -\|w\|_2,$$

where d is the distance function and w is the smallest translational vector, so that $\text{int}(P_i + w) \cap Q_j = \emptyset$. Here, $\text{int}(P_i + w)$ denotes the interior of the polyhedron $P_i + w$, which is the polyhedron P_i translated by w . An illustration is given in Fig. 2.

Fig. 2 The polyhedra P_i and Q_j overlap. The vector w is the smallest vector such that $P_i + w$ and Q_j come into contact



If the polyhedra are disjoint, then d is simply the Hausdorff distance. In summary, the distance function between two convex compact polyhedra is given by

$$d(P_i, Q_j) = \begin{cases} -\|w\|_2, & \text{if } P_i \cap Q_j \neq \emptyset, \\ \text{dist}(P_i, Q_j), & \text{otherwise,} \end{cases}$$

where $\text{dist}(\cdot, \cdot)$ is the Hausdorff distance.

Since robot r moves, the polyhedra in the approximation of r evolve in time. A motion of P_i is the composition of a rotation with a translation. Both geometrical transformations depend on the joint angles $q(t)$, see [14, 23]. Therefore, we subsequently write $P_i(q(t))$ to denote the polyhedron at time t .

The collision avoidance constraint at time t is then obtained by imposing that the minimum distance between the approximation of the robot and the obstacles is larger than a safety margin, that is

$$\min_{\substack{i=1, \dots, n_p \\ j=1, \dots, n_q}} d(P_i(q(t)), Q_j) \geq \varepsilon, \quad (4)$$

where $\varepsilon > 0$ is the safety margin.

Let τ_f be the travel time of the robot between v_t and v_h . Combining (3) with (4) leads to the following kinodynamic motion planning problem between v_t and v_h :

$$\min \tau_f \quad \text{w.r.t. } q, \dot{q} \in W_{1, \infty}^m([0, \tau_f]), u \in L_{\infty}^m([0, \tau_f]) \quad (\text{OCP})$$

s.t. • equations of motion:

$$\frac{d}{dt} q(t) = \dot{q}(t), \quad \text{a.e. in } [0, \tau_f],$$

$$\frac{d}{dt} \dot{q}(t) = M(q(t))^{-1} (G(q(t), \dot{q}(t)) + F(q(t), u(t))), \quad \text{a.e. in } [0, \tau_f],$$

• collision avoidance:

$$\min_{\substack{i=1, \dots, n_p \\ j=1, \dots, n_q}} d(P_i(q(t)), Q_j) \geq \varepsilon, \quad \text{a.e. in } [0, \tau_f],$$

• boundary conditions:

$$g(q(0)) = v_t, \quad \dot{q}(0) = 0, \quad g(q(\tau_f)) = v_h, \quad \dot{q}(\tau_f) = 0,$$

• box constraints:

$$\underline{q} \leq q(t) \leq \bar{q}, \quad \underline{\dot{q}} \leq \dot{q}(t) \leq \bar{\dot{q}}, \quad \underline{u} \leq u(t) \leq \bar{u}, \quad \text{a.e. in } [0, \tau_f],$$

where $\underline{q}, \bar{q}, \underline{\dot{q}}, \bar{\dot{q}}, \underline{u}$ and \bar{u} are given lower and upper bound values, which are specific to the robot. The function $g(q(t))$ gives the position of the barycenter of the last link of the robot at time t . This link will perform the task. See [23] for an explicit formulation of g .

As usual $L^\infty([0, \tau_f])$ denotes the Banach space of essentially bounded functions mapping from $[0, \tau_f]$ into \mathbb{R}^m and $W_{1,\infty}^m([0, \tau_f])$ denotes the Banach space of absolutely continuous functions with essentially bounded derivative that map from $[0, \tau_f]$ into \mathbb{R}^m .

The problem (*OCP*) is an optimal control problem where the state variables are q and \dot{q} , and the control variable is u [13]. Other cost functions, such as minimizing the energy consumption, can be defined in (*OCP*). In (*OCP*), the dynamic constraints are the equations of motion and the box constraints. The collision avoidance constraint and the boundary conditions define the kinematic constraints. Moreover, let us observe that the collision avoidance constraint is not continuously differentiable because of the distance function d .

Since (*OCP*) contains state constraints and the dimension of the state variable is small, we choose to use a direct method to solve it [13, 14, 36]. The method involves first discretizing the control problem and transforming it into a finite-dimensional nonlinear optimization problem. The control variables are approximated by B-splines of order 2 and the ordinary differential equations are integrated with the classical Runge-Kutta method of order 4. The resulting nonlinear optimization problem is nonsmooth and non-convex. Despite the nonsmoothness, the problem is solved by a sequential quadratic programming (SQP) method [16].

SQP methods are iterative methods. At each iteration, a quadratic programming (QP) sub-problem is being solved to find a search iteration. The objective function of (QP) is a local quadratic approximation of the Lagrange function. Since the Hessian matrix of the Lagrange function is not well defined at the points of non-differentiability of d , we use BFGS update formulas [3, 26] to replace it. The nonsmooth constraint with d is being linearized by using an approximate subgradient of d , which we obtain numerically by finite differences.

A trajectory between v_t and v_h is issued from the solution of (QP) by integrating the equations of motion. Therefore, a sequence of trajectories is associated to the SQP method. Without a good initial trajectory between v_t and v_h , even the SQP methods that are augmented by a globalization strategy (see [12, 29]) might not converge. However, for our particular optimal control problem, a good initial trajectory can be found. To this end, a two-step strategy has been developed. First, a path-planning algorithm is used to find intermediate points. These points indicate the direction for the robot to reach v_h . Then, an optimal control problem that does not take into account the obstacles, is defined to find the fastest trajectory that passes through the vicinity of the intermediate points. The resulting solution is the initial trajectory. A detailed description of the computation of the initial trajectory can be found in [21].

Another strategy to compute a good initial trajectory is given in [4]. However, the initial trajectory in this paper must be collision-free to be integrated in the algorithm developed by the authors to solve the optimal control problem. In our solver, the initial trajectory may collide with the static environment.

5 Detection of Conflicting Trajectories

This section concerns the detection of conflicts between two given trajectories which is performed in line 10 of Algorithm 1. Let r_1 and r_2 be two robots in R . For simplicity, let us assume in this section that each robot is a convex compact polyhedron. Let robot r_1 (resp. r_2) move between the locations v_i^1 and v_h^1 (resp. v_i^2 and v_h^2) during the time interval I_1 (resp. I_2). Finally, let a_1 (resp. a_2) be the arc whose tail node is v_i^1 (resp. v_i^2) and head node is v_h^1 (resp. v_h^2). The element $(r_1, a_1, I_1, r_2, a_2, I_2)$ belongs to the set \mathcal{C} of conflicting trajectories if and only if at least one collision is detected between the robots that move along their trajectory during their respective time interval. Our aim in this section is then to detect such collisions.

There exist two types of detection method. The *static* detection checks if there is a collision between two objects at each time step. The *dynamic* checking determines if for all configurations given on a continuous path a collision occurs between the objects. As Cameron pointed out in [7], the static detection is simple, but can miss a collision if the time discretization is too rough. On the other side, taking small time steps is time consuming. For this reason, we choose to use a dynamic method. We follow the method developed by Schwarzer et al. in [30]. This method is based on the comparison of lower bounds of the distance between the robots with an upper bound of the relative distance travelled by the points in the robots. The advantages of this method are its simplicity, its exactness and the automatic adaptation of the sampling resolution.

Let us recall that the motion of a robot from the tail to the head node of a given arc a was computed in (OCP). For this computation, we have considered the time interval $[0, \tau_f]$. To solve (OCP), a time discretization $(\tau_i)_{i=1}^N$ is used, with $\tau_0 = 0$, $\tau_N = \tau_f$ and N the number of time steps. In the scheduled tours resulting from (WCP), a time interval is associated with each arc: $I_a = [\underline{t}_a, \bar{t}_a)$. Hence, to detect a conflict between the arcs a_1 and a_2 , we need first to transform the time discretization used in (OCP) on the time interval I_1 and I_2 . The new time discretization for an arc a on I_a is defined as follows:

$$(\tau_i)_{i=1}^N \text{ with } t_i = \tau_i + \underline{t}_a, \quad i = 1, \dots, N.$$

It follows: $t_1 = \underline{t}_a$ and $t_N = \tau_N + \underline{t}_a = \tau_f + \underline{t}_a$.

Each robot has its own time discretization. Let $(t_i^1)_{i=1}^{N_1}$, resp. $(t_i^2)_{i=1}^{N_2}$, be the time discretization of robot r_1 , resp. r_2 , after transformation on the time interval I_1 , resp. I_2 . Let us classify the time steps $(t_i^1)_{i=1}^{N_1}$ and $(t_i^2)_{i=1}^{N_2}$ in an ascending order as illustrated in Fig. 3. The time is then decomposed on subintervals $[t_l, t_u]$ of the form: $[t_i^1, t_{i+1}^1]$, $[t_i^1, t_{j+1}^2]$, $[t_j^2, t_{i+1}^1]$ or $[t_j^2, t_{j+1}^2]$. We check on each such subinterval if a collision between the robots occurs.

Let us consider the time subinterval $[t_l, t_u]$. The idea of Schwarzer, Saha and Latombe is to compare upper bounds of the distance travelled by the robots

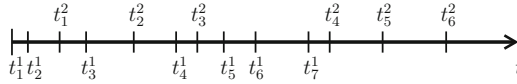


Fig. 3 Ascending order of the time steps $(t_i^1)_{i=1}^{N_1}$ and $(t_i^2)_{i=1}^{N_2}$. The case $t_1^2 > t_1^1$ means that robot r_2 stays at its initial position v_i^2 while r_1 is moving. Likewise, $t_7^1 < t_4^2$ indicates that robot r_1 has reached its final destination v_j^1 , whereas robot r_2 is still moving

during $[t_l, t_u]$ with a lower bound of the distance between both robots. Let us define the following quantities

- $\eta(t)$ is a non-trivial lower bound of the Hausdorff distance between the robots at time t . The relation $\eta(t) \leq \delta$, δ small and positive, means that the robots are colliding.
- $\lambda_i(t_a, t_b)$ is an upper bound on the length of the curves traced by all points in robot r_i , $i = 1, 2$, between t_a and t_b with $t_a, t_b \in [t_l, t_u]$.

Schwarzer, Saha and Latombe’s method is based on the following sufficient condition:

Two polyhedra r_1 and r_2 do not collide at any time $t \in [t_l, t_u]$ if

$$\lambda_1(t_l, t_u) + \lambda_2(t_l, t_u) < \eta(t_l) + \eta(t_u) . \tag{5}$$

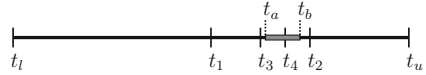
The reverse of the above condition is not true. We cannot say anything about the collision-freeness on $[t_l, t_u]$ when the inequality is not satisfied. In that case, the idea is to bisect the time interval into two subintervals $[t_l, t_m]$ and $[t_m, t_u]$ where $t_m = \frac{1}{2}(t_l + t_u)$. In the second step, we check if a collision occurs at t_m by computing $\eta(t_m)$. If $\eta(t_m)$ is positive, then the sufficient condition is applied on both subintervals $[t_l, t_m]$ and $[t_m, t_u]$. The collision is detected once the lower bound η is smaller than a given threshold $\delta > 0$.

One strength of this detection method is that the algorithm automatically decides whether a time interval needs to be bisected further. Moreover, the method can never fail. To prove this fact, let us observe first that $\lambda_i(t_a, t_b) \rightarrow 0$ when $|t_b - t_a| \rightarrow 0$, $i = 1, 2$. Then, let us distinguish the cases:

- If no collision occurs in the interval $[t_l, t_u]$, then there exists $\eta_{min} \geq \delta > 0$ such that $\eta(t) > \eta_{min}, \forall t \in [t_l, t_u]$. By bisecting, the length of the new time subintervals is always smaller. So, the left-hand side of Inequality (5) becomes smaller with the bisection whereas the right-hand side remains lower-bounded. Hence, Inequality (5) is satisfied.
- If the polyhedra collide, then there is a time subinterval $[t_a, t_b] \subseteq [t_l, t_u]$ such that $\eta(t) \leq \delta, \forall t \in [t_a, t_b]$ since the motion of the polyhedra is continuous. Then, by bisecting, Inequality (5) remains unsatisfied until the new middle of the time interval falls into $[t_a, t_b]$.

Let us illustrate this argument with the example in Fig. 4. The time interval $[t_l, t_u]$ is represented. The time interval $[t_a, t_b]$ when the collision occurs is in grey. The algorithm checks first if the polyhedra collide at t_l and t_u . Second step of the algorithm establishes that Inequality (5) is not satisfied. The first

Fig. 4 Convergence of the collision detection algorithm when a collision occurs



bisection is executed by computing $t_1 = \frac{1}{2}(t_l + t_u)$. No collision occurs at t_1 ($\eta(t_1) > \delta$). Inequality (5) is satisfied on $[t_l, t_1]$ but not on $[t_1, t_u]$. Hence, the middle point of $[t_l, t_u]$ is: $t_2 = \frac{1}{2}(t_1 + t_u)$. At that time, η is greater than δ . The bisection is then executed and we obtain the following subintervals $[t_1, t_2]$ and $[t_2, t_u]$. Inequality (5) is verified on $[t_2, t_u]$ but not on $[t_1, t_2]$. Next, let us compute $t_3 = \frac{1}{2}(t_1 + t_2)$ and check if $\eta(t_3)$ is bigger than δ . Let us do so until we reach $t_4 = \frac{1}{2}(t_3 + t_2)$. For that point, the value of $\eta(t_4)$ is smaller than δ . The collision is detected and the element $(r_1, a_1, I_1, r_2, a_2, I_2)$ is reported as in conflict.

Schwarzer et al. establish in [30] an upper bound λ_1, λ_2 for any kind of robots. The function η is defined as a non-trivial lower bound of the real distance between two polyhedra. A two-phase approach is considered which consists of a *broad phase* and a *narrow phase*. In the broad phase, the polyhedra are approximated by a simple bounding volume such as an axis-aligned box or a sphere and η is defined as the distance between the bounding volumes. As long as the bounding volumes are disjoint, the broad phase is applied. Once the bounding volumes overlap, the narrow phase is used. This phase computes the exact distance between the polyhedra. Thus the two-phase approach allows a minimal cost in the computation of η since the exact distance is determined only when the polyhedra are close to each other. Note that if the robots would have a more complex geometry, then a hierarchy of bounding volumes would be defined such as in [9, 17].

For the narrow phase, we use Lin and Canny's algorithm [24, 25]. This algorithm determines the closest pair of features between the polyhedra, where the features of a polyhedron are its vertices, its edges and its faces located on its boundary. We choose to follow Lin and Canny's algorithm since the approach is fast, easy to implement and perfectly suited when polyhedra move slightly between two time steps.

6 Numerical Experiments

In the following, we present some of the 2D-instances that have been solved using our implementation. Here, the polygon shaped robots move on the 2D-plane by avoiding the obstacles. The trajectories sketched in the following only represent the trace of the center of gravity. For clarity reasons, the rotation along the trajectories is not shown. Also, the actual timing of the robots is not presented explicitly, but it has been taken into account for the calculations.

The shape of a robot is visualized by its starting position, while the obstacles are represented by black polygons. All the weld points are assumed to have a processing time of 0 s. Since the robots are a two-dimensional polyhedron, the meaning of the

state and control variables differs from the one given in Sect. 4. The state variables are here the angle of rotation θ of the robot as well as the position r and the velocity \dot{r} of the center of gravity of the robot. The control variables are the velocity μ of the angle of rotation and the acceleration a of the center of gravity of the robot. As in (OCP), the equations of motion are ordinary differential equations, which are here given by:

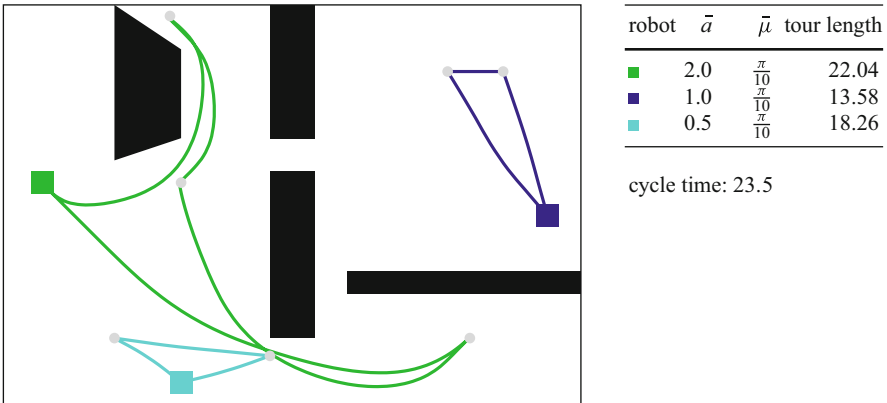
$$\frac{d}{dt}r(t) = \dot{r}(t), \quad \frac{d}{dt}\theta(t) = \mu(t), \quad \frac{d}{dt}\dot{r}(t) = a(t), \quad \text{a.e. } t \in [0, \tau_f].$$

For our numerical experiments, the lower and upper bounds of the control variables, which are:

- for the acceleration of the center of gravity of the robot: $\bar{a} = -\underline{a}$,
- for the velocity of the angle of rotation: $\bar{\mu} = -\underline{\mu}$,

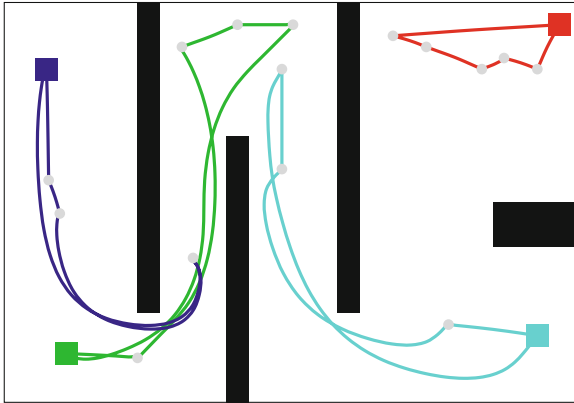
differ for the robots and the instances. Thus, their value are explicitly given. A higher acceleration \bar{a} corresponds to a faster robot $r \in R$.

The process of our approach is showcased for the following examples. To visualize the advantages of our approach the number of total arcs compared to the number of arcs for which the exact travel time has been calculated is given as well as the computation times for the continuous trajectory calculations and the entire process.



#arcs	#exact trajectories	cont. time	total time
168	34	27.1 s	29.5 s

Instance 1: Three robots, where especially the green and the much slower cyan robot need to be coordinated. The number of required updates could be reduced by a factor of five.



robot	\bar{a}	$\bar{\mu}$	tour length
■	1.0	$\frac{\pi}{10}$	26.59
■	1.0	$\frac{\pi}{10}$	26.28
■	1.0	$\frac{\pi}{10}$	25.95
■	0.5	$\frac{\pi}{10}$	26.52

cycle time: 26.7

#arcs	#exact trajectories	cont. time	total time
788	62	56.9 s	66.0 s

Instance 2: This example contains four robots. Due to a tight cycle time, three robots have to operate very close to each other. Less than 10 % of the connecting trajectories were calculated.

7 Conclusion

In this paper, the automatic reconfiguration of a welding cell using state-of-the-art mathematical techniques has been presented. The Welding Cell Problem involves calculating feasible robot movements in a workcell composed of several welding robots and tasks. A solution is feasible when the tasks are assigned between the robots and the motion between the tasks is planned in such a way that it is finished within the cycle time of the production process and is collision-free. For that purpose, techniques from discrete optimization were efficiently combined with effective algorithms to solve optimal control problems and to detect collisions.

Even if we have presented numerical results in two dimensions, the formulation and the resolution of the WCP are independent of the dimension of the workcell. The quality of our two-dimensional results and the efficiency of our method are very promising for a real three dimensional welding cell. The time to reconfigure a cell should be in this case a mere fraction of the manual configuration.

Acknowledgements This work has been supported by the DFG Research Center MATHEON—Mathematics for key technologies.

References

1. Achterberg, T.: Constraint integer programming. Ph.D. thesis, Technische Universität Berlin (2007)
2. Achterberg, T.: SCIP: Solving constraint integer programs. *Math. Program. Comput.* **1**, 1–41 (2009)
3. Betts, J.T.: Practical methods for optimal control using nonlinear programming. In: *Advances in Design and Control*, vol. 3. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2001)
4. Björkenstam, S., Gleeson, D., Bohlin, R., Carlson, J.S., Lennartson, B.: Energy efficient and collision free motion of industrial robots using optimal control. In: *IEEE*, pp. 510–515 (2013)
5. Björkenstam, S., Spensieri, D., Carlson, J.S., Bohlin, R., Gleeson, D.: Efficient sequencing of industrial robots through optimal control. *Procedia CIRP* **23**, 194–199 (2014). 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems
6. Bobrow, J.E.: Optimal robot path planning using the minimum-time criterion. *IEEE J. Rob. Autom.* **4**(4), 443–450 (1988)
7. Cameron, S.: A study of the clash detection problem in robotics. In: *International Conference on Robotics and Automation*, pp. 488–493 (1985)
8. Cameron, S.A., Culley, R.K.: Determining the minimum translational distance between two convex polyhedra. In: *Proceedings of International Conference on Robotics and Automation*, pp. 591–596 (1986)
9. Cohen, J.D., Lin, M.C., Manocha, D., Ponamgi, M.K.: I-collide: An interactive and exact collision detection system for large-scaled environments. In: *Symposium on Interactive 3D Graphics*, pp. 189–196 (1995)
10. Desrosiers, J., Lübbecke, M.: A primer in column generation. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.) *Column Generation*, pp. 1–32. Springer, New York (2005)
11. Donald, B., Xavier, P., Canny, J., Reif, J.: Kinodynamic motion planning. *J. ACM* **40**, 1048–1066 (1993)
12. Fletcher, R., Leyffer, S., Toint, P.: On the global convergence of a filter–SQP algorithm. *SIAM J. Optim.* **13**(1), 44–59 (2002)
13. Gerds, M.: Optimal control of ODEs and DAEs. In: *De Gruyter Textbook*. De Gruyter, Berlin (2012)
14. Gerds, M., Henrion, R., Hömberg, D., Landry, C.: Path planning and collision avoidance for robots. *Numer. Algebra Control Optim.* **2**, 437–463 (2012)
15. Gilbert, E.G., Johnson, D.W.: Distance functions and their application to robot path planning in the presence of obstacles. *IEEE J. Robot. Autom.* **RA-1**(1), 21–30 (1985)
16. Gill, P., Murray, W., Saunders, M.: SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Rev.* **47**, 99–131 (2005)
17. Gottschalk, S., Lin, M.C., Manocha, D.: Obbtree: a hierarchical structure for rapid interference detection. In: *SIGGRAPH*, A. (ed.) *Computer Graphics Proceedings, Annual Conference Series* (1996)
18. Hart, G.D., Anitescu, M.: An $O(m+n)$ measure of penetration depth between convex polyhedral bodies for rigid multibody dynamics (2010)
19. Kim, Y.J., Lin, M.C., Manocha, D.: DEEP: dual-space expansion for estimating penetration depth between convex polytopes. In: *IEEE Conference on Robotics and Automation*, pp. 921–926 (2002)
20. Landry, C., Henrion, R., Hömberg, D., Skutella, M., Welz, W.A.: Task assignment, sequencing and path-planning in robotic welding cells. In: *Proceedings of the 18th International Conference on Methods and Models in Automation and Robotics (MMAR '13)*, pp. 252–257 (2013)
21. Landry, C., Welz, W.A., Gerds, M.: Combining discrete and continuous optimization to solve kinodynamic motion planning problems. *Optim. Eng.* **17**(3), 533–556 (2016)

22. Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **59**(3), 345–358 (1992)
23. LaValle, S.M.: *Planning Algorithms*. Cambridge University Press, Cambridge (2006)
24. Lin, M.C.: Efficient collision detection for animation and robotics. Ph.D. thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley (1993)
25. Lin, M.C., Canny, J.F.: A fast algorithm for incremental distance calculation. In: *Proceedings 1991 IEEE International Conference on Robotics and Automation*, p. 1008 (1991)
26. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2006)
27. Rambau, J., Schwarz, C.: Solving a vehicle routing problem with resource conflicts and makespan objective with an application in car body manufacturing. *Optim. Methods Softw.* **29**, 353–375 (2014)
28. Saha, M., Sánchez-Ante, G., Latombe, J.C.: Planning multi-goal tours for robot arms. In: *IEEE*, pp. 3797–3803 (2003)
29. Schittkowski, K.: On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function 2. *Math. Oper. Stat. Ser. Optim.* **14**(2), 197–216 (1983)
30. Schwarzer, F., Saha, M., Latombe, J.: Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. *IEEE Trans. Robot.* **21**, 338–353 (2005)
31. Segeborn, J., Segerdahl, D., Ekstedt, F., Carlson, J.S., Carlsson, A., Söderberg, R.: A generalized method for weld load balancing in multi station sheet metal assembly lines. In: *International Mechanical Engineering Congress and Exposition*, pp. 491–499 (2011)
32. Segeborn, J., Segerdahl, D., Ekstedt, F., Carlson, J.S., Andersson, M., Carlsson, A., Söderberg, R.: An industrially validated method for weld load balancing in multi station sheet metal assembly lines. *J. Manuf. Sci. Eng.* **136**, 011002 (2013)
33. Skutella, M., Welz, W.A.: Route planning for robot systems. In: Hu, B., Morasch, K., Pickl, S., Siegle, M. (eds.) *Operations Research Proceedings 2010*, pp. 307–312. Springer, New York (2011)
34. Spensieri, D., Carlson, J.S., Ekstedt, F., Bohlin, R.: An iterative approach for collision free routing and scheduling in multirobot stations. In: *IEEE Transactions on Automation science and Engineering*, pp. 1–13 (2015)
35. Steinbach, M.C., Bock, H.G., Kostin, G.V., Longman, R.W.: Mathematical optimization in robotics: towards automated high speed motion planning. *Math. Ind* **7**, 303–340 (1997)
36. von Stryk, O., Schlemmer, M.: Optimal control of the industrial robot manutec r3. In: Bulirsch, R., Kraft, D. (eds.) *Computational Optimal Control*. ISNM International Series of Numerical Mathematics, vol. 115, pp. 367–382. Birkhäuser, Basel (1994)
37. Welz, W.A.: Robot tour planning with high determination costs – routing under uncertainty. Ph.D. thesis, Technische Universität Berlin (2014)

Numerical Approaches Towards Bilevel Optimal Control Problems with Scheduling Tasks

Konstantin D. Palagachev and Matthias Gerdts

Abstract In this paper, we consider the problem of scheduling N robots interacting with a moving target. Both, the sequence of the robots and their trajectories are unknown and subject to optimization. Such type of problems appear in highly automated production plants and in the simulation of virtual factories. The purpose of the paper is to provide a mathematical model and to suggest a numerical solution approach. Our approach is based on the formulation of the problem as a bilevel optimization problem, where the lower level problem is an optimal control problem, while the upper level problem is a finite dimensional mixed-integer optimization problem. We approach the problem by exploitation of necessary optimality conditions for the lower level problem and by application of a Branch & Bound method for the resulting single level optimization problem. Two settings are taken into account. Firstly, no state constraints are assumed on the lower level problem, thus the local minimum principle applies directly. Secondly, the problem setting is augmented by pure state constraints, which are being handled by virtual controls in order to regularize the problem.

1 Introduction and Formulation of Bilevel Scheduling Optimal Control Problem

Many modern manufacturing processes are automatized to a large degree using robots for, e.g., welding, painting, or transporting parts autonomously. In the future, the interaction of robots with other robots or human workers will become more and more important. In addition, the flexibility with respect to the tasks a robot has to perform will increase as well. For instance, a freely moving autonomous transportation robot in an automated factory will have to transport some load from one position, e.g. in a storehouse or at a production site, to another remote position in a storehouse or a different production site. This task requires a path planning

K.D. Palagachev • M. Gerdts (✉)

Institut für Mathematik und Rechneranwendung, Fakultät für Luft- und Raumfahrttechnik,
Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, München,
Germany

e-mail: konstantin.palagachev@unibw.de; matthias.gerdts@unibw.de

algorithm with the capability of avoiding collisions with other robots or objects. In addition, the transportation process may have several phases, for instance a transportation phase and an interaction phase with a human or another robot. This task can be modeled by a multi-phase optimal control problem. However, there is a second level since many of such transportation robots have to be coordinated in such a way, that the robots arrive just in time at the requested positions. This task leads to a scheduling problem, which determines the optimal starting times for the individual robots such that the overall time becomes minimal. The overall time on the other hand depends on the travel times of the robots. Hence, a coupled optimization problem with a scheduling problem at the upper level and an optimal control problem at the lower level has to be solved. Its solution helps to optimize the processes and to analyze and simulate processes during the virtual factory design of a new plant. Similar type of problems occur in automated ports of debarkation, but also in civil or military observation tasks, where the observation of a moving or fixed object has to be coordinated between several observers.

Let us consider the problem of scheduling N robots performing certain tasks while they interact with a target robot which moves on a fixed trajectory. The objective is to minimize the overall duration of the problem, guaranteeing that each robot interacts with the target. This setting is a particular example for a bilevel scheduling optimal control problem. Scheduling of the different robots can be formulated as a job shop problem with the task to find the optimal starting times t_i , $i = 1, \dots, N$, of the N robots. For given starting times t_i , $i = 1, \dots, N$, optimal control problems have to be solved in order to find time optimal trajectories of the robots and the durations of the robot actions. For simplicity we assume that the interaction between the robots and the target robot is instantaneous. To this end only an approach phase and a return phase have to be considered for each robot-target interaction. Thus the optimal control problems consist of two phases with durations $p_1^{(i)}$ and $p_2^{(i)}$, respectively, for $i = 1, \dots, N$. Hence, the i -th robot starts at time instance t_i , it interacts with the moving target robot at time instance $t_i + p_1^{(i)}$, and it returns to its initial position at time instance $t_i + p_1^{(i)} + p_2^{(i)}$. Please note that the durations $p_1^{(i)}$ and $p_2^{(i)}$ depend implicitly on the starting time t_i , since the distance between each robot and the target robot changes due to the target's motion.

In order to schedule the robots, we introduce the binary variables $w_{ij} \in \{0, 1\}$, with $i, j \in \{1, \dots, N\}$, $i < j$, where $w_{ij} = 1$ means that the i -th robot runs before the j -th one and $w_{ij} = 0$ vice versa. A natural condition to avoid collisions between robots is to impose that for each $i = 1 \dots, N - 1$, the i -th scheduled robot interacts with the moving target before the starting time of the next scheduled one.

Remark 1 Note that this condition is conservative, since it guarantees a collision avoidance only in a neighbourhood of the target and when the target is not too close to the starting position of the robots. Nevertheless, it prevents the problem of growing to much in dimensionality and complexity. However, using the techniques in Sect. 6, it is essentially possible to add further state constraints to avoid collision between the robots at all time. The computational effort would increase considerably, though.

As a result, the problem of scheduling the N robots can be formulated in the following way

Problem 1 (Upper Level Problem)

$$\text{Minimize} \quad \max_{1 \leq i \leq N} \{t_i + p_1^{(i)} + p_2^{(i)}\} \tag{1}$$

with respect to $t_i, p_1^{(i)}, p_2^{(i)} \in \mathbb{R}, w_{ij} \in \{0, 1\}$, subject to

$$t_i + p_1^{(i)} - t_j \leq (1 - w_{ij})M \tag{2}$$

$$t_j + p_1^{(j)} - t_i \leq w_{ij}M \tag{3}$$

$$t_i \geq 0 \tag{4}$$

where $i, j \in \{1, \dots, N\}$ with $i < j$.

In constraints (2)–(3) the constant M is supposed to be sufficiently large in order to ensure that just one of the constraints becomes active for a given w_{ij} . A typical choice is to choose M greater than the sum of all starting times t_i and all phase lengths $p_1^{(i)}$ and $p_2^{(i)}$. Note that $w_{ij} = 1$ implies that $t_j \geq t_i + p_1^{(i)}$ by (2), i.e. robot j starts after the interaction of robot i with the target robot. Likewise, $w_{ij} = 0$ implies $t_i \geq t_j + p_1^{(j)}$ by constraint (3).

The durations $p_1^{(i)}$ and $p_2^{(i)}$ for $i = 1, \dots, N$ are determined by suitably defined optimal control problems. In order to keep the problem simple, we assume that the dynamics of each robot are given by a system of linear ordinary differential equations. We denote by $x_0^{(i)} \in \mathbb{R}^{n_x}$ the starting position of the i -th robot and by $x_T \in W^{1,\infty}([0, +\infty), \mathbb{R}^{n_x})$ the function providing target robot’s position at time t . In this way, for each robot i the following parametric optimal control problem with input t_i has to be solved in order to obtain the durations $p_1^{(i)}$ and $p_2^{(i)}$:

Problem 2 (Lower Level Problem $OCP(t_i)$) Minimize

$$p_1^{(i)} + p_2^{(i)} + \frac{c}{2} \int_{t_i}^{t_i + p_1^{(i)} + p_2^{(i)}} \|u^{(i)}(t)\|^2 dt \tag{5}$$

with respect to $x^{(i)} \in W^{1,\infty}([t_i, t_i + p_1^{(i)} + p_2^{(i)}], \mathbb{R}^{n_x}), u^{(i)} \in L^\infty([t_i, t_i + p_1^{(i)} + p_2^{(i)}], \mathbb{R}^{n_u})$ and $p_1^{(i)}, p_2^{(i)} \in \mathbb{R}$ and subject to

$$\dot{x}^{(i)}(t) = A^{(i)}x^{(i)}(t) + B^{(i)}u^{(i)}(t) \quad a.e.in (t_i, t_i + p_1^{(i)} + p_2^{(i)}) \tag{6}$$

$$x^{(i)}(t_i) = x_0^{(i)} \tag{7}$$

$$x^{(i)}(t_i + p_1^{(i)}) = x_T(t_i + p_1^{(i)}) \tag{8}$$

$$x^{(i)}(t_i + p_1^{(i)} + p_2^{(i)}) = x_0^{(i)}. \tag{9}$$

Note that in (5), we have introduced a quadratic term penalizing the control effort with some scaling factor $c \geq 0$, while in (6) we have $A^{(i)} \in \mathbb{R}^{n_x \times n_x}$ and $B^{(i)} \in \mathbb{R}^{n_x \times n_u}$. As usual, the Banach space $L^\infty([t_0, t_f], \mathbb{R}^n)$ consists of all measurable functions $h : [t_0, t_f] \rightarrow \mathbb{R}^n$ with

$$\|h\|_\infty := \operatorname{ess\,sup}_{t_0 \leq t \leq t_f} \|h(t)\| < \infty,$$

where $\|\cdot\|$ denotes the Euclidean norm in \mathbb{R}^n . For $1 \leq r < \infty$ the Banach space $L^r([t_0, t_f], \mathbb{R}^n)$ consists of all measurable functions $h : [t_0, t_f] \rightarrow \mathbb{R}^n$ with

$$\|h\|_r := \left(\int_{t_0}^{t_f} \|h(t)\|^r \right)^{1/r} < \infty.$$

For $1 \leq r \leq \infty$ the Banach space $W^{1,r}([t_0, t_f], \mathbb{R}^n)$ consists of all absolutely continuous functions $h : [t_0, t_f] \rightarrow \mathbb{R}^n$ with $\|h\|_{1,r} := \max \{ \|h\|_r, \|\dot{h}\|_r \} < \infty$.

We emphasize that, due to the structure of the problem, an optimal scheduling sequence is only obtained, when each robot is moving along its optimal trajectory, given by the solution of Problem 2. It follows that the lengths of the phases for each robot, appearing as parameters in the Problem 1, are optimization variables in Problem 2. This in fact is a bilevel scheduling optimal control problem (BLSOCP), which can be stated in compact notion as follows:

$$\text{Minimize} \quad \max_{1 \leq i \leq N} \{t_i + p_1^{(i)} + p_2^{(i)}\}$$

with respect to $t_i, p_1^{(i)}, p_2^{(i)} \in \mathbb{R}, w_{ij} \in \{0, 1\}$ subject to (2)–(4) and

$$(p_1^{(i)}, p_2^{(i)}) \in \operatorname{argmin} \left\{ p_1^{(i)} + p_2^{(i)} + \frac{c}{2} \int_{t_i}^{t_i + p_1^{(i)} + p_2^{(i)}} \|u^{(i)}(t)\|^2 dt \mid \begin{array}{l} \text{constraints} \\ (6) - (9) \text{ hold} \end{array} \right\}$$

for every $i, j \in \{1, \dots, N\}$ with $i < j$.

From a historical point of view, this type of problems is closely related to the economical problem of Stackelberg [24] in the field of game theory. Further applications can be found in resources and weapons allocation [5, 6], network design problems [21, 23] and engineering problems. The latter are often modeled as mathematical programs with equilibrium constraints, see, e.g., [3, 10, 11]. For a comprehensive treatment of bilevel optimization problems, please refer to [8]. Necessary conditions for a class of bilevel optimal control problems are considered in [25, 26].

There are basically two main techniques for solving bilevel optimization problems. The first one keeps the bilevel structure and treats the lower level problem as a parametric optimization problem (or a black-box), which is being solved by suitable gradient-type methods whenever the solution algorithm for the upper level problem

requires it, see [17, Chapter 7] or [9]. Since the solution of the lower level problem in general depends in a non-smooth (or even discontinuous) way on the variables of the upper level problem, gradient-free optimization methods or bundle methods should be used in this approach.

The second technique, which is frequently used in practice, is based on the formulation of first order necessary optimality conditions for the lower level problem. The lower level problem is then replaced by its necessary conditions, which are considered as constraints in the upper level problem, see, e.g., [1, 18, 19] or [17, Chapter 5]. This reduces the bilevel problem into a single level nonlinear optimization problem. The drawback of this method is that in general necessary conditions are not sufficient for optimality and hence information is lost in the single level formulation and non-optimal solutions for the bilevel optimization may be obtained. The approach can be considered as an optimistic approach since the upper level player can choose among the possible stationary points of the lower level player the most suitable one.

In this paper, despite its potential drawback, we follow the second approach by exploiting the local minimum principle from optimal control for Problem 2. Note that the resulting single level problem is a mixed-integer nonlinear optimization problem. Finding a global solution is a challenging task. Floudas and Gümüş [16] proposed a global optimization approach based on the deterministic global optimization algorithm αBB , see [2] for details.

The paper is organized as follows. Section 2 exploits the minimum principle for the lower level problem and provides a single level reformulation of our bilevel scheduling optimal control problem. To this end no state or control constraints are permitted in the lower level problem. In Sect. 3, we recall the Branch and Bound algorithm, while in Sect. 4 we apply it to the discretized bilevel scheduling optimal control problem. Sections 5 and 6 deal with numerical results of the state constrained case.

2 Minimum Principle and Reduction to Single Level Problem

In this section, we recall a local minimum principle from [13, Chapter 3] for nonlinear optimal control problems and eventually apply it to Problem 2. To this end, the bilevel scheduling optimal control problem BLSOCP will be reduced to a single level mixed-integer nonlinear optimization problem.

Let $[t_0, t_f] \subset \mathbb{R}$ be a non-empty compact interval with $t_0 < t_f$ fixed. Let

$$\begin{aligned} f_0 &: [t_0, t_f] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R} & f &: [t_0, t_f] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x} \\ g &: [t_0, t_f] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_g} & \psi &: \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_\psi} \end{aligned}$$

be continuously differentiable functions and let g be twice continuously differentiable with respect to all arguments. Consider the following generic optimal control problem:

Problem 3

$$\text{Minimize} \quad \int_{t_0}^{t_f} f_0(t, x(t), u(t)) dt$$

with respect to $x \in W^{1,\infty}([t_0, t_f], \mathbb{R}^{n_x})$, $u \in L^\infty([t_0, t_f], \mathbb{R}^{n_u})$, subject to the constraints

$$\dot{x}(t) = f(t, x(t), u(t)) \quad \text{a.e.in } (t_0, t_f) \quad (10)$$

$$g(t, x(t), u(t)) \leq 0_{\mathbb{R}^{n_g}} \quad \text{a.e.in } (t_0, t_f) \quad (11)$$

$$\psi(x(t_0), x(t_f)) = 0_{\mathbb{R}^{n_\psi}}. \quad (12)$$

For Problem 3, the **augmented Hamilton function** $\hat{H} : [t_0, t_f] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_g} \times \mathbb{R} \rightarrow \mathbb{R}$ is defined by

$$\hat{H}(t, x, u, \lambda, \mu, \lambda_0) := \lambda_0 f_0(t, x, u) + \lambda^\top f(t, x, u) + \mu^\top g(t, x, u). \quad (13)$$

In a local minimum (x^*, u^*) of Problem 3, for notational convenience, we will use the abbreviations $f[t] := f(t, x^*(t), u^*(t))$, $f'_x[t] := f'_x(t, x^*(t), u^*(t))$, $\psi'_{x_0} := \psi'_{x_0}(x^*(t_0), x^*(t_f))$ and in a similar way $f'_u[t]$, $g[t]$, $g'_x[t]$, $g'_u[t]$, and ψ'_{x_f} .

Theorem 1 (see [13, Theorem 3.3.8]) *Let the following assumptions hold for Problem 3:*

- (i) (x^*, u^*) is a local minimum of Problem 3.
- (ii) $\text{rank}(g'_u[t]) = n_g$ almost everywhere in (t_0, t_f) and the pseudo-inverse of $g'_u[t]$,

$$g'_u[t]^+ = g'_u[t]^\top (g'_u[t]g'_u[t]^\top)^{-1},$$

is essentially bounded in (t_0, t_f) .

Then there exist multipliers

$$\lambda_0 \in \mathbb{R}, \quad \lambda \in W^{1,\infty}([t_0, t_f], \mathbb{R}^{n_x}), \quad \mu \in L^\infty([t_0, t_f], \mathbb{R}^{n_g}), \quad \sigma \in \mathbb{R}^{n_\psi}$$

such that the following conditions hold:

- (a) $\lambda_0 \geq 0$ and $(\lambda_0, \lambda, \mu, \sigma) \neq 0$
- (b) **Adjoint equation:** Almost everywhere in (t_0, t_f) we have

$$\dot{\lambda}(t) = -\nabla_x \hat{H}(t, x^*(t), u^*(t), \lambda(t), \mu(t), \lambda_0)$$

(c) **Transversality conditions:**

$$\lambda(t_0)^\top = -\sigma^\top \psi'_{x_0}, \quad \lambda(t_f)^\top = \sigma^\top \psi'_{x_f}$$

(d) **Stationarity of the Hamilton function:** Almost everywhere in (t_0, t_f) we have

$$\nabla_u \hat{H}(t, x^*(t), u^*(t), \lambda(t), \mu(t), \lambda_0) = 0_{\mathbb{R}^{n_u}}$$

(e) **Complementarity condition:** For every $i = 1, \dots, n_g$ and almost everywhere in (t_0, t_f) we have

$$\mu_i(t) \geq 0_{\mathbb{R}^{n_g}} \quad \text{and} \quad \mu(t)_i g_i[t] = 0.$$

Our goal now is to reduce the bilevel optimal control problem BLSOCP into a single level problem, by exploiting Theorem 1. We first observe that for each $i \in \{1, \dots, N\}$ the final time in $OCP(t_i)$ is free and subject to optimization, furthermore intermediate state constraints are present at time $t_i + p_1^{(i)}$ (i.e. equation (8)). We can transform the problem into a fixed-time optimal control problem, introducing the transformations $t_1^{(i)} : [0, 1] \rightarrow [t_i, t_i + p_1^{(i)}]$ and $t_2^{(i)} : [0, 1] \rightarrow [t_i + p_1^{(i)}, t_i + p_1^{(i)} + p_2^{(i)}]$, defined as

$$t_1^{(i)}(\tau) := t_i + \tau p_1^{(i)} \quad \text{and} \quad t_2^{(i)}(\tau) := t_i + p_1^{(i)} + \tau p_2^{(i)} \quad \forall \tau \in [0, 1]. \quad (14)$$

By definition $t_1^{(i)}$ and $t_2^{(i)}$ are differentiable, and for every $\tau \in (0, 1)$ we have

$$dt_1^{(i)}(\tau)/d\tau = p_1^{(i)} \quad \text{and} \quad dt_2^{(i)}(\tau)/d\tau = p_2^{(i)}.$$

Let us now define for every $x^{(i)} \in W^{1,\infty}([t_i, t_i + p_1^{(i)} + p_2^{(i)}], \mathbb{R}^{n_x})$ and $u^{(i)} \in L^\infty([t_i, t_i + p_1^{(i)} + p_2^{(i)}], \mathbb{R}^{n_u})$ the functions

$$\begin{aligned} x_1^{(i)}(\tau) &:= x^{(i)}(t_1^{(i)}(\tau)), & x_2^{(i)}(\tau) &:= x^{(i)}(t_2^{(i)}(\tau)), \\ u_1^{(i)}(\tau) &:= u^{(i)}(t_1^{(i)}(\tau)), & u_2^{(i)}(\tau) &:= u^{(i)}(t_2^{(i)}(\tau)). \end{aligned} \quad (15)$$

It is easy to check that $x_1^{(i)}, x_2^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$ and $u_1^{(i)}, u_2^{(i)} \in L^\infty([0, 1], \mathbb{R}^{n_u})$. In this way, we can prove (see Theorem 2 below) that the bilevel optimal control problem BLSOCP can be reduced to the following single level mixed-integer optimal control problem:

Problem 4 Minimize

$$\xi \quad (16)$$

with respect to $t_i, p_1^{(i)}, p_2^{(i)} \in \mathbb{R}$, $w_{ij} \in \{0, 1\}$, $x_1^{(i)}, x_2^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$, $\lambda_{x_1}^{(i)}, \lambda_{x_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$ and $\lambda_{p_1}^{(i)}, \lambda_{p_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R})$ subject to

$$t_i + p_1^{(i)} + p_2^{(i)} \leq \xi \quad (17)$$

$$t_i + p_1^{(i)} - t_j \leq (1 - w_{ij})M, \quad t_j + p_1^{(j)} - t_i \leq w_{ij}M, \quad t_i \geq 0 \quad (18)$$

$$\dot{x}_k^{(i)} = p_k^{(i)} \left[A^{(i)} x_k^{(i)} - \frac{1}{c} B^{(i)} (B^{(i)})^\top \lambda_{x_k}^{(i)} \right] \quad (19)$$

$$\dot{\lambda}_{x_k}^{(i)} = -p_k^{(i)} (A^{(i)})^\top \lambda_{x_k}^{(i)} \quad (20)$$

$$\dot{\lambda}_{p_k}^{(i)} = -1 - (\lambda_{x_k}^{(i)})^\top \left[A^{(i)} x_k^{(i)} - \frac{1}{2c} B^{(i)} (B^{(i)})^\top \lambda_{x_k}^{(i)} \right] \quad (21)$$

$$x_1^{(i)}(0) = x_2^{(i)}(1) = x_0^{(i)}, \quad x_1^{(i)}(1) = x_2^{(i)}(0), \quad x_1^{(i)}(1) = x_T(t_1^{(i)}(1)) \quad (22)$$

$$\lambda_{p_1}^{(i)}(0) = \lambda_{p_2}^{(i)}(0) = \lambda_{p_2}^{(i)}(1) = 0 \quad (23)$$

$$\lambda_{p_1}^{(i)}(1) = -[\lambda_{x_1}^{(i)}(1) - \lambda_{x_2}^{(i)}(0)]^\top \dot{x}_T(t_1^{(i)}(1)) \quad (24)$$

where constraints (17)–(24) hold for almost every $\tau \in (0, 1)$ and every $i, j = 1, \dots, N$ with $i < j$.

Theorem 2 Consider the bilevel optimal control problem BLSOCP with upper level Problem 1 and lower level Problems 2 with $c > 0$ for $i = 1, \dots, N$. Let a constraint qualification hold for Problems 2. Then replacing Problems 2 by their necessary conditions lead to Problem 4.

Remark 2 Please note that Problem 4 is not equivalent to the bilevel problem, since the latter is nonconvex. Moreover, we assume that the multiplier λ_0 (related to the objective function) is not zero. This is justified if a constraint qualification holds, see, e.g., [13, Lemma 3.3.10].

Proof First, we exploit transformations (14) and (15) in Problems 2 and notice that the parameters $p_1^{(i)}$ and $p_2^{(i)}$ can be treated as a state variables (i.e. functions in $W^{1,\infty}([0, 1], \mathbb{R})$) with $\dot{p}_1^{(i)}(\tau) = 0 = \dot{p}_2^{(i)}(\tau)$ for every $\tau \in (0, 1)$. Hence Problems 2 assume the following transformed form (TLLP):

for each $i \in \{1, \dots, N\}$ minimize

$$\int_0^1 p_1^{(i)}(\tau) \left\{ 1 + \frac{c}{2} \|u_1^{(i)}(\tau)\|^2 \right\} + p_2^{(i)}(\tau) \left\{ 1 + \frac{c}{2} \|u_2^{(i)}(\tau)\|^2 \right\} d\tau \quad (25)$$

with respect to $x_1^{(i)}, x_2^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$, $p_1^{(i)}, p_2^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R})$ and $u_1^{(i)}, u_2^{(i)} \in L^\infty([0, 1], \mathbb{R}^{n_u})$ subject to

$$\dot{x}_k^{(i)}(\tau) = p_k^{(i)} [A^{(i)} x_k^{(i)}(\tau) + B^{(i)} u_k^{(i)}(\tau)] \quad a.e.in (0, 1), k = 1, 2 \quad (26)$$

$$\dot{p}_k^{(i)}(\tau) = 0 \quad a.e.in (0, 1), k = 1, 2 \quad (27)$$

$$x_1^{(i)}(0) - x_0^{(i)} = 0_{\mathbb{R}^{n_x}}, \quad x_1^{(i)}(1) - x_2^{(i)}(0) = 0_{\mathbb{R}^{n_x}} \quad (28)$$

$$x_1^{(i)}(1) - x_T(t_i + p_1^{(i)}) = 0_{\mathbb{R}^{n_x}} \quad x_2^{(i)}(1) - x_0^{(i)} = 0_{\mathbb{R}^{n_x}}. \quad (29)$$

For simplicity, we will omit the dependency of the variables from τ . Our goal now is to transform TLLP into a boundary value problem, exploiting Theorem 1. Let $\lambda_{x_1}^{(i)}, \lambda_{x_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$, $\lambda_{p_1}^{(i)}, \lambda_{p_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R})$ and $\sigma \in \mathbb{R}^{4n_x}$ be the multipliers defined in Theorem 1. We observe that the augmented Hamilton function, related to TLLP is defined as:

$$\begin{aligned} \hat{H}^{(i)}[\tau] &:= \hat{H}^{(i)}(\tau, x_1^{(i)}, x_2^{(i)}, p_1^{(i)}, p_2^{(i)}, u_1^{(i)}, u_2^{(i)}, \lambda_{x_1}^{(i)}, \lambda_{x_2}^{(i)}) \\ &= p_1^{(i)} \left\{ 1 + \frac{c}{2} \|u_1^{(i)}\|^2 \right\} + p_2^{(i)} \left\{ 1 + \frac{c}{2} \|u_2^{(i)}\|^2 \right\} \\ &\quad + (\lambda_{x_1}^{(i)})^\top \left\{ p_1^{(i)} [A^{(i)} x_1^{(i)} + B^{(i)} u_1^{(i)}] \right\} + (\lambda_{x_2}^{(i)})^\top \left\{ p_2^{(i)} [A^{(i)} x_2^{(i)} + B^{(i)} u_2^{(i)}] \right\}. \end{aligned}$$

Note that in the formulation of the Hamilton function, we have assumed that $\lambda_0 \neq 0$. In this way, we can omit it, dividing the other multipliers by λ_0 .

Stationarity of the Hamilton function, $\nabla_{u_k} \hat{H}^{(i)} = 0_{\mathbb{R}^{n_u}}$, implies

$$p_k^{(i)} c u_k^{(i)} + p_k^{(i)} (B^{(i)})^\top \lambda_{x_k}^{(i)} = 0_{\mathbb{R}^{n_u}} \quad a.e.in (0, 1), k = 1, 2$$

and thus, assuming $p_k^{(i)} \neq 0$ and $c \neq 0$

$$u_k^{(i)}(\tau) = -\frac{1}{c} (B^{(i)})^\top \lambda_{x_k}^{(i)} \quad a.e.in (0, 1), k = 1, 2. \quad (30)$$

This provides an analytic expression for the controls $u_1^{(i)}, u_2^{(i)}$ involved in TLLP. Note that in (30), we have assumed that $p_k^{(i)} \neq 0$, which makes sense, since $p_k^{(i)} = 0$ corresponds to the degenerate case in which $x_0^{(i)} = x_T(t_i)$ (i.e. i -th robot's starting position is exactly the same as target's position at time t_i). By substituting the expressions of $u_1^{(i)}$ and $u_2^{(i)}$ from (30) in (26), we obtain (19).

Let us now derive the adjoint equations (20). For $k = 1, 2$ we have

$$\begin{aligned}\dot{\lambda}_{x_k}^{(i)}(\tau) &= -\nabla_{x_k} \hat{H}^{(i)}[\tau] = -p_k^{(i)} (A^{(i)})^\top \lambda_{x_k}^{(i)}, \\ \dot{\lambda}_{p_k}^{(i)}(\tau) &= -\nabla_{p_k} \hat{H}^{(i)}[\tau] = -1 - \frac{c}{2} \|u_k^{(i)}\|^2 - (\lambda_{x_k}^{(i)})^\top [A^{(i)} x_k^{(i)} + B^{(i)} u_k^{(i)}] \\ &= -1 - (\lambda_{x_k}^{(i)})^\top [A^{(i)} x_k^{(i)} - \frac{1}{2c} B^{(i)} (B^{(i)})^\top \lambda_{x_k}^{(i)}]\end{aligned}\quad (31)$$

almost everywhere in $(0, 1)$. Note that in (31), we substituted the control $u_k^{(i)}$ with its expression, given by Eq. (30).

Finally, we derive the boundary conditions (23)–(24). Let us define the function $\psi : \mathbb{R}^{2n_x} \times \mathbb{R}^2 \times \mathbb{R}^{2n_x} \times \mathbb{R}^2 \rightarrow \mathbb{R}^{4n_x}$ as

$$\begin{aligned}\psi(x_1^{(i)}(0), x_2^{(i)}(0), p_1^{(i)}(1), p_2^{(i)}(1), x_1^{(i)}(1), x_2^{(i)}(1), p_1^{(i)}(1), p_2^{(i)}(1)) \\ := \begin{bmatrix} x_1^{(i)}(0) - x_0^{(i)} \\ x_1^{(i)}(1) - x_2^{(i)}(0) \\ x_1^{(i)}(1) - x_T(t_i + p_1^{(i)}) \\ x_2^{(i)}(1) - x_0^{(i)} \end{bmatrix}.\end{aligned}$$

By the transversality conditions with $\sigma^{(i)} = (\sigma_1^{(i)}, \sigma_2^{(i)}, \sigma_3^{(i)}, \sigma_4^{(i)})^\top$ where $\sigma_k^{(i)} \in \mathbb{R}^{n_x}$ for $k = 1, \dots, 4$, we have

$$\lambda_{x_1}^{(i)}(0) = -(\sigma^{(i)})^\top \psi'_{x_1^{(i)}(0)} = -\sigma_1^{(i)}, \quad \lambda_{x_2}^{(i)}(0) = -(\sigma^{(i)})^\top \psi'_{x_2^{(i)}(0)} = \sigma_2^{(i)} \quad (32)$$

$$\lambda_{p_1}^{(i)}(0) = -(\sigma^{(i)})^\top \psi'_{p_1^{(i)}(0)} = 0, \quad \lambda_{p_2}^{(i)}(0) = -(\sigma^{(i)})^\top \psi'_{p_2^{(i)}(0)} = 0 \quad (33)$$

and

$$\lambda_{x_1}^{(i)}(1) = (\sigma^{(i)})^\top \psi'_{x_1^{(i)}(1)} = \sigma_2^{(i)} + \sigma_3^{(i)}, \quad \lambda_{x_2}^{(i)}(1) = (\sigma^{(i)})^\top \psi'_{x_2^{(i)}(1)} = \sigma_4^{(i)} \quad (34)$$

$$\lambda_{p_1}^{(i)}(1) = (\sigma^{(i)})^\top \psi'_{p_1^{(i)}(1)} = -\sigma_3^{(i)T} \dot{x}_T(t_1^{(i)}(1)), \quad \lambda_{p_2}^{(i)}(1) = (\sigma^{(i)})^\top \psi'_{p_2^{(i)}(1)} = 0. \quad (35)$$

Boundary conditions (23) are given by (33) and the second equation in (35). Finally, from (32) and (34), it follows that $\sigma_3^{(i)} = \lambda_{x_1}^{(i)}(1) - \lambda_{x_2}^{(i)}(0)$. Substituting its value in (35), we obtain

$$\lambda_{p_1}^{(i)}(1) = -(\lambda_{x_1}^{(i)}(1) - \lambda_{x_2}^{(i)}(0))^\top \dot{x}_T(t_i + p_1^{(i)}(1))$$

and thus Eq. (24).

3 Branch and Bound

In the previous section, we saw how the bilevel optimal control problem can be transformed into a single level optimal control problem, involving binary variables w_{ij} . In this section, we present a technique for solving Problem 4 numerically. It is based on the approximation of the original infinite dimensional problem by a finite dimensional one.

For the sake of simplicity, we restrict ourselves to the simplest discretization method: Euler's method on an equidistant grid

$$\mathbb{G}_I := \{kh \mid k = 0, \dots, I\}$$

with $I \in \mathbb{N}$ and step-size $h = 1/I$. The discretization of the states and multipliers of Problem 4 on \mathbb{G}_I is given by $x_k^{(i)} := (x_{k,0}^{(i)}, \dots, x_{k,I}^{(i)})^\top \in \mathbb{R}^{(I+1)n_x}$, $\lambda_{x_k}^{(i)} := (\lambda_{x_k,0}^{(i)}, \dots, \lambda_{x_k,I}^{(i)})^\top \in \mathbb{R}^{(I+1)n_x}$ and $\lambda_{p_k}^{(i)} := (\lambda_{p_k,0}^{(i)}, \dots, \lambda_{p_k,I}^{(i)})^\top \in \mathbb{R}^{(I+1)}$. Discretization of the dynamics (19)–(21) of Problem 4 by Euler's method yields

$$\begin{aligned} x_{k,m+1}^{(i)} &= x_{k,m}^{(i)} + h \left\{ p_k^{(i)} \left[A^{(i)} x_{k,m}^{(i)} - \frac{1}{c} B^{(i)} (B^{(i)})^\top \lambda_{x_{k,m}}^{(i)} \right] \right\} \\ \lambda_{x_{k,m+1}}^{(i)} &= \lambda_{x_{k,m}}^{(i)} + h \left\{ -p_k^{(i)} (A^{(i)})^\top \lambda_{x_{k,m}}^{(i)} \right\} \\ \lambda_{p_{k,m+1}}^{(i)} &= \lambda_{p_{k,m}}^{(i)} + h \left\{ -1 - \lambda_{x_{k,m}}^{(i)} \left[A^{(i)} x_{k,m}^{(i)} - \frac{1}{2c} B^{(i)} (B^{(i)})^\top \lambda_{x_{k,m}}^{(i)} \right] \right\} \end{aligned}$$

for $k = 1, 2$ and $m = 0, \dots, I-1$. The boundary conditions (22)–(24) read as follows:

$$\begin{aligned} x_{1,0}^{(i)} &= x_0^{(i)}, & x_{2,I}^{(i)} &= x_0^{(i)} \\ x_{1,I}^{(i)} &= x_{2,0}^{(i)}, & x_{1,I}^{(i)} &= x_T(t_i + p_1^{(i)}) \\ \lambda_{p_{1,0}}^{(i)} &= 0, & \lambda_{p_{2,0}}^{(i)} &= 0 \\ \lambda_{p_{1,I}}^{(i)} &= -(\lambda_{x_{1,I}}^{(i)} - \lambda_{x_{2,0}}^{(i)})^\top \dot{x}_T(t_i + p_1^{(i)}), & \lambda_{p_{2,I}}^{(i)} &= 0. \end{aligned}$$

The rest of Problem 4 remains the same.

Note that after discretization, Problem 4 becomes a finite dimensional mixed-integer nonlinear optimization problem, which can be written in the following compact form (MINLP):

$$\begin{aligned} & \text{Minimize } F(y) \\ \text{s.t.} \quad & G(y, w) \leq 0, \quad H(y) = 0 \\ & y \in \mathbb{R}^{n_y}, \quad w \in \{0, 1\}^{n_w} \end{aligned}$$

where $y := (\xi, t_i, p_k^{(i)}, x_{k,m}^{(i)}, \lambda_{x_{k,m}}^{(i)}, \lambda_{p_{k,m}}^{(i)})$, $w = (w_{ij} \mid i, j \in \{1, \dots, N\}, i < j)$, $n_y = 1 + 3 \cdot N + 4 \cdot N \cdot I \cdot n_x + 2 \cdot N \cdot I$ and $n_w = N \cdot (N - 1)/2$. Furthermore $F(y) := \xi$ and the functions G and H are defined as

$$G(y, w) := \begin{bmatrix} t_i + p_1^{(i)} + p_2^{(i)} - \xi \\ -t_i \\ t_i + p_1^{(i)} - t_j - (1 - w_{ij})M \\ t_j + p_1^{(j)} - t_i - w_{ij}M \end{bmatrix} \quad (36)$$

and

$$H(y) := \begin{bmatrix} x_{k,m+1}^{(i)} - x_{k,m}^{(i)} - h \left\{ p_k^{(i)} \left[A^{(i)} x_{k,m}^{(i)} - \frac{1}{c} B^{(i)} (B^{(i)})^\top \lambda_{x_{k,m}}^{(i)} \right] \right\} \\ \lambda_{x_{k,m+1}}^{(i)} - \lambda_{x_{k,m}}^{(i)} + h p_k^{(i)} (A^{(i)})^\top \lambda_{x_{k,m}}^{(i)} \\ \lambda_{p_{k,m+1}}^{(i)} - \lambda_{p_{k,m}}^{(i)} - h \left\{ 1 + \lambda_{x_{k,m}}^{(i)} \left[A^{(i)} x_{k,m}^{(i)} - \frac{1}{2c} B^{(i)} (B^{(i)})^\top \lambda_{x_{k,m}}^{(i)} \right] \right\} \\ x_{1,0}^{(i)} - x_0^{(i)} \\ x_{2,I}^{(i)} - x_0^{(i)} \\ x_{1,I}^{(i)} - x_{2,0}^{(i)} \\ x_{1,I}^{(i)} - x_T(t_i + p_1^{(i)}) \\ \lambda_{p_{1,0}}^{(i)} \\ \lambda_{p_{2,0}}^{(i)} \\ \lambda_{p_{1,I}}^{(i)} + \left[\lambda_{x_{1,I}}^{(i)} - \lambda_{x_{2,0}}^{(i)} \right]^\top \dot{x}_T(t_i + p_1^{(i)}) \\ \lambda_{p_{2,I}}^{(i)} \end{bmatrix} \quad (37)$$

where the expressions in (36) hold for $i, j \in \{1, \dots, N\}$ with $i < j$, while the expressions in (37) hold for $i \in \{1, \dots, N\}$, $k \in \{1, 2\}$ and $m \in \{0, \dots, I - 1\}$.

The presence of the binary vector w makes the problem NP-hard, furthermore no gradient based method can be applied directly to the problem. Schittkowski et al. [4] proposed a mixed-integer SQP algorithm for solving nonlinear mixed-integer mathematical programs, nevertheless their method applies to problems, in which the integer variable is not ‘‘categorical’’, i.e. small changes between different integer values does not affect significantly the model. This does not apply in our case.

Our approach for solving MINLP is based on the Branch and Bound Algorithm 1 in [12]. It is a tree-search algorithm, combined with a rule for pruning sub-trees. At each node of the tree, a continuous relaxation of the initial mixed-integer problem has to be solved. In case the optimal solution of the relaxed problem turns out to be feasible for MINLP, its objective function value provides an upper bound for the optimal objective function value of MINLP and the node is explored. Usually, the relaxed problem does not have an integer solution. In this case, the optimal objective function value serves as lower bound for the sub-tree emanating from the current node. If the optimal objective function value is less than the upper bound

provided by the best solution found so far, then all successors of the current node are generated by adding additional constraints (branching). Afterwards, all successors have to be investigated. If at some node the relaxed problem has an optimal objective function value, which is greater or equal to the upper bound, provided by the best solution found so far, then this node needs not to be explored any further and the sub-tree can be pruned, since the optimal objective function values of all nodes in the sub-tree are greater or equal to the value of the current node.

We now investigate how a relaxation of MINLP is done. Given the vectors $0 \leq \underline{w} \leq \bar{w} \leq 1$ with $\underline{w}, \bar{w} \in \{0, 1\}^{n_w}$, let $W(\underline{w}, \bar{w}) = \{w \in \mathbb{R}^{n_w} \mid \underline{w} \leq w \leq \bar{w}\}$ and let $NLP(\underline{w}, \bar{w})$ denotes the continuous nonlinear optimization problem

$$\begin{aligned} & \text{Minimize } F(y) \\ & \text{s.t.} \quad G(y, w) \leq 0, \quad H(y) = 0 \\ & \quad \quad y \in \mathbb{R}^{n_y}, \quad w \in W(\underline{w}, \bar{w}). \end{aligned}$$

Note that $\{0, 1\}^{n_w} \subset W(0, 1)$ and $W(0, 1)$ is called a relaxation of $\{0, 1\}^{n_w}$. Correspondingly, $NLP(\underline{w}, \bar{w})$ is called a relaxation of MINLP. Since the admissible set of the relaxation is larger than the one of MINLP, the optimal objective function value of the relaxed problem provides a lower bound for the optimal objective function value of MINLP.

Let us focus now on how to create new nodes. This is done by the following:

Branching Rule

Let the problem $NLP(\underline{w}, \bar{w})$ with $\underline{w} = (w_1, \dots, w_{n_w})^\top \in \{0, 1\}^{n_w}$, $\bar{w} = (\bar{w}_1, \dots, \bar{w}_{n_w})^\top \in \{0, 1\}^{n_w}$, $0 \leq \underline{w} \leq \bar{w} \leq 1$, $\underline{w} \neq \bar{w}$ be given. Let (y^*, w^*) be an optimal solution of $NLP(\underline{w}, \bar{w})$ such that w^* is not integral, i.e. $w^* \notin \{0, 1\}^{n_w}$. Then, branching consists of the following steps

1. Determine some index k , such that $\underline{w}_k < w_k^* < \bar{w}_k$. The choice of k can be done in several ways: first/last non-integer value of w^* , maximum/minimum distance of w_k^* to an integer.
2. Create two new subproblems $NLP(\underline{w}^{(1)}, \bar{w}^{(1)})$ and $NLP(\underline{w}^{(2)}, \bar{w}^{(2)})$ with

$$\begin{aligned} \underline{w}_i^{(1)} &= \underline{w}_i, & \bar{w}_i^{(1)} &= \begin{cases} \bar{w}_i, & \text{if } i \neq k, \\ 0, & \text{if } i = k, \end{cases} & i &= 1, \dots, n_w, \\ \underline{w}_i^{(2)} &= \begin{cases} \underline{w}_i, & \text{if } i \neq k, \\ 1, & \text{if } i = k, \end{cases} & \bar{w}_i^{(2)} &= \bar{w}_i, & i &= 1, \dots, n_w. \end{aligned}$$

A recursive application of the branching rule starting with $NLP(0, 1)$ generates a finite tree structure. Each node of the tree corresponds to a continuous nonlinear programming problem $NLP(\underline{w}, \bar{w})$. Each edge corresponds to imposing one additional constraint.

Following Leyffer [22], a node $NLP(\underline{w}, \overline{w})$ is explored or fathomed, i.e. no further branching has to be done, if one of the following events occur:

- $NLP(\underline{w}, \overline{w})$ is infeasible. This implies that all nodes of the subtree are infeasible as well.
- $NLP(\underline{w}, \overline{w})$ has an integral solution. The corresponding optimal objective function value serves as an upper bound for MINLP.
- The optimal objective function value of $NLP(\underline{w}, \overline{w})$ is greater than or equal to the upper bound found so far. Due to the branching rule, the optimal objective function values are monotonically non-decreasing for the nodes of the subtree and thus, the subtree can be pruned.

In addition, we need a rule to traverse the tree. In this article, the depth-first approach is adopted, in this way a feasible solution for MINLP and hence an upper bound is computed as soon as possible. Alternative strategies are breadth-first search or some problem-specific search strategies.

Note that Algorithm 1 requires to solve the sub-problems to global optimality. This is a difficult task in its own and could be achieved by using convex under-estimators, which, however, are difficult to obtain for our problems. Hence, in

Algorithm 1: Branch and Bound algorithm

<pre> 1: Let F_u be an upper bound for the optimal objective function value of MINLP (if none is known, let $F_u = +\infty$) 2: Let the root of the tree $NLP(0, 1)$ be active 3: while (there are active nodes) do 4: Select an active node $NLP(\underline{w}, \overline{w})$ 5: Solve $NLP(\underline{w}, \overline{w})$ (if possible) and let (y^*, w^*) be an optimal solution and $F^* =$ $F(y^*, w^*)$ be the optimal objective function value of $NLP(\underline{w}, \overline{w})$ 6: if (infeasible) then 7: Mark node as explored 8: end if 9: if ($w^* \in \{0, 1\}^{n_w}$) then 10: if ($F^* < F_u$) then 11: Save (y^*, w^*) as best solution and set $F_u = F^*$ 12: end if 13: Mark node as explored 14: end if 15: if ($F^* \geq F_u$) then 16: Mark node as explored 17: end if 18: if ($F^* < F_u$) then 19: Apply branching rule, mark all successors as active and mark current node as inactive 20: end if 21: end while </pre>
--

our computational studies we only considered local minima facing the danger that optimal solutions are pruned falsely. Nevertheless, since the combinatorial effort in our case studies is not high, we crosschecked the results by enumeration of all possible combinations. The rigorous construction of convex underestimators remains as a task for future research.

4 Unconstrained Test Problem

In order to test Algorithm 1, applied to Problem 4, we consider a simple scheduling problem, involving $N = 3$ robots moving in the two dimensional plane. Denoting with $(x_1^{(i)}, x_2^{(i)})$ the position of the i -th robot in the plane, its dynamics is given by

$$\begin{aligned} \dot{x}_1^{(i)}(t) &= x_3^{(i)}(t) & a.e.in (t_i, t_i + p_1^{(i)} + p_2^{(i)}) \\ \dot{x}_2^{(i)}(t) &= x_4^{(i)}(t) & a.e.in (t_i, t_i + p_1^{(i)} + p_2^{(i)}) \\ \dot{x}_3^{(i)}(t) &= u_1^{(i)}(t) & a.e.in (t_i, t_i + p_1^{(i)} + p_2^{(i)}) \\ \dot{x}_4^{(i)}(t) &= u_2^{(i)}(t) & a.e.in (t_i, t_i + p_1^{(i)} + p_2^{(i)}) \end{aligned}$$

with starting positions $x_0^{(1)} = (4, 0)^\top$, $x_0^{(2)} = (0, 4)^\top$ and $x_0^{(3)} = (-4, 0)^\top$ respectively. Finally, we assume that the target robot is moving along a circle centered in the origin with radius $R = 2$ [m] and speed $v = 1.4$ [m/s]. Thus, its position at any instance t is given by $x_T(t) = (R \cdot \cos(v/R \cdot t), R \cdot \sin(v/R \cdot t))^\top$.

We solve the previously described problem with $c = 0.1$, $M = 1000$ and $I = 25$ in the following way:

- Algorithm 1 handles the tree structure of the problem
- At each node of the tree, the relaxed problem is solved by the optimal control package OCPID-DAE1, publicly available at <http://www.optimal-control.de>

The optimal scheduling of the robots is (1) < (2) < (3) with starting times $t_1 = 0.00$, $t_2 = 1.46$, $t_3 = 2.75$ and phase durations $p^{(1)} = [1.46, 2.49]$, $p^{(2)} = [1.29, 1.94]$ and $p^{(3)} = [1.70, 1.62]$. We can observe that $p_1^{(i)} < p_2^{(i)}$ for $i = 1, 2$ while $p_1^{(3)} > p_2^{(3)}$. This is reasonable, since $p_2^{(1)}$ and $p_2^{(2)}$ do not influence the objective function, while $p_2^{(3)}$ does (being (3) the last scheduled robot).

The necessary time for solving the problem was 3.804 s (running on 2.4 GHz Intel Core i7 with 8 GB RAM). Figure 1 shows the (x_1, x_2) -positions of the robots in the plane as well as the controls of the three robots.

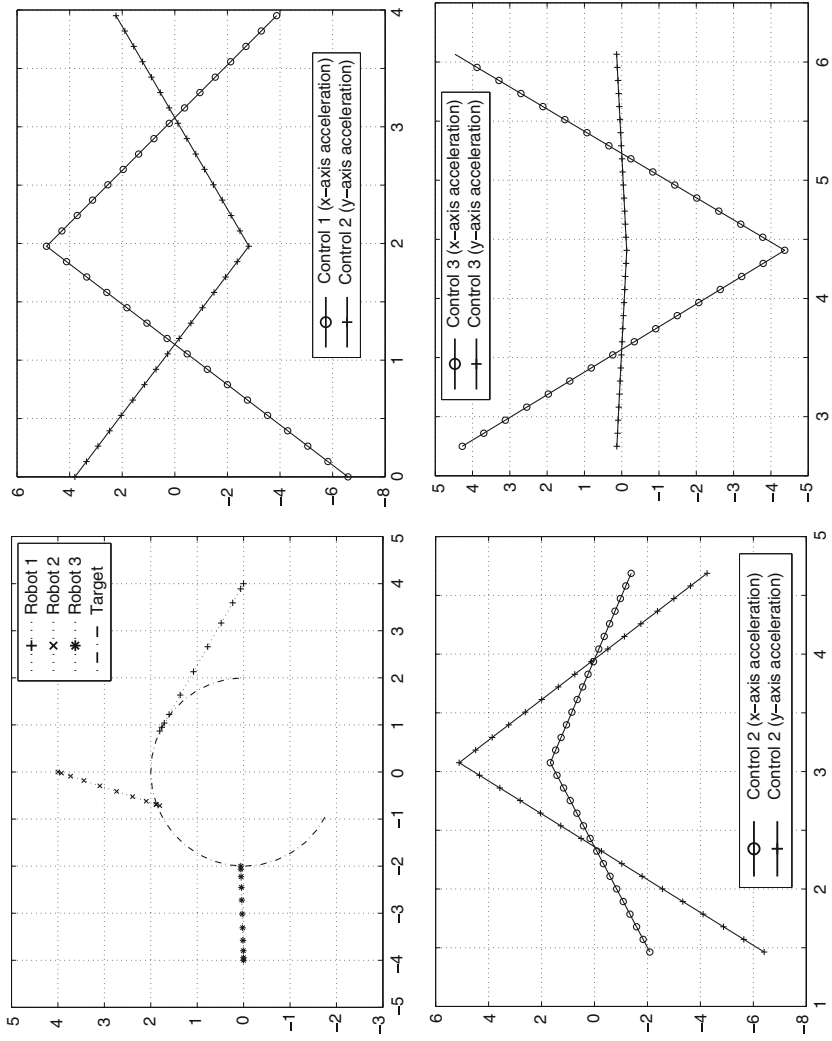


Fig. 1 Positions (top left) and controls of robots 1 (top right), 2 (bottom left), and 3 (bottom right)

5 Control and State Constrained Problem

In the previous sections, we imposed no control or state constraints on Problem 2, mainly due to the sake of simplicity. In fact, the presence of constraints leads to complementarity conditions, stated in Theorem 1, which are hard to solve numerically. Nevertheless, it is often necessary to impose constraints due to structural or safety reasons. In many cases, pure state constraints (i.e. constraints involving only the states of the problem) have to be considered, for instance if an obstacle has to be avoided or the speed has to be kept in a certain range. The main drawback of pure state constraints is that the multiplier λ in Theorem 1 loses its regularity. In fact, it can be proven (see [13, Section 3.2]) that λ is a function of bounded variation and the adjoint equations can only be written in an integral form, involving Riemann-Stieltjes integrals (see [13, p.64–66] for details). From a numerical point of view, this causes several difficulties, since efficient numerical schemes are required to handle the Riemann-Stieltjes integrals properly.

To overcome these difficulties, in this paper we adopt the concept of virtual control, introduced in [7, Chapter 4], [20] for optimal control problems subject to an elliptic partial differential equation. The idea is to embed the optimal control problem subject to pure state constraints into a family of optimal control problems subject to mixed control-state constraints using a regularization parameter $\alpha > 0$. For linear quadratic optimal control problems, it has been proven in [14] that the optimal solution of the regularized problem converges to the optimal solution of the pure state constrained problem in L^2 norm, as α tends to zero. For general nonlinear optimal control problems, the convergence of the regularized solution is still an open problem, but its formal proof is beyond the scope of this paper. Nevertheless the numerical simulations in the next section suggest that the results obtained from [14] can be extended to general nonlinear optimal control problem.

Let the pure state constraint

$$g^{(i)}(t, x^{(i)}(t)) \leq 0_{\mathbb{R}^{n_g}}$$

with a twice continuously differentiable function $g^{(i)} : [t_i, t_i + p_1^{(i)} + p_2^{(i)}] \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_g}$ be given. Using artificial control vectors $v_k^{(i)} \in L^\infty([0, 1], \mathbb{R}^{n_g})$ (called virtual controls) with $i = 1 \dots, N$, $k = 1, 2$, and a regularization parameter $\alpha > 0$, the virtual control concept embeds the pure state constraint into a family of mixed control constraints

$$g^{(i)}(t_k^{(i)}(\tau), x_k^{(i)}(\tau)) - \alpha v_k^{(i)}(\tau) \leq 0_{\mathbb{R}^{n_g}} \tag{38}$$

where $t_k^{(i)}$ is the time transformation, defined in (14), while $x_k^{(i)}$ is defined in (15).

Moreover, an additional penalty term $\frac{1}{2} \int_0^1 \|v_k^{(i)}(\tau)\|^2 d\tau$ is added to the objective function in order to drive $v_k^{(i)}$ to zero as α tends to zero. Using this virtual control concept and the transformations (14) and (15), we obtain a scaled, constrained version of Problems 2 with a mixed control state constraint for $\alpha > 0$:

Problem 5 Minimize

$$\int_0^1 p_1^{(i)} \left\{ 1 + \frac{c}{2} \|u_1^{(i)}\|^2 \right\} + p_2^{(i)} \left\{ 1 + \frac{c}{2} \|u_2^{(i)}\|^2 \right\} + \frac{1}{2} \left\{ \|v_1^{(i)}\|^2 + \|v_2^{(i)}\|^2 \right\} d\tau \quad (39)$$

with respect to $x_1^{(i)}, x_2^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$, $p_1^{(i)}, p_2^{(i)} \in \mathbb{R}$, $u_1^{(i)}, u_2^{(i)} \in L^\infty([0, 1], \mathbb{R}^{n_u})$, and $v_1^{(i)}, v_2^{(i)} \in L^\infty([0, 1], \mathbb{R}^{n_g})$, subject to

$$\dot{x}_k^{(i)} = p_k^{(i)} (A^{(i)} x_k^{(i)} + B^{(i)} u_k^{(i)}) \quad a.e. \tau \in [0, 1] \quad (40)$$

$$g^{(i)}(t_k^{(i)}, x_k^{(i)}) - \alpha v_k^{(i)} \leq 0_{\mathbb{R}^{n_g}} \quad a.e. \tau \in [0, 1] \quad (41)$$

$$x_1^{(i)}(0) - x_0^{(i)} = 0_{\mathbb{R}^{n_x}} \quad (42)$$

$$x_1^{(i)}(1) - x_2^{(i)}(0) = 0_{\mathbb{R}^{n_x}} \quad (43)$$

$$x_1^{(i)}(1) - x_T(t_i + p_1^{(i)}(1)) = 0_{\mathbb{R}^{n_x}} \quad (44)$$

$$x_2^{(i)}(1) - x_0^{(i)} = 0_{\mathbb{R}^{n_x}}. \quad (45)$$

Remark 3 Note that in order to obtain Problem 5, we have first exploited transformations (14) and (15) and then introduced the virtual control regularization (38). This leads to a slightly simpler formulation of the adjoint equations in Theorem 3 compared to those which we would have obtained if the virtual control regularization would have been applied prior to the time transformation.

We can now state a result similar to Theorem 2 for the constrained Problem 5.

Theorem 3 Consider the bilevel optimal control problem BLSOCP with upper level Problem 1 and lower level Problems 5 with $c, \alpha > 0$ for $i = 1, \dots, N$. Let a constraint qualification hold for Problems 5. Then, the corresponding single level problem, which is obtained by replacing Problems 5 by their necessary conditions, is the following mixed-integer optimal control problem:

Problem 6 Minimize

$$\xi \quad (46)$$

with respect to $t_i, p_1^{(i)}, p_2^{(i)} \in \mathbb{R}$, $w_{ij} \in \{0, 1\}$, $x_1^{(i)}, x_2^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$, $\lambda_{x_1}^{(i)}, \lambda_{x_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$, $\lambda_{p_1}^{(i)}, \lambda_{p_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R})$ and

$\mu_1^{(i)}, \mu_2^{(i)} \in L^\infty([0, 1], \mathbb{R}^{n_g})$ for $i, j \in \{1, \dots, N\}$ with $i < j$, subject to

$$t_i + p_1^{(i)} + p_2^{(i)} \leq \xi \quad (47)$$

$$t_i + p_1^{(i)} - t_j \leq (1 - w_{ij})M, \quad t_j + p_1^{(j)} - t_i \leq w_{ij}M, \quad t_i \geq 0 \quad (48)$$

$$\dot{x}_k^{(i)} = p_k^{(i)} \left[A^{(i)} x_k^{(i)} - \frac{1}{c} B^{(i)} (B^{(i)})^\top \lambda_{x_k}^{(i)} \right] \quad (49)$$

$$\dot{\lambda}_{x_k}^{(i)} = -p_k^{(i)} (A^{(i)})^\top \lambda_{x_k}^{(i)} - \nabla_x g^{(i)}(t_k^{(i)}, x_k^{(i)})^\top \mu_k^{(i)} \quad (50)$$

$$\begin{aligned} \dot{\lambda}_{p_k}^{(i)} = & -1 - (\lambda_{x_k}^{(i)})^\top \left[A^{(i)} x_k^{(i)} - \frac{1}{2c} B^{(i)} (B^{(i)})^\top \lambda_{x_k}^{(i)} \right] \\ & - \tau (\mu_k^{(i)})^\top \nabla_t g^{(i)}(t_k^{(i)}, x_k^{(i)}) - \delta_{k1} (\mu_2^{(i)})^\top \nabla_t g^{(i)}(t_2^{(i)}, x_2^{(i)}) \end{aligned} \quad (51)$$

$$\mu_k^{(i)} \geq 0, \quad (\mu_k^{(i)})^\top [g^{(i)} - \alpha^2 \mu_k^{(i)}] = 0, \quad g^{(i)} - \alpha^2 \mu_k^{(i)} \leq 0 \quad (52)$$

$$x_1^{(i)}(0) = x_2^{(i)}(1) = x_0^{(i)}, \quad x_1^{(i)}(1) = x_2^{(i)}(0), \quad x_1^{(i)}(1) = x_T(t_1^{(i)}(1)) \quad (53)$$

$$\lambda_{p_1}^{(i)}(0) = \lambda_{p_2}^{(i)}(0) = \lambda_{p_2}^{(i)}(1) = 0 \quad (54)$$

$$\lambda_{p_1}^{(i)}(1) = -(\lambda_{x_1}^{(i)}(1) - \lambda_{x_2}^{(i)}(0))^\top \dot{x}_T(t_1^{(i)}(1)). \quad (55)$$

Proof Let $\lambda_{x_1}^{(i)}, \lambda_{x_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R}^{n_x})$, $\lambda_{p_1}^{(i)}, \lambda_{p_2}^{(i)} \in W^{1,\infty}([0, 1], \mathbb{R})$, $\mu_1^{(i)}, \mu_2^{(i)} \in L^\infty([0, 1], \mathbb{R}^{n_g})$ and $\sigma^{(i)} \in \mathbb{R}^{4n_x}$ be the multipliers defined in Theorem 1, related to Problem 5. The Hamilton function reads

$$\begin{aligned} \hat{H}_\alpha^{(i)}[\tau] := & \hat{H}_\alpha^{(i)}(\tau, x_1^{(i)}, x_2^{(i)}, p_1^{(i)}, p_2^{(i)}, u_1^{(i)}, u_2^{(i)}, v_1^{(i)}, v_2^{(i)}, \lambda_{x_1}^{(i)}, \lambda_{x_2}^{(i)}, \mu_1^{(i)}, \mu_2^{(i)}) \\ = & p_1^{(i)} \left\{ 1 + \frac{c}{2} \|u_1^{(i)}\|^2 \right\} + p_2^{(i)} \left\{ 1 + \frac{c}{2} \|u_2^{(i)}\|^2 \right\} + \frac{1}{2} \left\{ \|v_1^{(i)}\|^2 + \|v_2^{(i)}\|^2 \right\} \\ & + (\lambda_{x_1}^{(i)})^\top \left\{ p_1^{(i)} [A^{(i)} x_1^{(i)} + B^{(i)} u_1^{(i)}] \right\} + (\lambda_{x_2}^{(i)})^\top \left\{ p_2^{(i)} [A^{(i)} x_2^{(i)} + B^{(i)} u_2^{(i)}] \right\} \\ & + (\mu_1^{(i)})^\top \left\{ g^{(i)}(t_1^{(i)}, x_1^{(i)}) - \alpha v_1^{(i)} \right\} + (\mu_2^{(i)})^\top \left\{ g^{(i)}(t_2^{(i)}, x_2^{(i)}) - \alpha v_2^{(i)} \right\}. \end{aligned}$$

From the stationarity of the Hamilton function, we have

$$u_k^{(i)} = -\frac{1}{c} (B^{(i)})^\top \lambda_{x_k}^{(i)} \quad a.e. \tau \in [0, 1], \quad k = 1, 2 \quad (56)$$

(see proof of Theorem 2 for details). Furthermore $0 = \nabla_{v_k} \hat{H}_\alpha$ implies $v_k^{(i)} - \alpha \mu_k^{(i)} = 0_{\mathbb{R}^{n_g}}$ and thus

$$v_k^{(i)} = \alpha \mu_k^{(i)} \quad a.e. \tau \in [0, 1], k = 1, 2. \quad (57)$$

By substituting the control function value $u_k^{(i)}$ from (56) in (40), we obtain (49). Let us now derive the adjoint equations (50)–(51). For $k = 1, 2$ from Theorem 1, we have

$$\dot{\lambda}_{x_k}^{(i)} = -\nabla_{x_k} \hat{H}_\alpha^{(i)} = -p_k^{(i)} (A^{(i)})^\top \lambda_{x_k}^{(i)} - \nabla_{x_k} g^{(i)}(t_k^{(i)}, x_k^{(i)})^\top \mu_k^{(i)}$$

and, recalling that $t_1^{(i)}(\tau) = t_i + \tau p_1^{(i)}$ and $t_2^{(i)}(\tau) = t_i + p_1^{(i)} + \tau p_2^{(i)}$,

$$\begin{aligned} \dot{\lambda}_{p_k}^{(i)} &= -\nabla_{p_k} \hat{H}_\alpha^{(i)} = -1 - \frac{c}{2} \|u_k^{(i)}\|^2 - (\lambda_{x_k}^{(i)})^\top [A^{(i)} x_k^{(i)} + B^{(i)} u_k^{(i)}] \\ &\quad - \tau (\mu_k^{(i)})^\top \nabla_{t_k} g^{(i)}(t_k^{(i)}, x_k^{(i)}) - \delta_{k1} (\mu_2^{(i)})^\top \nabla_{t_2} g^{(i)}(t_2^{(i)}, x_2^{(i)}) \\ &= -1 - (\lambda_{x_k}^{(i)})^\top [A^{(i)} x_k^{(i)} - \frac{1}{2c} B^{(i)} (B^{(i)})^\top \lambda_{x_k}^{(i)}] \\ &\quad - \tau (\mu_k^{(i)})^\top \nabla_{t_k} g^{(i)}(t_k^{(i)}, x_k^{(i)}) - \delta_{k1} (\mu_2^{(i)})^\top \nabla_{t_2} g^{(i)}(t_2^{(i)}, x_2^{(i)}) \end{aligned} \quad (58)$$

which are exactly Eqs. (50)–(51). Note that in (58), we have substituted $u_k^{(i)}$ and $v_k^{(i)}$ with their analytical expression from equation (56) and (57). The complementarity conditions in Theorem 1 read as

$$\mu_k^{(i)} \geq 0_{\mathbb{R}^{n_g}}, \quad g^{(i)} - \alpha v_k^{(i)} \leq 0_{\mathbb{R}^{n_g}}, \quad (\mu_k^{(i)})^\top (g^{(i)} - \alpha v_k^{(i)}) = 0$$

almost everywhere in $[0, 1]$. Introducing $v_k^{(i)}$ from (57) yields (52).

Finally, Eqs. (53)–(55) can be derived in the same way as in Theorem 2.

Problem 6 is a mixed-integer dynamic optimization problem with complementarity constraints. The presence of the complementarity constraints in (52) causes standard constraint qualifications to fail and a straightforward application of the Branch & Bound method to the discretized problem will cause numerical difficulties in solving the relaxed optimization problems, which will be MPCCs (mathematical programs with complementarity constraints). There are several ways to address MPCCs, e.g. by relaxation or penalization. Any of these approaches would be feasible. In our numerical studies, the best performance was obtained by applying the Lipschitz-continuous Fischer-Burmeister function

$$\Phi^{FB}(a, b) := \sqrt{a^2 + b^2} - a - b \quad (a, b \in \mathbb{R}) \quad (59)$$

componentwise to the complementarity constraints (52). The Fischer-Burmeister function has the property that $\Phi^{FB}(a, b) = 0$ holds, if and only if the complementarity conditions $a, b \geq 0$ and $ab = 0$ hold. To this end, (52) can be replaced by the equality constraints

$$\Phi^{FB} \left(-g_j^{(i)} + \alpha^2 \mu_{k_j}^{(i)}, \mu_{k_j}^{(i)} \right) = 0_{\mathbb{R}^{n_g}}, \quad j = 1, \dots, n_g$$

that have to hold almost everywhere in $(0, 1)$.

6 Constrained Test Problem

Let us consider the same scheduling problem, already stated in Sect. 4 with additional obstacle avoidance constraints. For simplicity, we assume that the obstacle is a circular body with center (x_o, y_o) and radius r_o . In order to have no collision with it, the following constraints have to be imposed on each robot:

$$r_o^2 - (x_1^{(i)}(t) - x_o)^2 - (x_2^{(i)}(t) - y_o)^2 \leq 0 \quad \forall t \in [t_i, t_i + p_1^{(i)} + p_2^{(2)}], \quad i = 1, \dots, N$$

Thus, we obtain a bilevel optimization problem, where the lower level problem has pure state constraints. By exploiting Theorem 3, we can obtain a mixed-integer optimal control problem, with relaxation parameter $\alpha > 0$, and $c = 0.1$ which has to be solved at each iteration of Algorithm 1. For this numerical test, we chose $\alpha = 10^{-6}$.

The starting positions of the three robots are the same as for the unconstrained problem (i.e. $x^{(1)} = [4, 0]$, $x^{(2)} = [0, 4]$ and $x^{(3)} = [-4, 0]$), the target moves along the same trajectory, while the obstacle has been placed along the target’s trajectory $((x_o, y_o) = [-2, 0]$ with $r_o = 1$), hence no interactions are allowed inside the circle. The optimal solution provides the same scheduling as for the unconstrained problem: $(1) < (2) < (3)$, with starting times $t_1 = 0.44$, $t_2 = 2.34$, $t_3 = 3.75$ and phase durations $p^{(1)} = [1.73, 2.75]$, $p^{(2)} = [1.40, 2.27]$ and $p^{(3)} = [1.48, 3.18]$. We observe that the second robot interacts with the target just before it enters the obstacle area, while the third robot interacts with the target as soon as it leaves the obstacle area. We believe this is the reason for the starting delay ($t_1 = 0.44 > 0$). The running time of the whole optimization procedure was 683 seconds, significantly greater than the unconstrained one (due to the presence of state constraints, which increase significantly the dimensions of the discretized problem).

Figure 2 shows the (x_1, x_2) -positions of the robots in the plane as well as the controls of the three robots.

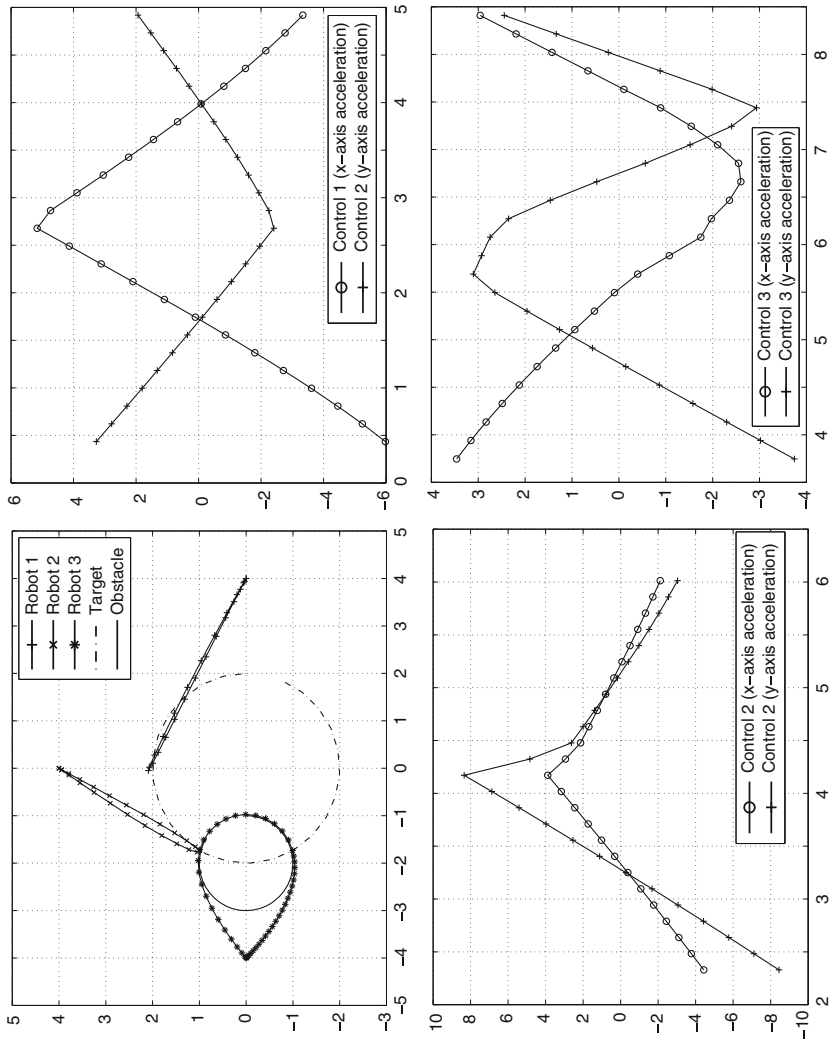


Fig. 2 Positions (top left) and controls of robots 1 (top right), 2 (bottom left), and 3 (bottom right)

7 Conclusions and Outlook

The paper investigates a class of bilevel optimization problems with a scheduling problem at the upper level and a control and state constrained optimal control problem at the lower level. The lower level problem is replaced by its necessary optimality conditions using a virtual control relaxation method and the resulting mixed-integer nonlinear program is then solved by a Branch & Bound method. Two simple numerical examples illustrate the performance of the method. With the presented approach, which has to be considered as a first step towards efficient solution techniques for the class of bilevel scheduling optimal control problems, it is in principle possible to consider more complex dynamics and more complex state constraints. Even more complex obstacle shapes can be handled as described in [15]. These extensions, which are imperative for practically relevant problem settings, of course add another level of complexity and much higher CPU times and robustness issues can be expected. To overcome these issues, further research addressing robustness, efficiency, and structure exploitation of the methods is necessary. On a theoretical level, the construction of convex underestimators in the Branch & Bound method is an open and challenging task. Moreover, alternative approaches such as the previously mentioned black-box approach or an approach using value functions for the lower level problem have to be investigated.

References

1. Albrecht, S.: Modeling and numerical solution of inverse optimal control problems for the analysis of human motions. PhD thesis, University of Technology Munich, Department of Mathematics, Munich (2013)
2. Androulakis, I.P., Maranas, C.D., Floudas, C.A.: α BB: a global optimization method for general constrained nonconvex problems. *J. Global Optim.* **7**, 337–363 (1995)
3. Bard, J.F.: Practical Bilevel Optimization. Nonconvex Optimization and Its Applications, vol. 30. Kluwer Academics, Dordrecht (1998)
4. Braak, G.V.D., Bünner, M.J., Schittkowski, K.: Optimal design of electronic components by mixed-integer nonlinear programming. *Optim. Eng.* **5**, 271–294 (2004)
5. Bracken, J., McGill, J.T.: Mathematical programs with optimization problems in the constraints. *Oper. Res.* **21**, 37–44 (1973)
6. Bracken, J., McGill, J.T.: Defence applications of mathematical programs with optimization problems in the constraints. *Oper. Res.* **22**, 1086–1096 (1974)
7. Cherednichenko, S., Krumbiegel, K., Rösch, A.: Error estimates for the Lavrientiev regularization of elliptic optimal control problems. *Inverse Probl.* **24**, 1–21 (2008)
8. Dempe, S.: Foundations of Bilevel Programming. Kluwer Academic Publishers, Dordrecht (2002)
9. Fisch, F.: Development of a framework for the solution of high-fidelity trajectory optimization problems and bilevel optimal control problems. PhD thesis, University of Technology Munich, Department of Flight Systems Dynamics, Munich (2010)
10. Fletcher, R., Leyffer, S.: Numerical experience with solving MPECs as NLPs. Department of Mathematics and Computer Science, University of Dundee, Dundee (2002)

11. Fletcher, R., Leyffer, S., Ralph, D., Scholtes, S.: Local convergence of SQP methods for Mathematical Programs with Equilibrium Constraints. *SIAM J. Optim.* **17**, 259–286 (2006)
12. Gerdtts, M.: Solving mixed-integer optimal control problems by Branch & Bound: a case study from automobile test-driving with gear shift. *Optim. Control Appl. Methods* **26**, 1–18 (2005)
13. Gerdtts, M.: *Optimal Control of ODE and DAEs*. Walter de Gruyter GmbH & Co. KG, Berlin, Boston (2012)
14. Gerdtts, M., Hüpping, B.: Virtual control regularization of state constrained linear quadratic optimal control problems. *Comput. Optim. Appl.* **51**, 867–882 (2012)
15. Gerdtts, M., Henrion, R., Hömberg, D., Landry, C.: Path planning and collision avoidance for robots. *Numer. Algebra Control Optim.* **2**, 437–463 (2012)
16. Gümüş, Z.H., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. *J. Global Optim.* **20**, 1–31 (2001)
17. Hatz, K.: Efficient numerical methods for hierarchical dynamic optimization with application to cerebral palsy gait modeling. PhD thesis, University of Heidelberg (2014)
18. Knauer, M.: Bilevel-Optimalsteuerung mittels hybrider Lösungsmethoden am Beispiel eines deckengeführten Regalbediengerts in einem Hochregallager, PhD thesis. Universität Bremen (2009)
19. Knauer, M.: Fast and save container cranes as bilevel optimal control problems. *Math. Comput. Model. Dyn.* **18**, 465–486 (2012)
20. Krumbiegel, K., Rösch, A.: On the regularization error of state constrained Neumann control problems. *Control Cybern.* **37**, 369–392 (2008)
21. LeBlanc, L.J.: Mathematical programming algorithms for large scale network equilibrium and network design problems. PhD thesis, Northwestern University, Evanston, IL (1973)
22. Leyffer, S.: Integrating SQP and Branch-and-Bound for mixed-integer nonlinear programming. *Comput. Optim. Appl.* **18**, 295–309 (2001)
23. Marcotte, P.: Network design problem with congestion effects: a case of bilevel programming. *Math. Program.* **34**, 23–36 (1986)
24. Stackelberg, H.: *The Theory of Market Economy*. Oxford University Press, Oxford (1952)
25. Ye, J.: Necessary conditions for bilevel dynamic optimization problems. *SIAM J. Control Optim.* **33**, 1208–1223 (1995)
26. Ye, J.: Optimal strategies for bilevel dynamic problems. *SIAM J. Control Optim.* **35**, 512–531 (1997)

Part III
Selected Production Technologies

Math-Based Algorithms and Software for Virtual Product Realization Implemented in Automotive Paint Shops

Fredrik Edelvik, Andreas Mark, Niklas Karlsson, Tomas Johnson, and Johan S. Carlson

Abstract We present a simulation framework that makes it possible to accurately simulate spray painting of e.g. a truck cab in only a few hours on a standard computer. This is an extreme improvement compared to earlier approaches that require weeks of simulation time. Unique algorithms for coupled simulations of air flows, electrostatic fields and charged paint particles make this possible. In addition, we demonstrate that the same framework can be used to efficiently simulate the laydown of sealing or adhesive material. In the virtual paint factory the production preparation process can be performed in the computer, which allows the engineers to replace physical prototypes with virtual ones to shorten the lead time in product development, and avoid future unforeseen technological and environmental problems that can be extremely costly if they are discovered at the end of the production line, or even worse by the costumer.

1 Introduction

Today, the margins of automotive manufacturers are moderate and competition is fierce. The industry furthermore faces shifts of paradigms regarding propulsion as well as styling, with environmental requirements ever-present. Effective product realization response is thus important. The rapid increase in computational power has made virtual tools an integrated part of the development of products and processes. Virtual prototyping stimulates industrial innovation and simulations offer an alternative to measurements, when these are too expensive or even impossible to perform. Furthermore, the risk for unforeseen costs and quality problems is reduced through the possibility to perform analyses and optimization in the early stages of product development. Today, although most development is done virtually, design

F. Edelvik (✉) • A. Mark • N. Karlsson • T. Johnson • J.S. Carlson
Fraunhofer-Chalmers Research Centre for Industrial Mathematics, Chalmers Science Park,
SE-412 88 Gothenburg, Sweden
e-mail: fredrik.edelvik@fcc.chalmers.se; andreas.mark@fcc.chalmers.se;
niklas.karlsson@fcc.chalmers.se; tomas.johnson@fcc.chalmers.se;
johan.carlson@fcc.chalmers.se

decisions are still based on experience rather than mathematical analysis. In this chapter the focus is on paint and surface treatment processes in automotive paint shops, in earlier works we have shown many examples of the power of mathematics for virtual product realization in production planning and robot lines [1, 5, 19, 20].

The surface treatment is the process in an automotive factory that consumes most energy, water and chemicals, and produces most waste and pollution. Roughly 40% of the energy in major OEM operations is used in the paint shop with an average consumption of 700–900 kWh per car body. Within the paint shop the dominating energy cost is the ventilation and heating of the air in the booth (50%) followed by the ovens (25%). The CO₂ emissions from a modern paint shop (BMW Shenyang) are around 140 kg/car, and considerably higher in older paint shops. The solvent based automotive paint is a major source of man-made volatile organic components (VOCs) and hazardous air pollutants including toluene, xylene, methyl ethyl ketone, and ethyl benzene, that may have long-term health effects and cause harm to the environment such as increasing the ozone concentration in the lower atmosphere. The length of a coating line from the body-in-white shop to the final assembly is usually between two and three kilometers. Roughly 60 cars are coated per hour and the dwell time in the paint shop is about 8–11 h [21]. This means that the paint shop not only has a large environmental impact it is also a bottleneck in production. On the other hand, due to its great complexity the painting process offers many approaches to improve the use of energy, material and reduce emissions that in turn have direct impact on sustainable automotive production.

Virtual tools are frequently used to support an effective product and production realization in other parts of the automotive factory, but that is not the case in the paint shop. In the paint shop the product preparation, when robot paths and process parameters are fine-tuned, is a slow and costly trial-and-error procedure, where a large number of prototypes are painted, washed and painted again etc. There is therefore a great need to improve the product preparation process and this is absolutely necessary to meet the future demands on fast adaption and tailored solutions for new material combinations and products. The possibility to perform systematic simulations is then essential and would contribute to sustainable production by reducing the number of prototypes that needs to be painted, and by making it possible to optimize the processes with respect to quality, cost and environmental impact. However, the surface treatment processes pose great challenges for mathematical modeling and simulation.

In this chapter we focus on spray painting and sealing. Although a few attempts have been made to simulate spray painting [2–4, 7, 8, 27] the simulation times were prohibitively long for the tools to be industrially useful and a systematic validation for realistic geometries was missing. In [12] we presented a novel framework that allows for accurate simulations of spray painting of a car in just a few hours on a standard computer. To achieve this, novel algorithms were developed for coupled simulations of air flow, electromagnetic fields, and charged paint droplets. Particularly important for the computational efficiency is the Navier-Stokes solver. Unique, immersed boundary methods are used to model the presence of objects in the fluid and they are combined with an adaptive Cartesian octree grid [10–13].

This enables modeling of moving objects at virtually no additional computational cost, and greatly simplifies preprocessing by avoiding the cumbersome generation of a body-conforming mesh. To validate the simulation framework, an extensive measurement campaign, including painting of plates and two car fenders for different process conditions and robot paths, was performed [12].

For the sealing application we first investigated an approach based on smoothed particle hydrodynamics (SPH) [17]. The advantage with SPH is that non-Newtonian fluids and free surface flow can be easily handled. The results showed a good agreement with experiments for laydown simulation with a hollow-cone nozzle but the approach suffered from long simulation times, mainly due to the fact that very short time steps were required to handle when the material collides with the target surface. Therefore, we shifted focus to the volume of fluid (VOF) method and developed a novel method that avoids the commonly used under relaxation and therefore can use a larger time step. In [14] the resulting width, thickness and shape of applied material on test plates as a function of time and spraying distance were compared to experiments with excellent agreement.

The outline of the rest of the chapter is as follows: in the next section the different paint shop processes and their mathematical challenges are described with focus on spray painting and sealing. In Sect. 3 we present the simulation framework which in Sect. 4 is validated with measurements for spray painting of a truck cab and the laydown of sealing material on a car. Finally, the last section summarizes the chapter and future work is discussed.

2 The Paint Shop Processes

The manufacturing process in an automotive paint shop is a process chain consisting of several different coating processes, see Fig. 1. In the pretreatment pollutants are removed from the body and the purpose is also to increase the adhesiveness of the

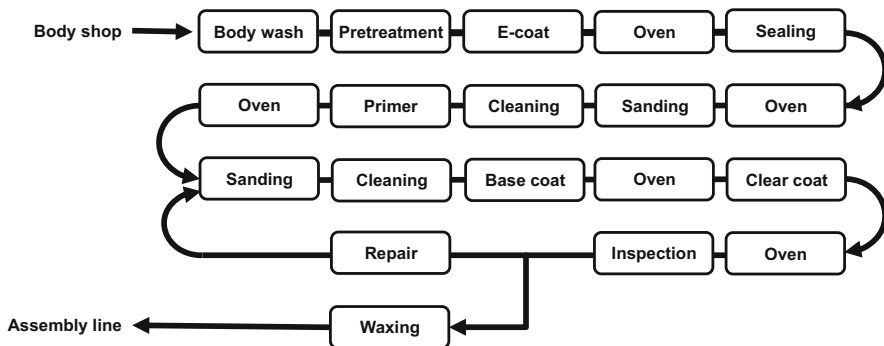


Fig. 1 Process steps in a modern automotive paint shop

paint film later applied. This is followed by electrocoating, which is a cathodic electrodeposition process used for corrosion protection. The grounded body is submerged into a bath with an electrolyte liquid and an electric voltage is applied to a number of anodes positioned in the bath. The resulting electric current causes a transport of the paint material in the bath to the body and a highly resistive layer is formed. The mathematical challenges include the complex multi-phase flow that if simulated can be used to predict the location of air pockets, which prevent the build-up of the corrosion protective layer, and also the fluid access and drainage during the process.

After oven curing of the E-coat layer, sealing and underbody protection material are applied to cover cavities and seams, where moisture otherwise might create a corrosive environment, and also to dampen noise. In the order of 50 m of material is used for a vehicle, see Fig. 2. The sealing material is non-Newtonian and shear thinning meaning that the viscosity decays with increasing shear rate. Due to the complexity of the process characterized by multi-phase and free surface flows, multi-scale phenomena, and large moving geometries, the current situation in the automotive industry is to rely on individual experience and physical validation for improving the sealing process.

The next step is the application of the different paint layers, where the standard is primer followed by base coat and clear coat layers. The most common technique is to use an Electrostatic Rotary Bell Sprayer (ERBS), see Fig. 3. Paint is injected at the center of a rotating bell; the paint forms a film on the bottom side of the bell and is atomized at the edge. The droplets are charged electrostatically and driven towards the target both by shaping air surrounding the rotating bell and by a potential difference in the order of 50–100 kV between paint applicator and target. Also here we are facing a very complex process characterized by multi-phase flow, large moving geometries, multi-scale phenomena and even multi-physics aspects.

The painted layers are cured in ovens, where the heat transfer typically is through convection or IR. The curing process is very energy intensive and in the convective ovens fans direct a hot air flow on the body such that its surface reaches



Fig. 2 A robot cell at Volvo Cars with two sealing robots applying material to the car body (*left*). Examples of interior sealing beads in the front passenger seat area applied by a flat bead sealing nozzle (*right*). Photos copyright: Volvo Cars



Fig. 3 Robotized spray painting of a Volvo truck cab using electrostatic rotary bells. Photo copyright: AB Volvo

a temperature of around 150°C in the clear coat oven and around 75°C in the base coat oven. The large temperature gradients and thin boundary layers make also this part of the paint shop a big challenge for modeling and simulation. Finally, before sending the painted body to the assembly shop, it passes through a visual inspection to ensure structure, gloss and that there are no defects. In addition, properties such as structure, color match and paint thickness are measured on a regular basis.

Although the paint and surface processes are undoubtedly very complex the rapid increase in computational power opens up new possibilities and making modeling and simulation an integrated part of the development would have many advantages. The risk for unforeseen costs and quality problems is reduced by offering the possibility to perform analyses and optimization in the early phases of product and process development and virtual prototyping stimulates industrial innovation. Furthermore, fast and efficient simulation can help to reduce the time required for introduction of new models, reduce the environmental impact and increase quality.

3 Multi-Physics Simulation Framework

The simulation framework consists of the Navier-Stokes solver, IBOFlow (Immersed Boundary Octree Flow Solver), which is based on a finite volume discretization on a Cartesian octree grid that can be dynamically refined and

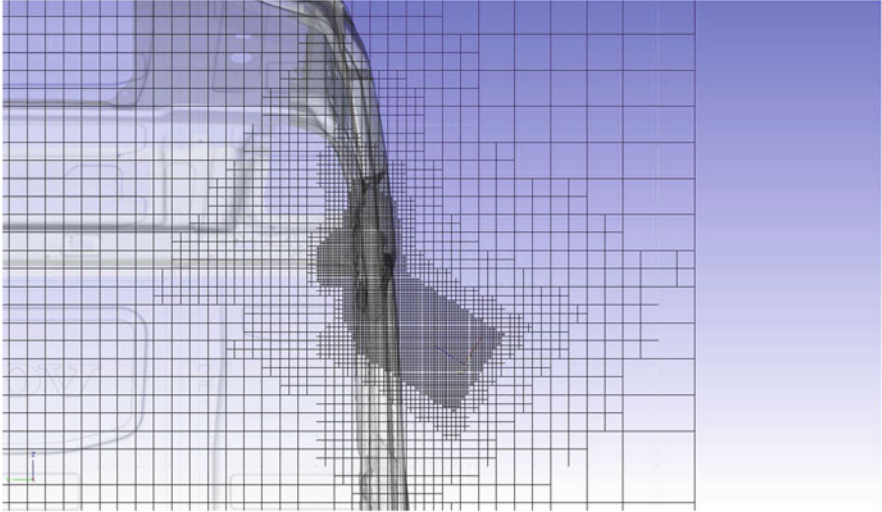


Fig. 4 A side view of the dynamic octree grid around a truck door frame and the paint applicator

coarsened, see Fig. 4 for an example. Unique immersed boundary methods are used to model the presence of objects in the fluid [11]. The electrostatic solver is based on the same discretization framework and immersed boundary conditions are used to set the voltages at the applicator and target geometry. The paint droplets are simulated as Lagrangian particles and their motion is given by the Basset-Boussinesq-Oseen (BBO) equation. The Sundials package is used to efficiently solve the BBO equation for the individual droplets and trace them from applicator to target.

3.1 Flow Solver

An incompressible fluid is modeled by the Navier-Stokes equations,

$$\nabla \cdot \bar{u} = 0, \quad (1)$$

$$\rho_f \frac{\partial \bar{u}}{\partial t} + \rho_f \bar{u} \cdot \nabla \bar{u} = -\nabla p + \mu \nabla^2 \bar{u} + \bar{s}, \quad (2)$$

where \bar{u} is the fluid velocity, ρ_f is the fluid density, p is the pressure, μ is the dynamic viscosity and \bar{s} is the droplet source term. The Navier-Stokes equations are discretized with the finite volume method on a dynamic Cartesian octree grid, which is automatically generated and allows adaptive grid refinements to follow moving objects. The equations are solved in a segregated way and the SIMPLEC method is used to couple the pressure and the velocity fields [25]. All variables are

stored in a co-located arrangement and the pressure weighted flux interpolation is used to suppress pressure oscillations [16]. The Forward Euler scheme is used for the temporal discretization and an adaptive fluid time step is employed such that the maximum Courant number based on the fluid velocity and the movement of the applicators are restricted. A standard $k - \epsilon$ turbulence model is utilized.

The internal boundary conditions are handled by the hybrid immersed boundary method [11]. In the method the fluid velocity is set to the local velocity of the object with an immersed boundary condition. Extrapolation and mirroring of the velocity close to the boundary are used to formulate an implicit boundary condition which is added to the operator for the momentum equations. The mirroring results in a fictitious fluid velocity field inside the immersed object. Mass conservation is ensured by excluding this velocity field in the discretized continuity equation. A thorough description of the method and an extensive validation can be found in [11].

The two-phase flow in the sealing application is modeled with the volume of fluid (VOF) method, where the local property of the fluid is dependent on the volume fraction. The volume fraction is transported with the local velocity field. To keep the interface between the sealing material and the air sharp a hybrid CICSAM convective scheme is adopted [24]. The sealing material is characterized by a Carreau model, in which the viscosity is dependent on the local shear rate,

$$\mu = (\mu_0 - \mu_\infty) + \left(1 + (\lambda\dot{\gamma})^2\right)^{0.5(N-1)} + \mu_\infty, \quad (3)$$

where μ_0 is the viscosity at zero shear rate, μ_∞ is the viscosity at infinite shear rate, λ is the relaxation time, $\dot{\gamma}$ is the shear rate and N is a power index. The four material parameters are determined from rheological experiments, where the sealing material is sheared for two minutes for shear rates between 1 and 450 s^{-1} , and to capture thixotropic effects, going down from 450 to 1 s^{-1} .

3.2 *Electrostatic Solver*

For an internally charged bell atomizer the electric field generated due to the potential difference between the bell and the grounded target is given by the following Poisson's equation [4]:

$$\nabla^2\phi = -\frac{\rho}{\epsilon}, \quad (4)$$

where ϕ is the potential, ρ is the droplet space charge density and ϵ is the air electrical permittivity. The electric field is given by $\vec{E} = -\nabla\phi$. The electrostatic solver also employs the finite volume method to solve the Poisson's equation. The equation is discretized on a dynamic octree grid and immersed boundary conditions are used to set the voltages at the applicator and target geometry. For the electrostatic solver it is important to refine the octree grid close to the edges of the target

geometry, where the electric fields are large. In general, the flow and electrostatic solvers use different octree grids.

3.3 Droplet Solver

The droplets are treated as point-like Lagrangian particles and by including only the gravity/buoyancy, drag and electrostatic forces the following BBO equation [15] is obtained for the motion of the droplets:

$$\rho_p \frac{d\bar{u}_p}{dt} = (\rho_p - \rho_f) \bar{g} - \bar{u}_r |\bar{u}_r| C_d \frac{\rho_f m_p}{\rho_p 2r_p} + \bar{E} q_p, \quad (5)$$

where ρ_p is the droplet density, \bar{u}_p is the droplet velocity, \bar{u}_r is the relative droplet and fluid velocity, C_d is the drag coefficient, m_p is the droplet mass, r_p is the droplet radius, q_p is the droplet charge and \bar{E} is the electric field. The drag coefficient is evaluated according to [18]

$$C_d = 24 \frac{1 + 0.15 Re_p^{0.687}}{Re_p}, Re_p = \frac{2r_p |\bar{u}_r| \rho_f}{\mu}, \quad (6)$$

where Re_p is the instantaneous droplet Reynolds number. The droplet break-up process is not simulated in this work and since the spray is dilute away from the near bell region droplet-droplet interaction can be neglected.

The dominant charging mechanism for an internally charged applicator is induction, which imparts a charge on the droplets leaving the bell. The droplet charge is assumed to be proportional to the droplet area, which means that the droplet charges are coupled to the droplet size distribution. The area charge density is difficult to measure and is therefore determined empirically by matching the measured paint thickness on the edges in the panel simulations. The fluid velocity and the electric field are interpolated to the location of the respective droplets and the resulting drag force and charge are interpolated back to the fluid grid, generating the droplet source term in (2) and charge density in (4). The BBO equation is solved for each droplet using the CVODE solver in the Sundials software package [6]. The efficient automatic time stepping in Sundials is employed but restricted by a local droplet Courant number of 0.5. In each local time step it is checked if the droplet collides with a triangulated object. This must be done accurately as many droplets travel close to the target before they collide. If the droplet collides with an object the droplet is removed from the droplet solver and stored as an impact in the local triangle. These impacts are then employed in the paint thickness estimation described in [23]. To include the local fluctuations of the fluid velocity field a droplet turbulence model is introduced. Each fluid time step the velocity field is reconstructed by adding a random turbulent noise with time correlated direction and magnitude determined from experimental data.

3.4 *Spray Simulation*

The multi-physics framework for simulation of spray painting consists of the three described components; the air flow solver, the electrostatic solver and the droplet solver. The air flow is coupled with the droplets through the momentum transfer and the drag force. The electrostatic potential is coupled with the droplets through the droplet space charge density and the electrostatic force. Finally, the air flow and the electrostatic potential are weakly coupled through the droplets.

The breakup process is currently not simulated and therefore some near-bell measurements are needed to generate the input to the paint thickness simulations. For the applicator used in the Results section the droplet size distribution, and the air and droplet velocities close to the bell, were measured for typical operating process parameters using a Spraytec RTS 5001 from Malvern Instruments and laser doppler anemometry (LDA), respectively. Based on these measurements so-called brushes are generated for each set of paint process parameters that are used, see [12] for a more thorough description.

In all simulations the paint target is put in the middle of the simulation box, whose size is adapted depending on the geometry and robot path. At the top of the box an inlet boundary condition corresponding to the down draft velocity in the paint booth is set and at the bottom an outlet boundary condition is set. On the other outer boundaries symmetry boundary conditions are employed. The droplets are injected in a plane below the applicator with sizes and velocities sampled from the brushes. The electrostatic potential is put to zero on the grounded target and to the high voltage on the applicator.

The fluid and the electrostatic base grid have cell size 100 mm and is automatically refined four times around the target. Further, the fluid grid is dynamically refined three times around and below the moving applicator, and the electrostatic grid is refined around the applicator. The grid resolution is set to ensure a grid-independent solution; see [12] for a grid dependency study. The time step is automatically determined from the smallest cell size and the flow velocities. The adaptive grid update is determined by the applicator's location.

One coupled multi-physics time step is summarized as:

1. Update position of applicators and target geometry
2. Calculate the adaptive fluid time step and update grid refinements
3. Connect immersed boundaries with octree grid
4. Solve the Navier-Stokes Eqs. (1), (2)
5. Solve the Poisson's equation (4)
6. Calculate the electrostatic field from the potential
7. Interpolate the velocities and the electrostatic field to the octree nodes
8. Inject new droplets
9. Solve the BBO equation (5) for the droplets
 - a. Interpolate velocity and electrostatic field to the droplet positions
 - b. Take a local time step

- c. Calculate impacts
 - d. Add contribution to momentum transfer source terms
 - e. Add contribution to electrostatic space charges
 - f. Iterate until local time equals the fluid time
10. Perform thickness estimation
 11. Iterate

Notice that the fluid solver and the electrostatic solver employ the same time step but each droplet has its own local time step.

3.5 Sealing Laydown Simulation

To perform a sealing laydown simulation the flow solver described in Sect. 3.1 is used with the VOF method to handle the multi-phase flow and the Carreau model to characterize the rheology of the material. The required input is a target geometry, the motion of the active sealing nozzle and pictures of the spray cone for different volume flows.

Due to the high Stoke's number the flow pattern of the sealing material between the nozzle and impact is independent on the flow direction and velocity of the surrounding air. Furthermore, the short application distance implies that gravity has little or no effect on the flow pattern in the air. But, when the material strikes the target's surface it begins to flow and the resulting deposition is highly dependent on the impact angle, material rheology, volume flow and the target's geometry. Therefore, in the air the flow dependent sealing spray pattern is reconstructed from experiments and the fluid flow solver simulates the multi-phase surface flow.

The experimental spray pattern and the corresponding reconstruction for three different volume flows are shown in Fig. 5. The reconstruction is based on ray tracing. The width of the spray pattern depends on both volume flow and distance from the applicator, and is interpolated from the experimental data. With the help of the ray tracing the impact zone is determined and the volume flow, angle and distance dependent impact pattern is predicted with high accuracy. A similar approach could be used for the other common nozzle type, the hollow-cone.

At the fluid cells corresponding to the impact positions close to the target, the material is inserted as a volumetric material source in the fluid solver and the velocity is set by immersed boundary conditions in the momentum equations. To ensure the no-slip boundary condition on the surface of the target the hybrid immersed boundary condition is adopted [11]. Furthermore, a reference pressure is set in the fluid domain.

To determine the rheology, i.e. the dependency between the viscosity and the shear rate, of the sealing material tests are performed in a rotary rheometer with a parallel disc. The Carreau model in (3) with parameters, viscosity at zero shear rate $\mu_0 = 886.23 \text{ Pa s}$, viscosity at infinite shear rate $\mu_\infty = 1.3418 \text{ Pa s}$, relaxation time $\lambda = 0.01 \text{ s}$, and power index $N = -200$, could be fitted to the experimental

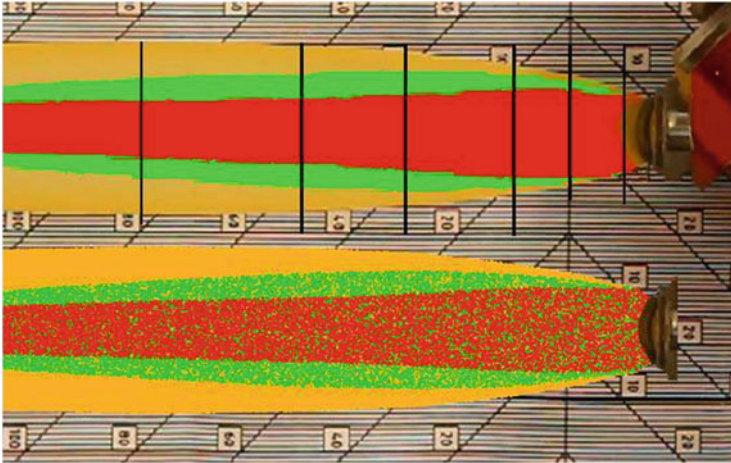


Fig. 5 *Top*: Picture of the static spray pattern for three different volume flows (*red* 10 ml/s, *green* 15 ml/s and *orange* 20 ml/s). *Bottom*: Simulated/reconstructed flow pattern for the different volume flows visualized with virtual droplets

data of the sealing material used at Volvo Cars in Torslanda, Gothenburg, with good agreement, see [14] for further details. The density of the sealing material, $\rho = 1.080 \text{ kg/m}^3$.

Therefore, one time step in the sealing laydown simulations can be summarized as:

1. Update position of applicators and target geometry
2. Calculate the adaptive fluid time step and update grid refinements
3. Connect immersed boundaries with octree grid
4. Solve the Navier-Stokes equations (1, 2)
5. Transport volume fraction with the local velocity
6. Interpolate the material volume fraction to the octree nodes
7. Reconstruct the surface of the sealing material on the target by a marching cube algorithm
8. Iterate

4 Results

4.1 Spray Painting

To validate the software the exterior painting of a Volvo truck cab, see Fig. 6, was simulated with the process settings used in production and the results were compared to measurements. The validation was performed in two steps. First,

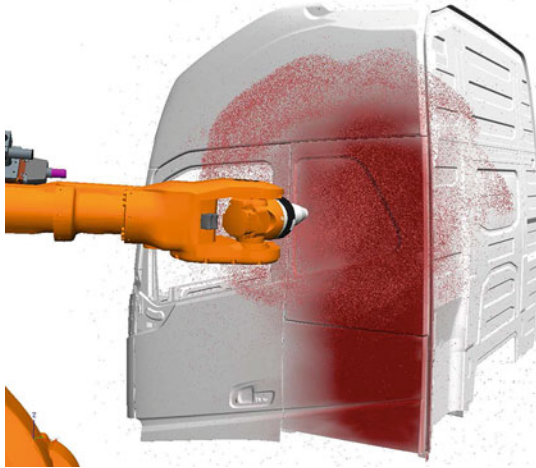


Fig. 6 Simulation of robotized spray painting of a Volvo truck cab in IPS Virtual Paint

Table 1 The brushes and the corresponding process parameters used for painting the truck cab

Name	Paint flow (ml/min)	Shape air 1 (l/min)	Shape air 2 (sl/min)	Rotation speed (rot/min)	High voltage (V)
Brush 4	420	240	100	40,000	70,000
Brush 23	390	100	400	40,000	10,000
Brush 26	370	240	100	40,000	50,000
Brush 39	200	300	100	40,000	50,000
Brush 1	370	240	100	40,000	70,000
Brush 2	390	240	100	40,000	70,000
Brush 13	450	240	100	40,000	70,000
Brush 15	470	240	100	40,000	70,000
Brush 16	150	300	100	40,000	50,000
Brush 18	220	240	100	40,000	70,000
Brush 19	355	240	100	40,000	50,000
Brush 22	350	240	100	40,000	10,000
Brush 24	260	240	100	40,000	70,000
Brush 49	330	240	100	40,000	70,000

The first four brushes have also been used to paint rectangular panels

brushes were validated and fine-tuned by painting rectangular panels with three identical strokes. Thickness measurements on the panels were available for four of the brushes, see Table 1. The additional twelve brushes were generated by interpolation and extrapolation from the first four brushes. Secondly, these brushes were used to simulate the painting of the truck cab.

The applicator used was a Dürr Ecobell 2 internally charged atomizer. Instead of the base coat and clear coat layers, see Fig. 1, Volvo Trucks uses a top coat paint

as the final layer. The thickness of the top coat layer is what we validate here. The robot speed and tool center point (TCP) distance varied slightly along the robot path of the truck cab, but were roughly 470 mm/s and 200 mm, respectively. The spray booth downdraft air velocity was set to -0.42 m/s.

The painted panels were of rectangular shape and size 200×1000 mm. The paint thickness of each panel was measured along three lines in the middle of the panel parallel to its long edge. The thickness value of each point in each line was furthermore an average of three probe measurements. Near bell input data for the simulations was generated from measurements, as described in Sect. 3.4. The input was fine-tuned for the process conditions used in the present campaign to generate accurate thickness results on the panels. In Fig. 7 a comparison between simulation results and measurements are shown.

Figure 3 illustrates the spray painting of a Volvo truck cab, whereas Fig. 6 shows the spray painting in the IPS Virtual Paint simulation software. The coverage of the robot path and the usage of each brush can be seen in Fig. 8. Note, that the major part of the truck cab was painted with brushes identical to, or in very close resemblance with, the brushes validated on the panels.

Painting of the interior of the truck cab was carried out before the exterior painting. The interior painting yields some overspray on the exterior of the cab as

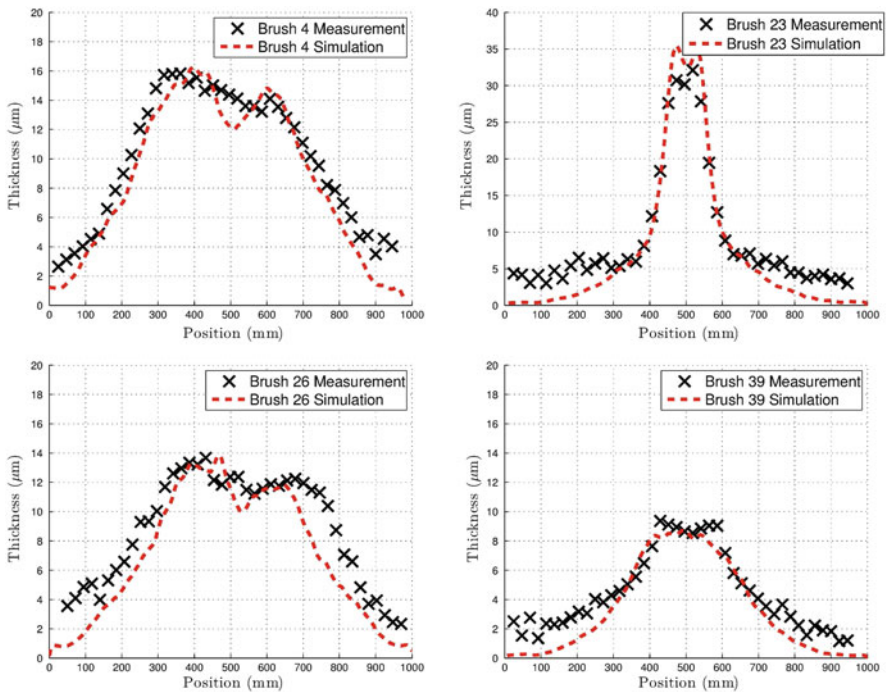


Fig. 7 Measured and simulated paint thicknesses for the first four brushes shown in Table 1 in order of ascending brush number. Note that the thickness is more than twice as large for Brush 23

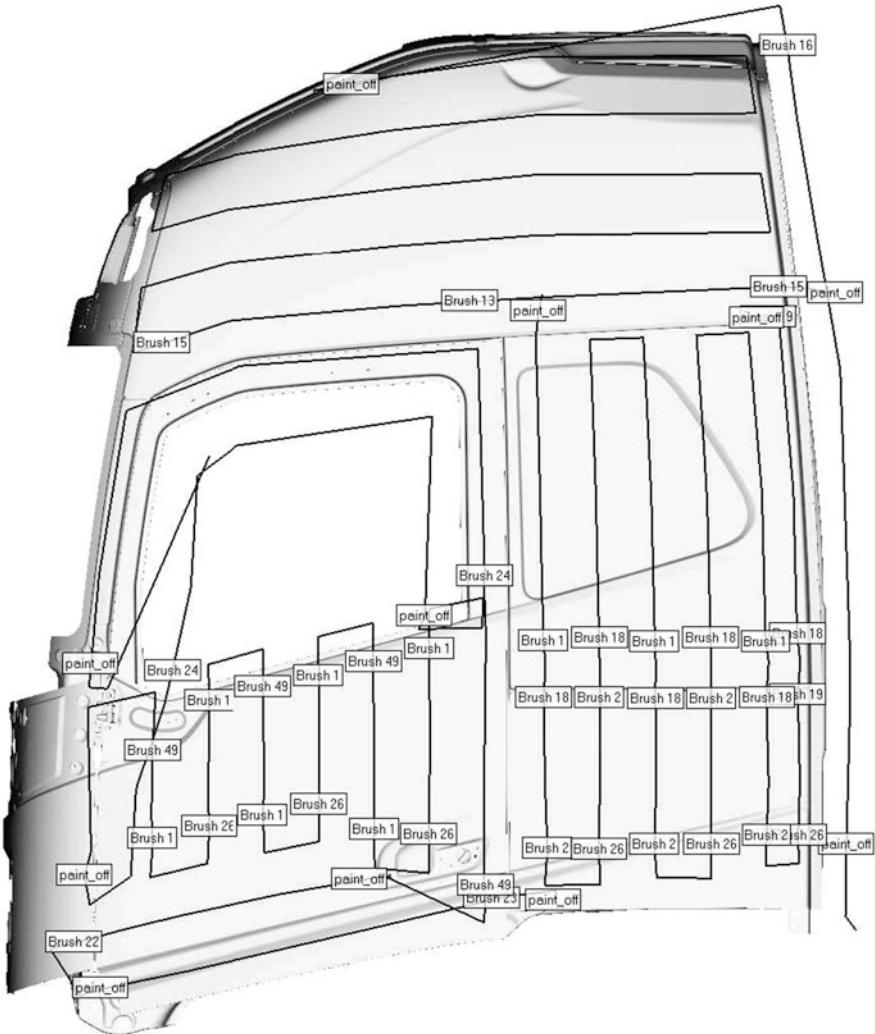


Fig. 8 The robot program, including brush triggers, used to paint the truck cab

can be seen to the right in Fig. 9. Therefore the parts affected by overspray have been excluded in the validation. Validation points were taken from three different regions of the truck cab exploiting different brushes and geometrical features, namely the roof side, the cab side and the cab door. The regions as well as the exact points chosen for validation can be seen to the left in Fig. 9.

The validation results for the cab roof side, cab side and cab door are presented in Figs. 10, 11 and 12, respectively. Notice in particular the agreement to overall thickness levels of the different parts. And furthermore the simulation software's

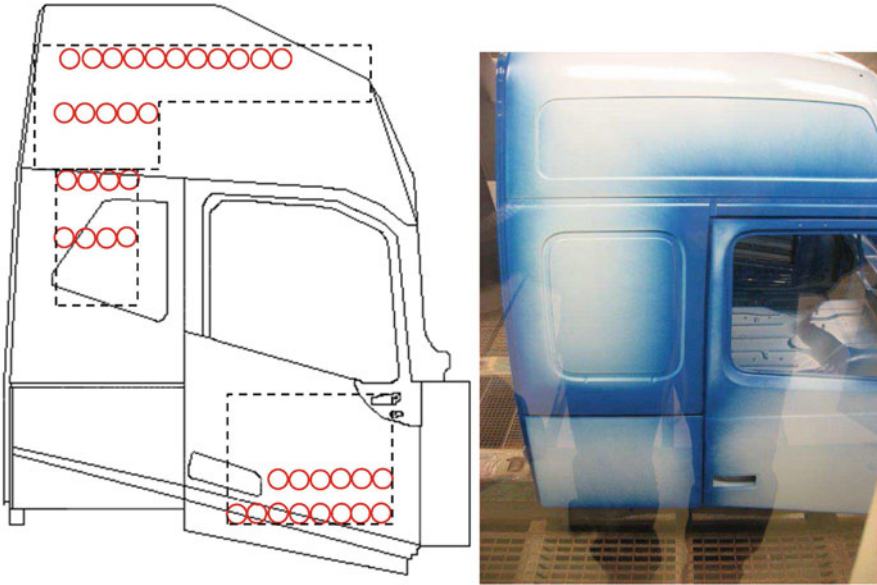


Fig. 9 The measurement points selected for validation, the dashed regions represent all the data available for validation (*left*). The *blue regions* on the cab surface are affected by overspray from interior painting and are therefore not included in the validation (*right*)

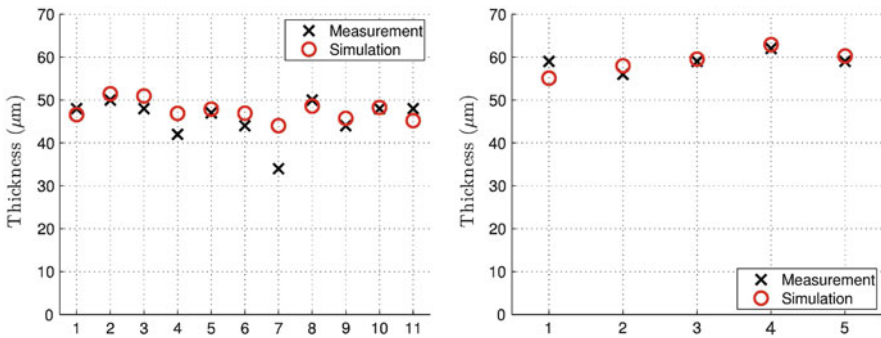


Fig. 10 Thickness results for the two lines on the roof side of the truck cab, see Fig. 9. The top line is shown to the *left* and the *bottom line* is shown to the *right*

accurate prediction of dips and peaks in thicknesses caused by features of the surface and robot paths, such as the dip and peak following a change in surface curvature on the lower part of the door in Fig. 12. The standard deviation of the difference between simulation results and measurements, the root mean square (RMS) difference and the RMS difference normalized to the mean measured thickness are presented in Table 2. The difference relative to the measured thickness is 4–7% for each region. AB Volvo has observed that the small process variations

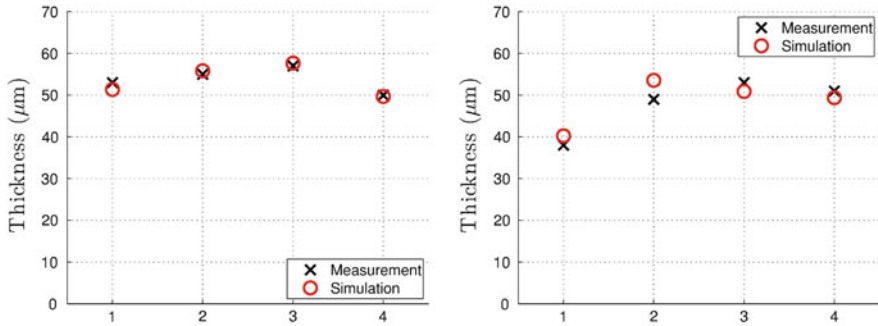


Fig. 11 Thickness results for the two lines on the cab side, see Fig. 9. The *top line* is shown to the left and the *bottom line* is shown to the right

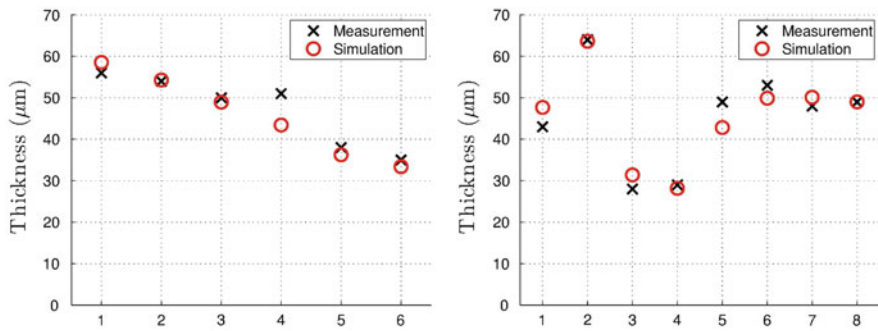


Fig. 12 Thickness results for the two lines on the cab door, see Fig. 9. The *top line* is shown to the left and the *bottom line* is shown to the right

Table 2 Numerical measures of the difference between experiments and simulations

Data set	Std	RMS	RMS rel
Side	2.268	2.146	0.042
Door	3.405	3.350	0.073
Roof	3.222	3.368	0.068
Total	3.166	3.143	0.064

The RMS rel is the RMS difference relative to the mean measured thickness

that are inevitable in practice, e.g. the position of the cab in the paint booth which is allowed to vary at most 20 mm and slightly uneven paint flow, and the error in the measurement probe, cause a thickness difference of 3–4 μm for truck cabs painted on the same line with the same color. Hence, this means that the difference between the simulations and measurements is in the same order as the thickness variation observed in production.

The speed of the simulations of the truck cab is remarkable. The full robot program for painting the cab (i.e. side, door and roof side) is performed by one robot and takes approximately 75 s in reality. The simulation run over night on a descent

desktop computer, and takes approximately 18 h on a six core CPU, single GPU work station. The fluid time step was set to 0.005 s. The number of computational cells for the flow solver was about 350,000 and approximately the same for the electrostatic solver, 300,000 computational paint particles per second were injected and their maximum lifetime were 2 s.

4.2 Sealing

To validate the simulation results obtained with the IPS Virtual Paint software they were compared to measurements for four sealing beads applied to a plate with different process conditions and a production bead on a Volvo V40 car. For the plates the volume flow, nozzle to plate distance (TCP distance) and nozzle velocity are stated in Table 3. In Fig. 13 the experimental and simulated beads are shown and in Fig. 14 the average bead widths are compared with excellent agreement.

The final verification was performed on a Volvo V40 car produced in the factory in Gent. An interesting bead in production in the rear wheel house area was chosen. In Fig. 15 the scanned experimental bead (yellow) is compared with the simulated one (green). Notice the good agreement in width, thickness and surface structure.

Table 3 The volume flow, nozzle to plate distance (TCP distance) and nozzle velocity for the four validation beads

Bead number	Volume flow [ml/s]	TCP distance [mm]	Velocity [mm/s]
1	30	35	400
2	40	35	400
3	50	35	400
4	30	50	400

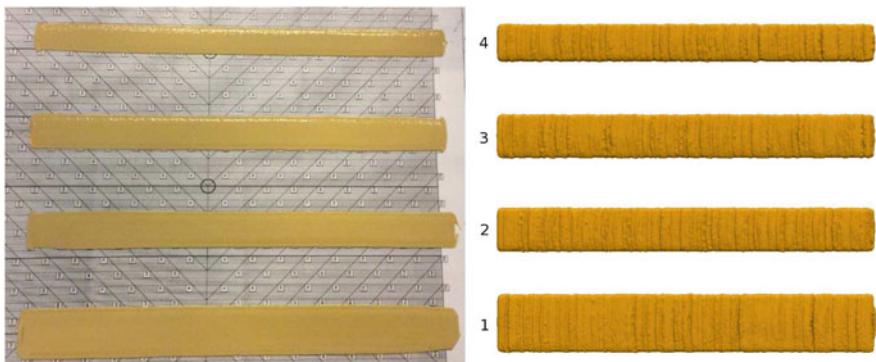


Fig. 13 The four experimental beads (*left*) and the corresponding simulated beads (*right*)

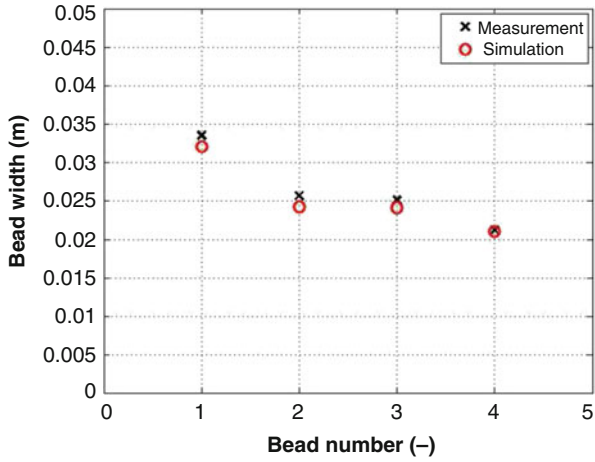


Fig. 14 Comparison between the measured and simulated bead widths

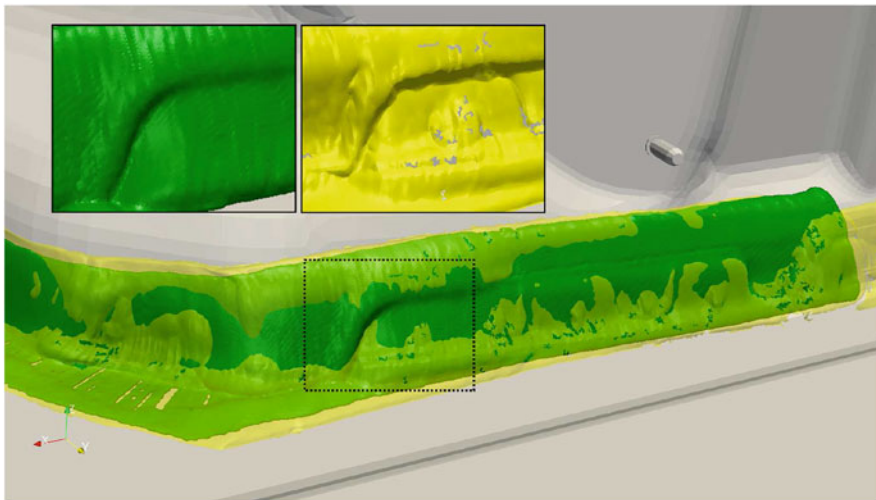


Fig. 15 Sealing bead verification in the rear wheel house area of a Volvo V40 car. The scanned experimental bead is shown in yellow and the simulated bead in green

5 Summary

In this paper a novel framework for simulation of electrostatic spray painting and sealing is presented. A systematic validation is performed including both plates, a car and a truck cab. Overall the agreement between simulations and measurements is very good. The framework is integrated in the IPS (Industrial Path Solutions) software for automatic path planning. The very efficient implementation gives a

major improvement of computational speed compared to earlier approaches and makes it possible to perform detailed simulations in just a few hours on a standard computer. This fact makes it feasible to include such detailed simulations in the production preparation process and offline programming of the paint robots. A large part of the production preparation process can therefore be performed in the computer, which allows the engineers to replace physical prototypes with virtual ones to shorten the lead time in product development, reduce the cycle time in production, and avoid future unforeseen technological and environmental problems that can be extremely costly if they are discovered at the end of the production line, or even worse by the customer.

The near future work on the simulation of electrostatic spray painting includes an extension to externally charged bells, where the foundation has been set with a novel finite volume based solver for three species negative corona discharge simulations [9, 26]. In the longer perspective the aim is to automate the product preparation by automatically generating robot paths and process conditions that guarantee a certain wanted paint coverage. This is obviously a big step compared to the current status and it is a step that cannot be taken without developing novel mathematical methods, algorithms and tools.

For the sealing application an extension to laydown of glue is ongoing [22]. The main difference compared to sealing is that glue has a more complex rheology. The complexity is increased by also considering the hemming process. Hemming is combined with glue to fixate inner and outer structures such as e.g. doors, hoods and trunk lids, and create sealed joints that prevent corrosion. Its usage will most likely increase in importance in automotive production since the method has the potential to join different materials in future lightweight structures. The simulation of the hemming process requires modeling of the complex fluid-structure interaction, where the glue interacts with the folded structure.

Acknowledgements This work was supported in part by the Swedish Governmental Agency for Innovation Systems, VINNOVA, through the FFI Sustainable Production Technology program, and in part by the Sustainable Production Initiative and the Production Area of Advance at Chalmers. The support is gratefully acknowledged. The test geometries used in the measurement campaign were provided by Volvo Car Corporation and AB Volvo.

References

1. Carlson, J.S., Spensieri, D., Söderberg, R., Bohlin, R., Lindkvist, L.: Non-nominal path planning for robust robotic assembly. *J. Manuf. Syst.* **32**, 429–435 (2013)
2. Domnick, J., Scheibe, A., Ye, Q.: The simulation of the electrostatic spray painting process with high-speed rotary bell atomizers. Part I: direct charging. *Part. Part. Syst. Charact.* **22**, 141–150 (2005)
3. Domnick, J., Scheibe, A., Ye, Q.: The simulation of the electrostatic spray painting process with high-speed rotary bell atomizers. Part II: external Charging. *Part. Part. Syst. Charact.* **23**, 408–416 (2006)

4. Ellwood, K.R.J., Braslaw, J.: A finite-element model for an electrostatic bell sprayer. *J. Electrostat.* **45**, 1–23 (1998)
5. Hermansson, T., Bohlin, R., Carlson, J.S., Söderberg, R.: Automatic assembly path planning for wiring harness installations. *J. Manuf. Sys.* **32**, 417–422 (2013)
6. Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S.: SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.* **31**, 363–396 (2005)
7. Huang, H., Lai, M.-C., Meredith, W.: Simulation of spray transport from rotary cup atomizer using KIVA-3V. In: *Proceedings of the 10th International KIVA Users Group Meeting, Detroit* (2000)
8. Im, K.-S., Lai, M.-C., Yu, S.-T.J., Matheson, R.R. Jr.: Simulation of spray transfer processes in electrostatic rotary bell sprayers. *J. Fluids Eng.* **126**, 449–456 (2004)
9. Johnson, T., Jakobsson, S., Wettervik, B., Andersson, B., Mark, A., Edelvik, F.: A finite volume method for electrostatic three species negative corona discharge simulations with application to externally charged powder bells. *J. Electrostat.* **74**, 27–36 (2015). doi:10.1016/j.elstat.2014.12.009
10. Mark, A., van Wachem, B.G.M.: Derivation and validation of a novel implicit second-order accurate immersed boundary method. *J. Comput. Phys.* **227**, 6660–6680 (2008)
11. Mark, A., Rundqvist, R., Edelvik, F.: Comparison between different immersed boundary conditions for simulation of complex fluid flows. *Fluid Dyn. Mater. Proc.* **7**, 241–258 (2011)
12. Mark, A., Andersson, B., Tafuri, S., Engström, K., Söröd, H., Edelvik, F., Carlson, J.S.: Simulation of electrostatic rotary bell spray painting in automotive paint shops. *Atomization Sprays* **23**, 25–45 (2013)
13. Mark, A., Svenning, E., Edelvik, F.: An immersed boundary method for simulation of flow with heat transfer. *Int. J. Heat Mass Transf.* **56**, 424–435 (2013)
14. Mark, A., Bohlin, R., Segerdahl, D., Edelvik, F., Carlson, J.S.: Optimisation of robotised sealing stations in paint shops by process simulation and automatic path planning. *Int. J. Manuf. Res.* **9**, 4–26 (2014)
15. Maxey, M.R., Riley, J.J.: Equation of motion for a small rigid sphere in a nonuniform flow. *Phys. Fluids* **26**, 883–889 (1983)
16. Rhie, C.M., Chow, W.L.: Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA JI* **21**, 1527–1532 (1983)
17. Rundqvist, R., Mark, A., Edelvik, F., Carlson, J.S.: Modeling and simulation of viscoelastic fluids using Smoothed Particle Hydrodynamics. *Fluid Dyn. Mater. Proc.* **7**, 259–278 (2011)
18. Schiller, L., Neumann, A.: Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung. *Verein Deutscher Ingenieure* **77**, 318–320 (1933)
19. Segeborn, J., Segerdahl, D., Ekstedt, F., Carlson, J.S., Andersson, M., Söderberg, R.: An industrially validated method for weld load balancing in multi station sheet metal assembly lines. *J. Manuf. Sci. Eng.* **136**, 011002 (2013)
20. Shellshear, E., Tafuri, S., Carlson, J. S.: A multi-threaded algorithm for computing the largest non-colliding moving geometry. *Comput.-Aided Des.* **49**, 1–7 (2014)
21. Streitberger, H.-J., Dössel, K.-F.: *Automotive Paints and Coatings*. Wiley-VCGH, Weinheim (2008)
22. Svensson, M., Mark, A., Edelvik, F., Kressin, J., Bohlin, R., Segerdahl, D., Carlson, J.S., Wahlborg, P.-J., Sundbäck, M.: Process simulation and automatic path planning of adhesive joining. In: *Procedia CIRP, 6th CIRP Conference on Assembly Technologies and Systems (CATS), Gothenburg*, vol. 44, pp. 298–303 (2016)
23. Tafuri, S., Ekstedt, F., Carlson, J.S., Mark, A., Edelvik, F.: Improved spray paint thickness calculation from simulated droplets using density estimation. In: *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Chicago*, vol. 2, pp. 339–347 (2012)
24. Ubbink, O.: Numerical prediction of two fluid systems with sharp interfaces. PhD thesis, Department of Mechanical Engineering, Imperial College of Science, London (1997)

25. van Doormaal, J.P., Raithby, G.D.: Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numer. Heat Transfer* **7**, 147–163 (1984)
26. Wettervik, B., Johnson, T., Jakobsson, S., Mark, A., Edelvik, F.: A domain decomposition method for three species modeling of multi-electrode negative corona discharge – with applications to electrostatic precipitators. *J. Electrostat.* **77**, 139–146 (2015). doi:10.1016/j.elstat.2015.08.004
27. Ye, Q., Shen, B., Tiedje, O., Domnick, J.: Investigations of spray painting processes using a airless spray gun. *J. Energy Power Eng.* **7**, 74–81 (2013)

Hot Blade Cuttings for the Building Industries

David Brander, Andreas Bærentzen, Anton Evgrafov, Jens Gravesen, Steen Markvorsen, Toke Bjerger Nørbjerg, Peter Nørtoft, and Kasper Steenstrup

Abstract The constructions of advanced architectural designs are presently very labour intensive, time consuming, and expensive. They are therefore only applied to a few prestige projects, and it is a major challenge for the building industry to bring the costs down and thereby offer the architects more variability in the (economically allowed) designs—i.e., to allow them to think out of the box. To address this challenge The Danish National Advanced Technology Foundation (now Innovation Fund Denmark) is currently supporting the BladeRunner project that involves several Danish companies and public institutions. The project aims to reduce the amount of manual labour as well as production time by applying robots to cut expanded polystyrene (EPS) moulds for the concrete to form doubly curved surfaces. The scheme is based upon the so-called Hot Wire or Hot Blade technology where the surfaces are essentially swept out by driving an Euler elastica through a block of EPS. This paper will be centered around the mathematical challenges encountered in the implementation of this idea. Since the elastica themselves are well known and described in the works of Euler et al. already in eighteenth century, these new challenges are mainly concerned with the rationalization of the architects' CAD drawings into surfaces that can be created via this particular sweeping and cutting technology.

D. Brander • A. Bærentzen • J. Gravesen • S. Markvorsen (✉) • K. Steenstrup
DTU Compute, Kongens Lyngby, Denmark
e-mail: dbra@dtu.dk; janba@dtu.dk; jgra@dtu.dk; stema@dtu.dk; khor@dtu.dk

A. Evgrafov
NTNU, Department of Mathematical Sciences, Trondheim, Norway
e-mail: anton.evgrafov@math.ntnu.no

T.B. Nørbjerg
Edlund A/S, Valby, Denmark
e-mail: toke.norbjerg@edlund.dk

P. Nørtoft
Aqubiq, Kongens Lyngby, Denmark
e-mail: peter.nortoft@aqubiq.com

1 The Need for Low Cost Procedures

A recurring theme in the architectural industry of today is a conflict between the design ambitions of the architect and the economic realities of fabrication processes. The desire to create unique and attractive designs, often motivated by the competitive industry climate, leads to the use of curved geometries and bespoke elements that can be conceived easily within modern CAD systems, but, in reality, are prohibitively expensive to build. This results in compromises at the so-called *rationalization* stage, where the design is adjusted within an engineering context for production purposes. A typical example is where the desired shape of a building leads to panels (or some other element) of perhaps 200 different shapes. Consultation with fabrication contractors then reveals that dramatic cost reductions can be achieved if the design is adjusted so that only 20 unique elements are used, with repetition, instead of the 200. Finally, budgetary considerations force a compromise of the original design.

The present project addresses this issue, in particular within the domain of the production of formwork for concrete constructions. The shape of the surface of a facade or other element is produced—possibly on location—in negative in several pieces of (easily transported and packed) expanded polystyrene (EPS) foam that is then used as a mould for concrete (in situ) casting. EPS can also be used in positive shape production for some applications by applying a coating and retaining the EPS as a structural element. For curved surfaces the currently available technology for shaping the EPS is computer numerically controlled milling, a slow, and therefore expensive, process. The *BladeRunner* project, supported by the Innovation Fund Denmark, is presently developing new processes, *robotic Hot Wire/Blade cutting*, for carving shapes out of EPS using a robotically controlled heated wire or blade. The technology is projected to reduce production time of architectural formwork by a factor of over 100, and to bring the cost of production for advanced shapes into the domain of financial feasibility.

2 Principles of Hot Blade Cuttings

The essential principle of both Hot Wire and Hot Blade cutting is very simple. A heated wire or blade, either of which we may think of as a “blade”, is moved relative to a block of EPS, carving out a surface through the block (Fig. 1). Either the block or the blade, or both, are controlled by a robot. For definiteness, we regard the block as fixed and the blade as moving.



Fig. 1 Robotic Hot Wire cutting in Odense, Denmark

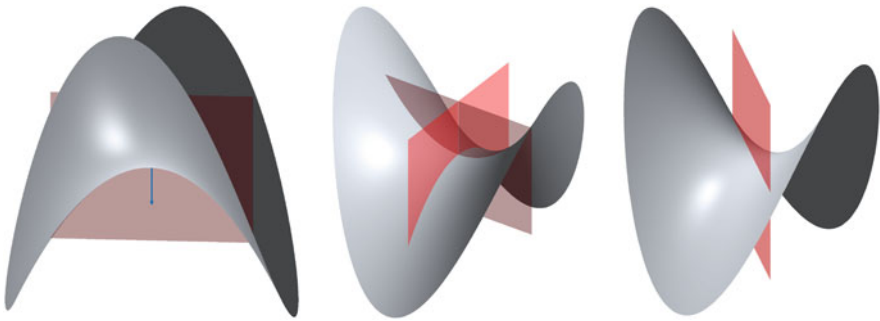


Fig. 2 A saddle surface. *Left*: The normal curvature defined by this intersection curve is positive if the downward pointing surface normal is chosen. *Middle*: The planes defining the principal curvatures at the center. *Right*: This normal section is a straight line; the normal curvature is zero in this direction

2.1 Hot Wire Cutting and Its Limitations

For the wire technology, the wire is held tight, forming a straight line, and thus sweeps out a *ruled surface*. This technology is limited in its ability to approximate general freeform surfaces. This can be seen by considering the *Gaussian curvature* of a surface, which is defined as follows: through any point p on a surface a curve is obtained by intersecting the surface with a plane perpendicular to the tangent plane at that point (Fig. 2). The *normal curvature* associated to the tangent direction of this curve is the curvature of this curve at the point p , with the sign determined by a fixed choice of surface normal vector (Fig. 2, left). The maximum and minimum values obtained from all possible tangent directions at p are called the *principal curvatures*, κ_1 and κ_2 , and their product is the Gaussian curvature $K = \kappa_1\kappa_2$. In the

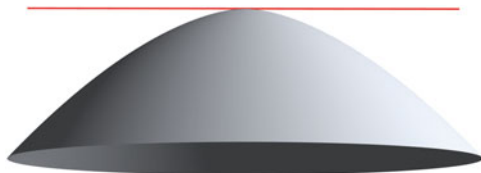


Fig. 3 At a point of positive Gaussian curvature the surface is bowl-shaped. No straight line tangent to the surface can approximate a curve in the surface to more than first (tangential) order

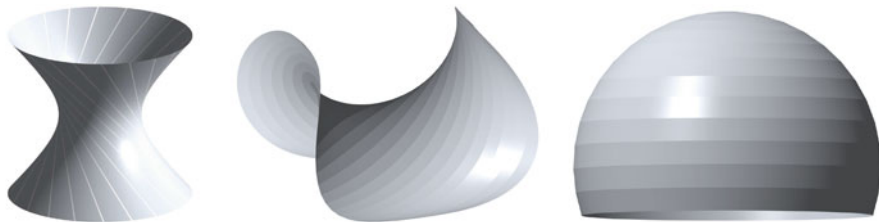


Fig. 4 *Left to right*: The hyperboloid (a ruled surface with automatic tangent matching along adjacent strips); approximation of a negatively curved surface by *strips* of ruled surfaces; Ruled strip approximation of a positively curved surface

saddle surface shown at Fig. 2, the principal curves bend in opposite directions away from the tangent plane and so κ_1 and κ_2 have opposite signs and $K < 0$.

If the Gaussian curvature is positive, then κ_1 and κ_2 at p have the same sign, and any other tangent direction at p has normal curvature κ_n with $\kappa_1 \geq \kappa_n \geq \kappa_2$. Therefore κ_n cannot be zero in this case. Now for an arbitrary arc-length parameterized curve γ in the surface the acceleration vector decomposes as $\gamma''(s) = \kappa_g(s)v(s) + \kappa_n(s)N(s)$, where N is the surface normal, and κ_n is the normal curvature in the direction of γ' . It follows that, if $\kappa_n \neq 0$, then the acceleration is non-zero and thus the curve cannot be a straight line. On the other hand, a ruled surface is defined to be a surface swept out by a smoothly varying family of straight lines: through every point of a ruled surface there is a straight line lying in the surface. Therefore, by the discussion above, a ruled surface cannot have positive Gaussian curvature; moreover, there is no chance of obtaining a good local approximation for a positively curved surface by a ruled surface (Fig. 3).

Figure 4 shows (center) an approximation of a negatively curved surface by ruled strips which can be realized by repeated hot wire cuttings. The ruling directions are chosen to be close to the *asymptotic directions*, namely directions where the normal curvature κ_n is zero. However, even for negatively curved surfaces, it is in general not possible to obtain a tangent continuous approximation—the tangent planes do not match along adjacent strips.

We refer to Sect. 7 for a concrete modern example of a relevant surface—a skater ramp—which clearly displays all curvatures and thence also the production challenges that we are addressing in this work.

2.2 Hot Blade Cutting

The blade concept is much more general than the wire concept illustrated above: the end points and the *tangents* at the endpoints of the blade (center-)curve can vary relative to each other during the sweeping. We will assume, however, that the curve lies in a plane, that is, that the end tangents are co-planar. This restriction makes the mechanical implementation of the process easier, both in terms of choosing the cross-sectional shape of the blade design and allowing for the possibility that only one edge of the physical blade is heated, as illustrated to the right in Fig. 8 below.

An elastic rod, of a fixed length and with end points and end tangents at a given position, assumes the shape of an Euler elastica (discussed below). These curves are well understood mathematically and are given in terms of elliptic functions. We refer to Sect. 3 below for a brief outline of the parametric representation of the family of all planar elastica.

In order to apply either of these technologies to a given CAD design, a *rationalization* of the relevant surface is necessary: the surface must first be *segmented* into pieces, each of which can be approximated within a given tolerance by a surface swept out by curves of the relevant type (lines or a family of elastica). Next, each segment is foliated by curves each of which is approximated by a curve of the type in question. Finally, the data for producing these curve sweepings is given to the robot control software.

Methods for rationalization for Hot Wire cutting have been given already in the literature (see below). Therefore, in this article, the rationalization project we are concerned with consists of both developing a segmentation algorithm for blade-cut surfaces, and an algorithm for approximating arbitrary spline curves by Euler elastica.

2.3 Previous Related Works

Pottmann and Flörey [5] developed a ruled surfaces segmentation algorithm using the fact that on ruled surfaces one of the asymptotic directions at a point must be tangent to the ruling, giving natural candidates for the ruling direction in the surface to be approximated. As such, this segmentation strategy does not generalize to the case of hot *blade* cutting, therefore a new strategy must be developed.

For the Hot Blade technology, some work has been done in the late 1990s to the early 2000s by a group at Delft: see [4, 6] and associated references. The use of the Hot Blade technology there is somewhat different, as the aim is to produce 3-dimensional solid rapid prototype models from EPS via a so-called “thick-layered fabrication” process. The solid is built up by stacking many thick slices, and the curved surface that needs to be cut is only a narrow strip around the boundary of each slice. Therefore, the segmentation problem is completely different from the surface segmentation problem that will apply to the BladeRunner process.

The work of the Delft group is concentrated on approximating the blade shape and algorithms for tool positioning. The approach they use for approximating the blade shape is to apply a numerical method to minimize the bending energy. Below we will use a different approach that takes advantage of the known analytic solutions for this problem to give an explicit parameterization of the space of solutions. This allows us to move easily in the space of solutions, calculate gradients, and use standard optimization packages to find an elastic curve that approximates an arbitrary given curve.

3 The Euler Elastica

We describe here a parameterization of the space of planar elastic curve segments. More details of this parameterization and further references can be found in [3]. An introduction to the theory of elastic curves, with historical references, can be found in [8]. Other works on the topic of elastic curves as splines are [1, 2, 10, 11].

3.1 The Euler-Lagrange Equation

We give here a brief derivation of the differential equation that determines the solutions to the elastica problem. The reader unfamiliar with the calculus of variations could take this derivation for granted and proceed directly to the solutions given in the next subsection. Let $\gamma : [0, \ell] \rightarrow \mathbb{R}^2$ be a plane curve segment parameterized by arc-length, and define an angle function $\theta(s)$ by $\dot{\gamma}(s) = (\cos \theta(s), \sin \theta(s))$. A curve segment of length ℓ starting at (x_0, y_0) and ending at (x_ℓ, y_ℓ) satisfies

$$x_\ell = x_0 + \int_0^\ell \cos \theta \, ds, \quad y_\ell = y_0 + \int_0^\ell \sin \theta \, ds. \quad (1)$$

Let κ denote the curvature $\theta'(s)$. An *elastica* is a curve that minimizes the *bending energy*

$$\frac{1}{2} \int_0^\ell \kappa(s)^2 \, ds. \quad (2)$$

The equations defining the elastica are obtained from a variational problem: suppose γ is an elastica from (x_0, y_0) to (x_ℓ, y_ℓ) with angle function $\theta(s)$. A smooth variation is given by the family γ_t with angle function $\theta_t(s) = \theta(s) + t\psi(s)$, where ψ is a differentiable function with $\psi(0) = \psi(\ell) = 0$. Applying the method of Lagrange multipliers we find that, if γ minimizes the energy among such curves, then the

angle function θ satisfies:

$$\frac{d^2\theta}{ds^2} + \lambda_1 \sin \theta - \lambda_2 \cos \theta = 0. \tag{3}$$

Setting $(\lambda_1, \lambda_2) = \lambda(\cos \phi, \sin \phi)$, with $\lambda \geq 0$, this becomes $\frac{d^2\theta}{ds^2} + \lambda \sin(\theta - \phi) = 0$. Note that $\lambda = 0$ if and only if κ is constant, i.e the curve $\boldsymbol{\gamma}$ is either a straight line segment or a piece of a circle. If $\lambda \neq 0$, set

$$\tilde{\boldsymbol{\gamma}}(s) = \sqrt{\lambda} R_{-\phi} \boldsymbol{\gamma} \left(\frac{s}{\sqrt{\lambda}} \right), \quad R_\phi = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}. \tag{4}$$

Then $\tilde{\boldsymbol{\gamma}}$ is a scaled and rotated version of $\boldsymbol{\gamma}$ and thus also an elastica. Its tangent angle is $\tilde{\theta}(s) = \theta(s/\sqrt{\lambda}) - \phi$ and it satisfies the normalized pendulum equation $\frac{d^2\tilde{\theta}}{ds^2} = -\sin \tilde{\theta}$. Hence we conclude that, up to a scaling and rotation of the ambient space, all arc-length parameterized elastica $\boldsymbol{\gamma} : [0, 1] \rightarrow \mathbb{R}^2$, with non-constant curvature κ , can be expressed as:

$$\boldsymbol{\gamma}(s) = \boldsymbol{\gamma}(0) + \int_0^s (\cos \theta(t), \sin \theta(t)) dt \tag{5}$$

where

$$\ddot{\theta} = -\sin \theta. \tag{6}$$

3.2 The Space of Elastic Curve Segments

We now find some suitable parameters to describe the space of elastic curve segments. First, it is well known that the solutions to (6) can be expressed in closed form via the elliptic functions sn, cn, and dn. These solutions can be found in Love [9]. There are two classes of solution curves: those with inflection points (i.e. points where $\kappa = \dot{\theta} = 0$) and those without inflections. Each class of solutions is a 1-parameter family (Fig. 5).

The *inflectional elastica* starting at $(0, 0)$ with initial angle $\theta(0) = 0$ and $\dot{\theta}(0) > 0$ is

$$\zeta_k(s) = \boldsymbol{\zeta}(s, k) = \begin{pmatrix} 2E(s, k) - s \\ 2k(1 - \text{cn}(s, k)) \end{pmatrix}, \quad \text{where } k = \dot{\theta}(0)/2.$$

A *segment* of such a curve is determined by the value k , a starting point s_0 and a length ℓ . Finally, adding a scaling S , a rotation ϕ and a translation (x_0, y_0) , we have a standard representation $\boldsymbol{\gamma}_{(k, s_0, L, S, \phi, x_0, y_0)} : [0, 1] \rightarrow \mathbb{R}^2$ for a segment of an inflectional

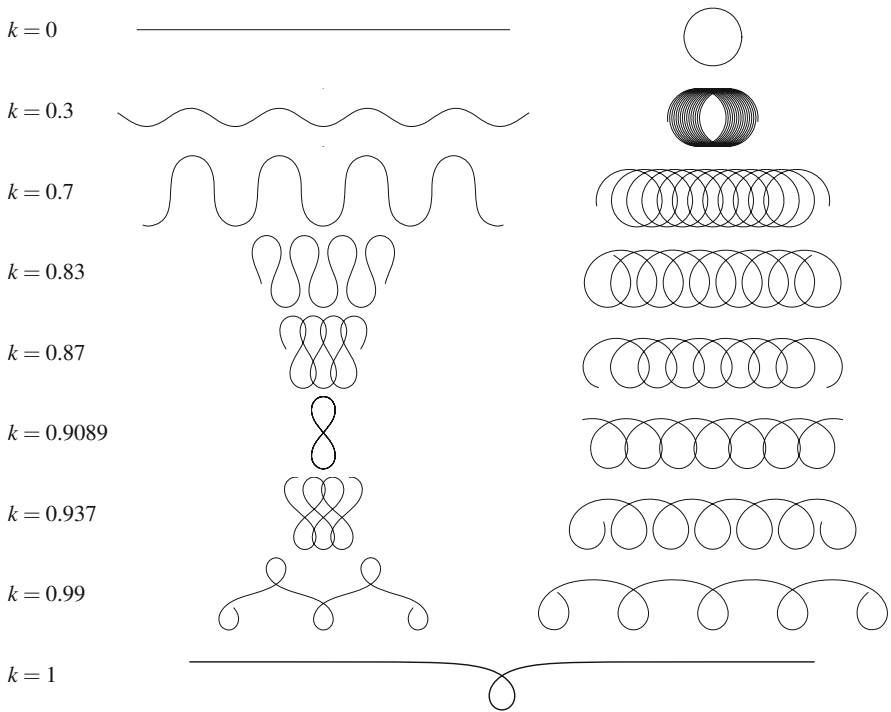


Fig. 5 Euler elastica. *Left*: Inflectional. *Right*: Non-inflectional. The respective elastica—with values of k ranging from 0 at the *top* to 1 at the *bottom*—are plotted

elastic curve:

$$\begin{aligned} \mathcal{Y}_{(k,s_0,\ell,S,\phi,x_0,y_0)}(t) &= SR_\phi(\zeta_k(s_0 + \ell t)) + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \\ &= SR_\phi\left(\begin{pmatrix} 2E(s_0 + \ell t, k) - (s_0 + \ell t) \\ 2k(1 - \text{cn}(s_0 + \ell t, k)) \end{pmatrix}\right) + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \end{aligned}$$

where

$$E(s, k) := \int_0^s \text{dn}^2(\tau, k) d\tau.$$

Note that the arc-length parameter in this representation is $s = S(s_0 + \ell t)$ and not t and that the length is $L = S\ell$.

Similarly, we obtain a standard representation of a *non-inflectional* elastic curve segment:

$$\mathcal{Y}_{(k,s_0,\ell,S,\phi,x_0,y_0)}(t) = SR_\phi \begin{pmatrix} (1 - \frac{2}{k^2})(s_0 + \ell t) + \frac{2}{k} E\left(\frac{s_0 + \ell t}{k}, k\right) \\ \frac{2}{k}(1 - \operatorname{dn}\left(\frac{s_0 + \ell t}{k}, k\right)) \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}.$$

4 Sweeping Surfaces with Euler Elastica

Figures 6 and 7 in this section illustrate examples of surfaces foliated by continuously varying segments of Euler elastica. These examples are constructed by parameterizing the space of planar elastica segments, as in the previous section, choosing a small number of sample curve segments, and then interpolating the data through the parameter space. Hence each surface is swept by a family of planar elastica.

In principle, all of these surfaces could be produced by robotic hot-blade cutting, but there are technical issues that depend on the practical implementation. For example, the surface on the left in Fig. 7 is a surface of revolution, but one end of the profile curve is much closer to the axis of rotation than the other. This means that the blade moves much more slowly on the inner end resulting in too much melting of the EPS. One solution is to segment the surface into several pieces, cut separately. Another is to approximate this surface by some other, non-rotational, family of elastic segments.

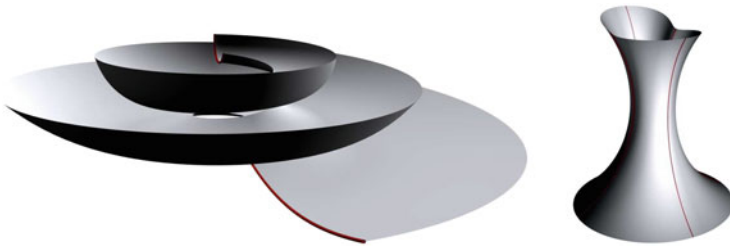


Fig. 6 Examples of surfaces swept by continuously varying elastic curve segments

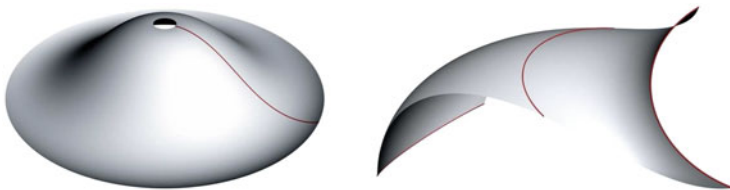


Fig. 7 Two technically problematic situations



Fig. 8 *Left*: A cylindrical rod can cut in any direction that is close to perpendicular to the tangent of the curve. *Right*: A flat (ribbon) blade design (with its hot edge indicated in red) moves well only in the directions given by the flat extension of the blade past the hot edge itself

Yet another restriction arises if a flat blade is used, rather than a cylindrical rod (see Fig. 8). With the flat blade design, the blade is curved in a plane perpendicular to the plane of the blade. If one edge of the blade is heated, the motion of the blade should be roughly in the direction of this edge, that is, approximately perpendicular to the plane of the curve, in order to cut a path through the material. Another way to say this is that the elastic curves should be as close as possible to geodesic curves (which are characterized by $|\kappa_n| = \kappa = \|\gamma''\|$) on the surface under construction. To require that these planar elastic curves are true geodesics would place too large a restriction on the uses of this method; so we apply a tolerance instead. Both surfaces shown in Fig. 6 are reasonable candidates for cutting with a flat blade like the blade to the right in Fig. 8. The surface to the right in Fig. 7 however, would be impractical with the given elastica foliation. The osculating plane spanned by γ' and γ'' of the elastic curve shown is very close to the tangent plane of the surface; thus the hot edge of the blade is pointing out of the surface, and the blade would not be able to progress in the required direction.

5 Approximation by Euler Elastica

In this section we consider the problem of approximating a given planar spline curve $\mathbf{x} : [0, 1] \rightarrow \mathbb{R}^2$ by a planar elastic curve. We present two different approaches to this problem. In the first we try to find the parameters $(k, s_0, \ell, S, \phi, x_0, y_0)$ of the elastica that has the best fit to the curve \mathbf{x} . This is a nonlinear optimization problem, and the final result depends crucially on a good initial guess. The second approach is purely numerical—we model the elastica with a spline on a much finer knot vector than the original curve, and then solve a constrained optimization problem minimizing the elastic energy under the constraint of being within some distance to the original curve.

5.1 Analytic Approach: Finding the Parameters for the Elastica

We describe here the essentials of the gradient driven analytic approach. For full details, see the article [3].

We wish to find the elastic curve segment which most closely resembles the given spline curve \mathbf{x} . We choose to minimize the L_2 -distance between the curves. For a given set of parameters, the elastic curve segment $\boldsymbol{\gamma}_{(k,s_0,\ell,S,\phi,x_0,y_0)}$ is parameterized with constant speed ℓS over the interval $[0, 1]$. The spline curve is itself also defined on $[0, 1]$, but its speed is not necessarily constant. Since the L_2 -norm compares points at corresponding parameter values, we need to reparameterize either the spline or the elastica for the L_2 -distance to be a good measure of the curves' resemblance to each other. The simplest way is to reparameterize the elastica using the arc length s of the spline which can be calculated as

$$s(t) = \int_0^t \frac{ds}{dt} d\tau = \int_0^t \|\mathbf{x}'(\tau)\| d\tau, \quad (7)$$

and the length of the spline is then $L = s(1)$. We now consider the minimization problem

$$\text{minimize}_{k,s_0,\ell,S,\phi,x_0,y_0} \mathcal{E}(k, s_0, \ell, S, \phi, x_0, y_0), \quad (8)$$

where

$$\mathcal{E} = \frac{1}{2} \int_0^1 \|\mathbf{x}(t) - \boldsymbol{\gamma}_{(k,s_0,\ell,S,\phi,x_0,y_0)}(s(t)/L)\|^2 \|\mathbf{x}'(t)\| dt \quad (9)$$

is the square of the L_2 -distance between the spline and the elastica segment.

We use a gradient driven optimization package IPOPT [13], so we need the partial derivatives of the objective function \mathcal{E} with respect to the parameters $(c_1, \dots, c_7) = (k, s_0, \ell, S, \phi, x_0, y_0)$, i.e.,

$$\frac{\partial \mathcal{E}}{\partial c_i} = - \int_0^1 \left\langle \frac{\partial \boldsymbol{\gamma}_c}{\partial c_i}(s(t)/L), \mathbf{x}(t) - \boldsymbol{\gamma}_c(s(t)/L) \right\rangle \|\mathbf{x}'(t)\| dt. \quad (10)$$

The optimization problem is non-convex, so there are several local minima for \mathcal{E} . Therefore the optimization gives different results depending on the initial values of the parameters, cf. Fig. 9. It is therefore necessary for us to have a good initial guess. We describe next our method for finding an initial guess. The full details can be found in [3].

We find the initial guess by considering the differential equation (3). If we let $u = \frac{1}{\lambda}(\lambda_2 x - \lambda_1 y)$ then the differential equation can be written as $\frac{d^2 u}{ds^2} = \lambda \frac{du}{ds}$, and

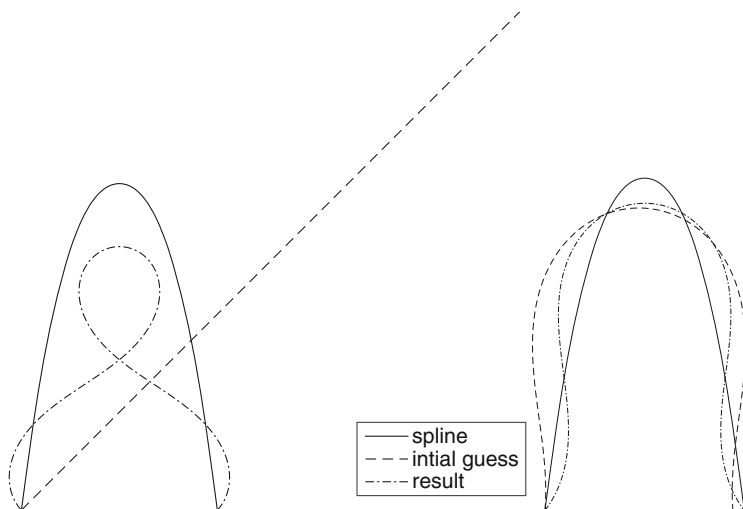


Fig. 9 Approximating a spline by elastica. The *solid line* is the spline, the *dashed curves* are the initial guess, the *dash-dotted curves* are the optimized approximations. To the *left* an arbitrary (bad) initial guess and to the *right* our guess

integrating this yields

$$\kappa = \frac{d\theta}{ds} = \lambda u + \alpha = \lambda_2 x - \lambda_1 y + \alpha . \tag{11}$$

Letting θ_u denote the angle between the u -axis and the tangent, we have

$$\cos \theta_u = \frac{1}{\lambda} \begin{pmatrix} \lambda_2 \\ -\lambda_1 \end{pmatrix} \cdot \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{du}{ds},$$

so

$$\frac{d \sin \theta_u}{du} = \frac{ds}{du} \frac{d \sin \theta_u}{ds} = \frac{1}{\cos \theta_u} \cos \theta_u \frac{d\theta_u}{ds} = \kappa = \lambda u + \alpha,$$

and thus

$$\sin \theta_u = \frac{1}{2} \lambda u^2 + \alpha u + \beta .$$

As $(\lambda_1, \lambda_2) = S^{-2}(\cos \phi, \sin \phi)$ we get estimates for the scale S and the angle ϕ by solving the first equation with respect to $\lambda_1, \lambda_2, \alpha$ in the least square sense. In a similar manner we can estimate β , and by analysing the resulting parabola we can determine whether we should use an elastica with or without inflections and estimate the parameter k . In the next step we determine which segment of the elastica we

should use, i.e., estimate s_0 and ℓ . We finally determine the translation (x_0, y_0) by a least square fit. If we want end point interpolation then we can achieve that by a final scaling, rotation, and translation.

5.2 A Purely Numerical Approach

We have described above a method for approximating a spline curve $\mathbf{x} : [0, 1] \rightarrow \mathbb{R}^2$ by a segment of an elastic curve, represented by an analytic solution in terms of elliptic functions. An alternative approach is to approximate the spline by another spline curve \mathbf{y} which is intended to be close to an elastica, in the sense that it minimizes the elastic energy. This approach could be advantageous for practical reasons. For example, existing CAD software and other mathematical software and algorithms already work with the data structure of splines.

We will use a refined knot vector for the new spline curve \mathbf{y} . By knot insertion we express both the target spline \mathbf{x} and the elastica approximation \mathbf{y} using the same basis functions (B-splines). This gives us control points \mathbf{x}_i and $\mathbf{y}_i, i = 1, \dots, n$, that we can compare. We now seek to minimize the bending energy (2) of the new spline curve \mathbf{y} , with control points \mathbf{y}_i , while staying close to the original curve \mathbf{x} , with control points \mathbf{x}_i . The difference between these spline curves is also a spline curve, with control points $\mathbf{x}_i - \mathbf{y}_i$, and the distance between the curves is captured by the distance between the control points. These points have coordinates $(\mathbf{x}_i - \mathbf{y}_i) \cdot \mathbf{e}_i$, where $\mathbf{e}_1 = (1, 0)$ and $\mathbf{e}_2 = (0, 2)$. That is, we consider the constrained optimization problem:

$$\text{minimize}_{\mathbf{y}_1, \dots, \mathbf{y}_n} \quad \frac{1}{2} \int_0^1 \kappa_y^2 ds, \tag{12}$$

$$\text{such that} \quad -\epsilon \leq (\mathbf{x}_i - \mathbf{y}_i) \cdot \mathbf{e}_j \leq \epsilon, \quad i = 1, \dots, n, j = 1, 2. \tag{13}$$

We need to constrain the problem additionally, e.g., by fixing the positions and tangents at the two end points. On top of this, we thus have an optimization, or sampling, over end points and tangents. For end point interpolation we simply put $\mathbf{y}_1 = \mathbf{x}_1, \mathbf{y}_n = \mathbf{x}_n$, and remove these two control points from both the optimization and the constraints. The tangent constraints just specify directions along which \mathbf{y}_2 and \mathbf{y}_{n-1} can move. The length could also be specified. In any case, we are no longer looking for the elastica that minimizes the distance to \mathbf{x} , but rather for an elastica that is ϵ -close \mathbf{x} . If none of the constraints are active at the end of the optimization we conclude that we have obtained an elastica which is closer to the target spline than ϵ . This is of course only true up to the discretization error resulting from using splines to model elastica. By refining the knot vector of the spline we obtain a smaller discretization error, and we can validate the solution by checking the differential equation (11). An example of this approach is shown in Fig. 10.

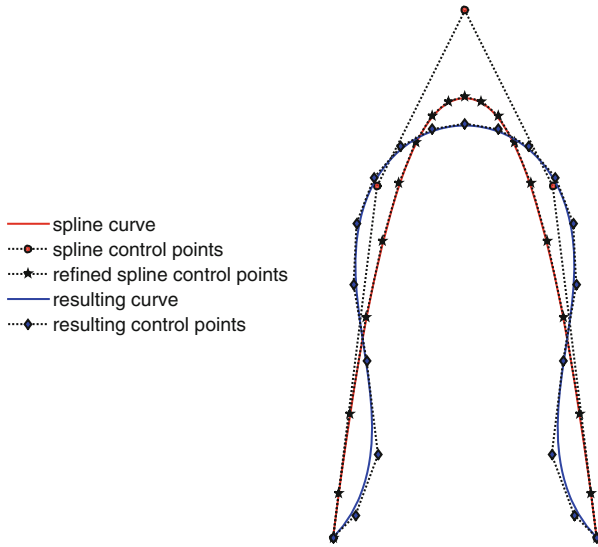


Fig. 10 A spline approximation of an elastica (*blue*) constrained by a target spline (*red*). The end positions and tangents have been fixed

A disadvantage of this method is that we cannot guarantee that our solution \mathbf{y} is close to an elastica—only that it has less bending energy than the input curve \mathbf{x} . For this reason, we have chosen to work with the analytic approach outlined in the previous subsection.

6 Surface Rationalization

Before a given CAD surface can be realized as a mould in the form of a collection of EPS blocks it needs to be divided into patches. Each individual patch is approximated by a surface swept by a hot blade, i.e., a surface foliated by planar elastic curves as described in the previous sections.

In fact, we need to consider two processes: *blocking* and *segmentation*. Blocking is the process of dividing a surface into blocks such that each block can be cut individually using either a hot wire or a hot blade. Segmentation on the other hand is the process of dividing the surface into patches swept by elastica or ruled patches. If blocking is performed before segmentation, we simply divide the 3D shape into blocks and then fit the best possible ruled or elastic surface patch to each block—possibly taking constraints between block boundaries into account. Doing segmentation first is arguably harder, but has certain benefits: knowing which segments intersect a given block can be used to inform the blocking procedure.

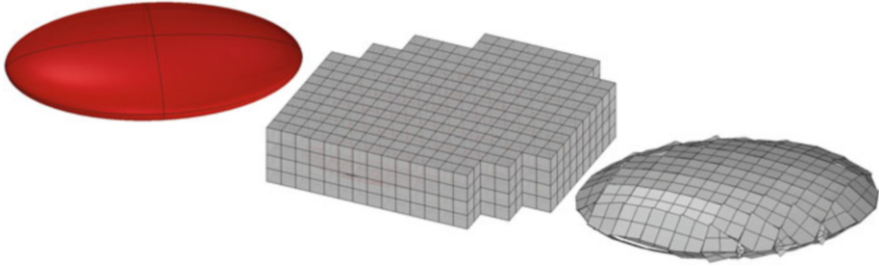


Fig. 11 A simple approach to the rationalization of the *red ellipsoid* by planar surfaces. No boundary conditions are enforced in this rationalization

In the following, we consider a more concrete approach to segmentation in the context where we assume that blocking has been performed first.

We first consider the problem of approximating a single surface by a surface foliated by planar elastic curves. One way to accomplish this is first to foliate the surface by planar curves and then approximate these planar curves by elastica. That is, we intersect the surface with a family of planes that are sufficiently overall transversal to the given surface, and thereby foliate the surface by planar curves. We then pick a finite number of these planes, approximate each of the corresponding planar sections with a segment of an elastica, calculate endpoints, end tangents, and lengths and interpolate this data to obtain an approximation of the original surface.

For the general case we imagine our CAD surface sitting inside a collection of EPS blocks. This divides the surface into a collection of pieces each of which is the intersection between a block and the full surface, see Fig. 11. We now approximate each of these pieces by an elastica swept surface while demanding that neighbouring surfaces fit together in a C^1 fashion. This can be a large global optimization problem, and at the end we check to see if the result is within the required tolerance. We then pick the blocks where the tolerance is exceeded, cut these blocks in half and redo the optimization.

In the more complex approach, where segmentation (of the CAD surface) is performed first, several options can be considered. One way is to fit the largest patch that upholds the tolerance criteria to the surface and remove this part to create a reduced surface. This procedure is repeated until the whole surface is removed, i.e., the original surface is covered by patches. Another approach is a patch-growing algorithm as in [7]: A number of patches grow on the surface and whenever two patches meet a competition determines the boundary between the patches. The determining force in the competition is the improvement on the Euler elastica sweep approximation, i.e., the resulting boundary is the one with the largest combined improvement.

For fabrication, each patch needs to be divided into blocks, and this can be difficult on the boundaries; either the blocks need to be cut smaller to align with the boundaries or multiple elastica sweeps are needed, i.e., the block can be cut more than once by the blade.

A third option is a hybrid of the above mentioned methods, where the knowledge from the patch methods guides the placement of the blocks.

7 Example

To illustrate the procedures, we consider the modelling and construction of the skater ramp shown in Fig. 12. This CAD surface consists of spline surfaces, some of which are doubly curved. The curved surfaces (see Fig. 13) are the ones that need special moulds. Here there are three different types: (1) three ruled parts (the “sides”), (2) two corners with negative curvature at the front of the image and (3) two corners with positive curvature at the back. We will approximate each corner by a surface swept by elastica. The ruled parts can be cut either by the hot wire following the rulings or by the hot blade approximating the curved cross section curve by an elastic curve.

For each corner, the control points give rise to a set of planar spline curves which foliate the surface (see Fig. 14). These curves can be approximated by elastica as described in Sect. 5.

If the splines are approximated independently, the control parameters for the resulting elastica might differ quite a lot between two adjacent curves. This is because, for a typical (uncomplicated) curve segment, there can be many different elastic curve segments that approximate it quite closely. To avoid large jumps in the control parameters we use the elastica that approximates the first spline curve as the initial guess for the optimization at the next spline, and so forth.

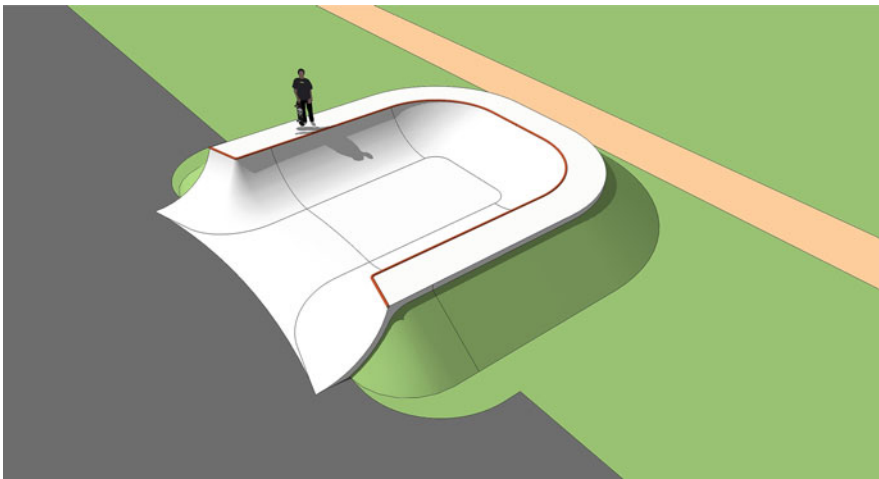


Fig. 12 The skater ramp example

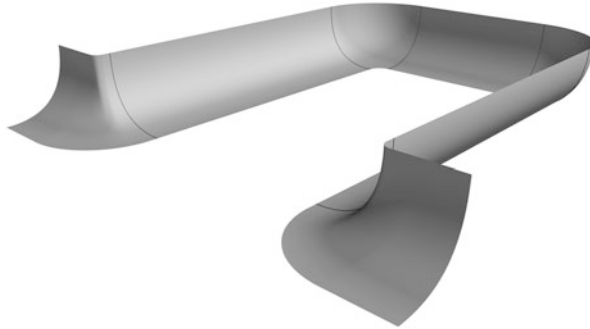


Fig. 13 The spline surfaces of the skater ramp must be approximated by elastica swept surfaces

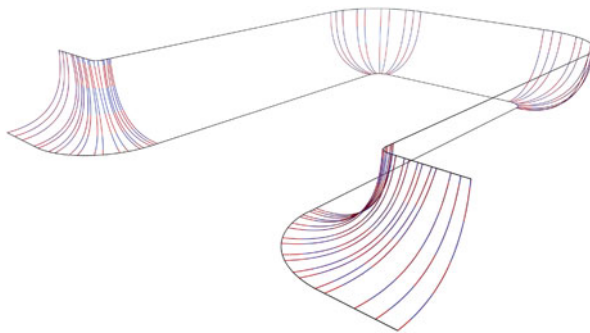


Fig. 14 The corner surfaces are foliated by planar spline curves (*blue*). Each of these are approximated by an elastic curve (*red*)

Table 1 The optimized value for the L_2 -distance for the two corner types

	Min	Max	Average
Negative	0.838846837	0.9107882	0.868163184
Positive	5.738445432	5.788943718	5.778703296

The minimal value corresponds to the elastic curve which best approximates the spline. The height of the ramp is 854.10 with the lengths of the spline curves varying between 1342.6 and 1459.8

The optimization is performed with constraints: the approximating elastic curve is in each case required to have the same length and the same end points as the original spline curve. The resulting elastic curves can also be seen in Fig. 14.

Our optimization algorithm minimizes the square on the L_2 -distance between the spline curve and the elastica, see (9). Table 1 shows some of these distance values.

For a visual inspection and evaluation of the result, in Fig. 15 we have plotted the spline and the approximating elastica in the worst case (i.e. highest L_2 distance). For the corner with negative curvature the curves are nearly indistinguishable. For the positively curved corner, there is clearly a difference, but the overall shape is the

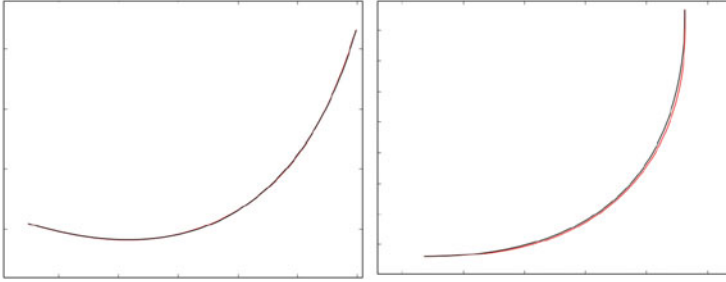


Fig. 15 The original planar spline curve (*black*) on *top* of the approximating elastica (*red*). These are the worst cases for the corners with negative curvature (*left*) and positive curvature (*right*)

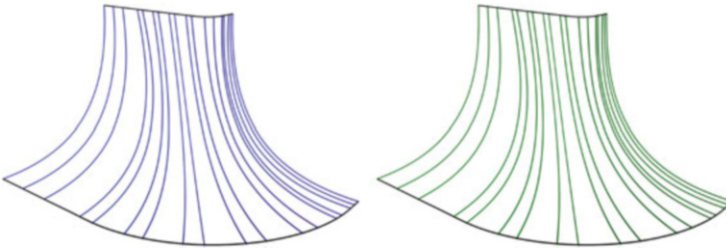


Fig. 16 *Left*: The surface is foliated by planar spline curves. *Right*: The surfaces are foliated by elastic curves each one of which approximates the corresponding spline in the figure to the *left*

same, and the approximation is certainly within any conceivable tolerance for this particular application (Fig. 16).

8 Conclusion

Our work on approximating arbitrary spline curves by elastic curves, illustrated here by the test case of the skater ramp, indicates that the problem of approximating most of the CAD surfaces used in architecture by panels of surfaces swept by planar elastica is feasible, and that it can be effectively implemented into the work flow of modern robotics enhanced constructions of buildings and other manifestations of architectural design, see also the report in [12].

The utility of the technology depends now on the technical problem of designing blades that can be heated and used to cut EPS in a consistent, robust and predictable way. We have received positive experimental results and optimistic input from our project partners showing that this blade technology can indeed be developed and made operational on the scale needed.

In addition to the application to final production architecture, we anticipate that the theoretical framework described here will also have other industrial applications.

For example, rapid prototyping is an important part of the innovation pipeline. Prototypes that are currently produced in EPS using CNC milling could be produced much more quickly using hot blade cutting.

As is evident from our present description and also from the discussion of the state of the art in Sect. 2.3 the full implementation of the various assets of the robotic Hot Wire/Blade cutting idea needs—to mention but one momentum, at least for the Building Industries—an almost paradigmatic shift of attention away from the classical use of relatively complicated scaffoldings and laths for the concrete shuttering of facade elements that are not just off-the-shelf items. In comparison with the proposed applications of the Hot Blade Cutting Technology, the classical way of facade production is often very labor intensive, and often it even demands an extra time-consuming postprocessing, a fairing by hand, in order to obtain the desired smoothness of the facades and surfaces. On a final note we should also mention that this particular MaDiFa concept, that we have presented in this chapter, offers one more genuine quality, which will also reduce transportation and logistics costs considerably, namely that in this case the digital factory in question is essentially mobile and can be set up for EPS cutting, assembling, and shaping on any location.

Acknowledgements This work was completed with the support of Innovation Fund Denmark, project number 128-2012-3.

References

1. Birkhoff, G., Boor, C.D.: Piecewise polynomial interpolation and approximation. In: Garabedian, H.L. (ed.) *Approximation of Functions*, Proceedings of the General Motors Symposium 1964, pp. 164–190. Elsevier Publishing Co., Amsterdam (1965)
2. Borbély, A., Johnson, M.: Elastic splines I: existence. *Constr. Approx.* **40**, 189–218 (2014)
3. Brander, D., Gravesen, J., Nørbjerg, T.B.: Approximation by planar elastic curves. *Adv. Comput. Math.* (2016). doi:10.1007/s10444-016-9474-z
4. Broek, J.J., Horváth, I., de Smit, B., Lennings, A.F., Rusák, Z., Vergeest, J.S.M.: Free-form thick layer object manufacturing technology for large-sized physical models. *Autom. Constr.* **11**, 335–347 (2002)
5. Flöry, S., Pottmann, H.: Ruled surfaces for rationalization and design in architecture. In: *Proceedings of the ACADIA*, pp. 103–109 (2010)
6. Horváth, I., Kovács, Z., Vergeest, J.S.M., Broek, J.J., de Smit, A.: Free-form cutting of plastic foams: a new functionality for thick-layered fabrication of prototypes. In: *Proceedings of the TCT'98 Conference*, Nottingham, pp. 229–237 (1998)
7. Julius, D., Kraevoy, V., Sheffer, A.: D-charts: quasi-developable mesh segmentation. In: *Computer Graphics Forum*, Proceedings of the Eurographics, pp. 581–590 (2005)
8. Levien, R.: From spiral to spline; optimal techniques for interactive curve design. Ph.D. thesis, UC Berkeley (2009)
9. Love, A.E.H.: *A Treatise on the Mathematical Theory of Elasticity*. Cambridge University Press, Cambridge (1906)
10. Malcolm, M.: On the computation of nonlinear spline functions. *SIAM J. Numer. Anal.* **14**, 254–282 (1977)

11. Mehlum, E.: Nonlinear splines. In: Computer Aided Geometric Design (Proceedings of the Conference), University of Utah, Salt Lake City, UT, 1974, pp. 173–207 (1974)
12. Søndergaard, A., Feringa, J., Nørbjerg, T.B., Steenstrup, K., Brander, D., Gravesen, J., Markvorsen, S., Bærentzen, A., Petkov, K., Hattel, J., Clausen, K., Jensen, K., Knudsen, L., Kortbek, J.: Robotic hot-blade cutting. In: Reinhardt, D., Saunders, R., Burry, J. (eds.) *Robotic Fabrication in Architecture, Art and Design 2016*, pp. 150–164. Springer International Publishing, Cham. ISBN: 978-3-319-26378-6, doi:10.1007/978-3-319-26378-6_11
13. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program. Ser. A* **106**, 25–57 (2006)

Model-Based Design of Self-Correcting Forming Processes

M. Krüger, M. Borzykh, U. Damerow, M. Gräler, and A. Trächtler

Abstract In this paper, a self-correcting strategy for a metal forming process is presented. This strategy entails continuously observing the properties of the product and ensuring that these properties stay in the required tolerances. This reduces the scrap rate and the need to adapt the configuration of the machine manually, both aspects leading to an increase of the productivity and efficiency of the process. For the development of the strategy, a structured design method for mechatronic systems is adapted. During the whole development process, it uses intensively mathematical models of system dynamics to ensure a high quality of the results. There are two commonly used model types to describe the behavior of a forming process: finite-element models and multibody system models. For the development of such a mathematical model, the process has to be examined. It is analyzed regarding its disturbances, influences and possibilities to take action. In this paper, the bending-process is modeled as a multibody system. It describes the most significant influences and is fundamental to develop the self-correcting closed-loop control. The presented model includes a good compromise between computation time and accuracy. After introducing the structured developed closed-loop control, the implementation and the results are presented.

1 Introduction

Industry is confronted with growing requirements regarding quantity, quality, flexibility and costs. Companies are thus equipped with highly automated production systems, in which sensors detect slightest variations between reference values and actual tool path and powerful control units take action to correct these errors. In a lot of processes, uncertainties which lead to varying quality of the produced products exist. This is caused by unpredictable disturbances and changing system properties.

M. Krüger • M. Borzykh (✉) • M. Gräler • A. Trächtler
Fraunhofer Research Institution for Mechatronic Systems Design IEM, Paderborn, Germany
e-mail: mikhail.borzykh@iem.fraunhofer.de

U. Damerow
Chair of Forming- and Machining Technology, University of Paderborn, Paderborn, Germany

In particular, offline calculated reference values are no longer optimal in view of the real process. Therefore, more flexibility and intelligent control of production processes are necessary in order to react to quality changes.

Metal parts are often produced using metal-forming processes, especially for large volume production runs. Challenges in metal-forming processes are development of robust, adaptable and intelligent forming processes, which are able to react to unforeseen disturbances. We call such a process a self-correcting forming processes. Nowadays, most common mechanical manufacturing processes are equipped with measuring, drive and control engineering elements. Hence, they are mechatronic systems. However, metal-forming processes are usually developed with classical domain-specific design methods instead of using a model-based domain-spanning development process as is well established for mechatronic systems [10]. Metal-forming processes are most often optimized offline with FEM (finite element method) and optimal trajectories are calculated for a particular set of parameters. The static trajectories are realized by means of closed-loop controlled actuators. However, there is usually no feedback of product properties and hence no possibility to react to unpredictable disturbances like temperatures, different material behavior or tool wear [8]. These effects cause the static optimized trajectories to diverge from the optimum which results in deviations of product properties, e.g. workpiece dimensions (see Fig. 1), causing suboptimal operation point. Due to that, a lot of produced parts do not match the quality requirements like geometric tolerances, which implies lower productivity, higher scrap rate and higher costs for the producing company.

In the following sections we present how to design intelligent self-correcting forming processes. The design includes several aspects of the metal-forming process. Existing sensors and actuators can either be used or substituted. If necessary, additional devices can be integrated. Also, an intelligent control strategy has to be designed. Mathematical process models are mandatory for design of such complex systems. Models have to include all relevant dynamic effects in order to find a reliable self-correcting strategy. Further, a well-structured, model-based design process has to be used. The transfer of the design methodology for mechatronic systems to self-correcting metal-forming processes is also presented.

Online process control despite facing unpredictable disturbances requires the relevant product properties to be observable. Only with this information is it possible to react on deviations to hold the product properties within tolerances (see Fig. 1 for

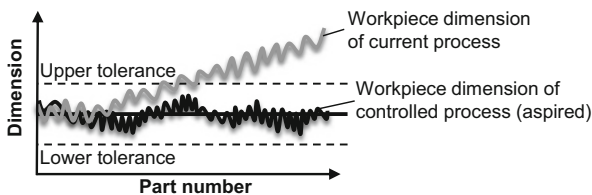


Fig. 1 Non-controlled and controlled processes

an illustration). However, this is still not very common in metal-forming processes despite having highly automated machines.

We demonstrate modeling and design of self-correcting forming processes using a punch-bending process as application example. Here, product properties of the individual parts are checked manually and randomly by an operator. If the part's properties leave the required tolerances (see Fig. 1), the configuration of the punch-bending machine has to be changed. The failure to reproduce the shape of the element within allowable tolerances is caused by varying shape or strength of the semi-finished material (flat wire) as well as the thermal and dynamical behavior and wears phenomena of the punch-bending machine itself or of the punch-bending tool. Currently, the machine's parameters have to be adapted manually by the operator and his personal experience. Further, these targeted interventions are only possible with a stopped punch-bending machine, making this procedure time consuming, especially if more than one iteration step is necessary. If the machine is to change its configuration by itself, it requires certain knowledge about the process. This knowledge is gained by a simulation model, which is also used to design the control strategy.

This contribution is structured as follows: In the next section we present the general idea of self-correcting metal-forming processes and give a short introduction of the model-based design methodology. This is followed by a brief description of the bending process. The following two sections present our modeling approach for the bending process and the design of the self-correcting control strategy. The contribution ends with a short conclusion.

2 Self-Correcting Metal-Forming Processes

Nowadays, production systems including metal forming are extensively equipped with sensors, actuators and digital information processing (e.g. with PLCs), thereby following the trend towards intelligent, networked production systems. Development activities in metal forming focus on two criteria: accuracy of produced parts and flexibility of machines and tools. Although there are lots of sensors integrated in a forming machine, it is difficult to measure or predict product properties like geometric dimensions online. First and foremost, sensors are used to measure signals necessary for control of actuators, e.g. velocity or position of punches or servo motors. Today, control of metal forming is mostly realized as open-loop control [8] (see Fig. 2). Process planning is done offline in a quasi-static way, i.e. trajectories of presses or punches are designed offline. The result is a target tool-path that is realized by means of available actuators or by adjusting tools manually. Local controllers are used to ensure realization of desired tool-path with sufficient precision. A significant drawback of this approach is that there is no possibility to react on unforeseen changes like fluctuations in material properties, tool wear or varying temperatures, just to name a few examples. Machine dynamics are mostly neglected, too.

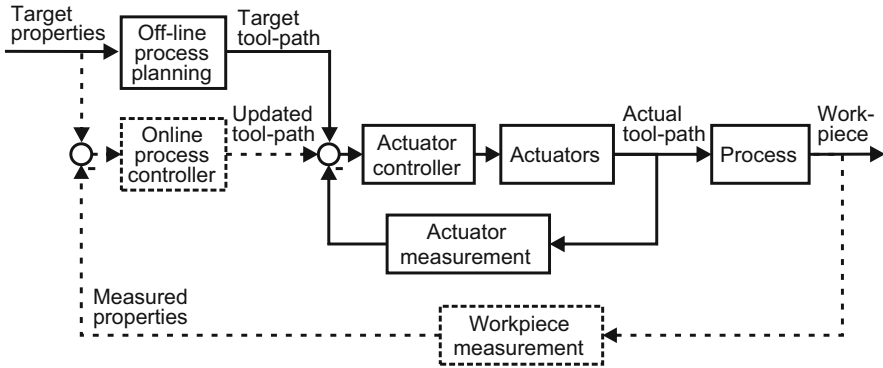


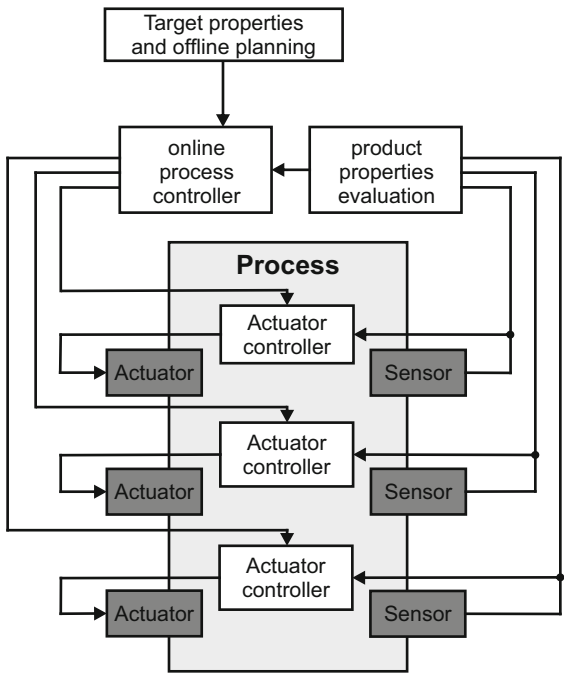
Fig. 2 Open-loop control vs. closed-loop control of metal forming processes, (based on [8])

A closed loop control of the production properties is necessary in order to cope with such time-varying, unpredictable fluctuations of the process. The existing local controllers can still be used, but they are not sufficient as they can only react on deviations from the offline tool-path and not on unforeseen disturbances. Hence, the product properties have to be controlled by means of a closed-loop control strategy (see dotted line in Fig. 2). This results in a hierarchical control architecture. On the top level there is an online process controller which is used to control deviations in product properties and computes reference values for actuators. These reference values are then controlled by means of several local controllers on a subordinate level. Figure 3 illustrates this hierarchical control architecture.

A closed-loop control of product properties necessitates the relevant properties to be measurable or observable. A lot of research has been done in development of condition monitoring systems [5]. Today, it is possible to measure the desired product properties like geometric shape in several metal forming machines. This information is used to enhance the quality by rejecting produced parts with poor properties [1]. Usually there is no feedback to the control unit and this technique is also not state-of-the-art. In most processes, the product properties are inspected offline after production as a separate step of quality control. Especially for large volume production, only a small subset of produced parts are inspected, e.g. following some statistical approach.

Self-Correcting forming processes are metal forming process which are equipped with such a hierarchical control architecture. Similar hierarchical control strategies have already been implemented successfully in lots of application examples, see [4] for more details. Such hierarchical systems are complex systems with several actuators and sensors. Particularly for metal forming processes, there are non-negligible couplings between different tools as they influence product properties over a much wider area than the zone of contact. Appropriate models of the process dynamics are necessary in order to design a self-correcting control strategy. Such models have to include all relevant dynamical effects, either resulting from varying material properties, machine dynamics or the forming process itself. In general, a

Fig. 3 Hierarchical closed-loop control of metal forming processes



time-discrete model of process dynamics is given by

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \tag{1a}$$

$$\mathbf{y}_{i+1} = g(\mathbf{x}_i, \mathbf{u}_i), \tag{1b}$$

with \mathbf{y} the product properties that have to be controlled, \mathbf{x} the state of the system and \mathbf{u} the inputs. i denotes the discrete time step e.g. assuming the production of one part per step. For some processes, it is not sufficient to control the product properties at discrete time steps, but to have some continuous property evolution. In such cases, a continuous model analogue to (1) is needed. The goal of modeling is the derivation of a distinct mathematical description of some physical behavior. The model can be based on physical equations or measurements through experiments. Due to efficiency, the model should be as simple as possible, meaning just the significant effects should be modeled with sufficient precision.

In order to develop a closed-loop control of product properties, several questions have to be answered, such as: What are the properties that have to be controlled? How can current product properties be measured? Do additional sensors need to be implemented? How can the process be adjusted for the properties to become controllable? Are the existing actuators sufficient or are additional ones necessary? What kind of model has to be developed to design the online process controller? A structured, model-based system development is the best way to answer these questions adequately.

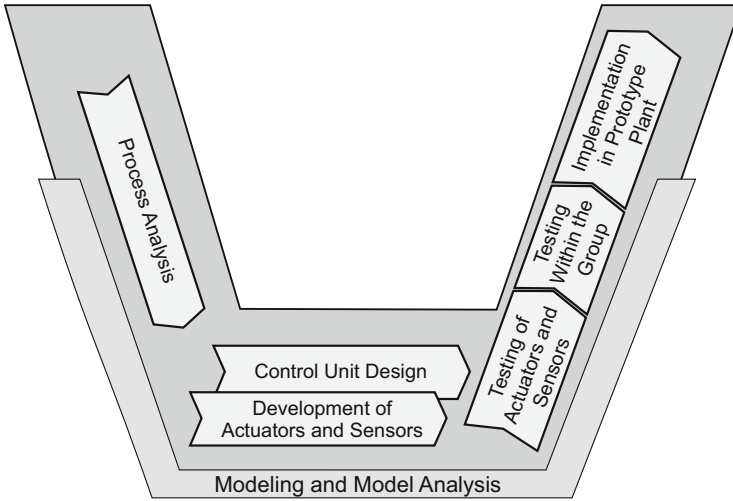


Fig. 4 V-model for development of mechatronic systems adapted to metal-forming processes

Modern metal forming machines are mechatronic systems. They consist of sensors, actuators and information processing that are used to realize some desired behavior of a basic system (the metal forming). Looking at the system hierarchy, the local controllers are mechatronic function modules as defined in [7]. They also consist of actuators, sensors, information processing and some physical system as part of the forming process. A modularization of the system into several mechatronic subsystems can be achieved using this structuring principle. This reduces the complexity of the system and is one of the first steps of designing a self-correcting forming process.

The VDI guideline 2206 [10] describes a structured, well established design method for mechatronic systems including mathematical models of system dynamics. This design method can be adapted to the design of metal-forming processes (see Fig. 4). One of the basic principles in design of mechatronic systems is the intensive use of mathematical models during the whole design process. This is still a challenge for metal-forming processes, as usually only quasi-static models of the mechanical domain are used. These models, however, are not suited for dynamic simulations and design of closed-loop control. An alternative modeling approach is presented within this contribution.

Model-based design starts with Process Analysis. Basic information and requirements of the system are determined e.g. product properties to be controlled or relevant effects like temperature dependency, varying material behavior. A modeling approach has also to be chosen and a first ideal model of the process dynamics has to be built. This model and probably additional experiments are used to analyse the process. The process analysis answers most of the basic questions. The inputs and outputs as well as an ideal control strategy are developed and tested using the

process model. In this phase the actual sensors and actuators are not chosen yet, but the signals that have to be measured and how the process can be controlled is verified. In mechatronic systems design the result of this phase is called the principle solution.

Domain-specific design and development is the next step of the design process. In the context of self-correcting forming processes, the main steps are Control Unit Design and Development of Actuators and Sensors. Suitable sensors and actuators have to be chosen, considering the system requirements as well as the principle solution. If any proper sensors or actuators are not available, they have to be developed. The control strategy has to be detailed in view of additional requirements corresponding to particular sensors and actuators e.g. measurement noise or actuator limits. Relevant effects for the process dynamics are also added to the process model, so that it can be further used for control design.

The last phase of the design process is System Integration. At first, sensors and actuators are tested separately. Afterwards, smaller groups are tested e.g. consisting of one actuator and sensor. Particular testbeds may be used for this task. Also, the control unit can be tested by means of hardware-in-the-loop (HiL) techniques. After successful verification of the groups, they are integrated in the target forming machine and extensive tests have to be carried out.

3 Bending Process

The model-based design of self-correcting forming processes is demonstrated on a punch-bending process. Typical parts made with a punch-bending process are e.g. contact springs used in the electrical connection technology. In our case, a servo-controlled stamping and forming machine is used which is running at up to 60 parts per minute. The slide units driving the punches in the tool are driven by servo motors and their motion behavior can be adapted variably without necessity of additional actuators. Nowadays, these machines are usually driven with fixed process parameters during production. The produced part we look at is a so-called contact spring. It is made of a high-strength copper alloy flat wire and produced in two process steps as shown in Fig. 5. In the first step, two punches and dies shape the flat wire. In the next step, the shaped flat wire is bent around a bending core and

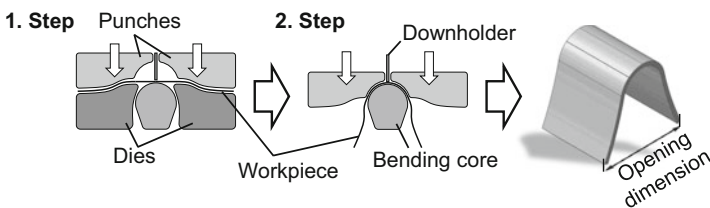


Fig. 5 Production steps of the contact spring

the critical dimension, here the opening length, is adjusted. Using the fixed machine parameters, the critical dimension leaves the tolerances very often, which leads to high scrap rates. The reasons for this can be a variation in the properties of the semi-finished product, or wear phenomena on the punch-bending machine itself or on the punch-bending tool. The only way to avoid deviations in the opening length is a manual adjustment of the machine. Additionally, the choice of the most appropriate process parameters is based on the operator's experience. The manual adjustment is a time-consuming and expensive procedure right at the early stages of a production scenario. In addition, the trend towards reduced part sizes with tight tolerances, made of high strength materials, is drastically increasing the requirements regarding the production process.

4 Modelling of Bending Process

As seen in Fig. 4, the design method for mechatronic systems intensively uses mathematical models during every phase. In this section, the development of the bending process model is shown. Modelling opens optimization possibilities to reach the quality goal without the necessity of time consuming and expensive experiments. Furthermore, the model can be used to perform numerical optimizations to ensure the best operating point during the whole process. Additionally, failures can be injected in the model and the dependencies investigated. That allows finding and testing a robust closed-loop control strategy to correct the behavior of the machine in a very structured way.

For further process analysis and the following design of the correction strategy, a modeling of the forming process and feed-back control is required. In general, finite element (FE) models and multibody systems (MBS) are used to describe a system [12]. These models lead to the following nonlinear differential equations of motion

$$\dot{\hat{\mathbf{x}}}(t) = \bar{f}(\hat{\mathbf{x}}(t), \hat{\mathbf{u}}(t)) \quad (2a)$$

$$\hat{\mathbf{y}}(t) = \bar{g}(\hat{\mathbf{x}}(t), \hat{\mathbf{u}}(t)) , \quad (2b)$$

which describe the time-continuous dynamic behavior for one workpiece of the punch-bending process. In this equation $\hat{\mathbf{x}}(t)$ are the states of the system, $\hat{\mathbf{u}}(t)$ the inputs and $\hat{\mathbf{y}}(t)$ the outputs, which in case of metal forming are the relevant product properties. There are already a variety of different simulation models for the purpose of metal forming. In most cases, these are quasi static models, which allow to design and optimize the process offline [9]. Especially in the last few years, the FE method has been extensively used to reduce expensive experiments [3]. The precision of the results is good, as long as only minor unpredictable disturbances or variations of the process occur. The computing time of FE simulations adds up to several hours, as a result of a high amount of degrees of freedom (see Fig. 6a). Due to that, these kind of models are inappropriate for tasks like online optimization or control design. The

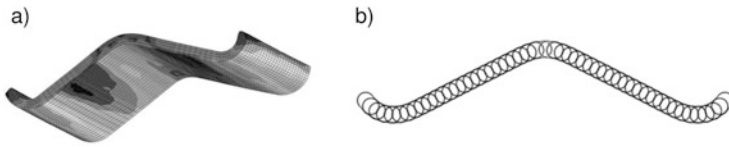


Fig. 6 Simulation models (a) FE model (b) MBS model

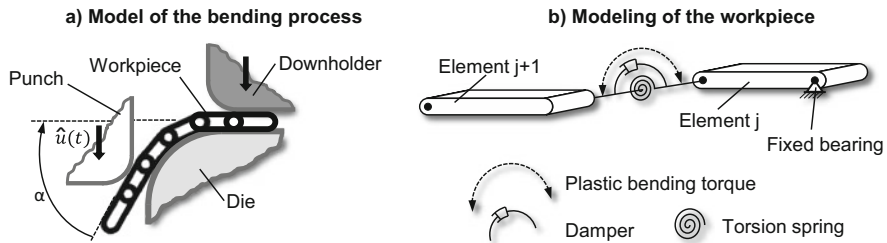


Fig. 7 Setup of the MBS model: (a) Modell of the bending process; (b) Modelling of the workpiece

computing time can be reduced by combining the response surface method (RSM) with such an FE model [11] to minimize the necessary number of simulations. This method also can be used to create an empirical model out of measurements. In [2] this method is used to react online to variations of the process. The adaption of the process takes place between the strokes of the punch. Furthermore, extensions to FE models are possible. For instance, in [13] a criterion for the creation of crinkles during the deep drawing process is implemented.

The quasi static models of metal forming processes do not allow the description of the whole system. Dynamic effects, which are induced by the behavior of the machine or by the electrical drives, are hardly considered in these kind of commonly used simulation models. These are effects which are usually implemented in MBS models. In the example of the deep drawing process [9], a crack was predicted with the quasi static simulation. In reality, this crack did not appear because the assumptions of the simulation did not fit to reality. This fact was not considered in the initial model. The false prediction is corrected with a co-simulation, which includes the dynamic behavior of the system as well. Unfortunately, this co-simulation was even more computationally intensive than the pure FE simulation, making it inappropriate for control design as well. Obviously, there is a compromise between computing time and accuracy which is necessary to be able to design a closed-loop controller. In this paper an MBS model for the bending processes is presented, which is suitable for control design purposes.

In Fig. 6, both types of models are shown. In comparison to FE models, the level of discretization in multibody system models is significantly rougher. In this particular case, the discretization is further decreased by employing a two-dimensional approach. The model contains m discrete rigid bodies (see Fig. 7), which are connected by spring-damper elements. Through these elements, the

elastic deformation of the workpiece is modeled. As a consequence, the dimension of the state vector $\hat{\mathbf{x}}(t)$ and the computing time are appreciably lower in comparison to an FE model.

The state vector $\hat{\mathbf{x}}(t)$ consists of time dependent variables of the system. In this particular case, it includes the relative angles and relative angular velocities between the elements. The angle between the rigid bodies j and $j + 1$ (see Fig. 7b) is called α_j . As a result, the state vector is defined as

$$\hat{\mathbf{x}}(t) = [\alpha_1 \dot{\alpha}_1 \dots \alpha_j \dot{\alpha}_j \dots \alpha_m \dot{\alpha}_m]^T \quad (3)$$

and is dependent on the tool geometry and the punch position $\hat{\mathbf{u}}(t)$, as indicated in Eq. (2). To consider plastic deformation as well, a plastic bending torque is calculated in the model. In contrast to the spring-damper torque, the plastic bending torque is calculated by a nonlinear relation of material properties and the relative angle between two rigid bodies. The sum of all relative angles equals the total angle α (see Fig. 7a):

$$\alpha = \sum_{j=1}^m \alpha_j \quad (4)$$

One of the goals of the following section is to find a function, which describes the remaining total angle of plasticity dependent on the applied total angle α .

4.1 Model of Plastic Deformation

The model is based on the elementary bending theory by Ludwik [6] which showed sufficiently good results. This model assumes a workpiece to be built of a chain whose joints are connected by a spring-damping system and torsional elements representing material properties (see Fig. 7b). Geometries of punches were imported in the MBS-model directly from CAD-data. The movement trajectories and velocity profiles of the punch and timing of the bending process were taken from the real process and fed into the model.

During the forming process, the total angle α is composed of the elastic α_{el} and plastic α_{pl} portions. For the continuous system, it means that

$$\alpha = \alpha_{el} + \alpha_{pl}. \quad (5)$$

Originally, Ludwik-law describes the deformation of a continuous beam. In this approximation, however, it is used for the total angle between two of the discrete elements, so that

$$\alpha = \sum_{j=1}^m \alpha_{el,j} + \alpha_{pl,j} \quad (6)$$

holds. More precisely, the elementary bending theory is applied repeatedly to the discrete elements. After the forming process is completed, the workpiece springs back by the elastic component and only the plastic part remains. According to Ludwik-law, the remaining total plastic angle α_{pl} can be calculated as a function of the total angle.

$$\alpha_{pl} = K \alpha = \sum_{j=1}^m K_j \alpha_j \tag{7}$$

The spring-back coefficient

$$K_j = 1 - \frac{r_{m,j} M_{B,j}}{E I} \in [0, +\infty) \tag{8}$$

can be determined from middle radius of curvature $r_{m,j}$, bending moment $M_{B,j}$, modulus of elasticity E and inertia geometric parameter I . The approximated radius of curvature $r_{m,j}$ for each joint can be found from element length L_e and relative angle α_j :

$$r_{m,j} = \frac{L_e}{2} \frac{1}{\tan(\alpha_j/2)}. \tag{9}$$

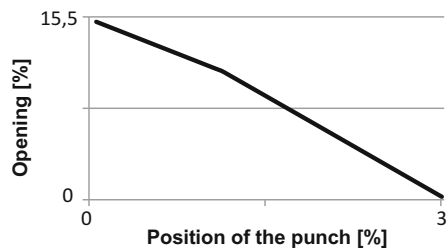
The inertia geometric parameter

$$I = \frac{b (S_0)^3}{12} \tag{10}$$

is dependent on the element width b and element thickness S_0 . The bending moment and elastic modulus are also variable and can be calculated depending on the flow curve and stress-strain diagram of the material.

Equipped with the described model, the first step of the design method, Process Analysis, can be started. An analysis of model and simulations revealed that the punch position (u) at the second bending step has the highest influence on the opening dimension of the contact spring. Figure 8 shows the change of the opening dimension at different values of the position of the punch by constant value

Fig. 8 Simulation results:
Influence of the punch position on the opening dimension



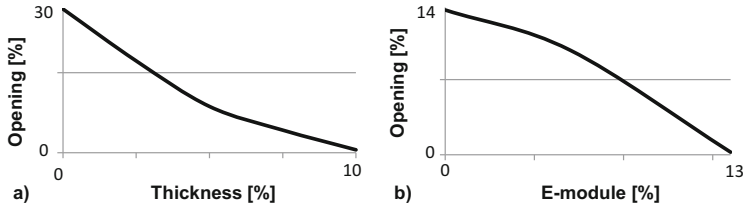


Fig. 9 Simulation results: (a) Influence of the material thickness on the opening, (b) Influence of E-module of the materials on the opening

of thickness—0.82 mm and elastic modulus—130 GPa. In this case, by position variation of 3% from initial value of 14.22 mm, the opening dimension change is approximately 15.5%.

Additionally, the influence of the material properties like thickness S_0 and elastic modulus E of a workpiece was analyzed. Both variables are varied within the tolerances specified in the data sheet. Figure 9a shows the change of the opening dimension at different thickness of workpiece with machine settings held constant and elastic modulus of 130 GPa. By thickness variation of 10% from initial value of 0.78 mm, the opening dimension change is approximately 30%. The influence of physical properties of the flat band on the bending process was also investigated. Figure 9b shows the change of the opening dimension at different values of the elastic modulus with machine settings held constant and thickness of workpiece of 0.82 mm. In this case, by elastic modulus variation of 13% from initial value of 123 GPa, the opening dimension change is approximately 14%.

As the result of the Process Analysis, it can be seen that the thickness of a workpiece as well as the punch position from the second bending step exert the greatest influence on the opening dimension. The thickness is an unpredictable disturbance that has to be compensated for by the self-correcting controller. Instead, the punch position can be controlled by giving varying reference values to the servo motors. Hence, it can be used as a controller output of the self-correcting controller.

5 Self-Correcting Control Strategy

Development of the correction strategy, sensors and actuators are the next steps of the design method. The development of actuators is not necessary as the bending machine is equipped with NC axes so that the movement of the punches can be controlled online. The position of the punch tool can be read from the machine control unit, too. The analysis showed a positioning accuracy of the NC-axis of 0.02 mm being absolutely sufficient. At a production speed of 60 parts per minute the machine dynamics can be neglected, too. In order to measure the opening dimension, additional sensors are necessary. In our case, commercially available systems were used. For the measurement of the opening dimension an optical

camera system has been integrated into the bending tool. The influence of varying material thickness can be reduced if it can be also be measured online. In the bending process, the thickness measurement is realized by a force measurement, because the relationship between thickness and force change is linear. Force sensors are more robust and less expensive. Hence, they are better suited to the application example than optical sensors. The influence of elastic modulus will be neglected due to minimal influence on the opening dimension and difficulties to measure it online. The dynamic model (2) is also used for design of the self-correcting strategy. The change of the opening dimension y is a non-linear function, which is dependent on material thickness S_0 , elastic modulus E and punch position u .

For designing the self-correcting strategy the results from model-based analysis were taken. For constant thickness S_0 and constant elastic modulus E the dependency between opening dimension and punch position can be computed using the dynamic model. To do so, a linearly increasing punch position $\hat{u}(t)$ is used as input. The corresponding opening dimension is given by an evaluation of the equations of motions

$$\hat{y}(t) = \bar{g}(\hat{\mathbf{x}}(t), \hat{u}(t)). \quad (11)$$

The punch position u is the control variable of the self-correcting controller. The parameters E and S_0 are disturbance variables. So the self-correcting control strategy consists of a feedback control of the opening dimension. Additionally, the influence of the thickness is compensated for by means of a disturbance compensation that is based on the force sensor. In order to build up a self-correcting strategy, it is necessary to measure the opening dimension for each workpiece online, especially if it begins to drift from the desired value to one of the tolerance limits.

The opening dimension can only be measured after the current part is produced. Hence, the current opening dimension is used for correcting the opening dimension of the next part. This is possible as the change rate of the opening dimension are slight enough from one part to the next one. For disturbance compensation, the situation is different. The thickness can be approximated using the measured punch force F_i of the first bending step (cf. Fig. 5) and there is enough time between the measurement and the second bending step to adjust the punch position. Hence, thickness variations can be compensated for separately for each produced part. For determination of the feedback control structure and design of control parameters, the nonlinear system response (11) of the opening dimension is linearized at a particular operating point

$$\left. \frac{\partial \hat{y}(t)}{\partial \hat{u}(t)} \right|_{t=t_{end}} = k, \quad (12)$$

so that a linear relationship between the opening dimension and the punch position can be achieved. The result is a linear system

$$y_i = k \cdot u_i, \quad (13)$$

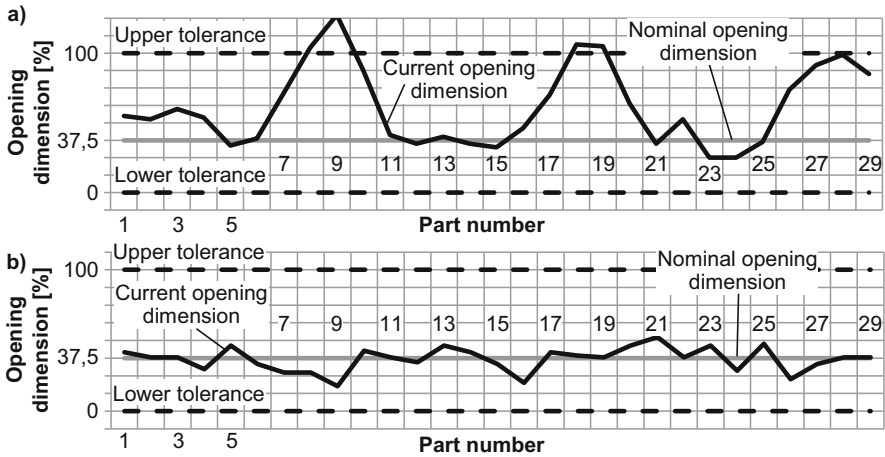


Fig. 11 Measured trend of the opening dimension without (a) and with (b) the self-correcting strategy

keeping the opening dimension of the contact spring within the tolerance range of ± 1 mm, a measurement accuracy of about 0.02 mm is indispensable. Additionally, the measurement device has to be fast enough to detect the opening dimension of each part at a production speed of 60 parts per minute.

The self-correcting strategy has been implemented into the existing machine control system. The required interfaces to measurement devices are also present in the control unit. Implementation of the correction step is achieved by adjusting the stroke of the machine axis at the second bending step for each part. A first verification of the self-correcting strategy under production conditions was carried out.

Using a production speed of 60 parts per minute the opening dimension as well as the punch force could be measured reliable and the closed-loop control showed a stable behavior. The opening dimension of the contact spring can be held within the tolerances. Figure 11a shows the opening dimension in non-controlled process. The dispersion of opening dimension is more than 80% of tolerances. Using the self-correcting control strategy, the dispersion of opening dimension was reduced to 40% of tolerances (see Fig. 11b). Furthermore, the current opening dimension is close to the nominal value.

7 Conclusion

This contribution illustrates how a mathematical description of a metal forming process can be used to enhance process quality in a digital factory. In this paper, a model-based design method for design of self-correcting controller is presented

an demonstrated using a bending process as application example. Using the bending model the forming process was analyzed and the appropriate correcting strategy was developed. The results of implementation of this strategy show that the current value of the opening dimension can be hold close to the desired value during a production run. Furthermore, it was possible to reduce the deviation of the opening dimension significantly. The same approach for developing corrective strategies is currently successfully used in other forming processes.

References

1. An, Q., Hortig, D., Merklein, M.: Infrared thermography as a new method for quality control of sheet metal parts in the press shop. *Arch. Civil Mech. Eng.* **12**, 148–155 (2012)
2. Annen, C., Wahl, M., Hora, P.: Planning and Control of Sheet Metal Forming Processes using LS-OPT. 9th LS-DYNA Forum (2010)
3. Danckert, J.: Analysis of deep drawing, ironing and backward can extrusion with main emphasis on residual stresses and processes robustness. The Sheet Metal Forming Group, Department of Production, Aalborg University Denmark. Doctoral Thesis (2005)
4. Gausemeier, J., Rammig, F. J., Schäfer, W. (eds.): *Design Methodology for Intelligent Technical Systems. Lecture Notes in Mechanical Engineering.* Springer, Berlin (2014)
5. Groche, P., Brenneis, M.: Manufacturing and use of novel sensoric fasteners for monitoring forming processes. *Measurement* **53**, 136–144 (2014)
6. Lange, K.: *Umformtechnik. Handbuch für Industrie und Wissenschaft*, vol. 3, pp. 244–306. Blechbearbeitung, Berlin (1990)
7. Lückel, J., Hestermeyer, T., Liu-Henke, X.: Generalization of the cascade principle in view of a structured form of mechatronic systems. In: *International Conference on Advanced Intelligent Mechatronics*, Como (2001)
8. Polyblank, A., Allwood, J., Duncan, S.: Closed-loop control of product properties in metal forming: a review and prospectus. *J. Mater. Process. Technol.* **214**, 2333–2348 (2014)
9. Schenke, C., Penter, L., Hardtmann, A.: *Systemsimulation in der Umformtechnik. ZWF-Zeitschrift für wirtschaftlichen Fabrikbetrieb*, issue 108, pp.741–748. Carl Hanser Verlag, Munich (2013)
10. VDI-Guideline 2206: *Design methodology for mechatronic systems* (2004)
11. Volk, M., Nardin, B., Dolsak, B.: Determining the optimal area-dependent blank holder forces in deep drawing using the response surface method. *Adv. Prod. Eng. Manag.* **2**, 71–82 (2014)
12. Wittenburg, J.: *Dynamics of Multibody Systems*, 2nd edn. Springer, Berlin (2007)
13. Wurster, K., Liewald, M., Blaich, C., Mihm, M.: Procedure for automated virtual optimization of variable blank holder force distributions for deep-drawing processes with LS-Dyna and optiSLang. 8th Weimar Optimization and Stochastic Days (2011)

Discrete Cosserat Rod Models Based on the Difference Geometry of Framed Curves for Interactive Simulation of Flexible Cables

Joachim Linn and Klaus Dreßler

Abstract For software tools currently used in industry for computer aided design (CAD), digital mock-up and virtual assembly there is an increasing demand to handle not only *rigid* geometries, but to provide also capabilities for realistic simulations of *large deformations of slender flexible structures* in real time (i.e.: *at interactive rates*). The theory of *Cosserat rods* provides a framework to perform physically correct simulations of arbitrarily large spatial deformations of such structures by *stretching, bending and twisting*. The kinematics of Cosserat rods is described by the *differential geometry of framed curves*, with the differential invariants of rod configurations corresponding to the strain measures of the mechanical theory. We utilize ideas from the *discrete* differential geometry of framed curves in combination with the variational framework of Lagrangian mechanics to construct *discrete Cosserat rod models* that behave qualitatively correct for rather coarse discretizations, provide a fast computational performance at moderate accuracy, and thus are suitable for interactive simulations. This geometry based discretization approach for flexible 1D structures has industrial applications in design and digital validation. We illustrate this with some application examples from automotive industry.

1 Introduction

Standard software tools currently used in industry for CAD, digital mock-up and virtual assembly can only handle *rigid* geometries. However, there is an increasing demand for a realistic, yet easy-to-use simulation of *large deformations of slender flexible structures*, preferably in real time (i.e.: *at interactive rates*). Typical examples of such structures from automotive industry are tubes, hoses, single cables, or wiring harnesses collecting many cables within a compound structure (see Fig. 1).

J. Linn (✉) • K. Dreßler
Fraunhofer ITWM, Kaiserslautern, Germany
e-mail: joachim.linn@itwm.fraunhofer.de; klaus.dressler@itwm.fraunhofer.de

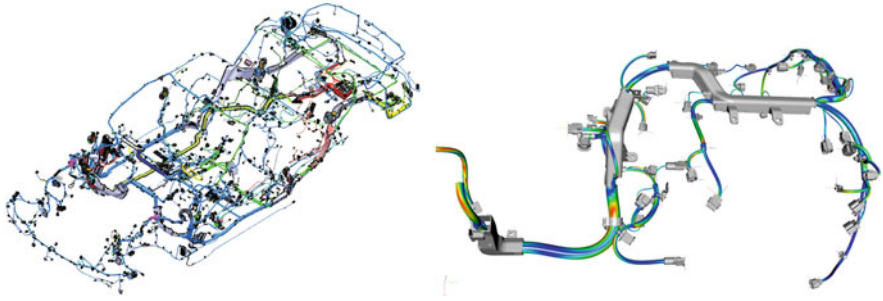


Fig. 1 Overview of the system of cables installed in a car (*left*), and simulation model of a wiring harness in the *IPS Cable Simulation* software (*right*)

The theory of *Cosserat rods* [1, 26, 29] provides a framework for structural models that are suitable for physically correct simulations of deformations of slender flexible objects by *stretching*, *bending* and *twisting*. Due to the slenderness of the geometry—i.e.: typical cross section diameters d are small relative to typical lengths L of the considered structures, such that $d/L \ll 1$ holds—such deformations may possibly imply large spatial displacements and rotations, while the local strains always remain small. Cosserat rod models, also denoted as *geometrically exact* models due to the possibility of a kinematically exact treatment of large rigid body motions, are particularly well suited to handle such large deformations.

In computational mechanics, such models are usually discretized via nonlinear finite elements [13]. This approach is tailored to provide very accurate simulation results. However, due to their algorithmic and algebraic complexity, discrete models constructed via nonlinear FE are technically complicated and in general computationally far too demanding for doing fast simulations compatible with rendering at 25 Hz (at least), simultaneous to an interactive modification of the boundary conditions by the user, either via the graphical user interface of a desktop computer, or via a data glove (or similar devices) within an augmented reality (AR) environment, unless such simulations are executed on highly performant computer hardware, using many processors with multiple cores, and highly parallelized algorithms. Therefore, if one aims at interactive simulations on ordinary desktop computers available to a broader range of users, the development of a different approach is required.

The kinematics of Cosserat rods is closely related to the *differential geometry of framed curves* [3, 7], with the differential invariants of rod configurations corresponding to the strain measures of the mechanical theory [1]. We propose to utilize ideas from the *discrete* differential geometry of framed curves [2, 5, 27, 30] to construct the *discrete kinematics* of Cosserat rod models in a way that preserves the essential geometric properties independent of the coarseness of the discretization.

Different from a nonlinear FE approach aiming at weak solutions of the mechanical equilibrium equations [34], we consider Cosserat rod models within the variational framework of *Lagrangian mechanics* in terms of the kinetic and

elastic energy of the rod [16, 20, 21]. In particular, as the elastic energy density of a rod is given as a quadratic form in the strain measures, we obtain the discrete elastic energy by an approach which we denote as *geometric finite differences*, providing a discretization of the strain measures that preserves their essential geometric properties, in combination with simple quadrature rule to approximate the integrated energy density. Due to the geometric discretization of the strains, discrete rod models constructed according to our discrete Lagrangian mechanics approach behave qualitatively correct even for very coarse discretizations, provide an ultrafast computational performance at moderate accuracy, and thus are suitable for interactive simulations.

In our article, we introduce the basic ideas of our geometry based discretization approach for flexible slender structures as sketched above. The mathematical backbone of our construction of discrete Cosserat rod models is provided by the difference geometry of framed curves in the spirit of Sauer's approach [27] to discrete Frénet curve theory. We present an extension of Sauer's ideas to construct the basic constituents of the discrete geometry of Cosserat curves in Euclidian space, including proper definitions of discrete curvatures, discrete generalized Frénet equations with geometrically exact solutions in terms of finite rotations, all summarized in the formulation of a principal theorem of discrete Cosserat curve theory. On this basis, the construction of discrete Cosserat rod models formulated in terms of discrete elastic energy functions, defined as quadratic forms of the invariants of discrete Cosserat curves, can be obtained in a straightforward manner.

Our discrete formulation of geometrically exact rods turns out to be particularly useful for a seamless integration into a CAE software environment as *IPS Cable Simulation*. As the models and algorithms are formulated in terms of elementary concepts of computational geometry, one can achieve the computational performance necessary for a true interaction of the user with the software in real time, which is a key feature in practical applications. We illustrate this aspect by presenting some typical application examples of assembly simulations of cables performed in automotive industry for design and digital validation purposes.

2 Notational Conventions

In this section we collect a few facts of linear algebra to introduce some notational conventions inspired by the ones given in [1, 14, 25].

2.1 Euclidian Point Space \mathcal{E}^3 and Its Vector Space \mathbb{E}^3

We denote three-dimensional Euclidian point space by \mathcal{E}^3 , its associated Euclidian vector space by \mathbb{E}^3 and use bracket notation $\langle \cdot, \cdot \rangle$ to denote its scalar product. All vectors $\mathbf{w} \in \mathbb{E}^3$ are written in boldface roman letters. By definition, they provide

parallel displacements $q = p + \mathbf{w}$ of points $p, q \in \mathcal{E}^3$. This explains the operation $+$: $\mathcal{E}^3 \times \mathbb{E}^3 \rightarrow \mathcal{E}^3$ on Euclidian space $(\mathcal{E}^3, \mathbb{E}^3, +)$ in a memnonic way, and likewise introduces the difference $q - p = \mathbf{w}$ of points as a proper operation. The distance of points in \mathcal{E}^3 is measured by the length $\|\mathbf{w}\| = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle} =: \|q - p\|$ of their displacement vectors. A fixed cartesian coordinate frame of \mathbb{E}^3 is defined by choosing a fixed origin $\mathcal{O} = \mathbf{0}$ and a fixed right-handed orthonormal triple $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ of basis vectors. Any vector quantity may be decomposed with respect to the fixed basis $\{\mathbf{e}_k\}_{k=1,2,3}$ in the form $\mathbf{w} = \sum_{k=1}^3 w_k \mathbf{e}_k$, where the real numbers $w_k = \langle \mathbf{w}, \mathbf{e}_k \rangle$ denote the cartesian components of $\mathbf{w} \in \mathbb{E}^3$. The position vector $\mathbf{x}(p)$ of a point $p \in \mathcal{E}^3$ is given by $p = \mathcal{O} + \mathbf{x}(p)$, with its cartesian components $x_k(p) = \langle \mathbf{x}(p), \mathbf{e}_k \rangle$.

2.2 Linear Mappings in \mathbb{E}^3

We denote linear mappings $\mathbf{A} : \mathbb{E}^3 \rightarrow \mathbb{E}^3$ within Euclidian vector space by upper case upright serifless letters and use dot notation $\mathbf{w} \mapsto \mathbf{A} \cdot \mathbf{w}$ to indicate their operation on vectors. The composition $(\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{w} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{w})$ of linear mappings is written in the same style. The identity \mathbf{I} maps all vectors onto themselves.

A linear mapping is completely determined by its values $\mathbf{v}_k = \mathbf{A} \cdot \mathbf{e}_k$ on the fixed basis and may be written in invariant form as a sum¹ $\mathbf{A} = \sum_{k=1}^3 \mathbf{v}_k \otimes \mathbf{e}_k \equiv \mathbf{v}_k \otimes \mathbf{e}_k$ of tensor products defined as $(\mathbf{a} \otimes \mathbf{b}) \cdot \mathbf{w} = \langle \mathbf{b}, \mathbf{w} \rangle \mathbf{a}$. The corresponding representation of the identity in terms of the fixed basis is given by $\mathbf{I} = \mathbf{e}_k \otimes \mathbf{e}_k$. Occasionally we use the notation $\mathbf{A} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, which identifies the linear mapping \mathbf{A} with the triple of vectors obtained as images of the fixed basis.

The determinant $\det(\mathbf{A})$ of a linear mapping is an invariant and equals the determinant of its representing matrix w.r.t. an arbitrary basis. The cross product $\mathbf{u} \times \mathbf{v}$ of vectors may be defined invariantly via the identity $\langle \mathbf{u} \times \mathbf{v}, \mathbf{w} \rangle = \det((\mathbf{u}, \mathbf{v}, \mathbf{w})) = [\mathbf{u}, \mathbf{v}, \mathbf{w}]$ which is required to hold for arbitrary vectors, and likewise explains their scalar valued triple product. The identity $\tilde{\mathbf{u}} \cdot \mathbf{v} = \mathbf{u} \times \mathbf{v}$ establishes the one-to-one correspondence between vectors \mathbf{u} and skew-symmetric mappings $\tilde{\mathbf{u}} = -\tilde{\mathbf{u}}^T$, represented by *tilde notation*.

2.3 Orthogonal Mappings

Linear mappings that preserve length are denoted as *orthogonal*: for orthogonal mappings \mathbf{R} the identity $\|\mathbf{w}\| = \|\mathbf{R} \cdot \mathbf{w}\|$ must hold for all vectors $\mathbf{w} \in \mathbb{E}^3$. This implies the orthonormality $\langle \mathbf{a}_i, \mathbf{a}_j \rangle = \delta_{ij}$ of the column vectors $\mathbf{a}_k = \mathbf{R} \cdot \mathbf{e}_k$ of an

¹We make frequent use of *Einstein's summation convention*, with Latin indices i, j, k, \dots running from 1 to 3, and Greek indices α, β, \dots from 1 to 2.

orthogonal mapping. This characteristic property may be equivalently formulated in a more compact form by the identities $\mathbf{R}^T = \mathbf{R}^{-1}$ or $\mathbf{R}^T \cdot \mathbf{R} = \mathbf{I} = \mathbf{R} \cdot \mathbf{R}^T$ which hold by definition for any orthogonal mapping. Orthogonal mappings \mathbf{R} that preserve the orientation of the fixed basis are characterized by $\det(\mathbf{R}) = 1$ and denoted as *proper orthogonal*. The orthogonal and proper orthogonal linear mappings on \mathbb{E}^3 form Lie groups $\mathbf{O}(3)$ and $\mathbf{SO}(3)$ respectively. Their Lie algebra is the set $\mathfrak{so}(3) \simeq \mathbb{R}^3 \simeq \mathbb{E}^3$ of skew-symmetric linear mappings.

2.4 Quaternions

Following ch. 7 of [11], we denote Hamilton’s algebra of *quaternions* by \mathbb{H} . We identify the orthonormal basis $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ of $\mathfrak{N}\mathbb{H} \simeq \mathbb{E}^3$ with the fixed basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ of \mathbb{E}^3 . Denoting the base vector of $\mathfrak{R}\mathbb{H} \simeq \mathbb{R}$ as $\mathbf{e}_0 = 1$, we may represent arbitrary quaternions invariantly as² $\mathbf{q} = q + \mathbf{q}$, with scalar part $q = \mathfrak{R}(\mathbf{q}) \in \mathbb{E}^1 \simeq \mathbb{R}$ and vector part $\mathbf{q} = \mathfrak{N}(\mathbf{q}) \in \mathbb{E}^3 \simeq \mathbb{R}^3$. The product of two arbitrary quaternions \mathbf{p} and \mathbf{q} is given by the formula: $\mathbf{p} \circ \mathbf{q} = pq - \langle \mathbf{p}, \mathbf{q} \rangle + p\mathbf{q} + q\mathbf{p} + \mathbf{p} \times \mathbf{q}$. Using the notation $\mathbf{q}^* = q - \mathbf{q}$ for conjugate quaternions, the scalar product $\langle \cdot, \cdot \rangle_{\mathbb{H}}$ of $\mathbb{H} \simeq \mathbb{E}^4$ may be obtained by $\langle \mathbf{p}, \mathbf{q} \rangle_{\mathbb{H}} = \frac{1}{2}(\mathbf{p} \circ \mathbf{q}^* + \mathbf{q} \circ \mathbf{p}^*) = pq + \langle \mathbf{p}, \mathbf{q} \rangle$, such that $|\mathbf{q}| = \sqrt{\mathbf{q}^* \circ \mathbf{q}} = \sqrt{q^2 + \mathbf{q}^2}$ yields the modulus of a quaternion. All non-zero quaternions have a unique inverse $\mathbf{q}^{-1} = \mathbf{q}^*/|\mathbf{q}|^2$, such that $\mathbf{q} \circ \mathbf{q}^{-1} = 1 = \mathbf{q}^{-1} \circ \mathbf{q}$ holds. Quaternions $\mathbf{q} = \mathbf{q}$ with a scalar part $\mathfrak{R}(\mathbf{q}) = 0$ are called *pure* (or *vector*) quaternions and identified with vectors in \mathbb{E}^3 . As the product of two vector quaternions is given by the simplified formula $\mathbf{p} \circ \mathbf{q} = -\langle \mathbf{p}, \mathbf{q} \rangle + \mathbf{p} \times \mathbf{q}$, their scalar and cross products in \mathbb{E}^3 may be written in terms of quaternion products as: $\langle \mathbf{p}, \mathbf{q} \rangle = -\frac{1}{2}(\mathbf{p} \circ \mathbf{q} + \mathbf{q} \circ \mathbf{p})$, and $\mathbf{p} \times \mathbf{q} = \frac{1}{2}(\mathbf{p} \circ \mathbf{q} - \mathbf{q} \circ \mathbf{p})$.

Proper rotations $\mathbf{R} \in \mathbf{SO}(3)$ may be represented by *unimodular* (or *rotational*) quaternions $\hat{\mathbf{q}} = q + \mathbf{q}$, satisfying $|\hat{\mathbf{q}}|^2 = q^2 + \mathbf{q}^2 = 1$ and therefore located on the unit sphere $S^3 \subset \mathbb{E}^4$, by means of the *Euler map* $\hat{\mathbf{q}} \mapsto \mathbf{R} = \mathfrak{E}(\hat{\mathbf{q}})$ implicitly defined via its operation on vectors $\mathbf{v} \in \mathbb{E}^3 \simeq \mathfrak{N}\mathbb{H}$ as: $\mathbf{R}(\hat{\mathbf{q}}) \cdot \mathbf{v} = \hat{\mathbf{q}} \circ \mathbf{v} \circ \hat{\mathbf{q}}^*$. Thus, the pair $\pm \hat{\mathbf{q}}$ represents the same proper rotation $\mathbf{R}(\hat{\mathbf{q}}) = \mathbf{R}(-\hat{\mathbf{q}})$, consistent with the fact that $S^3 \simeq \mathbf{SU}(2)$ yields a double covering of $\mathbf{SO}(3)$. The definition of $\mathfrak{E}(\hat{\mathbf{q}})$ implies the formulas $\mathbf{R}(\hat{\mathbf{p}}) \cdot \mathbf{R}(\hat{\mathbf{q}}) = \mathbf{R}(\hat{\mathbf{p}} \circ \hat{\mathbf{q}})$ for the composition of rotations and $\mathbf{R}(\hat{\mathbf{q}})^T = \mathbf{R}(\hat{\mathbf{q}}^*)$ for the inverse rotation, as $\mathbf{R}(1) = \mathbf{I}$ holds. According to Euler, each proper rotation may be represented as $\mathbf{R} = \exp(\vartheta \tilde{\mathbf{u}})$, i.e.: a rotation by an angle ϑ around an axis determined by the unit vector $\hat{\mathbf{u}}$, with uniquely determined $\vartheta \in (0, 2\pi)$ and $\hat{\mathbf{u}} \in S^2$ for $\mathbf{R} \neq \mathbf{I}$. The corresponding rotational quaternion is given by $\hat{\mathbf{q}} = \exp(\vartheta/2 \hat{\mathbf{u}}) = \cos(\vartheta/2) + \sin(\vartheta/2) \hat{\mathbf{u}}$, such that $\exp(\vartheta \tilde{\mathbf{u}}) = \mathfrak{E}(\pm \exp(\vartheta/2 \hat{\mathbf{u}}))$ holds identically.

²It is always clear from the context whether a term $q + \mathbf{q}$ refers to the addition of the real and imaginary parts of a quaternion or the parallel displacement of a point in \mathcal{E}^3 by a vector.

2.5 Geometric Curves in Euclidian Space

We regard *geometric curves* as *simple arcs* [30] corresponding to smooth, one-dimensional connected submanifolds.³ Thus, the mapping $\mathcal{C} \ni p \mapsto \xi(p) \in \mathbb{R}$ of the points p on a geometric curve $\mathcal{C} \subset \mathbb{E}^3$ to their real *coordinates* ξ is (at least once) differentiable and invertible, and the inverse mapping $\xi \mapsto p(\xi)$ from open intervals in \mathbb{R} into \mathbb{E}^3 provides a local *parametrization* of the curve. By joining the open intervals of local parametrizations, we obtain a larger one $(a, b) \subset \mathbb{R}$ corresponding to a global parametrization $\phi : [a, b] \rightarrow \mathcal{C}$ of the geometric curve, such that $(a, b) \ni \xi \mapsto p = \phi(\xi)$ yields all interior points of \mathcal{C} , and the two boundary points of \mathcal{C} are given by $\phi(a)$ and $\phi(b)$. The position vectors $\mathbf{x}(p) \in \mathbb{E}^3$ of curve points are then given by a *parameter curve* $\xi \mapsto \mathbf{r}(\xi) := \mathbf{x}(\phi(\xi))$ in \mathbb{E}^3 .

3 Framed Curves and Cosserat Rods

The theory of *Continuum Mechanics of solid bodies* [14, 31] provides proper physical models to simulate deformations of flexible parts. A continuum-mechanical model of a material body consists of three main constituents: *kinematics*, *equilibrium equations* and *constitutive laws*. Summarized briefly, the general programme of continuum mechanics aims at determining *equilibrium configurations* of a body subject to certain *boundary conditions*, such that all external *forces* acting on the body are *in equilibrium* with the internal ones (resulting from deformations of its shape) and inertial effects, making use of *constitutive laws* that relate local changes of shape, measured in terms of *strains*, to *stresses* that encode information on the corresponding local forces.

The theoretical framework provided by Continuum Mechanics as sketched above is a rather complex one, in particular if one is interested to model large (*finite*) deformations of parts w.r.t. their shape in an undeformed state, in contrast to infinitesimally small ones that can be treated by the well known standard models and numerical methods of Linear Elasticity. This seems to be discouraging in view of our goal to simulate large deformations of cables and tube-like parts fast enough to permit interactive action for the users with the simulation model. Fortunately the slender geometry of the parts considered provides the possibility to reduce the continuum model analytically to an object which is well known (and likewise well understood) in classical differential geometry, namely: a *framed curve*.

³One-dimensional connected (sub)manifolds are either *simple arcs* diffeomorphic to an interval, or *simple loops* diffeomorphic to a circle (see the appendix of Milnor's booklet [24] for a proof).

3.1 Basic Differential Geometry of Framed Parameter Curves

More precisely, we consider so called *Cosserat curves*, consisting of [1]

- a space curve $\mathbf{r}(s)$ corresponding to the *centerline* of the rod, and
- a *moving frame* $\mathbf{R}(s) = \mathbf{a}^{(j)}(s) \otimes \mathbf{e}_j$ of orthonormal directors,

where the pair $\{\mathbf{a}^{(1)}(s), \mathbf{a}^{(2)}(s)\}$ spans the local cross section of the rod at the position $\mathbf{r}(s)$, such that $\mathbf{a}^{(3)} = \mathbf{a}^{(1)} \times \mathbf{a}^{(2)}$ equals the unit length *cross section normal vector*, as sketched in Fig. 2.

As $\mathbf{r}(s) \in \mathbb{E}^3$ and $\mathbf{R}(s) \in \text{SO}(3)$, a Cosserat curve may be interpreted as a parameter curve in the manifold $\mathbb{E}^3 \times \text{SO}(3)$ of rigid body configurations in Euclidian space. The curve parameter s is usually assumed to correspond to the *arc length* of $\mathbf{r}(s)$, such that the tangent vector $\mathbf{t}(s) = \mathbf{r}'(s)$ has unit length. The frame $\mathbf{R}(s)$ is called *adapted* to the curve if $\mathbf{a}^{(3)}(s) = \mathbf{t}(s)$ holds. While in the setting of classical differential geometry of framed curves mainly adapted frames are considered, *non-adapted* frames are of primary interest in the kinematical theory of *geometrically exact rods*, where adapted frames merely occur as a special case, recoverable from a kinematically more general Cosserat curve by the *Euler–Bernoulli* constraint $\mathbf{r}'(s) = \mathbf{a}^{(3)}(s)$, enforcing cross sections to remain orthogonal to the tangent vector of an inextensible centerline curve.

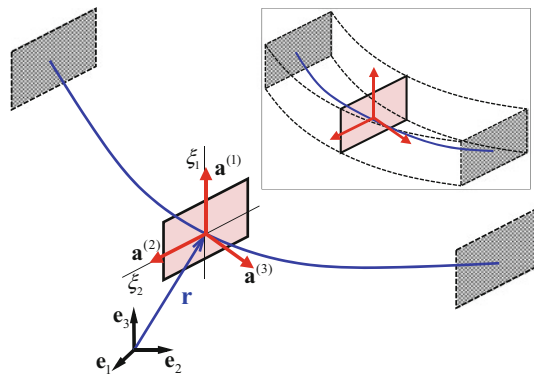


Fig. 2 Centerline curve $\mathbf{r}(s)$ and attached moving frame $\mathbf{R}(s) = \mathbf{a}^{(k)}(s) \otimes \mathbf{e}_k$ of a *Cosserat curve*, describing the geometry of the configurations of a prismatic rod in Euclidian space. The volumetric geometry is generated by sliding the cross section spanned by the frame directors $\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}\}$ along the centerline. The position vectors of the material points in the rod volume are parametrized by: $\mathbf{x} = \mathbf{r}(s) + \xi_\alpha \mathbf{a}^{(\alpha)}(s)$

3.1.1 Frénet Curves and Ribbons

The most well known adapted frame in elementary differential geometry of space curves is the *Frénet frame* $(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)}) = (\mathbf{n}, \mathbf{b}, \mathbf{t})$, consisting of the *principal normal* and *binormal* vectors defined as $\mathbf{n}(s) := \mathbf{t}'(s)/\kappa(s)$ and $\mathbf{b}(s) := \mathbf{t}(s) \times \mathbf{n}(s)$ on intervals of non-zero Frénet curvature $\kappa(s) := \|\mathbf{t}'(s)\|$. More general, one may consider parameter curves on oriented surfaces patches, with the (likewise adapted) *Darboux frame* of the curve defined by the curve tangent \mathbf{t} and the unit length normal vector field \mathbf{N} of the surface at the curve points. This leads to the notion of a *ribbon* (or *surface strip*), defined as a surface patch of infinitesimally small width around a curve $\mathbf{r}(s)$, oriented by a unit vector $\mathbf{N}(s)$ orthogonal to the tangent vector $\mathbf{t}(s)$ of the curve, with an adapted frame given by: $(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)}) = (\mathbf{N}, \mathbf{t} \times \mathbf{N}, \mathbf{t})$.

From a slightly different point of view, one may consider an arbitrary frame field $\mathbf{R}(s)$, given as a parameter curve in $\mathbf{SO}(3)$, and recover its corresponding space curve by integration: $\mathbf{r}'(s) = \mathbf{a}^{(3)}(s) \Leftrightarrow \mathbf{r}(s) = \mathbf{r}_0 + \int_0^s \mathbf{a}^{(3)}(\zeta) d\zeta$. The evolution of the adapted frame of a ribbon along its curve is determined by the generalized Frénet equations $\partial_s \mathbf{a}^{(k)}(s) = \boldsymbol{\kappa}(s) \times \mathbf{a}^{(k)}(s)$. The curvatures $\kappa^{(k)}(s)$ are defined as the components of the *Darboux vector* $\boldsymbol{\kappa} = \kappa^{(j)} \mathbf{a}^{(j)} = \frac{1}{2} \mathbf{a}^{(j)} \times \partial_s \mathbf{a}^{(j)} = \kappa \mathbf{b} + \kappa^{(3)} \mathbf{t}$ w.r.t. the directors of the moving frame. If they are given as continuous functions of arc length, they provide a complete set of differential invariants that determine the geometry of a ribbon up to a global rigid body motion.

3.1.2 Cosserat Curves and Quaternion Frames

Cosserat curves may be considered as natural generalizations of ribbons by omitting the requirement of an adaption of the frame to the curve. In the context of the kinematics of geometrically exact rods one proceeds even one step further by considering regular curves that are not necessarily parametrized by arc length: If one resolves the tangent vector $\mathbf{r}'(\zeta)$ w.r.t. the directors of the moving frame $\mathbf{R}(\zeta) = \mathbf{a}^{(k)}(\zeta) \otimes \mathbf{e}_k$, then the components $\Gamma^{(k)}(\zeta) := \langle \mathbf{a}^{(k)}(\zeta), \mathbf{r}'(\zeta) \rangle$ of the tangent vector, together with the curvatures $K^{(k)}(\zeta)$, which are implicitly given by the frame equations $\partial_\zeta \mathbf{a}^{(k)} = \boldsymbol{\kappa} \times \mathbf{a}^{(k)}$ and associated Darboux vector $\boldsymbol{\kappa}(\zeta) = K^{(j)}(\zeta) \mathbf{a}^{(j)}(\zeta)$, provide a complete set of differential invariants that determine a Cosserat curve up to a global rigid body motion.

The proof of this statement, which constitutes the *principal theorem* of the differential geometry of Cosserat curves, may be obtained by a straightforward adaption of the corresponding one for ribbons (see [1]): For given curvature functions $K^{(j)}(\zeta)$, the frame equations become a system of linear ODEs for the frame directors that can be integrated for an arbitrary initial value $\mathbf{R}_0 \in \mathbf{SO}(3)$ according to the theory of ordinary differential equations. Due to the special algebraic structure of the frame equations, the scalar products of the frame directors are conserved (i.e.: $\langle \mathbf{a}^{(i)}(\zeta), \mathbf{a}^{(j)}(\zeta) \rangle = \delta_{ij}$), such that the solution $\mathbf{R}(\zeta) = \mathbf{a}^{(j)}(\zeta) \otimes \mathbf{e}_j$ always remains in $\mathbf{SO}(3)$. For given $\boldsymbol{\Gamma}(\zeta) := \Gamma^{(j)}(\zeta) \mathbf{e}_j$ and known $\mathbf{R}(\zeta)$, the tangent vector $\mathbf{r}'(\zeta) = \boldsymbol{\Gamma}^{(j)}(\zeta) \mathbf{a}^{(j)}(\zeta) = \mathbf{R}(\zeta) \cdot \boldsymbol{\Gamma}(\zeta)$ can then be considered as a known

function that can subsequently be integrated, which finally yields the space curve $\mathbf{r}(\zeta) = \mathbf{r}_0 + \int_0^\zeta \mathbf{R}(\xi) \cdot \boldsymbol{\Gamma}(\xi) d\xi$ for an arbitrarily chosen initial value \mathbf{r}_0 .

Note that as $\|\mathbf{r}'(\zeta)\| = \|\boldsymbol{\Gamma}(\zeta)\|$ holds, the differential of arc length is given by $ds = \|\boldsymbol{\Gamma}(\zeta)\| d\zeta$, such that one may always reparametrize a regular Cosserat curve by its arc length function $s(\zeta)$, with a corresponding rescaling of the curvatures according to: $\kappa^{(k)}(s) = K^{(k)}(\zeta)/\|\boldsymbol{\Gamma}(\zeta)\|$. For curves parametrized by arc length $\boldsymbol{\Gamma}^{(k)}(s) = \langle \mathbf{a}^{(k)}(s), \mathbf{t}(s) \rangle$ are the direction cosines of the tangent vector w.r.t. the local frame axes. Ribbons consisting of regular curves parametrized by arc length with adapted frames correspond to the special case of constant $\boldsymbol{\Gamma}_0 = (0, 0, 1)^T \equiv \mathbf{e}_3$. Frénet curves may in turn be considered as special cases of ribbons, with their Darboux vector given by $\boldsymbol{\kappa} = \kappa \mathbf{b} + \tau \mathbf{t}$.

The formulation of Cosserat rod models as presented in [20] is based on *quaternionic Cosserat curves*, where the moving frame $\mathbf{R}(s)$ is represented equivalently by a moving unimodular (*rotational*) quaternion field $\hat{\mathbf{q}}(s)$, characterized by the identity $\mathbf{R}(s) \cdot \mathbf{v} = \hat{\mathbf{q}}(s) \circ \mathbf{v} \circ \hat{\mathbf{q}}^*(s)$ holding for arbitrary vectors $\mathbf{v} \in \mathbb{E}^3 \simeq \Im\mathbb{H}$. The generalized Frénet equations can be written equivalently in terms of a derivative equation $\mathbf{R}'(s) = \mathbf{R}(s) \cdot \dot{\mathbf{K}}(s)$ for the moving frame, using the skew matrix $\dot{\mathbf{K}} \in \mathfrak{so}(3)$ associated to the *material Darboux vector* $\mathbf{K}(s) = K^{(j)}(s) \mathbf{e}_j = \mathbf{R}^T(s) \cdot \boldsymbol{\kappa}(s)$.

The corresponding derivative equation for the equivalent quaternion frame is then given by $\hat{\mathbf{q}}'(s) = \frac{1}{2}\boldsymbol{\kappa}(s) \circ \hat{\mathbf{q}}(s) = \frac{1}{2}\hat{\mathbf{q}}(s) \circ \mathbf{K}(s)$. As $\Re(\mathbf{K}) = 0 = \Re(\boldsymbol{\kappa})$, any solution of this ODE has constant modulus. In particular $|\hat{\mathbf{q}}(s)| \equiv 1$ holds for any solution of the frame equation starting from an initial value $\hat{\mathbf{q}}_0 \in S^3$. The recovery formula for the centerline by integration in terms of a solution $\hat{\mathbf{q}}(s)$ of the quaternionic frame equation—determined by the given curvature vector $\mathbf{K}(s)$ and initial value $\hat{\mathbf{q}}_0$ —and given $\boldsymbol{\Gamma}(s)$ may then be reformulated in terms of quaternionic quantities as: $\mathbf{r}(s) = \mathbf{r}_0 + \int_0^s \hat{\mathbf{q}}(\zeta) \circ \boldsymbol{\Gamma}(\zeta) \circ \hat{\mathbf{q}}^*(\zeta) d\zeta$. This implies an equivalent formulation of the principal theorem for quaternionic Cosserat curves, which are likewise determined by given functions $\mathbf{K}(s)$ and $\boldsymbol{\Gamma}(s)$ up to a global rigid body motion.

3.2 Elastic Energy of a Cosserat Rod

Static equilibria of deformed elastic structures can be computed as minima of their elastic energy, subject to the assumed boundary conditions. As we intend to model slender flexible structures as elastic Cosserat rods, we need to specify a corresponding elastic energy function. For linear elastic material behaviour, the elastic (*stored*) energy function of a 3D body is a quadratic form of its Green–Lagrange strain tensor $\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I})$, where $\mathbf{F} = d\Phi(\mathbf{X})$ is the deformation gradient computed as the derivative of the positions $\mathbf{x} = \Phi(\mathbf{X})$ of the material points in the deformed body volume w.r.t. their positions \mathbf{X} in the undeformed body (see [14] for details).

If one computes the deformation gradient and Green–Lagrange strain tensors for the deformed configurations $\mathbf{x} = \mathbf{r}(s) + \xi_\alpha \mathbf{a}^{(\alpha)}(s)$ of a Cosserat rod w.r.t. its

undeformed reference configuration $\mathbf{X} = \mathbf{r}_0(s) + \xi_\alpha \mathbf{a}_0^{(\alpha)}(s)$ given by a smooth regular curve $\mathbf{r}_0(s)$ parametrized by arc length and its adapted frame $\mathbf{R}_0(s) = \mathbf{a}_0^{(k)}(s) \otimes \mathbf{e}_k$ with $\mathbf{r}'_0(s) = \mathbf{a}_0^{(k)}(s)$, one obtains [22] an exact closed form expression for \mathbf{E} which depends on the differences $\mathbf{K}(s) - \mathbf{K}_0(s)$ and $\mathbf{\Gamma}(s) - \mathbf{\Gamma}_0$ (with $\mathbf{\Gamma}_0 = (0, 0, 1)^T$) of the invariants of the framed curve in their deformed and undeformed configurations.

For slender rod geometries, one may always assume that the local strains remain small, although the deformations of the rod configuration correspond to large (*finite*) rotations and displacements in space. In this case one may approximate the exact expression for \mathbf{E} by taking only the leading order terms in the differences of the invariants into account. The resulting approximated energy density can then be integrated analytically over the cross section coordinates (ξ_1, ξ_2) in closed form, which finally yields [22] the elastic energy \mathcal{W}_{el} of a Cosserat rod as a quadratic functional in the differences $\mathbf{K} - \mathbf{K}_0$ and $\mathbf{\Gamma} - \mathbf{\Gamma}_0$ of the invariants, given by the sum $\mathcal{W}_{el} = \mathcal{W}_{es} + \mathcal{W}_{bt}$ of the two integrals

$$\mathcal{W}_{es} = \frac{1}{2} \int_0^L ds [EA] (\Gamma^{(3)}(s) - 1)^2 + [GA_\alpha] \Gamma^{(\alpha)}(s)^2, \quad (1)$$

$$\mathcal{W}_{bt} = \frac{1}{2} \int_0^L ds [EI_\alpha] \left(K^{(\alpha)}(s) - K_0^{(\alpha)}(s) \right)^2 + [GJ] \left(K^{(3)}(s) - K_0^{(3)}(s) \right)^2. \quad (2)$$

The first term (1) represents the elastic energy related to rod deformations by longitudinal *extension* combined with transverse *shearing*, the second term (2) accounts for the elastic energy stored in *bending* and *torsional* deformations of the rod.

The parameters $[EA]$, $[GA_\alpha]$, $[EI_\alpha]$ and $[GJ]$ quantify the *effective stiffness properties* of the local cross section of the rod related to the respective deformation mode. They may be constants, or vary along the rod as functions of s . In the simple case of a homogeneous and isotropic material characterized by the elastic moduli E and G , they are given as products of the moduli and geometric quantities (i.e.: area A , area moments I_α , polar moment J) of the cross section.

To discretize the energy integrals (1) and (2) we need a discrete model of framed curves with discrete versions of their invariants \mathbf{K} and $\mathbf{\Gamma}$.

4 The Difference Geometry of Framed Curves

In this section we inductively develop our approach to construct the discrete kinematics of geometrically exact rod models by “*geometric finite differences*”. We use concepts and results of the differential geometry of framed curves in three-dimensional Euclidian space, and introduce ideas from *Discrete Differential*

*Geometry (DDG)*⁴ to construct their discrete counterparts in a particular way, such that essential properties of the continuum theory are preserved in the discrete setting.

Below we consider basic concepts of the elementary differential geometry of parameter curves in Euclidian space, as presented in standard texts (e.g. do Carmo’s book [9]), from the *geometric* viewpoint emphasized throughout Blaschke’s books [3, 4], and outline some essential ideas how to transfer the continuous concepts to the discrete setting, following and extending ideas of Sauer [27].

4.1 Discrete Arc Length and Edge Tangent Vectors

The mutual distance of points on a smooth curve can be measured by *unbending* the curve to a straight line, such that the distance of the same points on the straightened curve equals their Euclidian distance in space.

This procedure of continuously “unrolling” a smooth curve to the real axis can be understood most easily for the simplified case of a *discrete approximation* of the curve by a *polygonal arc*, given by a sequence of curve points $p_j = \phi(\xi_j)$ obtained from a given discretization $a =: \xi_0 < \xi_1 < \dots < \xi_n := b$ of the parameter interval with position vectors $\mathbf{r}_j := \mathbf{r}(\xi_j) \equiv \mathbf{x}(p_j)$. The corresponding polygonal arc is the piecewise linear curve in \mathcal{E}^3 defined as the union $\mathcal{P}_n[p_0, \dots, p_n] := \cup_{j=1}^n [p_{j-1}, p_j]$ of edges $[p_{j-1}, p_j] := \{p \in \mathcal{E}^3 | p = p_{j-1} + \lambda(p_j - p_{j-1}), 0 \leq \lambda \leq 1\}$ that are spanned by pairs of adjacent points (*vertices*) p_{j-1} and $p_j = p_{j-1} + \mathbf{l}_{j-\frac{1}{2}}$, linked by *edge vectors* $\mathbf{l}_{j-\frac{1}{2}} := p_j - p_{j-1} = \mathbf{r}_j - \mathbf{r}_{j-1}$ of length $\ell_{j-\frac{1}{2}} := \|\mathbf{l}_{j-\frac{1}{2}}\| = \|\mathbf{r}_j - \mathbf{r}_{j-1}\|$.

Then, the distance of any pair (p_k, p_l) of vertices (with $k < l$), measured along the path of the polygonal arc \mathcal{P}_n , is given by the sum $\sum_{j=k+1}^l \ell_{j-\frac{1}{2}}$ of edge lengths in between, which equals the Euclidian distance of (p_k, p_l) if the polygonal arc is straightened out to a line. If the discretization is refined, the polygonal arc \mathcal{P}_n approximates the curve \mathcal{C} with increasing accuracy, provided the curve is sufficiently smooth (i.e.: at least Lipschitz continuous). According to the (approximate) identity

$$\sum_{j=k+1}^l \ell_{j-\frac{1}{2}} = \sum_{j=k+1}^l \left\| \frac{\mathbf{r}_j - \mathbf{r}_{j-1}}{\xi_j - \xi_{j-1}} \right\| (\xi_j - \xi_{j-1}) \approx \sum_{j=k+1}^l \|\mathbf{r}'(\xi_{j-1/2})\| (\xi_j - \xi_{j-1}),$$

the sum of edge lengths may be interpreted as a discrete approximation of the continuous integral $\int_{\xi_k}^{\xi_l} \|\mathbf{r}'(\xi)\| d\xi = \sum_{j=k+1}^l \int_{\xi_{j-1}}^{\xi_j} \|\mathbf{r}'(\xi)\| d\xi$ by evaluating the integral over the intervals $[\xi_{j-1}, \xi_j]$ of length $h_{j-1/2} := \xi_j - \xi_{j-1}$ approximately by

⁴DDG is an interdisciplinary field in applied mathematics which emerged rather recently at the borderline of differential geometry and discrete computational geometry, with the majority of applications in computer graphics. For a survey, we refer to the articles collected in the book [6].

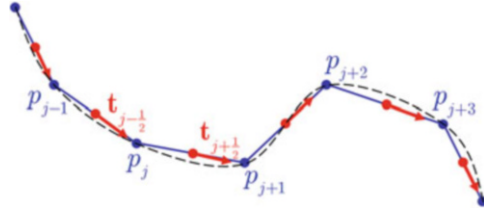


Fig. 3 Polygonal arc $\mathcal{P}_n[p_0, \dots, p_n]$ approximating a smooth regular geometric curve \mathcal{C} : The vertices $p_j \in \mathcal{C}$ define the edges $[p_{j-1}, p_j]$ of length $\ell_{j-1/2} > 0$, with unit length tangent vectors $\mathbf{t}_{j-1/2} = (p_j - p_{j-1})/\ell_{j-1/2}$ located at edge centers $\bar{p}_{j-1/2}$

the midpoint rule according to $\int_{\xi_{j-1}}^{\xi_j} \|\mathbf{r}'(\xi)\| d\xi \approx \|\mathbf{r}'(\xi_{j-1/2})\| h_{j-1/2} + O(h_{j-1/2}^3)$, and approximating the derivative $\mathbf{r}'(\xi)$ at the midpoints $\xi_{j-1/2} := \frac{1}{2}(\xi_{j-1} + \xi_j)$ by a central finite difference as $\mathbf{r}'(\xi_{j-1/2}) \approx (\mathbf{r}_j - \mathbf{r}_{j-1})/h_{j-1/2} + O(h_{j-1/2}^2)$.

Likewise, the position vector $\mathbf{r}_{j-1/2} := \frac{1}{2}(\mathbf{r}_{j-1} + \mathbf{r}_j) \equiv \mathbf{x}(\bar{p}_{j-1/2})$ of the *edge center* approximates the position vector $\mathbf{r}(\xi_{j-1/2})$ at the midpoint of the parameter interval with second order accuracy. Thus, considering the polygonal approximation of a curve naturally leads to the concept of *edge based tangent vectors* $\mathbf{l}_{j-1/2}/h_{j-1/2}$, with unit length edge tangent vectors given by $\mathbf{t}_{j-1/2} := \mathbf{l}_{j-1/2}/\ell_{j-1/2}$, both located at *edge centers*. Requiring that consecutive vertices p_j and p_{j+1} must not coincide, which in turn implies non-zero edge vectors (i.e.: $\|\mathbf{l}_{j+1/2}\| > 0 \Leftrightarrow p_j \neq p_{j+1}$) and unit vectors $\mathbf{t}_{j-1/2}$ well defined for all edges, corresponds to the definition of *discrete regularity* of a polygonal arc (see Fig. 3).

While the mapping $k \mapsto p_k$ of integer indices to vertex points in Euclidian space may be interpreted as a *discrete geometric curve* that induces a corresponding mapping $k \mapsto \mathbf{r}_k \equiv \mathbf{x}(p_k)$ of indices to position vectors, a *discrete parameter curve* is defined by the mapping $\xi_k \mapsto \mathbf{r}(\xi_k)$ induced by a discretization of the domain of a smooth parameter curve. Therefore, the discrete counterpart of arc length parametrisation corresponds to the case $h_{k-1/2} = \ell_{k-1/2}$ of grid constants equal to edge lengths, with *discrete arc length parameters* defined as $\varsigma_k := \varsigma_0 + \sum_{j=1}^k \ell_{j-1/2}$, marking the vertex positions of the polygonal arc straightened out parallel to the real axis. The main concepts introduced in this section may be summarized as follows:

A *discrete geometric curve* is a mapping $\mathbb{Z} \ni j \mapsto p_j \in \mathcal{E}^3$ of integer indices to points in Euclidian space. The discrete curve is *regular* iff $p_j \neq p_{j-1}$ holds for all vertex pairs (p_{j-1}, p_j) . A discrete regular geometric curve has well defined unit tangent vectors $\mathbf{t}_{j-1/2} = (p_j - p_{j-1})/\ell_{j-1/2}$ on all edges (see Fig. 3).

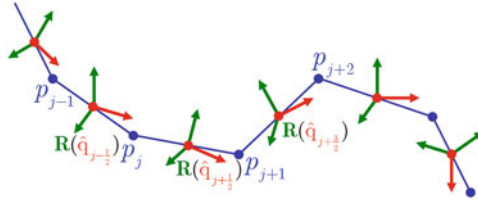


Fig. 4 Polygonal arc $\mathcal{P}_n[p_0, \dots, p_n]$ and edge based quaternionic frames $\hat{q}_{j-1/2} \equiv R_{j-1/2} = \mathfrak{E}(\hat{q}_{j-1/2})$ of a discrete Cosserat curve. In general, the frames are not adapted to the edges of \mathcal{P}_n , i.e.: $\mathbf{a}_{j-1/2}^{(3)} \neq \mathbf{t}_{j-1/2}$. A discrete ribbon is a special case of a discrete Cosserat curve with adapted frames

4.2 The Difference Geometry of Discrete Cosserat Curves

Our discussion of discrete regular geometric curves with edge centered unit tangent vectors indicates the path to introduce Cosserat curves in the discrete setting:

A discrete Cosserat curve is defined as a polygonal arc $\mathcal{P}_n[p_0, \dots, p_n]$ corresponding to a regular discrete geometric curve, augmented by a set $\{R_{j-1/2}\}_{j=1, \dots, n}$ of orthonormal frames $\text{SO}(3) \ni R_{j-1/2} = \mathbf{a}_{j-1/2}^{(k)} \otimes \mathbf{e}_k$ located at edge centers $\bar{p}_{j-1/2}$. The frames are represented by rotational quaternions as $R_{j-1/2} = \mathfrak{E}(\hat{q}_{j-1/2})$ in terms of the Euler map $\mathfrak{E} : S^3 \rightarrow \text{SO}(3)$ implicitly defined via its operation $\mathfrak{E}(\hat{q})\mathbf{v} = \hat{q} \circ \mathbf{v} \circ \hat{q}^*$ on vectors, with frame directors given by: $\mathbf{a}_{j-1/2}^{(k)} = R_{j-1/2} \cdot \mathbf{e}_k = \hat{q}_{j-1/2} \circ \mathbf{e}_k \circ \hat{q}_{j-1/2}^*$ (see Fig. 4).

4.2.1 Curvature Angles and Discrete Curvatures

For each pair of frames $R_{j\pm 1/2}$, there is a unique spatial difference rotation⁵ connecting these frames as $R_{j+1/2} = \mathbf{w}_j \cdot R_{j-1/2}$. According to Euler's theorem, this rotation $\mathbf{w}_j = R_{j+1/2} \cdot R_{j-1/2}^T$ can be written as $\mathbf{w}_j = \exp(\vartheta_j \tilde{\mathbf{u}}_j)$, i.e.: a rotation by an angle ϑ_j around the axis given by the unit vector $\hat{\mathbf{u}}_j$. The corresponding rotational quaternion connects the quaternion frames via $\hat{q}_{j+1/2} = \hat{\mathbf{w}}_j \circ \hat{q}_{j-1/2}$ and is given by:

$$\hat{q}_{j+1/2} \circ \hat{q}_{j-1/2}^* =: \hat{\mathbf{w}}_j = \cos(\vartheta_j/2) + \sin(\vartheta_j/2) \hat{\mathbf{u}}_j = \exp(\vartheta_j/2 \hat{\mathbf{u}}_j).$$

⁵As the group operation on $\text{SO}(3)$ is a product, the terminology *quotient rotation* would be more appropriate. However, we prefer the term *difference rotation* to emphasize the analogy to FD discretization of vectors in Euclidian space.

The *material difference rotation* given by $\mathbf{W}_j := \mathbf{R}_{j-1/2}^T \cdot \mathbf{R}_{j+1/2}$ may be obtained from the spatial one by a pull back rotation with either of the frames $\mathbf{R}_{j\pm 1/2}$, i.e.: $\mathbf{W}_j = \mathbf{R}_{j\pm 1/2}^T \cdot \mathbf{w}_j \cdot \mathbf{R}_{j\pm 1/2}$, and can be written as a rotation $\mathbf{W}_j = \exp(\vartheta_j \hat{\mathbf{U}}_j)$ by the angle ϑ_j about the back rotated axis⁶ $\hat{\mathbf{U}}_j := \mathbf{R}_{j\pm 1/2}^T \cdot \hat{\mathbf{u}}_j$. In terms of rotational quaternions, the equivalent relations for $\hat{\mathbf{W}}_j = \hat{\mathbf{q}}_{j\pm 1/2}^* \circ \hat{\mathbf{w}}_j \circ \hat{\mathbf{q}}_{j\pm 1/2}$ read:

$$\hat{\mathbf{q}}_{j-1/2}^* \circ \hat{\mathbf{q}}_{j+1/2} =: \hat{\mathbf{W}}_j = \cos(\vartheta_j/2) + \sin(\vartheta_j/2) \hat{\mathbf{U}}_j = \exp(\vartheta_j/2 \hat{\mathbf{U}}_j) .$$

Extraction of the *material rotation vector* $\vartheta_j \hat{\mathbf{U}}_j = 2 \log(\hat{\mathbf{W}}_j)$ from the quaternionic difference rotation $\hat{\mathbf{W}}_j$ then leads to the following definition of *curvature angles*:

$$\psi_j^{(k)} := \langle \mathbf{e}_k, 2 \log(\hat{\mathbf{W}}_j) \rangle = \vartheta_j \langle \mathbf{e}_k, \hat{\mathbf{U}}_j \rangle \Leftrightarrow \vartheta_j \hat{\mathbf{U}}_j = \psi_j^{(k)} \mathbf{e}_k . \quad (3)$$

As $\hat{\mathbf{U}}_j := \mathbf{R}_{j\pm 1/2}^T \cdot \hat{\mathbf{u}}_j$ and $\mathbf{a}_{j\pm 1/2}^{(k)} = \mathbf{R}_{j\pm 1/2} \cdot \mathbf{e}_k$ hold, one obtains the angles $\psi_j^{(k)}$ equivalently by decomposing $\vartheta_j \hat{\mathbf{u}}_j = 2 \log(\hat{\mathbf{w}}_j)$ w.r.t. the frame directors, i.e.: $\psi_j^{(k)} = \langle \mathbf{a}_{j\pm 1/2}^{(k)}, 2 \log(\hat{\mathbf{w}}_j) \rangle = \vartheta_j \langle \mathbf{a}_{j\pm 1/2}^{(k)}, \hat{\mathbf{u}}_j \rangle$. The set $\{\vartheta_j \hat{\mathbf{U}}_j\}_{j=1, \dots, n}$ of material rotation vectors (or equivalently: the set $\{\psi_j^{(k)}\}_{j=1, \dots, n}^{k=1, 2, 3}$ of curvature angles) corresponds to the discrete data of the set of quaternion frames $\{\hat{\mathbf{q}}_{j-1/2}\}_{j=1, \dots, n}$ of a discrete Cosserat curve. The frames can be reconstructed iteratively by the material difference rotations as $\hat{\mathbf{q}}_{j-1/2} \mapsto \hat{\mathbf{q}}_{j+1/2} = \hat{\mathbf{q}}_{j-1/2} \circ \exp(\vartheta_j/2 \hat{\mathbf{U}}_j)$, or likewise equivalently by spatial ones according to the algorithm

$$\hat{\mathbf{q}}_{j-1/2} \rightarrow \hat{\mathbf{w}}_j = \hat{\mathbf{q}}_{j-1/2} \circ \exp(\vartheta_j/2 \hat{\mathbf{U}}_j) \circ \hat{\mathbf{q}}_{j-1/2}^* \rightarrow \hat{\mathbf{q}}_{j+1/2} = \hat{\mathbf{w}}_j \circ \hat{\mathbf{q}}_{j-1/2} , \quad (4)$$

for $j = 1, \dots, n-1$, starting at $\hat{\mathbf{q}}_{1/2}$ chosen as initial value, and proceeding edgewise in ascending order.

Discrete material curvatures $K_j^{(k)}$ can then be defined by dividing the curvature angles by the discrete arc length distance $\Delta \zeta_j$ between the edge centers $\bar{p}_{j\pm 1/2}$:

$$K_j^{(k)} := \frac{\psi_j^{(k)}}{\Delta \zeta_j} = \Delta \zeta_j^{-1} \langle \mathbf{e}_k, 2 \log(\hat{\mathbf{W}}_j) \rangle = \frac{\vartheta_j}{\Delta \zeta_j} \langle \mathbf{e}_k, \hat{\mathbf{U}}_j \rangle . \quad (5)$$

If the discrete Cosserat curve approximates a smooth one, the vector $\mathbf{K}_j = K_j^{(k)} \mathbf{e}_k$ of discrete material curvatures converges to the material Darboux vector $\mathbf{K}(p_j)$ in the limit $\Delta \zeta_j \rightarrow 0$. In this case, the curvature angles can be interpreted as integrated values, approximating the integrals $\int_{p_{j-1/2}}^{p_{j+1/2}} \langle \mathbf{e}_k, 2\hat{\mathbf{q}}^* \circ d\hat{\mathbf{q}}_p \rangle \approx \psi_j^{(k)}$ of the curvature

⁶Note that the vectors $\hat{\mathbf{u}}_j$ and $\hat{\mathbf{U}}_j$ are eigenvectors of the difference rotations \mathbf{w}_j and \mathbf{W}_j with eigenvalue 1, which implies the identities $\mathbf{R}_{j-1/2}^T \cdot \hat{\mathbf{u}}_j = \mathbf{R}_{j+1/2}^T \cdot \hat{\mathbf{u}}_j$ and $\mathbf{R}_{j-1/2} \cdot \hat{\mathbf{U}}_j = \mathbf{R}_{j+1/2} \cdot \hat{\mathbf{U}}_j$.

1-form $\mathbf{t}(p) \mapsto 2\hat{\mathbf{q}}^*(p) \circ d\hat{\mathbf{q}}_p(\mathbf{t}(p)) = \mathbf{K}(p)$, and the discrete curvatures $K_j^{(k)} = \psi_j^{(k)} / \Delta\zeta_j$ with $\Delta\zeta_j \approx \int_{p_{j-1/2}}^{p_j+1/2} ds$ correspond to approximate integral averages.

Alternative discrete curvature expressions that approximate $\mathbf{K}(p_j) \approx \mathbf{K}_j$ according to the relations $\vartheta_j \approx 2 \sin(\vartheta_j/2) \approx 2 \tan(\vartheta_j/2)$ valid for small angles can be obtained directly from the imaginary and real parts $\Im(\hat{\mathbf{W}}_j) = \sin(\vartheta_j/2) \hat{\mathbf{U}}_j$ and $\Re(\hat{\mathbf{W}}_j) = \cos(\vartheta_j/2)$ of the material difference rotation. The simplest examples are: $\Delta\zeta_j \mathbf{K}_j \approx 2\Im(\hat{\mathbf{W}}_j) = 2 \sin(\vartheta_j/2) \hat{\mathbf{U}}_j$ and $\Delta\zeta_j \mathbf{K}_j \approx 2\Im(\hat{\mathbf{W}}_j) / \Re(\hat{\mathbf{W}}_j) = 2 \tan(\vartheta_j/2) \hat{\mathbf{U}}_j$. A particularly interesting variant, which appears within the derivation of discrete derivative equations for quaternionic frames, is given by:

$$\Delta\zeta_j \mathbf{K}_j \approx 4 \Im(\hat{\mathbf{W}}_j) / (1 + \Re(\hat{\mathbf{W}}_j)) = 4 \tan(\vartheta_j/4) \hat{\mathbf{U}}_j. \tag{6}$$

4.2.2 Material Edge Components, Edge Tangent Cosines and Shear Angles

The second set of data of a discrete Cosserat curve is given by the components

$$\ell_{j-1/2}^{(k)} := \langle \mathbf{a}_{j-1/2}^{(k)}, \mathbf{l}_{j-1/2} \rangle = \ell_{j-1/2} \langle \mathbf{a}_{j-1/2}^{(k)}, \mathbf{t}_{j-1/2} \rangle \tag{7}$$

of the edge vectors $\mathbf{l}_{j-1/2} = \ell_{j-1/2} \mathbf{t}_{j-1/2}$ w.r.t. the local frame, which we denote as *material edge components*. The squared edge lengths can be obtained as the sum $\ell_{j-1/2}^2 = \sum_{k=1}^3 (\ell_{j-1/2}^{(k)})^2$ of the squared material edges components. Considering $\{\ell_{j-1/2}^{(k)}\}_{j=1, \dots, n}^{k=1, 2, 3}$ as given data, the edge lengths $\ell_{j-1/2}$ and the discrete arc length parameters $\zeta_k = \zeta_0 + \sum_{j=1}^k \ell_{j-1/2}$ can be computed from these.

According to (7) the components $\ell_{j-1/2}^{(k)}$ are given as products of the edge lengths and the *edge tangent cosines*

$$\Gamma_{j-1/2}^{(k)} := \frac{\ell_{j-1/2}^{(k)}}{\ell_{j-1/2}} = \langle \mathbf{a}_{j-1/2}^{(k)}, \mathbf{t}_{j-1/2} \rangle = \langle \mathbf{e}_k, \hat{\mathbf{q}}_{j-1/2}^* \circ \mathbf{t}_{j-1/2} \circ \hat{\mathbf{q}}_{j-1/2} \rangle. \tag{8}$$

The *shear angles* $\gamma_{j-1/2}^{(k)} = \arccos(\Gamma_{j-1/2}^{(k)})$ measure the angles enclosed by the edge vector and the frame axes. The edge tangent cosines (8) converge to the direction cosines $\langle \mathbf{a}^{(k)}(p), \mathbf{t}(p) \rangle$ of the unit tangent vector $\mathbf{t}(p)$ w.r.t. the frame directors $\mathbf{a}^{(k)}(p)$ at $p = p_{j-1/2}$ in the limit $\ell_{j-1/2} \rightarrow 0$.

The set $\{p_j\}_{j=0, \dots, n}$ of vertices of the polygonal arc \mathcal{P} can be obtained by summation of the edge vectors $\mathbf{l}_{j-1/2}$ according to $p_j = p_0 + \sum_{i=1}^j \mathbf{l}_{i-1/2}$, starting from an arbitrarily chosen initial value $p_0 \in \mathcal{E}^3$. As $\mathbf{l}_{j-1/2} = \langle \mathbf{a}_{j-1/2}^{(k)}, \mathbf{l}_{j-1/2} \rangle \mathbf{a}_{j-1/2}^{(k)} = \ell_{j-1/2}^{(k)} \mathbf{a}_{j-1/2}^{(k)}$ and $\mathbf{a}_{j-1/2}^{(k)} = \hat{\mathbf{q}}_{j-1/2} \circ \mathbf{e}_k \circ \hat{\mathbf{q}}_{j-1/2}^*$, the sum for the computation of the

vertices from edge vectors may be rewritten in the form

$$p_j - p_0 = \sum_{i=1}^j \ell_{i-1/2} \mathbf{t}_{i-1/2} = \sum_{i=1}^j \ell_{i-1/2}^{(k)} \hat{\mathbf{q}}_{i-1/2} \circ \mathbf{e}_k \circ \hat{\mathbf{q}}_{i-1/2}^* \quad (9)$$

as an expression in terms of known quantities.

The sum (9) yields a discrete approximation of the integral $p_j - p_0 = \int_{p_0}^{p_j} dp$ with $dp = \mathbf{t}(p) ds$, which can be rewritten equivalently as $\int_{p_0}^{p_j} \hat{\mathbf{q}} \circ \boldsymbol{\Gamma} \circ \hat{\mathbf{q}}^* ds$ in terms of the quaternion frame $\hat{\mathbf{q}}(p)$ and $\boldsymbol{\Gamma}(p) = \mathbf{R}^T(p) \cdot \mathbf{t}(p) = \hat{\mathbf{q}}^*(p) \circ \mathbf{t}(p) \circ \hat{\mathbf{q}}(p)$. The second sum term in (9) corresponds to an approximation of the integrals $\int_{p_{i-1}}^{p_i} \hat{\mathbf{q}} \circ \boldsymbol{\Gamma} \circ \hat{\mathbf{q}}^* ds$ by the midpoint rule as $\ell_{i-1/2} \hat{\mathbf{q}}_{i-1/2} \circ \boldsymbol{\Gamma}_{i-1/2} \circ \hat{\mathbf{q}}_{i-1/2}^*$, with $\boldsymbol{\Gamma}_{i-1/2} := \boldsymbol{\Gamma}_{i-1/2}^{(k)} \mathbf{e}_k$. The discrete integration formula provided by (9) is equivalent to the difference equation

$$(p_j - p_{j-1})/\ell_{j-1/2} = \hat{\mathbf{q}}_{j-1/2} \circ \boldsymbol{\Gamma}_{j-1/2} \circ \hat{\mathbf{q}}_{j-1/2}^* \quad (10)$$

discretizing the derivative equation $dp = \hat{\mathbf{q}} \circ \boldsymbol{\Gamma} \circ \hat{\mathbf{q}}^* ds$.

4.2.3 Discrete Generalized Frénet Equations

The directors $\mathbf{a}_{j\pm 1/2}^{(k)}$ of adjacent frames $\mathbf{R}_{j\pm 1/2} = \mathbf{a}_{j\pm 1/2}^{(k)} \otimes \mathbf{e}_k$ are connected by spatial difference rotations $\mathbf{w}_j = \exp(\vartheta_j \tilde{\mathbf{u}}_j) = \text{cay}(\tan(\vartheta_j/2) \tilde{\mathbf{u}}_j)$ and satisfy the equations $\mathbf{a}_{j+1/2}^{(k)} = \mathbf{w}_j \cdot \mathbf{a}_{j-1/2}^{(k)}$, which may be rewritten equivalently as:

$$\mathbf{a}_{j+1/2}^{(k)} - \mathbf{a}_{j-1/2}^{(k)} = 2 \tan(\vartheta_j/2) \hat{\mathbf{u}}_j \times \frac{1}{2} (\mathbf{a}_{j-1/2}^{(k)} + \mathbf{a}_{j+1/2}^{(k)}) . \quad (11)$$

Divided by $\Delta\zeta_j$, this coupled system of difference equations corresponds to a geometric FD discretization of the generalized Frénet equations $d\mathbf{a}_p^{(k)} = \boldsymbol{\kappa} \times \mathbf{a}^{(k)}$ satisfied by the frame directors of a Cosserat curve.

According to (5) the FD approximation of the material Darboux vector at p_j is given by $\mathbf{K}_j = (\vartheta_j/\Delta\zeta_j) \hat{\mathbf{u}}_j \approx \mathbf{K}(p_j)$. The corresponding spatial Darboux vector then results from a forward rotation according to $\boldsymbol{\kappa}_j = \mathbf{R}_{j\pm 1/2} \cdot \mathbf{K}_j$ and is therefore given by: $\boldsymbol{\kappa}_j = (\vartheta_j/\Delta\zeta_j) \hat{\mathbf{u}}_j \approx \boldsymbol{\kappa}(p_j)$. The discrete spatial Darboux vector identified from (11) equals $\Delta\zeta_j^{-1} 2 \tan(\vartheta_j/2) \hat{\mathbf{u}}_j \approx \boldsymbol{\kappa}_j$, which provides an alternative consistent FD approximation of $\boldsymbol{\kappa}(p_j)$ that coincides with $\boldsymbol{\kappa}_j$ in the limit $\Delta\zeta_j \rightarrow 0$.

Averaged Frame Director Expansion of the Discrete Spatial Darboux Vector

For $\mathbf{R}_{j\pm 1/2} \in \text{SO}(3)$ the matrix $\mathbf{R}_{j-1/2} + \mathbf{R}_{j+1/2} = \mathbf{R}_{j-1/2} \cdot (\mathbf{I} + \mathbf{W}_j)$ is regular if the eigenvalues of \mathbf{W}_j are different from -1 , which is always the case for curvature

angles $|\psi_j^{(k)}| < \pi$. Then the *averaged frame directors* $\bar{\mathbf{a}}_j^{(k)} := \frac{1}{2}(\mathbf{a}_{j-1/2}^{(k)} + \mathbf{a}_{j+1/2}^{(k)})$ associated to vertex p_j , which appear on the r.h.s. of (11), are linearly independent and form a basis of \mathbb{E}^3 . Although they do not form an orthonormal frame, they become orthonormal in the limit $\Delta\zeta_j \rightarrow 0$, as $\lim_{\Delta\zeta_j \rightarrow 0} \bar{\mathbf{a}}_j^{(k)} = \mathbf{a}^{(k)}(p_j)$ holds. The *discrete generalized Frénet equations* (11) can be rewritten with greater formal similarity to their continuous counterpart by expanding the spatial Darboux vector $\boldsymbol{\kappa}_j \sim \hat{\mathbf{u}}_j$ w.r.t. the *dual basis vectors* $\bar{\mathbf{b}}_j^{(k)}$ implicitly defined as $\langle \bar{\mathbf{b}}_j^{(k)}, \bar{\mathbf{a}}_j^{(l)} \rangle = \delta_{kl}$ and explicitly given by $\bar{\mathbf{b}}_j^{(k)} = (\bar{\mathbf{a}}_j^{(l)} \times \bar{\mathbf{a}}_j^{(m)}) / \langle \bar{\mathbf{a}}_j^{(l)} \times \bar{\mathbf{a}}_j^{(m)}, \bar{\mathbf{a}}_j^{(3)} \rangle$ for cyclic permutations (klm) of (123). By construction $\lim_{\Delta\zeta_j \rightarrow 0} \bar{\mathbf{b}}_j^{(k)} = \mathbf{a}^{(k)}(p_j)$ holds. The representation $\boldsymbol{\kappa}(p_j) \approx \Delta\zeta_j^{-1} 2 \tan(\vartheta_j/2) \langle \bar{\mathbf{b}}_j^{(k)}, \hat{\mathbf{u}}_j \rangle \bar{\mathbf{a}}_j^{(k)}$ of the discrete spatial Darboux vector is obtained by using the identity $\hat{\mathbf{u}}_j = \langle \bar{\mathbf{b}}_j^{(k)}, \hat{\mathbf{u}}_j \rangle \bar{\mathbf{a}}_j^{(k)}$ and implies the corresponding approximation $K^{(k)}(p_j) \approx \Delta\zeta_j^{-1} 2 \tan(\vartheta_j/2) \langle \bar{\mathbf{b}}_j^{(k)}, \hat{\mathbf{u}}_j \rangle$ of the material curvatures at the vertices. Inserting this expansion into (11) leads to the equivalent reformulation

$$\mathbf{a}_{j+1/2}^{(k)} - \mathbf{a}_{j-1/2}^{(k)} = 2 \tan(\vartheta_j/2) \left[\langle \bar{\mathbf{b}}_j^{(m)}, \hat{\mathbf{u}}_j \rangle \bar{\mathbf{a}}_j^{(m)} - \langle \bar{\mathbf{b}}_j^{(l)}, \hat{\mathbf{u}}_j \rangle \bar{\mathbf{a}}_j^{(l)} \right]$$

of these equations for cyclic permutations (klm) of (123), as the Darboux vector component $\sim \bar{\mathbf{a}}_j^{(k)}$ is canceled due to the cross product. Dividing both sides of this FD equation for $\mathbf{a}_{j\pm 1/2}^{(k)}$ by $\Delta\zeta_j$ and taking the limit $\Delta\zeta_j \rightarrow 0$ yields the well known form $d\mathbf{a}_p^{(k)} = \boldsymbol{\kappa} \times \mathbf{a}^{(k)} = K^{(m)}\mathbf{a}^{(m)} - K^{(l)}\mathbf{a}^{(l)}$ of the generalized Frénet equations of the continuum theory of framed curves.

Discrete Derivative Equations for Quaternion Frames

An analogous discretisation of the derivative equation $d\hat{\mathbf{q}}_p = \frac{1}{2}\boldsymbol{\kappa} \circ \hat{\mathbf{q}}$ for the quaternion frames can be obtained from the Cayley transform⁷ and its inverse

$$\mathbf{q} \mapsto \hat{\mathbf{p}} = \text{cay}(\mathbf{q}) = \frac{1 + \mathbf{q}}{1 - \mathbf{q}}, \quad \hat{\mathbf{p}} \mapsto \mathbf{q} = \text{cay}^{-1}(\hat{\mathbf{p}}) = \frac{\hat{\mathbf{p}} - 1}{\hat{\mathbf{p}} + 1}$$

mapping vectors to rotational quaternions and vice versa. For $\hat{\mathbf{p}} = \cos(\alpha) + \sin(\alpha)\hat{\mathbf{e}} = \exp(\alpha\hat{\mathbf{e}})$ one obtains $\mathbf{q} = \text{cay}^{-1}(\hat{\mathbf{p}}) = \tan(\alpha/2)\hat{\mathbf{e}}$, such that $\exp(\alpha\hat{\mathbf{e}}) = \text{cay}(\tan(\alpha/2)\hat{\mathbf{e}})$ holds. Applied to the rotation $\hat{\mathbf{w}}_j = \exp(\vartheta_j/2\hat{\mathbf{u}}_j) = \hat{\mathbf{q}}_{j+1/2} \circ \hat{\mathbf{q}}_{j-1/2}^*$

⁷The notation $(1 + \mathbf{q})/(1 - \mathbf{q})$ captures the fact that $(1 + \mathbf{q}) \circ (1 - \mathbf{q})^{-1} = (1 - \mathbf{q})^{-1} \circ (1 + \mathbf{q})$ holds for all $\mathbf{q} \in \mathbb{E}^3$. Likewise the identity $(\hat{\mathbf{p}} - 1) \circ (\hat{\mathbf{p}} + 1)^{-1} = (\hat{\mathbf{p}} + 1)^{-1} \circ (\hat{\mathbf{p}} - 1)$ valid for all $-1 \neq \hat{\mathbf{p}} \in S^3$ is abbreviated by $(\hat{\mathbf{p}} - 1)/(\hat{\mathbf{p}} + 1)$.

connecting adjacent quaternion frames this yields the desired result:

$$\hat{\mathbf{q}}_{j+1/2} - \hat{\mathbf{q}}_{j-1/2} = \frac{1}{2} \left[4 \tan \left(\frac{\vartheta_j}{4} \right) \hat{\mathbf{u}}_j \right] \circ \frac{1}{2} (\hat{\mathbf{q}}_{j-1/2} + \hat{\mathbf{q}}_{j+1/2}). \quad (12)$$

Starting from $\hat{\mathbf{q}}_{j+1/2} = \hat{\mathbf{q}}_{j-1/2} \circ \hat{\mathbf{W}}_j$ with $\hat{\mathbf{W}}_j = \exp(\vartheta_j/2 \hat{\mathbf{U}}_j)$ one obtains the equivalent difference equation

$$2(\hat{\mathbf{q}}_{j+1/2} - \hat{\mathbf{q}}_{j-1/2}) = \frac{1}{2}(\hat{\mathbf{q}}_{j-1/2} + \hat{\mathbf{q}}_{j+1/2}) \circ \left[4 \tan \left(\frac{\vartheta_j}{4} \right) \hat{\mathbf{U}}_j \right], \quad (13)$$

which provides a geometric FD discretization of the equivalent form $d\hat{\mathbf{q}}_p = \frac{1}{2}\hat{\mathbf{q}} \circ \mathbf{K}$ of the derivative equation for the quaternion frames in terms of the material Darboux vector $\mathbf{K} = \hat{\mathbf{q}}^* \circ \boldsymbol{\kappa} \circ \hat{\mathbf{q}}$. The FD approximation of the material Darboux vector $\mathbf{K}(p_j)$ implied in (13) is the one already mentioned in (6). The FD formula for the spatial Darboux vector $\boldsymbol{\kappa}$ results from a forward rotation $\hat{\mathbf{u}}_j = \hat{\mathbf{q}}_{j\pm 1/2} \circ \hat{\mathbf{U}}_j \circ \hat{\mathbf{q}}_{j\pm 1/2}^*$ of the material rotation axis to the spatial one. The discrete derivative equations (12) and (13) are the equivalent quaternionic counterparts of the discrete generalized Frénet equations (11).

4.2.4 The Principal Theorem for Discrete Cosserat Curves

At this point, we have obtained all ingredients necessary for the statement and constructive proof of the **principal theorem of discrete Cosserat curve theory**.

The *basic data* of a discrete Cosserat curve, consisting of a polygonal arc $\mathcal{P}_n[p_0, \dots, p_n]$ and the set $\{\hat{\mathbf{q}}_{j-1/2}\}_{j=1, \dots, n}$ of quaternion frames representing orthonormal frames $\mathbf{R}_{j-1/2} = \mathfrak{E}(\hat{\mathbf{q}}_{j-1/2})$ located at edge centers $\bar{p}_{j-1/2}$, are:

- the set $\{\psi_j^{(k)}\}_{j=1, \dots, n-1}^{k=1, 2, 3}$ of *curvature angles*, and
- the set $\{\ell_{j-1/2}^{(k)}\}_{j=1, \dots, n}^{k=1, 2, 3}$ of *material edge components*.

The essential parts of the *discrete integration procedure*, resulting in a reconstruction of a discrete Cosserat curve from its basic data and chosen initial conditions, have already been outlined in the previous paragraphs of this section and is summarized compactly in the following algorithm:

1. A rotational quaternion $\hat{\mathbf{q}}_0 \in S^3$ representing an orthonormal frame $\mathbf{R}_0 = \mathfrak{E}(\hat{\mathbf{q}}_0) \in \mathbf{SO}(3)$ and a point $p_0 \in \mathcal{E}^3$ are selected as arbitrarily chosen initial values.
2. Starting from $\hat{\mathbf{q}}_{1/2} := \hat{\mathbf{q}}_0$, the quaternion frames are generated iteratively from the material rotation vectors $\vartheta_j \hat{\mathbf{U}}_j = \psi_j^{(k)} \mathbf{e}_k$ defined by the given curvature angles via sequential rotations

$$\hat{\mathbf{q}}_{j+1/2} = \hat{\mathbf{q}}_{j-1/2} \circ \exp(\vartheta_j/2 \hat{\mathbf{U}}_j) \quad \text{for } j = 1, \dots, n-1$$

according to (4). The sequence $\{\hat{\mathbf{q}}_{j-1/2}\}_{j=1,\dots,n}$ is an exact solution of the discrete derivative equations [(12), (13)] uniquely determined by its initial value.

3. Starting from p_0 , the vertices are obtained iteratively by summation as

$$p_j = p_0 + \sum_{i=1}^j \ell_{i-1/2}^{(k)} \hat{\mathbf{q}}_{i-1/2} \circ \mathbf{e}_k \circ \hat{\mathbf{q}}_{i-1/2}^* \text{ for } j = 1, \dots, n$$

according to (9) from the known quaternion frames and the given material edge components. The sequence $\{p_j\}_{j=0,\dots,n}$ is an exact solution of the discrete derivative equations (10) uniquely determined by its initial value p_0 .

This algorithm provides a constructive proof of the principal theorem for discrete Cosserat curves, which may be formulated as follows:

For given discrete data consisting of the sets of *curvature angles* $\{\psi_j^{(k)}\}_{j=1,\dots,n}^{k=1,2,3}$ and *material edge components* $\{\ell_{j-1/2}^{(k)}\}_{j=1,\dots,n}^{k=1,2,3}$, the solutions of the discrete derivative equations (10) and (12), (13) determine the quaternion frames $\{\hat{\mathbf{q}}_{j-1/2}\}_{j=1,\dots,n}$ and the vertices $\{p_j\}_{j=0,\dots,n}$ of a discrete Cosserat curve up to an overall rigid body motion in \mathcal{E}^3 .

5 Discrete Elastic Energy of Quaternionic Cosserat Rods

We discretize the continuum model of an elastic Cosserat rod by approximating its elastic energy integrals (1) and (2) by suitable quadrature rules, making use of the discrete curvatures $\{K_j^{(k)}\}_{j=1,\dots,n-1}^{k=1,2,3}$ and extensional and shearing strains $\{\Gamma_{j-1/2}^{(k)}\}_{j=1,\dots,n}^{k=1,2,3}$ defined in the previous Sects. 4.2.1 and 4.2.2. Here we briefly outline this approach described in detail in our article in [20].

We start with a discretization $0 =: s_0 < s_1 < \dots < s_n := L$ of the interval domain $[0, L]$ of the arc length parameter s of the reference curve $\mathbf{r}_0(s)$ into subintervals $[s_{j-1}, s_j]$ of length $h_{j-1/2} := s_j - s_{j-1}$. The distance between interval midpoints $s_{j\pm 1/2} = \frac{1}{2}(s_j + s_{j\pm 1})$ is given by $\Delta s_j := s_{j+1/2} - s_{j-1/2} = \bar{h}_j$, where $\bar{h}_j := \frac{1}{2}(h_{j-1/2} + h_{j+1/2})$ is the average of the grid constants $h_{j\pm 1/2}$ adjacent to s_j .

5.1 Discrete Extensional and Shear Energy

As the discrete extensional and shear strains $\Gamma_{j-1/2}^{(k)}$ are edge based quantities, an approximation of the energy integral (1) by midpoint quadrature is the natural

choice to obtain a discrete version of \mathscr{W}_{es} . The pull back of the strains to the reference configuration is obtained by a rescaling with the factors $\ell_{j-1/2}/h_{j-1/2} \approx \|\mathbf{r}'(s_{j-1/2})\|$, according to

$$\bar{\Gamma}_{j-1/2}^{(k)} := \frac{\ell_{j-1/2}}{h_{j-1/2}} \Gamma_{j-1/2}^{(k)} = \langle \mathbf{l}_{j-1/2}, \mathbf{a}_{j-1/2}^{(k)} \rangle / h_{j-1/2} = \langle \bar{\Gamma}_{j-1/2}, \mathbf{e}_k \rangle, \quad (14)$$

where $\bar{\Gamma}_{j-1/2} := \hat{\mathbf{q}}_{j-1/2}^* \circ (\mathbf{l}_{j-1/2}/h_{j-1/2}) \circ \hat{\mathbf{q}}_{j-1/2}$ is the material vector obtained from rotating the discrete edge tangent vector $\mathbf{l}_{j-1/2}/h_{j-1/2}$ back to the local frame. The discrete approximation of \mathscr{W}_{es} can be written in compact form as

$$\mathscr{W}_{es} \approx \mathscr{W}_{es}^{(D)} := \frac{1}{2} \sum_{j=1}^n h_{j-1/2} \langle \Delta \bar{\Gamma}_{j-1/2}, \mathbf{C}_{es} \cdot \Delta \bar{\Gamma}_{j-1/2} \rangle, \quad (15)$$

where $\Delta \bar{\Gamma}_{j-1/2} := \bar{\Gamma}_{j-1/2} - \Gamma_0$ with $\Gamma_0 = (0, 0, 1)^T$, and the shear and extensional stiffness parameters collected in the diagonal matrix $\mathbf{C}_{es} := \text{diag}([GA_1], [GA_2], [EA])$.

The condition of vanishing discrete transverse shear strains $\bar{\Gamma}_{j-1/2}^{(\alpha)} \equiv 0$ implies $\bar{\Gamma}_{j-1/2}^{(3)} = \ell_{j-1/2}/h_{j-1/2}$, such that (14) reduces to the extensional energy

$$\mathscr{W}_{ext}^{(D)} := \frac{1}{2} \sum_{j=1}^n h_{j-1/2} [EA] (\ell_{j-1/2}/h_{j-1/2} - 1)^2$$

of a *discrete extensible Kirchhoff rod* model [21], which approximates its continuum counterpart given by $\mathscr{W}_{ext} := \frac{1}{2} \int_0^L ds [EA] (\|\mathbf{r}'(s)\| - 1)^2$. Additionally imposing the *inextensibility condition* $\ell_{j-1/2} \equiv h_{j-1/2}$ on the edges implies $\mathscr{W}_{ext}^{(D)} \equiv 0 \equiv \mathscr{W}_{es}^{(D)}$.

5.2 Discrete Bending and Torsion Energy

The discrete curvatures $K_j^{(k)}$ are vertex based quantities, such that a discrete approximation of \mathscr{W}_{es} can be obtained from the energy integral (2) by (non-equidistant) trapezoidal quadrature. The pull back of the curvatures originally defined w.r.t. discrete arc length to the reference configuration implies a rescaling by the factor $\Delta \varsigma_j / \Delta s_j \approx \|\mathbf{r}'(s_j)\|$, i.e.: $\bar{K}_j^{(k)} := (\Delta \varsigma_j / \Delta s_j) K_j^{(k)}$, equivalent to the definition

$$\bar{K}_j^{(k)} := \langle \bar{\mathbf{K}}_j, \mathbf{e}_k \rangle, \quad \bar{\mathbf{K}}_j := (\Delta s_j)^{-1} 2 \log \hat{\mathbf{W}}_j = (\vartheta_j / \Delta s_j) \hat{\mathbf{U}}_j = \bar{K}_j^{(k)} \mathbf{e}_k \quad (16)$$

of discrete pulled back material curvatures. The discrete approximation of \mathcal{W}_{bt} can then be written in compactly as

$$\mathcal{W}_{bt} \approx \mathcal{W}_{bt}^{(D)} := \frac{1}{2} \sum_{j=0}^n \Delta s_j \langle \Delta \bar{\mathbf{K}}_j, \mathbf{C}_{bt} \cdot \Delta \bar{\mathbf{K}}_j \rangle, \tag{17}$$

with the curvature differences $\Delta \bar{\mathbf{K}}_j := \bar{\mathbf{K}}_j - \mathbf{K}_{0j}$ between the deformed and reference configurations, the diagonal matrix $\mathbf{C}_{bt} := \text{diag}([EI_1], [EI_2], [GJ])$ of bending and torsional stiffness parameters, and the grid constants of the half-edges near the boundary defined as $\Delta s_0 := h_{1/2}/2$ and $\Delta s_n := h_{n-1/2}/2$.

5.3 Boundary Conditions

The definition of material curvature vectors $\bar{\mathbf{K}}_0$ and $\bar{\mathbf{K}}_n$ at the boundary vertices is a new element in the discrete model that did not have to be considered in the theory of discrete Cosserat curves up to this point. These discrete curvatures are defined in the deformed configuration in terms of the material difference rotations $\hat{\mathbf{W}}_0 := \hat{\mathbf{q}}_0^* \circ \hat{\mathbf{q}}_{1/2}$ and $\hat{\mathbf{W}}_n := \hat{\mathbf{q}}_{n-1/2}^* \circ \hat{\mathbf{q}}_n$ connecting the *boundary frames* $\hat{\mathbf{q}}_0$ and $\hat{\mathbf{q}}_n$ to the frames on the adjacent edges, with analogous definitions for the boundary curvatures of the reference configuration.

The boundary frames of both configurations are determined by *boundary conditions* imposed on the discrete rod model, which have to be formulated as separate conditions, or can directly be built into the discrete energy $\mathcal{W}_{bt}^{(D)}$. Likewise, boundary conditions on the vertices p_0 and p_n influence the discrete energy $\mathcal{W}_{es}^{(D)}$.

5.4 Discrete Equilibrium Equations

With the boundary conditions built into the respective terms of the discrete energies (15) and (17), the total discrete elastic energy given by

$$\mathcal{W}_{el}^{(D)}(\mathcal{X}_f) := \mathcal{W}_{es}^{(D)}(\mathcal{X}_f) + \mathcal{W}_{bt}^{(D)}(\mathcal{X}_f) \tag{18}$$

become functions of the free vertex positions $\{\mathbf{r}_j\}$ and quaternion frames $\{\hat{\mathbf{q}}_{j-1/2}\}$ collected in the set $\mathcal{X}_f := \{\mathbf{r}_j\} \cup \{\hat{\mathbf{q}}_{j-1/2}\}$ of free variables.

Static equilibrium configurations of a discrete Cosserat rod subject to given boundary conditions can be obtained by *minimizing the discrete elastic energy* of the rod. The *discrete equilibrium equations*

$$\nabla_{\mathcal{X}_f} \mathcal{W}_{el}^{(D)}(\mathcal{X}_f) = 0 \Leftrightarrow \frac{\partial \mathcal{W}_{el}^{(D)}}{\partial \mathbf{r}_j}(\mathcal{X}_f) = \mathbf{0}, \quad \frac{\partial \mathcal{W}_{el}^{(D)}}{\partial \hat{\mathbf{q}}_{j-1/2}}(\mathcal{X}_f) = \mathbf{0} \tag{19}$$

provide a necessary condition for such energy minima. The discrete equilibrium equations (18) is a coupled nonlinear system of algebraic equation, which can be solved numerically by Newton’s method. Alternatively, one can find approximate equilibrium configurations by energy minimization, using nonlinear conjugate gradients or Quasi-Newton type optimization methods like BFGS.

The system (18) is actually a special case of the dynamic equilibrium equations for a semi-discrete model of time dependent discrete quaternionic Cosserat rods [20], which are obtained as the (abstractly written) Euler–Lagrange equations

$$\partial_t (\nabla_{\partial_t \mathcal{X}_f} \mathcal{L}^{(D)}(\mathcal{X}_f, \partial_t \mathcal{X}_f)) - \nabla_{\mathcal{X}_f} \mathcal{L}^{(D)}(\mathcal{X}_f, \partial_t \mathcal{X}_f) = 0$$

for the Lagrangian function $\mathcal{L}^{(D)}(\mathcal{X}_f, \partial_t \mathcal{X}_f) := \mathcal{I}_{kin}^{(D)}(\mathcal{X}_f, \partial_t \mathcal{X}_f) - \mathcal{W}_{el}^{(D)}(\mathcal{X}_f)$ defined as the difference of the kinetic and potential energy according to the general concepts of Lagrangian mechanics (see [18, 20] for further details).

5.5 The Geometric Nature of the Discrete Elastic Energy

Up to this point, the various steps to obtain a discrete energy function from a continuum functional and to determine equilibrium configurations as energy minima did not involve any specific aspects of the discrete geometry of the underlying model: The energy integrals are discretized by standard quadrature rules, the discrete equilibrium equations result from the gradient of the discrete energy w.r.t. the free variables, and the numerical solution of the equations is done by Newton or Quasi-Newton methods.

The essential ingredient to this procedure is provided by the fact that the curvatures and strains are well defined for arbitrarily large deformations, completely independent of the coarseness of the discretization. The discrete curvatures increase with increasing values of the angles, up to the very limit of degenerate configurations. The energy vanishes exactly only in the reference configuration and does not possess any “artificial” minima otherwise.

Not all discretizations share these important properties. For example, a finite element approach using linear interpolation of nodal $\mathbf{SO}(3)$ frames yields discrete curvatures $\sim \sin(\vartheta_j) \hat{\mathbf{u}}_j$ (see the discussion in Remark 8 of [16]), which becomes extremal at $\vartheta_j = \pm\pi/2$ and then decreases in modulus for larger values of ϑ_j , up to the value zero taken in the degenerate case $\vartheta_j = \pm\pi$. For coarse discretizations, it can easily happen that bending or torsion angles $|\vartheta_j| \approx \pi/2$ occur. A model based on this discretization of curvature would obviously produce unphysical results, as zig-zag type configurations with angles $\pi/2 \leq |\vartheta_j| \leq \pi$ become energetically favourable.

The discussion of this example illustrates that our efforts to construct a proper discrete theory of framed curves by geometric finite differences yields the essential contribution to a rod model required to behave qualitatively correct even for very

coarse discretizations. In this sense, one can state that our discrete Cosserat rod model inherits its structural stability from its underlying kinematic model of discrete Cosserat curves, completely independent of the coarseness of the discretization, and therefore displaying qualitatively correct physical behaviour at arbitrarily coarse discretizations.

5.6 Academic Test Examples

We conclude this section on discrete models of elastic Cosserat rods by showing some typical results obtained in two well known academic benchmark examples that illustrate the behaviour of our discrete Cosserat rod model in certain boundary value problems that are also relevant for our practical applications, and also show an illustrative comparison of a simulation with a corresponding laboratory experiment.

Figure 5 shows the performance of a recent implementation [19] of our discrete rod model in classical examples for inextensible rods due to Euler [12] and Kirchhoff [17] that are analytically solvable in closed form by Kirchhoff's kinetic analogy (see Art. 260, 262 and 270 in Ch. XIX of [23] and Dill [8] for details and historical remarks).

The coarse discretization in the *Elastica* example with only four edges has been deliberately fixed to equidistant edge lengths to induce visible differences between the discrete and the analytical solutions in configurations at larger deflections. However, these differences remain rather small, although an equidistant discretization is certainly not optimal in this example. Even on the level of kinetic quantities (i.e.: forces and moments), the “coarse” discrete solution captures the trend of the analytical curves of the bending moment and the tension force not only qualitatively, but displays a quantitative agreement with the corresponding analytical values that could hardly be expected. Nevertheless, the slight S-shaped variation of the analytical curve for the transverse shear force is captured only very roughly. For a finer equidistant discretization with 20 edges both the kinematic as well as the kinetic quantities of the analytical solution are captured almost perfectly .

The example of the Kirchhoff helix shows that equally positive results can be obtained for spatial deformations involving a combination of bending and torsion of the rod. In the classical form of Kirchhoff's problem for helical equilibrium configurations of elastic rods, the latter are shown to result from special combinations of the material components of forces and moments applied at both ends of the rod, which determines the solution up to an overall rigid body motion. The solution shown in Fig. 5 results from a slightly different variant of the boundary value problem, where the vertices and frames at both ends were clamped and varied along the known analytical solution of the continuum problem. Here, only the configuration data are shown, perfectly matching the analytical centerline curves for both the coarser (10 edges) and finer (50 edges) discretization. The plots of the kinetic quantities, which are all straight horizontal lines corresponding to constant material components of

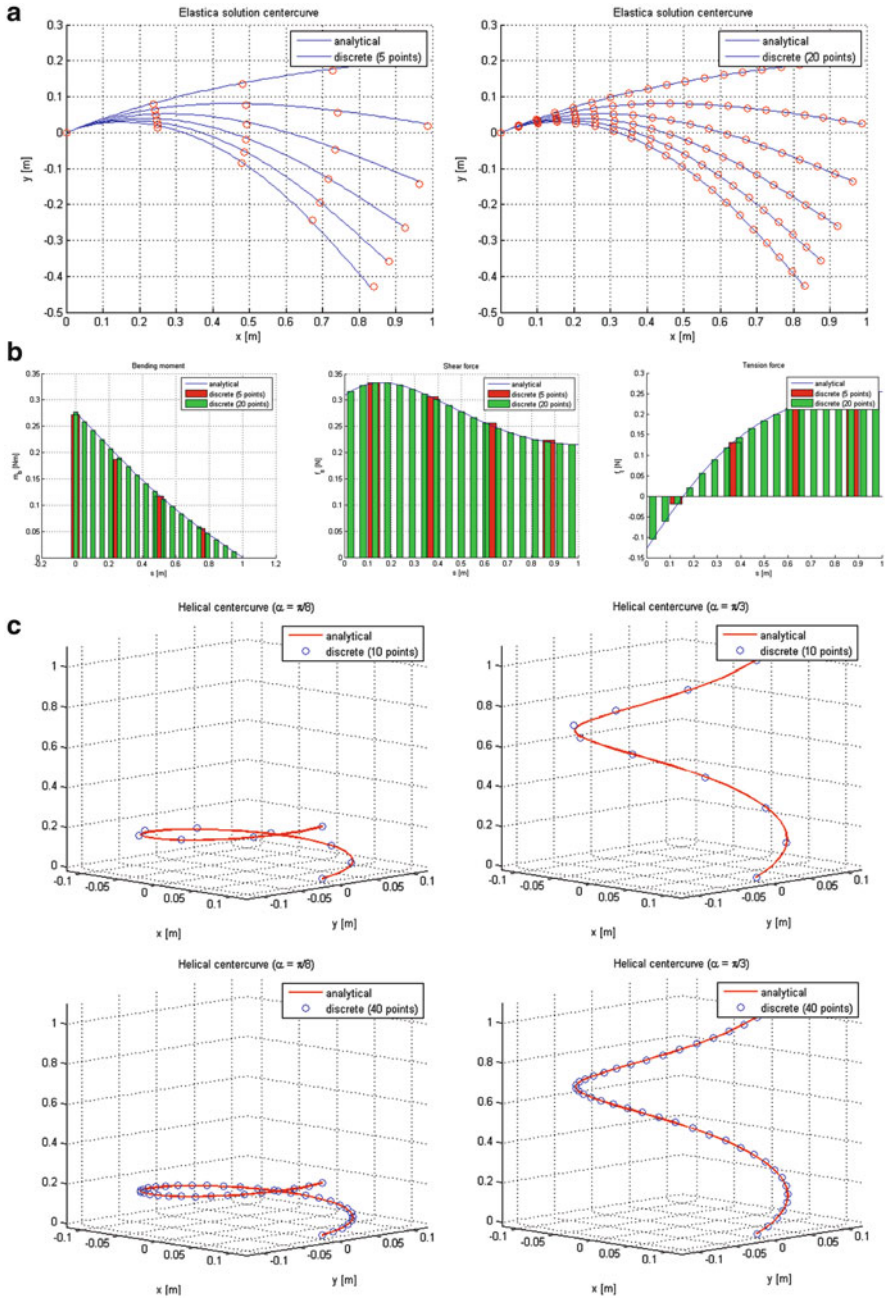


Fig. 5 (a) Plane bending of a clamped cantilever beam loaded by a vertical end force, with analytical solution curves given by *Euler's Elastica* model. (b) Bending moment (*left*), transverse shear force (*middle*), and tension force (*right*) as a function of arc length for the clamped cantilever

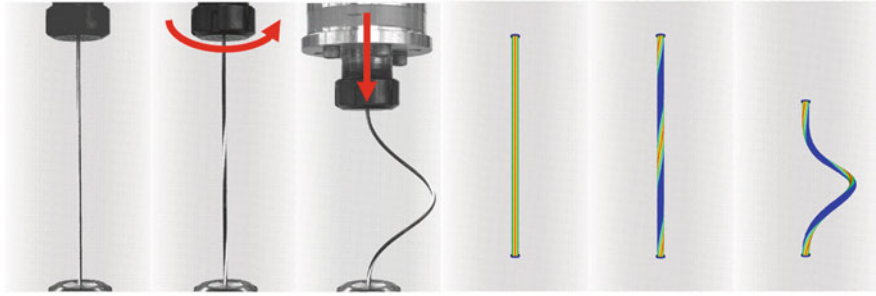


Fig. 6 Spatial configurations of a clamped cable after twisting and axial displacement of the clamped ends. *Left*: Laboratory experiment; *Right*: Results in *IPS Cable Simulation*

the forces and moments relative to the local frames, all captured accurately by the discrete solution, are omitted here.

The laboratory experiment [10] shown in Fig. 6 is conceptually similar to the Kirchhoff helix example with clamped ends, but more general, as it involves non-constant torsion and bending, which is closer to realistic cases encountered in assembly simulation. Although the real cable that has been used as a specimen in the experiment clearly displayed viscoplastic material properties measured for larger local deformations, on top of the basic elastic constitutive behaviour that dominates in the regime of small local strains, the computation of equilibrium configurations performed with the discrete Cosserat rod model in the software *IPS Cable Simulation* properly captures the complex spatial deformation behaviour of the cable in the real experiment if appropriately averaged effective elastic cable properties are used.

Concerning the computing times we would like to remark that already a decade ago it was possible to simulate the entire deformation sequences of the two examples discussed in this section (as well as others, see [21]) on a standard desktop computer within the range of milliseconds. This provides the basis for true real time interaction with the simulation model by an interactive change of the boundary conditions, or likewise a change of model parameters, like e.g. the length and stiffness parameters of the rod. Also immediate adaptations of the (in general: non-equidistant) discretization in reaction to external constraints induced by various types of clips or contact with rigid geometries in the environment can be handled algorithmically in a very efficient manner.

←
Fig. 5 (continued) beam example shown in (a): the values obtained from the discrete model with coarser (5 points) and finer (20 points) discretization approximate the analytical solution (*blue curves*) obtained from *Euler's Elastica* model. (c) Two helical configurations with smaller and larger pitch angle of a steel string in space, modeled as an *inextensible Kirchhoff rod*, with analytical solutions given by Kirchhoff's constant curvature helices, matched by discrete constant curvature solutions of the same boundary value problem with coarser (*top*) and finer (*bottom*) discretization

6 Application Examples from Vehicle Industry

The software package *IPS Cable Simulation* contains different variants of discrete Cosserat models, augmented by numerous productivity features that are useful for a variety of applications in industry, like enhanced CAD, digital mock-up, or the simulation of assembly (or disassembly) processes on desktop computers as well as in virtual (VR) or augmented (AR) reality environments. In *IPS Cable Simulation* the rod models have been enhanced w.r.t. a variety of different elements, such that external constraints induced by various types of clips, restricting spatial motions of one to six of the local rigid body d.o.f. of the rod, frictionless contact interaction with rigid geometries in the environment, or self contact and contact with other rods can be handled efficiently.

The user can interact with the discrete elastic rod model of a cable or hose e.g. by grabbing one of its ends with the mouse (or more sophisticated VAR devices), or likewise a clip attached at some intermediate location, and change its position and orientation in space interactively. These changes are captured by the user interface and translated by the software into a sequential change of boundary conditions. This sequence is then passed to the simulation model, which computes corresponding deformation sequences by solving the mechanical equilibrium equations sufficiently fast for rendering deformed configuration at interactive rates (i.e.: 25 GHz or faster).

The seamless integration of discrete Cosserat rod models into the IPS software is supported by their formulation in terms of elementary geometric quantities (i.e.: vertex positions and orthonormal frames) that can be handled highly efficient by the computational geometry methods and algorithms already implemented in IPS. As an example, frames translated along polygonal paths in space are fundamental objects in rigid body path planning, which historically has been one of the core capabilities of the IPS software.

6.1 *Representative Application Examples from Automotive Industry*

In what follows, we present some examples, all taken from joint projects with AUDI AG, to illustrate the typical usage of our discrete rod models integrated in the IPS software in automotive industry. Of course, simple screenshots from the software as the ones shown here are not able to convey the user experience in interactive operation with the software in a realistic way, and even the more detailed information provided by video⁸ sequences of the software in operation at real application cases can hardly be more than a substitute. Nevertheless, the

⁸We refer to the various videos shown on the website www.flexstructures.com for a visualization of deformation sequences of cables in different application cases.

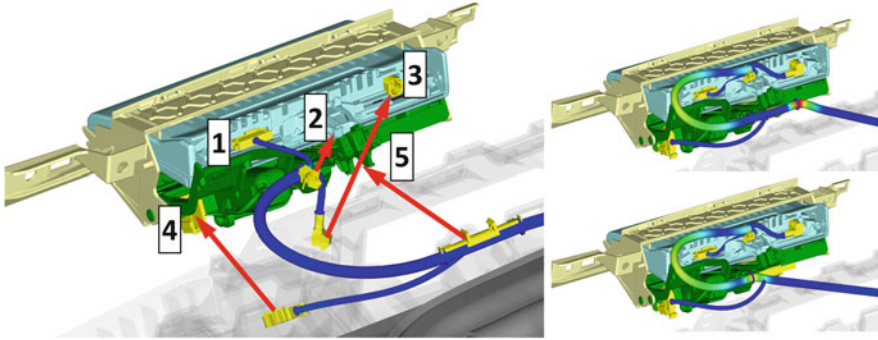


Fig. 7 *Left:* Assembly sequence of various clips and connectors to be mounted at head-up display. *Right:* Variation of the position and the type of a clip to lower the level of strain in the cable

principle way of usage of the simulation software and the benefits obtained in industrial applications should become clear from our descriptive exposition.

Figure 7 shows some screenshots taken from the simulation of the assembly sequence of various clips and connectors to be mounted at a head up display. The numbers indicate the assembly sequence, and the arrows point out the direction of the movement of the various clips and plugs towards their final mounting positions. The engineer working with the simulation software would typically grab individual clips, plugs or connectors with the mouse and perform the assembly operation for all on the virtual model sequentially, just as it would later be done by the worker in the real process. In this way, the engineer can validate if it is possible to assemble all cables without mutual crossing, and if the cable lengths are sufficient to avoid extensional straining and sharp bending at clips. If not, the length of a cable can be changed interactively to its optimal length. Also, the function of various clip types as well as their positioning can be checked and varied to avoid unfavourable configurations. The two pictures to the right of Fig. 7 show such a variation of the position and the type of a clip to lower the level of strain in the cable. Note that the assembly of the connector labeled as No. 3 requires a clockwise rotation by 90° for plug-in. The effect of torsional straining of the attached cable can be computed in terms of the torsional moment acting on the cable in its mounted configuration, in combination with the bending strains that are simultaneously acting. The combined level of stresses and strains can be assessed by St. Venant type stress functions that superpose forces and moments of the rod model into local 3D stress tensor values, which can be evaluated e.g. in terms of von Mises stress values.

Figure 8 displays another issue discovered and solved during the digital validation of the functionality of the same head up display: The originally suggested design of one of the plastic parts in the kinematic mechanism was done in a compact way to assure its mechanical stability with a minimal amount of plastic material, with a rather sharp rectangular kink at its upper right corner. Despite the slight rounding at its corner, the simulation of the operation of the whole mechanism including the cables showed that during sequential moving out and in of the display

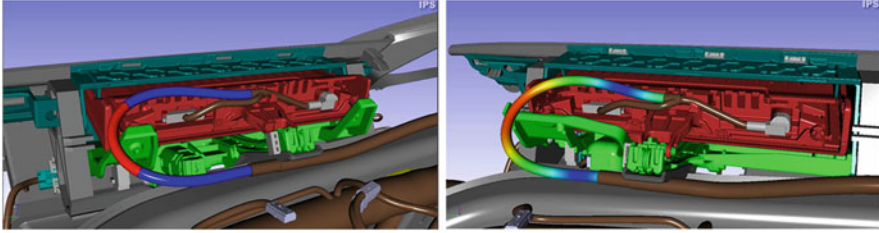


Fig. 8 Redesign of a plastic part in the kinematic mechanism of the same head-up display. *Left:* Former design with a sharp corner, where the cable may get hooked, which might cause damage. *Right:* Improved design, where the cable can slide along reproducibly always in the same stable way, such that potential damage due to hook-up or clamping of the cable is avoided

the cable had to slide across the corner, with a sudden change of its configuration due to buckling. The whole situation turned out to be rather sensitive to small changes in the cable length. All in all, it became clear that the initial design was not robust enough, and the increased bending strains occurring in the cable in its configuration hooked up at the corner of the plastic part might later cause damage due to the frequent repetitive loading. Therefore the design of the plastic part had to be changed. The picture to the right of Fig. 8 shows the final design solution: The vertical edge at the respective corner was bent upwards and elongated, such that the sharp right angle is eliminated, and the edge functions as a rail on which the cable can slide in a stable way during the forward and backward operation of the kinematic mechanism. Clearly, the simulation of the whole system with the physically correct deformation of flexible cables in contact with the rigid geometries in its environment provided the essential insights leading to an improved design. With this software, the functionality of the whole system of the headup display with all cables connected can be simulated and validated digitally in an early design phase. Traditionally, one would manufacture hardware prototypes of the simulated design to perform a physical validation. However, the possibility of simulation with physically correct models allows for functional validation without physical prototypes.

The third example shown in Fig. 9 addresses the optimization of the length of a flexible grommet to avoid damage during the repeated opening and closing operation of the hatchback and has been discussed in detail in [15], introducing a (at that time) new method of comparative durability assessment based on the prediction of local stresses during quasi-static deformation sequences of flexible cables or hoses. In this case, the grommet joining the chassis and hatchback, which conducts various cables inside, becomes substantially deformed during the opening and closing of the hatchback, such that bending and torsional strain occur at a considerable level. The issue could be resolved in a simulation study by an elongation of the length of the grommet tube, which lowers the level of strains, but still avoids unwanted clamping at the nearby hinge.

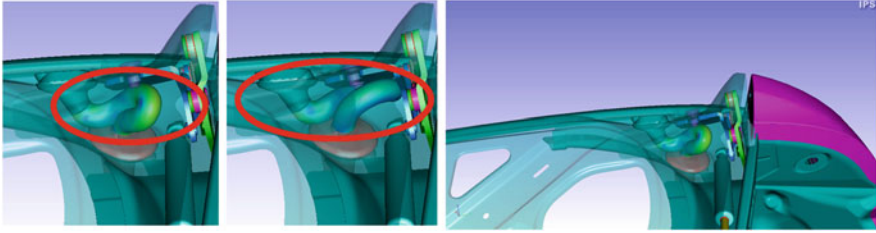


Fig. 9 Optimization of the length of a flexible grommet to avoid damage during the repeated opening and closing operation of the hatchback. *Left*: Design variant with a shorter length and higher bending strains. *Middle*: Alternative elongated design variant, showing lower bending strains in the shut state of the hatchback

7 Conclusion

Software tools used in industry for design and digital validation require a seamless integration of efficient models that are able to handle the simulation of large spatial deformations of slender elastic structures like cables and hoses by bending, twisting and extension at interactive rates.

In our article, we presented the construction of discrete elastic rod models of Cosserat type within the framework of Lagrangian mechanics on the kinematic basis of discrete models of framed curves in Euclidian space. We obtain discrete framed curves from their continuum counterparts by a discretization method we denote as *geometric finite differences*, which can be regarded as an extension of Sauer's "*difference geometry*" approach to Frénet curves. These discrete Cosserat curves, which consist of polygonal arcs augmented by orthonormal frames located at edge centers and represented by rotational quaternions, provide a family of discrete kinematical models that qualitatively capture all essential features of Cosserat curves in the continuum, independent of the coarseness of the discretization.

As a consequence, the discrete bending and twisting curvatures as well as the discrete extensional and transverse shear strains obtained from the difference geometry of Cosserat curves yield discrete approximations of the elastic energy of continuum Cosserat rods that model the deformation behaviour of slender elastic structures physically correct for arbitrarily coarse discretizations. As the basic kinematical constituents of our discrete Cosserat rod models are vertex positions and orthonormal frames, i.e.: elementary geometric quantities that can be handled very efficiently by methods and algorithms of computational geometry, they can be integrated in a computationally efficient and seamless way in geometry-oriented software packages, with enhancements to support the interaction with the CAD geometries in the environment, as well as user interaction in real time.

Our application examples from automotive industry illustrate the principle way of usage to solve practical tasks in design and digital validation with the software *IPS Cable Simulation*. As an outlook, we would like to mention that by the same approach we are able to integrate fully dynamical models of cables that account for inertial and viscous effects into our software environment as well as in other CAE

software packages for multibody system dynamics simulations [28]. Moreover, our geometric discretization approach may likewise be applied to obtain discrete models of flexible surfaces (see [32, 33] for basic research work).

Acknowledgements We thank F. Hoefl and C. Loris from *Audi AG* and *flexStructures GmbH* for the IPS application examples, and the members of the IPS team at department MDF of ITWM and *flexStructures GmbH* for contributing corresponding presentation slides and movies for our talk given at the MaDiFa workshop at WIAS, Berlin. T. Hermansson and the other members of the IPS development team at FCC have done superior work to integrate our discrete rod models into the IPS software and add a multitude of further developments to make them useful for industrial applications. Last, but not least we would like to thank our former colleague and long time collaborator Holger Lang (now at the chair LTD, university of Erlangen) for joint basic development work on the Lagrangian mechanics and discrete difference geometry of Cosserat rods.

References

1. Antman, S.S.: *Nonlinear Problems of Elasticity*. Springer, Berlin (2005)
2. Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., Grinspun, E.: Discrete elastic rods. *ACM Trans. Graph. (SIGGRAPH)* **27**(3), 63:1–63:12 (2008)
3. Blaschke, W.: *Vorlesungen über Differentialgeometrie. I. Elementare Differentialgeometrie* (3. Aufl.). Springer, Berlin (1930)
4. Blaschke, W., Leichtweiß, K.: *Elementare Differentialgeometrie* (5. Aufl.). Springer, Berlin (1974)
5. Bobenko, A.I., Suris, Y.B.: Discrete time Lagrangian mechanics on Lie groups, with an application on the Lagrange top, *Commun. Math. Phys.* **204**, 147–188 (1999)
6. Bobenko, A.I., Schröder, P., Sullivan, J.M., Ziegler, G.M. (eds.): *Discrete Differential Geometry*. Oberwolfach Seminars, vol. 38. Birkhäuser, Basel (2008)
7. Cartan, H.: *Formes différentielles — Applications élémentaires au calcul des variations et à la théorie des courbes et des surfaces*. Hermann, Paris (1967)
8. Dill, H.E.: Kirchhoff's theory of rods. *Arch. Hist. Exact Sci.* **44**(1), 1–23 (1992)
9. do Carmo, M.P.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs (1976)
10. Dörlich, V., Linn, J., Scheffer, T., Diebels, S.: Towards viscoplastic constitutive models for Cosserat rods. *Archive Mech. Eng.* **63**(2), 215–230 (2016)
11. Ebbinghaus, H.-D., Hermes, H., Hirzebruch, F., Koecher, M., Mainzer, K., Neukirch, J., Prestel, A., Remmert, R.: *Zahlen* (3. Auflage). Springer, Berlin (1992)
12. Euler, L.: Addimentatum I de curvis elasticis, p. 247 ff. In: *Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes, sive solutio problematis isoperimetrici lattissimo sensu accepti*. Opera Omnia I, vol. 24, pp. 231–297. Bousquet, Lausanne (1744)
13. Géradin, M., Cardona, A.: *Flexible Multibody Dynamics: A Finite Element Approach*. Wiley, New York (2001)
14. Gurtin, M.E.: *An Introduction to Continuum Mechanics*. Academic, New York (1981)
15. Hoefl, F., Stephan, T., Hermanns, O.: Eine neue Methode zur vergleichenden örtlichen Beanspruchungsanalyse für Kabel und Schläuche. SIMVEC Berechnung und Simulation im Fahrzeugbau 2010. VDI-Berichte Nr. **2107**, 297–309 (2010). ISBN 978-3-18-092107-5
16. Jung, P., Leyendecker, S., Linn, J., Ortiz, M.: A discrete mechanics approach to the Cosserat rod theory - Part 1: static equilibria. *Int. J. Numer. Methods Eng.* **85**(1), 31–60 (2011)
17. Kirchhoff, G.: Über das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes. *J. Math. (Crelle) Band* **56** (1859). Also: pp. 285–316 in *Gesammelte Abhandlungen*. Johann Ambrosius Bart, Leipzig (1882)

18. Lang, H., Arnold, M.: Numerical aspects in the dynamic simulation of geometrically exact rods. *Appl. Numer. Math.* **62**, 1411–1427 (2012)
19. Linn, J., Hermansson, T., Andersson, F., Schneider, F.: Kinetic Aspects of Discrete Cosserat Rods. In: Proceedings of the 8th ECCOMAS Thematic Conference on Multibody Dynamics, Prague, June 19–22 (2017)
20. Lang, H., Linn, J., Arnold, M.: Multibody dynamics simulation of geometrically exact Cosserat rods. *Multibody Sys. Dyn.* **25**(3), 285–312 (2011)
21. Linn, J., Stephan, T., Carlsson, J., Bohlin, R.: Fast simulation of quasistatic rod deformations for VR applications. In: Bonilla, L.L., Moscoso, M., Platero, G., Vega, J.M. (eds.) *Progress in Industrial Mathematics at ECMI 2006*, pp. 247–253. Springer, Berlin (2008)
22. Linn, J., Lang, H., Tuganov, A.: Geometrically exact Cosserat rods with Kelvin–Voigt type viscous damping. *Mech. Sci.* **4**, 79–96 (2013)
23. Love, A.E.H.: *A Treatise on the Mathematical Theory of Elasticity*, 4th edn. Cambridge University Press, Cambridge (1927)
24. Milnor, J.W.: *Topology from the Differentiable Viewpoint*, The University Press of Virginia, Charlottesville (1965)
25. Podio-Guidugli, P.: A primer in elasticity. *J. Elast.* **58**(1), 1–104 (2000)
26. Reissner, E.: On one-dimensional large-displacement finite-strain beam theory. *Stud. Appl. Math.* **52**, 87–95 (1973)
27. Sauer, R.: *Differenzgeometrie*. Springer, Berlin (1970)
28. Schulze, M., Dietz, S., Burgermeister, B., Tuganov, A., Lang, H., Linn, J., Arnold, M.: Integration of nonlinear models of flexible body deformation in multibody system dynamics. *Trans. ASME J. Comput. Nonlinear Dyn.* **9**, 011012:1–011012:10 (2014)
29. Simo, J.C.: A finite strain beam formulation: the three dimensional dynamic problem – Part I, *Comput. Methods Appl. Mech. Eng.* **49**(1), 55–70 (1985)
30. Sullivan, J.M.: Curves of finite total curvature (pp. 137–161), Curvatures of smooth and discrete surfaces (pp. 175–188), in [6], Part II: Curvatures of Discrete Curves and Surfaces
31. Truesdell, C., Noll, W.: *The Non-linear Field Theories of Mechanics*, 3rd edn. Antman, S.S. (ed.) Springer, Berlin (2004)
32. Weischedel, C., Tuganov, A., Hermansson, T., Linn, J., Wardetzky, M.: Construction of discrete shell models by geometric finite differences. In: Eberhard, P., Ziegler, P. (eds.) *Proceedings of the 2nd Joint International Conference on Multibody System Dynamics (IMSD2012)*, Stuttgart (2012). ISBN 978-3-927618-32-9
33. Weischedel, C.: A discrete geometric view on shear-deformable shell models. PhD thesis, Georg-August-Universität Göttingen (2012)
34. Wriggers, P.: *Nonlinear Finite Elements*. Springer, Berlin (2008)

The Production of Filaments and Non-woven Materials: The Design of the Polymer Distributor

Christian Leithäuser and René Pinnau

Abstract We present results from the joint research project ProFil (Stochastic Processes for the Production of Filaments and Non-wovens), which were derived for the optimal design of the polymer distributor. In particular, one is interested in designs which prevent the cooling and degeneration of the polymer due to long occupation times. Since this is directly related to the wall shear stress distribution the questions arise, which wall shear stresses are attainable and how the corresponding design can be computed numerically. Employing the concept of approximate controllability we can answer the first one and characterize the set of attainable wall shear stresses. Further, we present a new numerical approach based on conformal mappings which allows for an optimization in the supremum norm and for an additional incorporation of state constraints. Finally, we show how the real industrial problem is solved by a least-squares optimization using shape gradients.

1 Introduction

The motivation for this contribution originates at Fraunhofer ITWM (Fraunhofer-Institute for Industrial Mathematics) with its long history of research on the industrial processes of fiber spinning and production of non-woven materials. Synthetic fibers and non-wovens have become increasingly important in recent years and find applications in a broad variety of products: The range goes from hygienic products, like diapers, over various filter materials towards high-tech applications like battery separators.

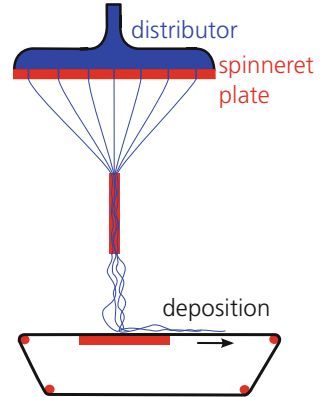
C. Leithäuser

Fraunhofer ITWM, Transport Processes, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
e-mail: christian.leithaeuser@itwm.fraunhofer.de

R. Pinnau (✉)

Department of Mathematics, TU Kaiserslautern, Gottlieb-Daimler-Straße, 67663 Kaiserslautern, Germany
e-mail: pinnau@mathematik.uni-kl.de

Fig. 1 Illustration of the spunbond production process for non-woven materials



The production process for non-woven materials is illustrated in Fig. 1: In a first step molten polymer with high viscosity is pressed from an extruder through a distributor geometry onto the spinneret plate. This is a plate containing small capillaries which are used for spinning the polymer into fibers. Turbulent air flow is applied for drawing and swirling the fibers. Finally, many fibers are deposited on a moving belt to form a non-woven material.

From 2010 until 2013, the joint research project ProFil (Stochastic Processes for the Production of Filaments and Non-wovens), which was funded by the Federal Ministry of Education and Research (BMBF), focused on the modeling, simulation and optimization of the whole process chain for the production of non-woven materials. See [18] for a state of the art overview of models and applications for the spinning process at Fraunhofer ITWM. Here, we deal with the first step of the process, especially with polymer flow in the distributor geometry and the following problem: The polymer is routed through tubes into the geometry which distributes it onto the spinneret plate. However, this is often a time-critical step because the polymer can degenerate or cool down if its occupation time is too long, which results in a poor fiber quality. It can lead to fiber breakage or even blockage of capillaries or parts of the distributor. In particular, in regions close to the walls, polymer can stagnate or solidify.

Research and practical applications at the Fraunhofer ITWM have shown that this can be greatly improved by designing geometries which avoid regions with low wall shear stress. A low wall shear stress means that there is a large boundary layer with low velocity, where polymer degeneration can take place. Increasing the wall shear stress results in a reduction of this layer and an improvement of the fiber quality. Hence, we need to find a geometry for a polymer flow distributor with an improved wall shear stress distribution. Mathematically we are dealing with a shape optimization problem [10, 11, 16]. See [12, 13] for a similar application of the wall shear stress problem from hemodynamics.

Intrinsically, this is a shape optimization problem with supremum norm cost function, because even a small stagnation area can have negative effects. Therefore, the wall shear stress must be controlled on every part of the distributor surface. However, a supremum norm cost function is not differentiable and therefore solving such a shape optimization problem is a challenge of its own.

The flow in the distributor is modelled by the Stokes equations and it turns out that the biharmonic stream function formulation allows for a straight forward calculation of the wall shear stress (Sect. 2). We present a new algorithm based on conformal mappings, which allows for optimization in the supremum norm and further the incorporation of state constraints (Sect. 3). The numerical results are very promising and yield even bounds on the deviation from the desired wall shear stress distribution. Naturally, the question arises if all desired wall shear stresses are attainable (Sect. 4). Using the concept of approximate controllability, we study the underlying operator and can characterize the set of attainable wall shear stress distributions. Based on these theoretical results, we can expect that also a standard shape optimization using the shape gradient of a least-squares functional will yield promising results (Sect. 5). Here, we can allow for more degrees of freedom in the shape compared to the previous approach.

2 Modelling

In this section we present the mathematical model for the flow in the polymer distributor. We begin by defining the wall shear stress operator $\mathbf{S} : \mathcal{D} \rightarrow L^2(\Gamma_0^w)$, which maps any shape Ω from the set of admissible shapes \mathcal{D} to the corresponding wall shear stress. The set \mathcal{D} consists of all open and bounded domains $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, which are admissible for our shape design problem. Precise definitions for \mathcal{D} depend on the actual setup and are provided later. For $\Omega \in \mathcal{D}$ let \mathbf{n} denote the outward pointing unit normal vector.

Since flow velocities are rather slow while the viscosity of the polymer melt is high, we use the incompressible Stokes equations to model the fluid inside the spin pack. For $\Omega \in \mathcal{D}$ let $\mathbf{u}(\Omega) : \Omega \rightarrow \mathbb{R}^d$ denote the vector valued velocity and let $p(\Omega) : \Omega \rightarrow \mathbb{R}$ denote the scalar valued pressure. Then, Stokes equations read

$$\begin{aligned} -\Delta \mathbf{u}(\Omega) + \nabla p(\Omega) &= 0 & \text{in } \Omega \\ \operatorname{div} \mathbf{u}(\Omega) &= 0 & \text{in } \Omega. \end{aligned} \tag{1}$$

We consider three types of boundary conditions which are needed for the underlying application. Therefore, let the boundary Γ decompose into inflow Γ^{in} , wall Γ^w and outflow Γ^{out} . On the inflow we define a normal velocity profile $w_\Omega : \Gamma^{in} \rightarrow \mathbb{R}$ and set

$$\mathbf{u}(\Omega) = w_\Omega \mathbf{n} \quad \text{on } \Gamma^{in}. \tag{2}$$

On the wall we apply a no-slip and vanishing normal velocity condition, i.e.,

$$\mathbf{u}(\Omega) = 0 \quad \text{on } \Gamma^w. \quad (3)$$

We use an outflow boundary condition to model the spinneret plate as a porous medium. This means that we do not perform a full simulation for the spinneret plate, which is a complex geometry with many small-scale capillaries, but we cover it through the boundary condition:

$$\begin{aligned} \mathbf{u}(\Omega) \times \mathbf{n} &= 0 && \text{on } \Gamma^{out} \\ p - \eta_{out}(\mathbf{n} \cdot \mathbf{u}(\Omega)) &= -g_{out} && \text{on } \Gamma^{out} \end{aligned} \quad (4)$$

where η_{out} and g_{out} are constants modeling the porous medium. The first condition is a no-slip condition and the second is a Darcy law modelling the porous medium.

Altogether, this yields the following boundary value problem: Find velocity and pressure $(\mathbf{u}(\Omega), p(\Omega))$ such that

$$\begin{aligned} -\Delta \mathbf{u}(\Omega) + \nabla p(\Omega) &= 0 && \text{in } \Omega \\ \operatorname{div} \mathbf{u}(\Omega) &= 0 && \text{in } \Omega \\ \mathbf{u}(\Omega) &= w_{\Omega} \mathbf{n} && \text{on } \Gamma^{in} \\ \mathbf{u}(\Omega) &= 0 && \text{on } \Gamma^w \\ p(\Omega) - \eta_{out}(\mathbf{n} \cdot \mathbf{u}(\Omega)) &= -g_{out} && \text{on } \Gamma^{out} \\ \mathbf{u}(\Omega) \times \mathbf{n} &= 0 && \text{on } \Gamma^{out}. \end{aligned} \quad (5)$$

When dealing with the Stokes problem our goal is to study and optimize the wall shear stress which we denote by $\sigma(\Omega) : \Gamma^w \rightarrow \mathbb{R}$. It is defined by

$$\sigma(\Omega) = |\nabla \times \mathbf{u}(\Omega)| \quad \text{on } \Gamma^w. \quad (6)$$

2.1 Wall Shear Stress Operator

Let us define a wall shear stress operator \mathbf{S} which maps any admissible shape to the wall shear stress. However, the following problem arises when trying to define such an operator: For every $\Omega \in \mathcal{D}$ the wall shear stress is a function in $L^2(\Gamma^w)$, where Γ^w is the wall boundary of Ω . Thus the space does depend on the actual choice of $\Omega \in \mathcal{D}$.

In order to get a well-defined operator we use the concept of a reference domain. Let $\Omega_0 \in \mathcal{D}$ be a fixed reference domain with boundary Γ_0 . Assume that for every $\Omega \in \mathcal{D}$ there exists a predefined diffeomorphism

$$G_\Omega : \Gamma_0 \rightarrow \Gamma \tag{7}$$

from the reference boundary to the boundary of Ω . This map does then induce a pull-back operator

$$\begin{aligned} G_\Omega^* : L^2(\Gamma) &\rightarrow L^2(\Gamma_0) \\ f &\mapsto f \circ G_\Omega \end{aligned} \tag{8}$$

and we define the wall shear stress operator

$$\begin{aligned} \mathbf{S} : \mathcal{D} &\rightarrow L^2(\Gamma_0^w) \\ \Omega &\mapsto G_\Omega^* \sigma(\Omega). \end{aligned} \tag{9}$$

We can also use the diffeomorphism G_Ω to derive the inflow boundary condition w_Ω from a given condition $w_0 := w_{\Omega_0}$ on the reference domain by defining

$$w_\Omega := w_0 \circ G_\Omega^{-1}. \tag{10}$$

2.2 Stream Function Formulation

To get the more convenient stream function formulation, we restrict the setting to $d = 2$ and assume that the Darcy boundary Γ^{out} is empty. In this case, Γ^{in} acts as the in- and outflow boundary, by prescribing a given velocity profile on the inflow as well as on the outflow. For this setup the wall shear stress can also be computed by means of the stream function $\Psi(\Omega)$ and vorticity $\omega(\Omega)$ solving the following biharmonic problem (cf. [1]):

$$\begin{aligned} \Delta \Psi(\Omega) &= -\omega(\Omega) && \text{in } \Omega \\ \Delta \omega(\Omega) &= 0 && \text{in } \Omega \\ \Psi(\Omega) &= g_\Omega && \text{on } \Gamma \\ \partial_{\mathbf{n}} \Psi(\Omega) &= 0 && \text{on } \Gamma \end{aligned} \tag{11}$$

The relation between velocity and stream function is

$$\mathbf{u}(\Omega) = \begin{pmatrix} \partial_2 \Psi(\Omega) \\ -\partial_1 \Psi(\Omega) \end{pmatrix}. \tag{12}$$

Here, g_Ω with $\partial_s g_\Omega = 0$ on Γ^w defines the boundary conditions in the following way:

$$\mathbf{n} \cdot \mathbf{u}(\Omega) = \partial_s \Psi(\Omega) = \partial_s g_\Omega \quad \text{on } \Gamma, \quad (13)$$

which vanishes on Γ^w by definition and

$$\boldsymbol{\tau} \cdot \mathbf{u} = -\partial_{\mathbf{n}} \Psi = 0 \quad \text{on } \Gamma, \quad (14)$$

where $\boldsymbol{\tau} = (-n_2, n_1)$ is the tangential vector. Again, we define

$$g_\Omega := g_0 \circ G_\Omega^{-1} \quad (15)$$

where $g_0 := g_{\Omega_0}$ is given for the reference domain Ω_0 .

The wall shear stress is equal to the vorticity at the boundary

$$\sigma(\Omega) = \omega(\Omega) \quad \text{on } \Gamma^w. \quad (16)$$

Without loss of generality we have dropped the absolute value from the definition of the wall shear stress [cf. (6)], because the wall shear stress is scalar valued in the two-dimensional case.

3 Supremum Norm Shape Optimization

Our goal is to control the wall shear stress pointwise in order to prevent any stagnation zones. Therefore, for a given target wall shear stress $\sigma_d \in C^0(\Gamma_0^w)$ we want to compute an optimal solution to the following shape optimization problem with supremum norm cost function:

$$\text{minimize}_{\Omega \in \mathcal{D}} \quad \|\sigma_d - \mathbf{S}(\Omega)\|_{C^0(\Gamma_0^w)}, \quad (17)$$

i.e., we minimize the maximal pointwise error between wall shear stress and target wall shear stress. The norm is defined by $\|f\|_{C^0(\Gamma_0^w)} := \sup_{x \in \Gamma_0^w} |f(x)|$.

Let $\Omega_0 \subset \mathbb{R}^2$ be a simply connected bounded reference domain of class $C^{4,1}$ and let the boundary Γ_0 decompose into the in- and outflow parts Γ_0^{in} and the wall parts Γ_0^w . For simplicity, there is no boundary Γ_0^{out} of Darcy type. For this setup the outflow boundary is part of Γ_0^{in} and is realized by prescribing an outflow velocity with opposite sign.

3.1 Conformal Maps

Conformal maps are a special class of diffeomorphisms which are angle preserving. They can be used to pull-back a shape-dependent problem to a fixed reference domain, leading to an ordinary optimization problem. The shape information is then hidden in a so called conformal parameter, which is a scalar function living on the reference domain.

Definition 1 Let $\Omega_0, \Omega_\alpha \subset \mathbb{R}^2$ be two-dimensional domains. Then a k -diffeomorphism $T = (T_1, T_2) : \Omega_0 \rightarrow \Omega_\alpha, k \in \mathbb{N}$, is called conformal map, if it fulfills the Cauchy-Riemann equations

$$\begin{aligned} \partial_1 T_1 &= \partial_2 T_2 \\ \partial_2 T_1 &= -\partial_1 T_2 \end{aligned} \tag{18}$$

on Ω_0 . Therefore, it is possible to identify conformal maps with holomorphic complex functions. We define the conformal parameter $\alpha \in C^{k-1}(\bar{\Omega}_0)$ such that

$$e^{2\alpha} = \det(DT), \tag{19}$$

where DT is the Jacobian of T . In the following we write $T_\alpha : \Omega_0 \rightarrow \Omega_\alpha$ for a conformal map corresponding to the conformal parameter α . Such a conformal map exists if and only if $\Delta\alpha = 0$ (see [5]).

Conformal maps can also be defined in higher dimensions, however, already in three dimensions the set of reachable domains is negligibly small. On the other hand in two dimensions the Riemann Mapping Theorem states that all simply connected domains can be reached from a simply connected reference domain by conformal deformations.

Theorem 1 (Riemann Mapping Theorem, [14]) *Let $\Omega_0, \Omega_1 \subsetneq \mathbb{R}^2$ be two sufficiently regular simply connected domains. Then, there exists a conformal map $T : \Omega_0 \rightarrow \Omega_1$.*

For this reason the conformal deformations approach is essentially limited to problems on two-dimensional domains.

We define the set of admissible shapes as all shapes which are reachable by certain conformal deformations of a given reference domain Ω_0 :

$$\mathcal{D} := \{\Omega_\alpha = T_\alpha(\Omega_0); \alpha \in \mathcal{A}\} \tag{20}$$

Here, \mathcal{A} denotes the set of admissible conformal parameters, e.g., defined via:

$$\mathcal{A} := \left\{ \alpha \in H^4(\Omega_0); \Delta\alpha = 0; \partial_{\mathbf{n}}\alpha|_{\Gamma_0^{in}} = 0; \dots \right. \tag{21}$$

$$\left. \int_{\Gamma_0^k} e^\alpha ds = \int_{\Gamma_0^k} 1 ds \text{ for every part } \Gamma_0^k \text{ of } \Gamma_0^{in} \right\}$$

The Neumann constraint in the definition of \mathcal{A} makes sure that the curvature of the Γ^{in} boundary is preserved, i.e., straight inflow parts stay straight. The integral constraint makes sure that for every $\Omega_\alpha \in \mathcal{D}$ and every connected component Γ_0^k of Γ_0^{in} the length of Γ_0^k and $\Gamma_\alpha^k = T_\alpha(\Gamma_0^k)$ is equal. These conditions are necessary, because the conformal parameter has a global influence.

3.2 Supremum Norm Shape Optimization Problem

We can now formulate the shape optimization problem (17) using the biharmonic formulation (11). For numerical reasons we have also added a regularization term to the cost function with $\varepsilon > 0$. This yields

$$\begin{aligned}
 & \text{minimize}_{(\Omega_\alpha, \Psi, \omega) \in M_1} \quad \|\sigma_d - T_\alpha^* \omega\|_{C^0(\Gamma_0^w)} + \varepsilon \|\alpha\|_{H^1(\Omega_0)}^2 \\
 & \quad \text{with } M_1 = \mathcal{D} \times H^2(\Omega_\alpha) \times H^2(\Omega_\alpha) \\
 & \text{subject to } \Delta \Psi = -\omega && \text{in } \Omega_\alpha \\
 & \quad \Delta \omega = 0 && \text{in } \Omega_\alpha \\
 & \quad \Psi = g_0 \circ T_\alpha^{-1} && \text{on } \Gamma_\alpha \\
 & \quad \partial_{\mathbf{n}} \Psi = 0 && \text{on } \Gamma_\alpha.
 \end{aligned} \tag{22}$$

Thus, the task is to minimize the supremum norm distance between wall shear stress $\sigma = \omega|_{\Gamma_\alpha^w}$ and target wall shear stress σ_d .

3.2.1 Problem on the Reference Domain

We eliminate the shape-dependence by applying the conformal pull-back operator T_α^* to the whole system. This yields a new optimization problem on the reference domain Ω_0 which is equivalent to (22). Instead of $\Omega_\alpha \in \mathcal{D}$ the conformal parameter $\alpha \in \mathcal{A}$ acts as the control.

$$\begin{aligned}
 & \text{minimize}_{(\alpha, \Psi, \omega) \in M_2} \quad \|\sigma_d - \omega\|_{C^0(\Gamma_0^w)} + \varepsilon \|\alpha\|_{H^1(\Omega_0)}^2 \\
 & \quad \text{with } M_2 = \mathcal{A} \times H^2(\Omega_0) \times H^2(\Omega_0) \\
 & \text{subject to } \Delta \Psi = -e^{2\alpha} \omega && \text{in } \Omega_0 \\
 & \quad \Delta \omega = 0 && \text{in } \Omega_0 \\
 & \quad \Psi = g_0 && \text{on } \Gamma_0 \\
 & \quad \partial_{\mathbf{n}} \Psi = 0 && \text{on } \Gamma_0.
 \end{aligned} \tag{23}$$

Remark 1 For analytical reasons the regularity condition $\mathcal{A} \subset H^4(\Omega_0)$ is necessary to guarantee that the optimization problem is well-defined, i.e., that $\omega \in C^0(\Gamma_0^w)$ holds. However, numerically a $H^1(\Omega_0)$ regularization term is sufficient.

3.2.2 Problem with State Constraints

In order to eliminate the supremum norm from the cost functional, we use a technique discussed, e.g., in [3]. We replace the cost functional by a scalar variable $\delta \in \mathbb{R}$ together with additional inequality constraints, which make sure that the distance between ω and σ_d does not grow bigger than δ . This yields

$$\begin{aligned}
 & \text{minimize}_{(\delta, \alpha, \Psi, \omega) \in M_3} && \delta + \varepsilon \|\alpha\|_{H^1(\Omega_0)}^2 \\
 & \text{with } M_3 = \mathbb{R} \times \mathcal{A} \times H^2(\Omega_0) \times H^2(\Omega_0) \\
 & \text{subject to } \Delta \Psi = -e^{2\alpha} \omega && \text{in } \Omega_0 \\
 & \Delta \omega = 0 && \text{in } \Omega_0 \\
 & \Psi = g_0 && \text{on } \Gamma_0 \\
 & \partial_n \Psi = 0 && \text{on } \Gamma_0 \\
 & \sigma_d - \omega \leq \delta && \text{on } \Gamma_0^w \\
 & -\sigma_d + \omega \leq \delta && \text{on } \Gamma_0^w.
 \end{aligned} \tag{24}$$

This is a nonlinear constrained optimal control problem (see [4]) given on a fixed domain. All geometric information is hidden in the conformal parameter $\alpha \in \mathcal{A}$ and the optimal shape can be recovered later after the optimal α has been computed.

3.2.3 Discretization

In order to solve (24) we derive a full discretization using finite elements, which leads to a Nonlinear Programming Problem (NLP). Note, that additional constraints are hidden in the definition of the set of admissible conformal parameters \mathcal{A} , which are also discretized. For details we refer to [9]. We apply existing methods to solve the NLP. For the following result we have made use of the interior point solver LOQO (see [17]). The reconstruction of Ω_{opt} is done by solving another optimization problem (see [9]).

3.3 Numerical Results

We test the supremum norm shape optimization algorithm by optimizing a two-dimensional polymer distributor. Figure 2 shows the optimal shape Ω_{opt} and the corresponding optimal wall shear stress, together with the target wall shear stress σ_d and the error region, i.e., the region between $\sigma_d - \delta$ and $\sigma_d + \delta$. The black markers help to draw a connection between geometry and arc length plot.

Using the presented approach it is possible to derive optimal shapes for two-dimensional shape optimization problems with supremum norm cost functions. However, when moving on to higher dimensional geometries, problems arise: On the one hand we cannot use conformal mappings and on the other hand the complexity increases due to the rising number of degrees of freedom. We could use more general mappings, but this would add even more degrees of freedom. Therefore, this approach is not suitable for the design of industrial three-dimensional distributor geometries. In the following, we make use of the classical least-square shape optimization to design the industrial spin pack. However, before we do this we take a closer look at the underlying operator and try to answer the question which wall shear stress distributions are attainable.

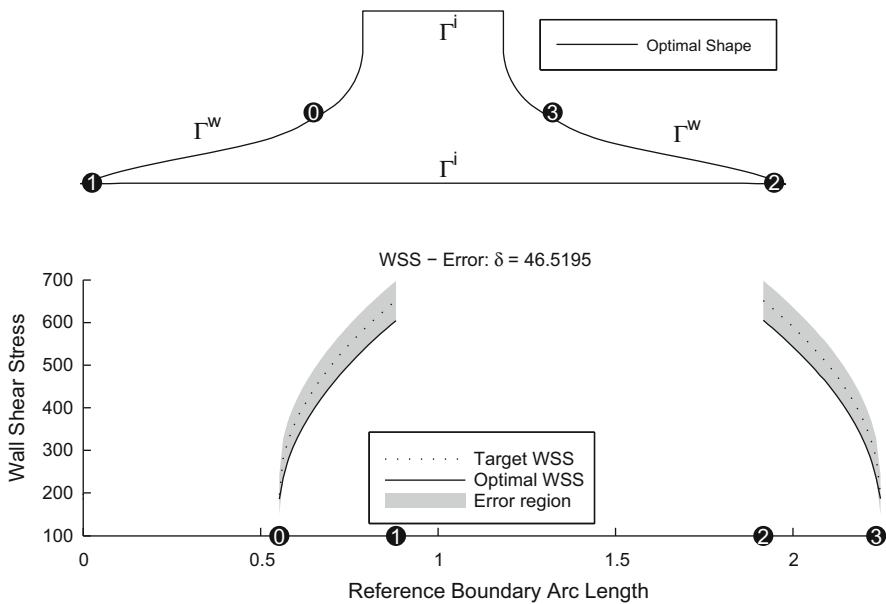


Fig. 2 Optimal wall shear stress distributor for supremum norm shape optimization

4 Linear Controllability

Even though the wall shear stress problem is intrinsically a supremum norm optimization problem, a least-square shape optimization approach may still be successful. Since a least-square cost function is differentiable, this approach is much more efficient, because we can make use of gradient based optimization. The motivation for studying the controllability of the operator is the following consideration: If a given target wall shear stress σ_d is reachable or at least close to reachable by the operator \mathbf{S} , we can expect similar results for different norms. Since the operator \mathbf{S} is highly nonlinear and therefore hard to study, we look at its linearization $d\mathbf{S}$. The idea to study the approximate controllability of linearized shape dependent operators originates in fact from [2]. In [6] we have followed a different approach to characterize the image space of a shape-dependent potential flow operator.

4.1 Geometric Setup

Instead of using conformal shape deformations, we choose a different geometric setup and consider normal boundary deformations. We use the biharmonic formulation (11). Let $\Omega_0 \subset \mathbb{R}^2$ be a bounded domain of class $C^{6,1}$ and let $g_0 \in H^{5+\frac{1}{2}}(\Gamma_0)$ be given with $\partial_s g_0|_{\Gamma_0^w} = 0$. See Remark 2 for a justification of the high regularity requirement. Let the boundary Γ_0 decomposes into the in- and outflow parts Γ_0^{in} and wall parts Γ_0^w .

We define

$$\Theta = \{\theta \in C^{4,1}(\mathbb{R}^2, \mathbb{R}^2); \|\theta\|_{C^{4,1}(\mathbb{R}^2, \mathbb{R}^2)} < 0.5\} \tag{25}$$

to be a ball around zero, where $C^{4,1}(\mathbb{R}^2, \mathbb{R}^2)$ denotes the space of 4-times differentiable functions from \mathbb{R}^2 to \mathbb{R}^2 with Lipschitz-continuous derivatives up to order 4 (see [19]). Let $\text{Id} \in C^{4,1}(\mathbb{R}^2, \mathbb{R}^2)$ denote the identity mapping. For $\theta \in \Theta$ we consider the map

$$\text{Id} + \theta : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \tag{26}$$

i.e., $(\text{Id} + \theta)(x) = x + \theta(x)$. From [15] we know that $\|\theta\|_{C^{4,1}(\mathbb{R}^2, \mathbb{R}^2)} < 0.5$ implies that $\text{Id} + \theta$ is a $(4, 1)$ -diffeomorphism. In order to define the set of admissible shapes let the space of admissible deformation directions be

$$\mathcal{V} := \{\mathbf{V} \in C^{4,1}(\mathbb{R}^2, \mathbb{R}^2); \mathbf{V}|_{\Gamma_0^{in}} = 0; \mathbf{V}|_{\Gamma_0^w} = v_n \mathbf{n}; v_n \in C^{4,1}(\mathbb{R}^2)\}. \tag{27}$$

Note, that since Ω_0 is assumed to be of class $C^{6,1}$ we have $\mathbf{n} \in C^{4,1}(\Gamma_0, \mathbb{R}^2)$. Hence, this definition makes sense. We only consider normal shape deformations, because infinitesimal tangential deformations would shift the boundary along itself and are therefore no real shape deformations. Let the intersection with Θ be denoted by

$$\Theta_{\mathcal{V}} := \Theta \cap \mathcal{V}. \quad (28)$$

We define the set of admissible shapes by

$$\mathcal{D} := \{\Omega_\theta = (\text{Id} + \theta)(\Omega_0); \theta \in \Theta_{\mathcal{V}}\}. \quad (29)$$

Thus, \mathcal{D} is a set of perturbations of the reference domain Ω_0 which leave Γ_0^{in} fixed and are normal on Γ_0^w .

With $G_{\Omega_\theta} := (\text{Id} + \theta)$ we study the operator

$$\begin{aligned} \mathbf{S} : \mathcal{D} &\rightarrow L^2(\Gamma_0^w) \\ \Omega_\theta &\mapsto (\omega(\Omega_\theta)|_{\Gamma_\theta^w}) \circ (\text{Id} + \theta), \end{aligned} \quad (30)$$

where the stream function $\Psi(\Omega_\theta)$ and vorticity $\omega(\Omega_\theta)$ are the solutions of

$$\begin{aligned} \Delta \Psi(\Omega_\theta) &= -\omega(\Omega_\theta) && \text{in } \Omega_\theta \\ \Delta \omega(\Omega_\theta) &= 0 && \text{in } \Omega_\theta \\ \Psi(\Omega_\theta) &= g_0 \circ (\text{Id} + \theta)^{-1} && \text{on } \Gamma_\theta \\ \partial_{\mathbf{n}} \Psi(\Omega_\theta) &= 0 && \text{on } \Gamma_\theta. \end{aligned} \quad (31)$$

Definition 2 The corresponding linearized wall shear stress operator $d\mathbf{S}$ is defined by

$$\begin{aligned} d\mathbf{S} : \mathcal{V} &\rightarrow L^2(\Gamma_0^w) \\ \mathbf{V} &\mapsto \frac{d\mathbf{S}(\Omega_\theta)}{d\theta}(0)\mathbf{V}, \end{aligned} \quad (32)$$

i.e., it is the derivative of $\mathbf{S}(\Omega_\theta) = \mathbf{S}((\text{Id} + \theta)(\Omega_0))$ with respect to θ in direction $\mathbf{V} \in \mathcal{V}$ evaluated at $\theta = 0$.

We have shown in [8] that the linearized operator $d\mathbf{S}$ is well-defined and can be computed by

$$\begin{aligned} d\mathbf{S} : \mathcal{V} &\rightarrow L^2(\Gamma_0^w) \\ \mathbf{V} &\mapsto \omega'(\mathbf{V})|_{\Gamma_0^w} + \partial_{\mathbf{n}}\omega(\Omega_0)(\mathbf{n} \cdot \mathbf{V}), \end{aligned} \quad (33)$$

where $\Psi'(\mathbf{V})$ and $\omega'(\mathbf{V})$ are the shape derivatives (cf. [16]) in direction \mathbf{V} and can be derived as the solution of

$$\begin{aligned}
 \Delta\Psi'(\mathbf{V}) &= -\omega'(\mathbf{V}) && \text{in } \Omega_0 \\
 \Delta\omega'(\mathbf{V}) &= 0 && \text{in } \Omega_0 \\
 \Psi'(\mathbf{V}) &= 0 && \text{on } \Gamma_0 \\
 \partial_{\mathbf{n}}\Psi'(\mathbf{V}) &= 0 && \text{on } \Gamma_0^{in} \\
 \partial_{\mathbf{n}}\Psi'(\mathbf{V}) &= (\mathbf{n} \cdot \mathbf{V})\omega(\Omega_0) && \text{on } \Gamma_0^w.
 \end{aligned}
 \tag{34}$$

Definition 3 (Approximate Controllability) Let $F : X \rightarrow Y$ be a linear operator. Then, F is approximately controllable if and only if $\text{im } F$ lies dense in Y .

Our goal is to prove the following result about the approximate controllability of the linearized wall shear stress operator:

Theorem 2 *Let Ω_0 be bounded and of class $C^{6,1}$ and assume that $\mathbf{S}(0) \neq 0$ on Γ_0^w . Then, the operator $d\mathbf{S} : \mathcal{V} \rightarrow L^2(\Gamma_0^w)/\mathcal{Z}_{\partial_{\mathbf{n}}}$ is approximately controllable. Here $\mathcal{Z}_{\partial_{\mathbf{n}}} = \{\partial_{\mathbf{n}}\phi|_{\Gamma_0^w} \in L^2(\Gamma_0^w); \phi \in H^4(\Omega_0) \text{ solution of (35)}\}$ is a finite dimensional subspace of $L^2(\Gamma_0^w)$.*

Remark 2 The assumptions of this section include a very high regularity requirement of $C^{6,1}$ for the reference domain Ω_0 . For the well-definedness of the operator \mathbf{S} itself, $C^{4,1}$ would suffice, because this would provide the existence of the trace of $\omega(\theta)$. It is also true that in many parts of this section the regularity assumptions can be relaxed by applying weak arguments. However, a key part for the final proof is the V -coercivity of the bilinear form corresponding to the uniqueness problem (35), which due to [19] does require $c_{11} \in C^1(\bar{\Omega}_0)$ for the coefficient of the boundary form. And by definition of that coefficient this requires $C^{6,1}$ for Ω_0 . However, since this coefficient is only needed on Γ_0^w , the regularity requirement can probably be weakened for the other boundary parts.

For the proof of Theorem 2 we need the following uniqueness lemma, which follows from [19, Theorem 17.11] (cf. [8]):

Lemma 1 *Assume that Ω_0 is bounded and of class $C^{4,1}$ and $c_{11} \in C^1(\bar{\Omega}_0)$. We consider*

$$\begin{aligned}
 \Delta\Delta\phi &= 0 && \text{in } \Omega_0 \\
 \phi &= 0 && \text{on } \Gamma_0 \\
 \partial_{\mathbf{n}}\phi &= 0 && \text{on } \Gamma_0^{in} \\
 \Delta\phi + c_{11}\partial_{\mathbf{n}}\phi &= 0 && \text{on } \Gamma_0^w
 \end{aligned}
 \tag{35}$$

and define $\mathcal{Z} := \{\phi \in H^4(\Omega_0); \phi \text{ solves (35)}\}$. Then, \mathcal{Z} is a finite dimensional subspace of $H^4(\Omega_0)$.

Proof of Theorem 2 Define

$$H_{i=0}^{\frac{5}{2}}(\Gamma_0) = \{\mu \in H^{\frac{5}{2}}(\Gamma_0); \mu = 0 \text{ on } \Gamma_0^{in}\} \tag{36}$$

and for $\mu \in H_{i=0}^{\frac{5}{2}}(\Gamma_0)$ let $\phi(\mu) \in H^4(\Omega_0)$ be the solution of the adjoint problem

$$\begin{aligned} \Delta\Delta\phi(\mu) &= 0 && \text{in } \Omega_0 \\ \phi(\mu) &= 0 && \text{on } \Gamma_0 \\ \partial_{\mathbf{n}}\phi(\mu) &= \mu && \text{on } \Gamma_0. \end{aligned} \tag{37}$$

For $(\mathbf{V}, \mu) \in \mathcal{V} \times H_{i=0}^{\frac{5}{2}}(\Gamma_0)$ integration by parts yields

$$\begin{aligned} 0 &= \int_{\Omega_0} \Delta\Delta\Psi'(\mathbf{V})\phi(\mu) \, dx \\ &= \int_{\Omega_0} \Delta\Psi'(\mathbf{V})\Delta\phi(\mu) \, dx - \int_{\Gamma_0} \Delta\Psi'(\mathbf{V})\partial_{\mathbf{n}}\phi(\mu) \, ds \\ &= \int_{\Omega_0} \Psi'(\mathbf{V})\Delta\Delta\phi(\mu) \, dx - \int_{\Gamma_0} \Delta\Psi'(\mathbf{V})\partial_{\mathbf{n}}\phi(\mu) \, ds + \int_{\Gamma_0} \partial_{\mathbf{n}}\Psi'(\mathbf{V})\Delta\phi(\mu) \, ds \end{aligned} \tag{38}$$

and we get the identity

$$- \int_{\Gamma_0^w} \omega'(\mathbf{V})\mu \, ds = \int_{\Gamma_0^w} (\mathbf{n} \cdot \mathbf{V})\omega(0)\Delta\phi(\mu) \, ds. \tag{39}$$

Now, assume that $\mu \in \text{im}(d\mathbf{S})^\perp \cap H_{i=0}^{\frac{5}{2}}(\Gamma_0)$, i.e.,

$$\int_{\Gamma_0^w} d\mathbf{S}(\mathbf{V})\mu \, ds = 0 \quad \text{for all } \mathbf{V} \in \mathcal{V}. \tag{40}$$

We conclude

$$\begin{aligned} 0 &= \int_{\Gamma_0^w} d\mathbf{S}(\mathbf{V})\mu \, ds \\ &= \int_{\Gamma_0^w} \omega'(\mathbf{V})\mu \, ds + \int_{\Gamma_0^w} \partial_{\mathbf{n}}\omega(0)(\mathbf{n} \cdot \mathbf{V})\mu \, ds \\ &= \int_{\Gamma_0^w} (\mathbf{n} \cdot \mathbf{V})(-\omega(0)\Delta\phi(\mu) + \partial_{\mathbf{n}}\omega(0)\partial_{\mathbf{n}}\phi(\mu)) \, ds. \end{aligned} \tag{41}$$

Since $\{\mathbf{n} \cdot \mathbf{V}; \mathbf{V} \in \mathcal{V}\}$ is dense in $L^2(\Gamma_0^w)$ we derive

$$-\omega(0)\Delta\phi(\mu) + \partial_{\mathbf{n}}\omega(0)\partial_{\mathbf{n}}\phi(\mu) = 0 \quad \text{on } \Gamma_0^w. \tag{42}$$

Because of $\omega(0) = \mathbf{S}(0) \neq 0$ on Γ_0^w , we can define

$$c_{11} := -\frac{\partial_{\mathbf{n}}\omega(0)}{\omega(0)} \in C^1(\Gamma_0^w). \tag{43}$$

This yields the uniqueness problem

$$\begin{aligned} \Delta\Delta\phi(\mu) &= 0 && \text{in } \Omega_0 \\ \phi(\mu) &= 0 && \text{on } \Gamma_0 \\ \partial_{\mathbf{n}}\phi(\mu) &= 0 && \text{on } \Gamma_0^{in} \\ \Delta\phi(\mu) + c_{11}\phi(\mu) &= 0 && \text{on } \Gamma_0^w. \end{aligned} \tag{44}$$

Define

$$\mathcal{L} := \{\phi(\mu) \in H^4(\Omega_0); \phi(\mu) \text{ is solution of (44)}\} \tag{45}$$

and

$$\mathcal{L}_{\partial_{\mathbf{n}}} := \{\mu = \partial_{\mathbf{n}}\phi|_{\Gamma_0^w}; \phi \in \mathcal{L}\}. \tag{46}$$

By Lemma 1 we know that \mathcal{L} is a finite dimensional subspace of $H^4(\Omega_0)$. Then, $\mathcal{L}_{\partial_{\mathbf{n}}}$ is a finite dimensional subspace of $H_{i=0}^{\frac{5}{2}}(\Gamma_0)|_{\Gamma_0^w}$ and thus of $L^2(\Gamma_0^w)$. Hence we conclude that $d\mathbf{S}$ is approximately controllable as a mapping to $L^2(\Gamma_0^w)/\mathcal{L}_{\partial_{\mathbf{n}}}$.

4.2 Conclusions

We were able to prove that the linearized wall shear stress operator is approximately controllable up to a finite dimensional subspace. Even though we have studied the linearizations, we can draw conclusions for the actual operator. Having the approximate controllability property for the linearization means that we can change the wall shear stress into almost every direction by applying infinitesimal shape deformations. This does suggest that the space of reachable wall shear stress profiles is rather large. But if a given target wall shear stress is close to reachable, we can hope that optimal shapes obtained by least-square shape optimization are also good in the supremum norm sense.

5 Least-Squares Shape Optimization

Encouraged by the controllability results, we use classical least-squares shape optimization to solve the real three-dimensional industrial problem. Since a least-squares cost function is differentiable, we can compute the gradient via a system of adjoint equations. This enables us to solve the optimization problem using gradient based optimization.

5.1 Optimization Problem

Let $\sigma_d \in H^1(\mathbb{R}^3)$ be a given target wall shear stress. Our goal is to find a shape whose wall shear stress is close to the target wall shear stress in the least-squares sense. We want to solve

$$\begin{aligned}
 & \text{minimize}_{\Omega \in \mathcal{D}} \quad J(\Omega) = \int_{\Gamma^w} (|\nabla \times \mathbf{u}| - \sigma_d)^2 ds & (47) \\
 & \text{subject to} \quad -\Delta \mathbf{u} + \nabla p = 0 & \text{in } \Omega \\
 & \quad \quad \quad \text{div } \mathbf{u} = 0 & \text{in } \Omega \\
 & \quad \quad \quad \mathbf{u} = w_0 \mathbf{n} & \text{on } \Gamma^{in} \\
 & \quad \quad \quad \mathbf{u} = 0 & \text{on } \Gamma^w \\
 & \quad \quad \quad p - \eta_{out}(\mathbf{n} \cdot \mathbf{u}) = -g_{out} & \text{on } \Gamma^{out} \\
 & \quad \quad \quad \mathbf{u} \times \mathbf{n} = 0 & \text{on } \Gamma^{out},
 \end{aligned} \tag{48}$$

with

$$\mathcal{D} := \{\Omega \subset \mathbb{R}^3 \text{ sufficiently regular, } \Gamma^{in} = \Gamma_0^{in}, \Gamma^{out} = \Gamma_0^{out}\}. \tag{49}$$

Thus, \mathcal{D} consists of all domains Ω whose in- and outflow boundaries agree with a fixed reference or initial domain Ω_0 .

5.2 Gradient

In a similar way as in Sect. 4 we apply deformations of the wall boundaries. That is for a domain $\Omega_k \in \mathcal{D}$ we apply deformations from the space

$$\mathcal{V}_k := \{\mathbf{V} \in H^2(\mathbb{R}^3, \mathbb{R}^3); \mathbf{V}|_{\Gamma_0^{in} \cup \Gamma_0^{out}} = 0\}. \tag{50}$$

Thus, for a deformation $\theta \in \mathcal{V}_k$ the domain Ω_k is deformed into the domain $\Omega_\theta := (\text{Id} + \theta)(\Omega_k)$. And by definition of \mathcal{V}_k the in- and outflow boundaries of Ω_θ remain unchanged.

We use a gradient descent method to solve the shape optimization problem and therefore the gradient must be computed. The gradient $\mathbf{V}_k \in H^2(\Omega_k)$ can be derived using the following variational formulation (for details see [5]):

$$(\mathbf{V}_k, v)_{H^2(\Omega_0)} = dJ(v) \quad \text{for all } v \in \mathcal{V}_k, \tag{51}$$

with

$$\begin{aligned} dJ(\mathbf{V}) = & - \int_{\Gamma_k^w} (\partial_{\mathbf{n}} \mathbf{u}(0) \times \mathbf{n}) \cdot \nabla \times \mathbf{v} (\mathbf{n} \cdot \mathbf{V}) \, ds \\ & + \int_{\Gamma_k^w} 2 \frac{\sigma(0) - \sigma_d}{\sigma(0)} \nabla \times \mathbf{u}(0) \cdot \partial_{\mathbf{n}} (\nabla \times \mathbf{u}(0)) (\mathbf{n} \cdot \mathbf{V}) \, ds \\ & - \int_{\Gamma_k^w} 2(\sigma(0) - \sigma_d) \partial_{\mathbf{n}} \sigma_d (\mathbf{n} \cdot \mathbf{V}) \, ds + \int_{\Gamma_k^w} \kappa (\sigma(0) - \sigma_d)^2 (\mathbf{n} \cdot \mathbf{V}) \, ds. \end{aligned} \tag{52}$$

Furthermore (\mathbf{v}, q) are the solution of the following adjoint Stokes problem:

$$\begin{aligned} -\Delta \mathbf{v} + \nabla q &= 0 && \text{in } \Omega_k \\ \text{div } \mathbf{v} &= 0 && \text{in } \Omega_k \\ \mathbf{v} &= 0 && \text{on } \Gamma_k^{in} \cup \Gamma_k^f \\ \mathbf{n} \cdot \mathbf{v} &= 0 && \text{on } \Gamma_k^w \\ \mathbf{v} \times \mathbf{n} &= 2 \frac{\sigma - \sigma_d}{\sigma} (\nabla \times \mathbf{u}(0)) && \text{on } \Gamma_k^w \\ q - \eta_{out} (\mathbf{n} \cdot \mathbf{v}) &= 0 && \text{on } \Gamma_k^{out} \\ \mathbf{v} \times \mathbf{n} &= 0 && \text{on } \Gamma_k^{out} \end{aligned} \tag{53}$$

This enables us to solve the optimization problem using a gradient descent method. For details we refer to [5].

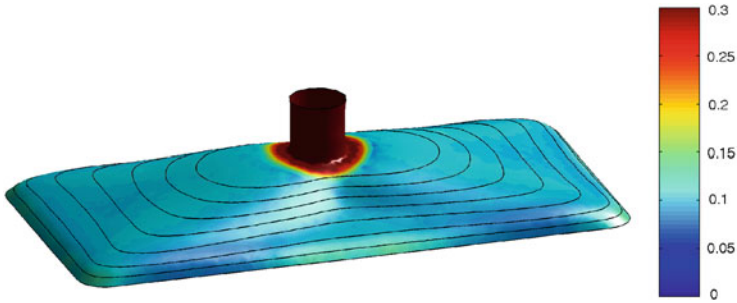


Fig. 3 Optimal distributor geometry for a uniform target wall shear stress of $\sigma_d = 0.1$ (nondimensional). The wall shear stress is indicated by *color*

5.3 Numerical Results

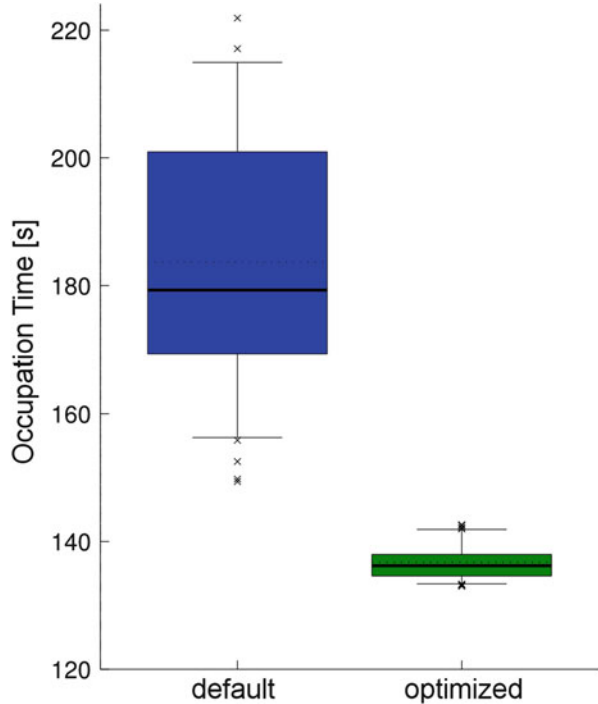
See Fig. 3 for an optimal distributor geometry for a uniform target wall shear stress of $\sigma_d = 0.1$ (nondimensional). The setup includes a parabolic velocity profile at the inflow tube and a Darcy type outflow condition to model the spinneret plate. We use an implementation based on COMSOL Multiphysics.

The wall shear stress of the optimal geometry is very close to the target wall shear stress. On the one hand this agrees with Theorem 2 and the interpretation that most wall shear stresses are attainable. On the other hand this provides us with a powerful tool to design new industrial distributor geometries while having extensive control over the wall shear stress.

6 Outlook

The results presented in this article have been obtained in the joint research project ProFil, which was funded by the Federal Ministry of Education and Research (BMBF). The utilization of the results concerning the spin pack is taking place in the ongoing project AUTOPOS funded by AiF (German Federation of Industrial Research Associations). AUTOPOS is a joint project between Fraunhofer ITWM, ITA (Institute for Textile Engineering, Aachen) and ten partners from industry. One of the goals is a complete redesign of industrial spin packs. Intermediate results (see [7]) from AUTOPOS show that a redesign of a spin pack can lead to a great improvement of the polymer occupation time: Figure 4 shows that both magnitude and variance of the occupation time is reduced.

Fig. 4 Occupation time for two spin pack designs. The bars show the deviations along different pathlines together with the median



Acknowledgements This work was supported by the German Federal Ministry of Education and Research (BMBF) grant no. 03MS606F and by the German Federation of Industrial Research Associations (AiF) grant no. 17629N.

References

1. Anderson, J., Wendt, J.: Computational Fluid Dynamics, vol. 206. McGraw-Hill, New York (1995)
2. Chenais, D., Zuazua, E.: Controllability of an elliptic equation and its finite difference approximation by the shape of the domain. *Numer. Math.* **95**, 63–99 (2003)
3. Grund, T., Rösch, A.: Optimal control of a linear elliptic equation with a supremum norm functional. *Optim. Methods Softw.* **15**, 299–329 (2001)
4. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: Optimization with PDE Constraints, vol. 23. Springer, Berlin (2009)
5. Leithäuser, C.: Controllability of shape-dependent operators and constrained shape optimization for polymer distributors. PhD Thesis, TU Kaiserslautern (2013)
6. Leithäuser, C., Feßler, R.: Characterizing the image space of a shape-dependent operator for a potential flow problem. *Appl. Math. Lett.* **25**(11), 1959–1963 (2012)
7. Leithäuser, C., Gramsch, S., Hietel, D., Wegener, R.: Modellierung und Simulation entlang der gesamten Vliesstoff-Prozesskette. In: Proceedings, vol. 28. Hofer Vliesstofftage (2013)
8. Leithäuser, C., Pinnau, R., Feßler, R.: Approximate controllability of linearized shape-dependent operators for flow problems. *ESAIM: Control Optim. Calc. Var.* **23**(3), 751–771 (2017)

9. Leithäuser, C., Pinnau, R., Feßler, R.: A numerical approach to shape optimization with state constraints. arXiv:1412.4350 (2014)
10. Mohammadi, B., Pironneau, O.: Shape optimization in fluid mechanics. *Annu. Rev. Fluid Mech.* **36**, 255–279 (2004)
11. Pironneau, O.: *Optimal Shape Design for Elliptic Systems*. Springer, Berlin (1984)
12. Quarteroni, A., Rozza, G.: Optimal control and shape optimization of aorto-coronary bypass anastomoses. *Math. Models Methods Appl. Sci.* **13**, 1801–1824 (2003)
13. Rozza, G.: On optimization, control and shape design of an arterial bypass. *Int. J. Numer. Methods Fluids* **47**, 1411–1419 (2005)
14. Schinzinger, R., Laura, P.: *Conformal Mapping: Methods and Applications*. Dover Publications, New York (2003)
15. Simon, J.: Differentiation with respect to the domain in boundary value problems. *Numer. Funct. Anal. Optim.* **2**, 649–687 (1980)
16. Sokolowski, J., Zolesio, J.: *Introduction to Shape Optimization: Shape Sensitivity Analysis*, vol. 16. Springer, Berlin (1992)
17. Vanderbei, R., Shanno, D.: An interior-point algorithm for nonconvex nonlinear programming. *Comput. Optim. Appl.* **13**, 231–252 (1999)
18. Wegener, R., Marheineke, N., Hietel, D.: Virtuelle Produktion von Filamenten und Vliesstoffen. In: Neunzert, N., Prätzel-Wolters, D. (eds.) *Mathematik im Fraunhofer-Institut Problemgetrieben - Modellbezogen - Lösungsorientiert*, pp. 105–165. Springer, Berlin (2014)
19. Wloka, J.: *Partial Differential Equations*. Cambridge University Press, Cambridge (1987)

Author Index

B

Bærentzen, A., 253–271
Bär, T., 117–132
Borzykh, M., 273–288
Brander, D., 253–271
Bukata, L., 135–159
Burget, P., 135–159

C

Carlson, J.S., 231–249

D

Damerow, U., 273–288
Damrath, F., 117–132
David-Henriet, X., 37–59
Dreßler, K., 289–318
Drótos, M., 3–18

E

Edelvik, F., 231–249
Evgrafov, A., 253–271

G

Gerdtts, M., 205–227
Ghezzi, L., 61–91
Göttlich, S., 21–35
Gräler, M., 273–288
Gravesen, J., 253–271

H

Hajba, T., 163–181
Hanzálek, Z., 135–159
Hardouin, L., 37–59
Herty, M., 21–35
Hömberg, D., 183–201
Horváth, Z., 163–181

J

Johnson, T., 231–249
Jósvai, J., 163–181

K

Karlsson, N., 231–249
Kiss-Tóth, C., 163–181
Kis, T., 3–18
Krüger, M., 273–288

L

Landry, C., 183–201
Leithäuser, C., 321–339
Linn, J., 289–318
Luckert, M., 21–35
Lüder, A., 93–112

M

Mark, A., 231–249
Markvorsen, S., 253–271

N

Nørbjerg, T.B., 253–271
Nørtoft, P., 253–271

P

Palagachev, K.D., 205–227
Pinnau, R., 321–339

R

Raisch, J., 37–59
Ron, M., 135–159

S

Schmidt, N., 93–112
Skutella, M., 183–201

Steenstrup, K., 253–271
Strahilov, A., 117–132
Šucha, P., 135–159

T

Trächtler, A., 273–288

V

Vielhaber, M., 117–132

W

Welz, W.A., 183–201

Subject Index

A

Activation cost, 83
Activities, 3–9, 11, 39, 90, 94, 103, 108, 131, 148, 152–154, 156, 275
Actuators, 96, 118, 120, 137, 274–279, 284
Admissible region, 81
Air flow, 125, 232, 234, 322
 solver, 239
(max, +)-algebra, 38
Ant-colony algorithms (ACO), 167–169
Assemble-to-order (ATO), 61, 73
Assembling, 140, 152, 153, 158, 271
Assembly operation, 125, 129, 315
Automated assembly systems, 117–132
Automotive manufacturing, 118, 120, 131, 231, 291

B

Bending energy, 258, 265, 266
Bilevel optimal control problem, 211, 212, 215, 222
Bilevel optimization problems, 208, 227
Bilevel scheduling optimal control problem (BLSOCP), 206, 208, 209
Bill of operation, 94, 98
Binary vectors, 8–9
Blocking, 29, 266, 267
Boolean decision variables, 88
Branch and bound algorithm, 216, 218
Branch-and-bound method, 8, 15, 89, 224
Branch-and-cut, 8, 9
Branch-and-price, 8–9, 190, 191
Branching, 15, 191, 217–219
Budget, 67, 68, 82, 86, 89–91

Budget constraint, 62, 85, 89
Buffers, 22–24, 26–28, 35, 38, 39, 53, 55–59, 164, 165, 176, 180–181

C

Cables, 125, 289–318
Cable simulation, 290, 291, 313, 314, 317
CAD, 136
 design, 257
 software, 145
CAE software, 291
Capacity, 4–7, 23–28, 39, 40, 63, 64, 163
Car manufacturers, 135
Cell, 17, 100, 102, 103, 106, 148, 149, 157, 158, 184, 204, 1367
Clearing function, 22, 25, 26, 28–30, 32, 33
Closed-loop, 58, 138, 274, 276–278, 280, 281, 286, 287
CNC milling, 271
Coating processes, 233
Collision, 122, 127, 184–188, 193–199, 206, 225
Collision-free, 137, 184, 185, 194, 196, 201
Column generation, 185, 190, 192, 193
Combinatorial optimization problems, 167, 169, 187
Computational geometry, 188, 291, 299, 314, 317
Computational mechanics, 290
Computer integrated manufacturing (CIM), 94
Concave, 22, 67, 78, 79, 81–83, 85–89, 91
Confidence level, 63, 70, 74
Conflict-free, 186–193
Conservation laws, 22, 23

- Constrained optimization, 64, 68, 81, 262, 265
 Constraint, 10, 40, 62, 80–82, 85, 87, 89,
 191–193, 195, 196, 207, 212, 217,
 221, 222, 224, 262
 Constraint integer program (CIP), 191, 193
 Constraint propagation, 15
 Constraint qualification, 212, 222
 Constructive heuristics, 9
 Control, 53–59
 theory, 38
 unit, 276
 Convex, 79–81, 87, 141, 154, 194, 195, 197,
 218, 219, 227
 Convex hull, 8
 Correlation, 139, 149–152
 Cosserat curves, 291, 295–297, 301–307, 309,
 311, 317
 Cosserat rod, 289–318
 Cost, 6, 9, 12, 61, 62, 64, 67, 68, 83–91, 118,
 128, 131, 150, 189, 196, 199, 232,
 233, 338
 Coupled simulations, 232
 Cplex, 156–158, 172, 178, 189
 Critical path, 169, 170
 Cross-covariance, 150
 Cumulated density function (CDF), 70
 Cycle time, 122, 125–127, 130, 136, 138, 142,
 184–187, 189, 200, 201, 249
- D**
- Darboux frame, 296
 Data filtering, 149
 Data-fitting, 31
 Data regularization, 77–82
 Dater, 43–45, 48, 177
 Decision variables, 6, 13, 88, 89
 Deep drawing process, 281
 Delay, 38, 54, 56, 225
 Deliveries, 46, 61, 64
 Demand Driven MRP (DDMRP), 63
 Depth-first approach, 218
 DES. *See* Discrete event simulation (DES)
 DES simulation, 22, 27, 28, 30–32, 34, 64
 Detection, 82, 149, 184, 185, 188, 197–199
 Digital factory, 3–18, 62, 122, 163, 164, 181,
 271, 287
 Digital planning, 118
 Digital validation, 291, 317
 Dijkstra, 187
 Dioid theory, 41–45
 Discrete event simulation (DES), 21–23, 64
 Disjunctive variables, 87
- Distance, 79, 80, 151, 157, 188, 194, 195,
 197–199, 206, 217, 233, 240, 243,
 247, 265, 269, 292, 299, 302, 307,
 328, 329
 Drag coefficient, 238
 Droplet, 236–240
 Droplet solver, 238
 Due dates, 11
 Dynamic(s), 21, 22, 31, 34, 35, 40, 47, 48,
 53, 54, 62–64, 69–72, 77, 122, 125,
 129–131, 138, 143, 207, 215, 219,
 227, 275–279, 284, 318
 consumption, 143
 model, 34, 139, 147, 285
- E**
- Economic order quantity (EOQ), 63
 ECUs. *See* Energy consumption units (ECUs)
 Effective stiffness properties, 298
 Elastica problem, 258
 Elastic curve segments, 258–261, 268
 Elastic modulus, 283–285
 Electrocoating, 234
 Electromagnetic fields, 232
 Electrostatic solver, 236–240, 247
 Energy, 135
 consumption, 130, 137, 144, 147, 153, 158
 function, 137, 140, 143, 147, 153, 158
 optimisation, 136, 137
 savings, 125, 139, 158
 signature, 126
 Energy consumption optimization, 117–132
 Energy consumption units (ECUs), 120, 125,
 126, 131
 Energy efficiency improvement measures
 (EIEMs), 118, 123–125
 Energy-efficient, 117, 118, 123, 125, 127, 128,
 131, 136
 Energy optimisation problem, 156
 Enterprise resource planning (ERP), 100
 EPS cutting, 271
 Euler-lagrange equation, 258–259
 Exact methods, 8
 External supplies, 46
- F**
- Factory data, 30
 Failures, 55, 163, 280
 Feasible domains, 7
 Feasible region, 87
 Feasible solution, 9, 184, 189, 191–193, 218

- Feedback, 53–59, 167, 274, 276, 280
 Feed-back control, 280, 285
 Fiber spinning, 321
 Finished products, 25, 46
 Fixed cost, 83
 Flexible grommet, 316, 317
 Flow, 25, 40, 154, 239, 240, 247, 283, 323, 331
 Flow shop problem, 165, 177
 Fluid-structure interaction, 249
 Flux function, 22, 23, 25, 29
 Force measurement, 285
 Framed curves, 289–318
 Free surface flows, 234
 Frénet frame, 296
 Function groups, 98, 106
- G**
- Gauss quadrature, 76
 Geodesic curves, 262
 Glue, 158, 249
 Goal function, 80, 81, 85
 Graph, 39, 40, 79, 141–143, 170, 185, 186
- H**
- Head-up display, 315, 316
 Hemming process, 249
 Heuristical methods, 9
 Heuristics, 8, 9, 15, 18, 64, 166–172, 177, 178
 Hierarchical control strategies, 276
 Hierarchy, 21, 97, 99–101, 103–109, 199, 278
 High-volume production, 21
 Historical data base, 71
 Hoses, 289, 316, 317
 Hot blade cutting, 254–258, 261, 271
 Hot wire cutting, 255–257
- I**
- Immersed boundary method, 236
 Inequality, 8, 46, 87
 Infeasible solutions, 9
 Inflectional elastica, 259
 Initial guess, 81, 262–264, 268
 Initial trajectory, 196
 Integer linear program (ILPs), 152, 156, 158, 189–191
 Integer programming, 87, 179
 Integer solutions, 8
 Intelligent control of production processes, 274
 In-teraction, 47, 97, 105, 122, 124, 205–207, 225, 238, 249, 293, 313, 314, 317
 Interarrival times, 27, 30–32
- Interface, 100, 101, 110, 237, 287, 290, 314
 Inventories, 23, 24, 61–63, 73, 90, 91
- J**
- Job shop problem, 206
 Just in Time (JIT), 64, 206
- K**
- Kanban, 64
 Key performance indicator (KPI), 82
- L**
- Lagrange multipliers, 156
 Lagrangian duality, 14
 Lagrangian particles, 236, 238
 Lagrangian relaxation, 11, 14–15, 139, 156–158
 Large scale, 23, 35, 164, 181
 Linear programming (LP) problem, 8, 67–69, 85, 152, 158
 (max, +)-linear systems, 38, 41
 Local search procedure, 168
 Logistic, 62–64, 72, 91, 271
 Lower bounds, 11, 14, 15, 86, 157, 176–181, 197
- M**
- Machine breakdowns, 55
 Machines, 4, 11, 12, 16, 24, 26, 28, 38–40, 56, 125, 137, 138, 165, 174, 176–181, 275, 276, 278, 279, 284
 Makespan, 165, 168, 169, 171, 174, 175, 177, 185
 Makespan minimization, 4
 Make-to-order (MTO), 73
 Manne family, 173
 Manufacturing data, 163, 181
 Manufacturing execution control (MES), 100
 Manufacturing resource planning (MRP II), 63
 Market demand, 63, 65, 67, 69–72
 Mass conservation, 25, 237
 Material flow, 21–35
 Material requirement planning (MRP), 63
 Mathematical process models, 274
 Mathematical programming, 64, 82, 87
 Measurements, 136–139, 143–145, 147, 149, 159, 231, 233, 239, 241–243, 245–248, 277, 281
 Mechatronic, 96, 118
 Mechatronical engineering, 93–112

- Mechatronic library, 103, 105
 Mechatronic systems, 118, 131, 274, 278–280
 Mixed-integer dynamic optimization problem, 224
 Mixed-integer linear programming (MILP), 5–7, 13–14, 67, 69, 88, 172, 178, 181
 Mixed-integer modeling, 34
 Mixed integer nonlinear optimization problem, 209, 215
 Mixed-integer optimal control problem, 211, 222, 225
 Model-based system development, 277
 Model driven engineering, 110
 Model pattern, 150, 151
 Monte Carlo (MC), 67, 68, 74–75
 Motion-planning, 184, 188, 195
 Movements, 91, 127, 136–141, 143–154, 187, 201
 Multi-commodity productive system, 91
 Multi-phase flow, 234, 240
 Multi-physics aspects, 234
 Multi-scale phenomena, 234
- N**
- Navier-Stokes solver, 232, 235
 Nonlinear optimization, 188, 196, 209, 215, 217, 262
 Nonlinear programming problem (NLP), 217, 218, 329
- O**
- Objective functions, 5–7, 12, 14–16, 166, 167, 177, 189, 192, 212, 216, 217, 222, 263
 Obstacle, 187, 193, 221, 225–227
 Obstacle avoidance, 225
 Online process control, 274
 Open-loop, 58, 275, 276
 Open-loop control vs. closed-loop control, 276
 Optimal control problems, 138, 185, 187, 188, 193, 196, 205–209, 211, 212, 215, 221, 222, 225, 227, 329
 Optimality, 5, 8, 9, 91, 157, 190, 209, 218, 227
 Optimality gap, 16
 Optimal scheduling, 163, 164, 206–209, 219, 227
 Optimal strategy, 91
- Optimization problem, 13, 14, 64, 78, 81, 84, 91, 108, 109, 137, 165, 189, 196, 206, 208, 209, 215, 217, 224, 225, 262, 263, 267, 322, 323, 326–329, 331, 336, 337
 Orders from customers, 46, 53
 Ordinary differential equations, 22, 196, 200, 296
 Orthogonal polynomials, 74
 Orthogonal polynomial sequence (OPS), 75
 Overlap, 5, 8, 14, 125, 194, 199
 Overloads, 22
- P**
- PACO algorithm, 167–169, 180
 PACO heuristics, 180, 181
 Paint droplets, 232, 236
 Palettes, 164, 165, 176, 180–181
 Parallel computing, 16
 Parameterization, 258
 Pareto front, 62, 84, 86, 88, 90
 Partial differential equations, 23, 34
 Partially ordered set, 41
 Peaks, 125, 128, 143, 149–151, 245
 Pearson's correlation coefficient, 150
 Performance vs. cost curve, 62, 67, 68, 83, 84, 87, 90
 Permutation flow shop problem (PFSP), 164, 165, 177
 Permutations, 165–167, 169, 174, 305
 Piece-wise constant (PWC), 79
 Piece-wise linear (PWL), 80
 Planning, 3–18, 181
 Point-to-point movements, 140, 153
 Polynomial chaos (PC), 74–77
 Polynomial chaos expansion (PCE), 76
 Power, 38, 89, 97, 98, 120, 125, 130, 136, 143, 146, 149
 consumption, 128, 130, 131, 135–159
 saving modes, 137–139, 142, 153, 156, 158
 Precedence constraints, 4, 5, 7, 11
 Preemption of activities, 11
 Prefilter, 53–56
 Probability density function (PDF), 65, 67, 69, 70
 Probability distribution, 30–32, 74
 Process analysis, 129, 278, 280, 284
 Process chain, 125, 233, 322
 Process dynamics, 276–279

Processing times, 24, 26, 55, 165, 166, 174, 178, 186
 Process quality, 107, 287
 Production lines, 35, 135, 136, 163, 164, 183
 Production plans, 3, 4
 Production rate, 22, 34, 57, 59, 183
 Programmable logic controllers (PLCs), 118
 Project scheduling, 4, 11
 Pull logic, 63
 Punch-bending machine, 275, 280

Q

Quadratic objective function, 16
 Quadratic programming (QP), 67, 68, 80, 196
 Quality control, 276

R

Rapid prototyping, 271
 Rare events, 67, 77
 Ray tracing, 240
 Realistic robot simulation (RRS), 136, 148
 Recurrence period, 77
 Relaxation, 8, 11, 14–15, 156, 180, 190, 216, 217, 224, 225, 227, 233, 237, 240
 Relaxed problem, 177, 216, 217, 219
 Renewable resources, 5, 6, 10, 12
 Reorder point (ROP), 63, 69–75, 77, 82, 83, 90
 Reorder time (ROT), 63, 70
 Repetition flow shop problem (R-PFSP), 164, 174, 176
 Residuated mapping, 46
 Residuation theory, 46
 Resource(s), 4–6, 10–12, 15, 16, 39, 67, 91, 94, 208
 consumption, 5
 leveling, 11–17
 Response surface method (RSM), 281
 Rheology, 240, 249
 Robot(s), 106, 118, 124, 125, 135–143, 148, 149, 155–159, 183–189, 193, 194, 197–201, 205–207, 219, 220, 225, 226, 234, 249
 cell, 234
 configuration, 144, 153
 Robot controller simulation (RCS), 136, 148
 Robotic cell, 136–138, 143, 147, 156–159
 Robotic lines, 135–159
 Robotic operations, 136, 137, 139, 143, 147–152

S

Safety stock, 63, 69, 70, 72–74
 Scale, 22, 264, 270
 Scheduled tours, 186–188, 190, 197
 Schedules, 3, 11, 141, 142, 155, 163, 166, 177, 186
 Scheduling problem, 3–18, 163, 206, 207, 219, 225, 227
 Sealing laydown simulation, 232, 240–241
 Search tree, 8, 15–17, 216
 Segmentation, 138, 257, 266, 267
 Self-correcting controller, 284–287
 Sensors, 96, 118, 131, 136, 273–279, 284, 285
 Separation procedures, 8
 Sequencing, 71, 128, 184, 185, 187–196
 Sequential quadratic programming (SQP), 196
 Service level, 68, 82, 90
 Shape optimization, 323, 326, 328–331, 335, 337
 Shaping, 234, 254, 271
 Shaving, 15
 Shortest path, 187
 Simplex method, 9, 87
 Simulation of assembly, 314
 Small scale, 23, 324
 Smoothed particle hydrodynamics (SPH), 233
 Spin pack, 330, 338, 339
 Spline curve, 262, 263, 265, 268–270
 Spray painting, 232, 233, 235, 239, 241–247
 Spray simulation, 239–240
 Stamping and forming machine, 279
 Standard deviations, 73, 150
 Static consumption, 143
 Steady–state, 22, 28
 Stochastic problem, 69–77, 90
 Stochastic simulation, 22
 Stochastic variables, 70, 74–77
 Stock level, 10, 63, 69, 70, 82, 83
 Stock-out, 63, 68–70, 82
 Stokes equations, 323
 Storage, 23, 27, 61, 98, 125, 183
 Structural models, 290
 Subgradient algorithm, 157
 Subgradient method, 15, 156
 Supply lead time, 63, 65, 67, 70, 71, 73
 Surface segmentation, 257
 Surrogate model, 72, 77
 Synchronizations, 38, 40, 46–53, 59, 142, 155
 System engineering, 93–112
 System integration, 103, 105, 279

T

Tabu list, 169, 171, 172
 Tabu search (TS), 169–172, 178, 180, 181
 Task assignment, 184
 Tasks, 11, 12, 15–17, 163, 184, 185, 187, 201
 Throughput, 53, 57–59, 109, 135, 138
 Timed event graphs (TEGs), 38, 39
 Tool center point (TCP), 243
 Torque, 146, 147, 281, 282
 Total processing time, 166
 Tour, 186, 187, 189–192, 200, 201
 Trajectory, 125, 131, 137–140, 146–148, 155, 158, 184, 185, 188, 193, 197, 206, 208, 225
 Transfer function matrix, 52–59
 Transport equation, 25
 Travel times, 185–188, 191–193, 206
 Truck cab, 233, 235, 241–246, 248
 Tubes, 289, 322
 Turbulent air flow, 322
 Two-phase flow, 237

U

Uncertainties, 55, 72, 74, 159, 273
 Uniform distribution, 71, 178
 Unit cost, 83
 Upper bounds, 15, 53, 86, 197, 200
 Upwind-scheme, 26

V

Validation, 103, 106, 107, 111, 118, 122, 126, 127, 131, 232, 237, 241, 244, 245, 247, 248, 291, 316, 317
 Variance propagation, 68, 73, 84
 Virtual assembly, 289
 Virtual commissioning (VC), 110, 122, 128, 129
 Virtual control, 221, 222, 227
 Virtual engineering, 120–122, 127–131
 Virtual paint software, 247
 Virtual validation, 120–122, 126–128, 131
 V-Model, 102, 103, 278
 Volume of fluid (VOF) method, 233

W

Wagner family, 173
 Wall shear stress, 322–326, 328, 330–333, 335, 336, 338
 Warehouses, 61
 Wave, 28
 Wear phenomena, 280
 Weighted average, 62, 85, 89
 Welding, 98, 101–103, 105, 106, 108–110, 132, 136, 148, 149, 152, 153, 157, 158, 184, 205
 Welding cell, 103, 106, 109, 148, 157, 183–201
 Wiring harness, 290
 Work in progress (WIP), 22, 25, 27, 131