Hans Georg Bock
Hoang Xuan Phu
Rolf Rannacher
Johannes P. Schlöder
Editors

Modeling, Simulation and Optimization of Complex Processes HPSC 2012

Springer

Modeling, Simulation and Optimization of Complex Processes - HPSC 2012

Hans Georg Bock • Hoang Xuan Phu •
Rolf Rannacher • Johannes P. Schlöder
Editors

Modeling, Simulation and Optimization of Complex Processes - HPSC 2012

Proceedings of the Fifth International Conference on High Performance Scientific Computing, March 5-9, 2012, Hanoi, Vietnam

Springer

Editors
Hans Georg Bock
Rolf Rannacher
Johannes P. Schlöder
Interdisciplinary Center
for Scientific Computing (IWR)
University of Heidelberg
Heidelberg
Germany

Hoang Xuan Phu
Vietnam Academy of Science and
Technology (VAST)
Hanoi
Vietnam

ISBN 978-3-319-09062-7 ISBN 978-3-319-09063-4 (eBook)
DOI 10.1007/978-3-319-09063-4
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014950820

Mathematics Subject Classification (2010): 34B15, 35Q92, 49K15, 49J15, 49M30, 65K05, 65L05, 68W10, 70E60, 93B30

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

High performance scientific computing is an interdisciplinary area that combines many fields such as mathematics and computer science as well as scientific and engineering applications. It is an enabling technology for both competitiveness in industrialized countries and for speeding up development in emerging countries. High performance scientific computing develops methods for modeling, computer aided simulation and optimisation of complex systems and processes. In practical applications in industry and commerce, science and engineering, it helps to conserve resources, to avoid pollution, to reduce risks and costs, to improve product quality, to shorten development times or simply to operate systems better. Topical aspects of scientific computing have been presented and discussed at the Fifth International Conference on High Performance Scientific Computing that took place in Hanoi on March 5–9, 2012. The conference has been organized by the Institute of Mathematics of the Vietnam Academy of Science and Technology (VAST), the Interdisciplinary Center for Scientific Computing (IWR) of the University of Heidelberg, Ho Chi Minh City University of Technology, and the Vietnam Institute for Advanced Study in Mathematics.

More than 270 participants from countries all over the world attended the conference. The scientific program consisted of in total more than 190 talks, a big part of them presented in 19 mini-symposia. Eight talks were invited plenary lectures given by Frank Allgöwer (Stuttgart), Ralf Borndörfer (Berlin), Ingrid Daubechies (Durham), Mats Gyllenberg (Helsinki), Karl Kunisch (Graz), Volker Schulz (Trier) and Christoph Schwab (Zurich).

Topics included mathematical modeling, numerical simulation, methods for optimization and control, parallel computing, software development, applications of scientific computing in physics, mechanics and biomechanics, material science, hydrology, chemistry, biology, biotechnology, medicine, sports, psychology, transport, logistics, communication networks, scheduling, industry, business and finance.

This proceedings volume contains 21 carefully selected contributions referring to lectures presented at the conference. We would like to thank all authors and the referees.

Special thanks go to the sponsors whose support significantly contributed to the success of the conference:

+ Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences
+ Interdisciplinary Center for Scientific Computing (IWR), Heidelberg
+ Daimler and Benz Foundation, Ladenburg
+ The International Council for Industrial and Applied Mathematics (ICIAM)
+ Berlin Mathematical School
+ Berlin-Brandenburg Academy of Sciences and Humanities
+ The Abdus Salam International Centre for Theoretical Physics, Trieste
+ Vietnam Academy of Science and Technology (VAST)
+ Institute of Mathematics, VAST
+ Faculty of Computer Science and Engineering, HCMC University of Technology

Heidelberg, Germany
May 2014

Hans Georg Bock
Hoang Xuan Phu
Rolf Rannacher
Johannes P. Schlöder

Contents

A Non-causal Inverse Model for Source Signal Recovery in Large-Domain Wave Propagation

Hunter M. Brown, Minh Q. Phan, and Stephen A. Ketcham

Abstract A non-causal inverse model for source signal recovery is formulated. The inverse model is derived from a causal forward model. For a dynamical system where the causal inverse is unstable, a non-causal inverse model can be used instead. Application of the non-causal inverse technique on a High Performance Computing (HPC) acoustic propagation model of an office and laboratory campus in Hanover, New Hampshire, USA is presented.

1 Introduction

In the area of short-duration large-domain acoustic and seismic signal propagation, highly accurate reduced-order models represent an enabling technology, [1–4]. Such models are derived from HPC-derived data, and can be used for rapid prediction of the dynamic responses without resorting to the time-consuming HPC simulation. Significant savings in computational resources and time can be achieved by this strategy, reducing what normally takes hours on a HPC supercomputer to minutes on a laptop. Current research efforts are being made to extend the use of reduced-order models beyond output prediction. These efforts include the problems of source signal recovery and source localization. By taking advantage of the knowledge of the dynamics of the environment represented by these reduced-order models, it is possible to address the source signal recovery and localization problems in a highly complex multi-path environment with non-line-of-sight sensors.

A source signal can be recovered by the use of inverse models. The original dynamical system is in continuous time, but we often use a discrete-time model to represent it. If the forward continuous-time dynamical system is asymptotically stable, then the poles of the discrete-time transfer function lie inside the unit circle on the complex plane. The discrete-time zeros, on the other hand, may lie outside

H.M. Brown • M.Q. Phan (✉)
Thayer School of Engineering, Dartmouth College, Hanover, NH 03755, USA
e-mail: hunter.m.brown@dartmouth.edu; minh.q.phan@dartmouth.edu

S.A. Ketcham
Cold Regions Research and Engineering Laboratory, Hanover, NH 03755, USA
e-mail: stephen.a.ketcham@erdc.usace.army.mil

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_1

the unit circle. This situation occurs when the pole-zero excess of the original continuous-time model is three or more, and the sampling interval is sufficiently small, [5–7]. A causal inverse for such a system is unstable because the zeros outside the unit circle become the unstable poles of the inverse transfer function. When this happens another method is needed to recover the source signal. In this paper we formulate a non-causal inverse model for the inverse problem. In our applications, the source signal recovery is performed off-line, hence the use of such a non-causal inverse model is acceptable.

First, we derive a causal inverse model in state-space form. The state-space representation is chosen for the inverse models because the HPC-derived models that are being used in our research are superstable models which are in state-space form, [8]. Next, the causal inverse derivation is extended to produce a non-causal inverse model that can be used for source signal recovery. Finally, numerical results using an HPC-derived model of an office and laboratory campus are provided to illustrate the validity of the developed non-causal inverse technique.

2 Mathematical Formulation

We now derive the inverse models in state-space forms. A causal inverse model is derived first, followed by its extension to a non-causal inverse model.

2.1 A Causal Inverse Model

Consider a forward system with at least as many independent outputs q as the number of independent inputs r,

$$x(k+1) = Ax(k) + Bu(k) \tag{1}$$

$$y(k) = Cx(k) + Du(k) \tag{2}$$

Suppose that the direct transmission term D is non-zero, and D has a left-inverse, denoted by D^+, such that $D^+D = I$. Then $u(k)$ can be solved from the measurement equation of the state-space model as

$$u(k) = D^+y(k) - D^+Cx(k) \tag{3}$$

Substituting (3) into the state equation (1) immediately produces the following causal inverse model in state-space form,

$$x(k+1) = A^*x(k) + B^*y(k) \tag{4}$$

$$u(k) = C^*x(k) + D^*y(k) \tag{5}$$

where the matrices that define the causal inverse model are:

$$A^* = A - BD^+C \qquad B^* = BD^+ \qquad C^* = -D^+C \qquad D^* = D^+ \tag{6}$$

The state-space model and its causal inverse model derived above assume the existence of a direct transmission term D which might not exist. If this is the case, by time-shifting the output measurements, an artificial D term can be created. In the acoustic signal propagation problem where the output sensors and the source are not collocated, there is a time delay before a signal at a source location reaches the i-th sensor location. This causes a number of Markov parameters to be zero or very small,

$$C_i B = C_i AB = \cdots = C_i A^{\alpha_i - 1} B \approx 0 \tag{7}$$

where C_i is the output influence matrix associated with the i-th output, and the time delay α_i can be different for each output location. Although these number of "zero" Markov parameters might be small relative to the duration of interest, it is still important to explicitly call out these time delays to avoid numerical ill-conditioning in the inverse solution. To remove these "zero" Markov parameters, the state-space model to use for the i-th output is

$$x(k+1) = Ax(k) + Bu(k) \tag{8}$$

$$y_i(k + \alpha_i + 1) = C_i A^{\alpha_i + 1} x(k) + C_i A^{\alpha_i} Bu(k) \tag{9}$$

By shifting the measurement forward by a sufficient number of time steps, we can create an artificial direct transmission term that relates $y_i(k + \alpha_i + 1)$ to $u(k)$. Outputs from different locations can be combined to produce a single measurement equation,

$$\begin{bmatrix} y_1(k+\alpha_1+1) \\ y_2(k+\alpha_2+1) \\ \vdots \\ y_q(k+\alpha_q+1) \end{bmatrix} = \begin{bmatrix} C_1 A^{\alpha_1+1} \\ C_2 A^{\alpha_2+1} \\ \vdots \\ C_q A^{\alpha_q+1} \end{bmatrix} x(k) + \begin{bmatrix} C_1 A^{\alpha_1} B \\ C_2 A^{\alpha_2} B \\ \vdots \\ C_q A^{\alpha_q} B \end{bmatrix} u(k) \tag{10}$$

As long as the matrix that multiplies $u(k)$ in (10) above has a left-inverse, then the procedure previously applied to (2) can be performed on the new measurement equation (10). Strictly speaking, the resultant inverse model will be non-causal, but this non-causality is not fundamental as it is entirely caused by inverting transfer functions with time delays. This inverse is numerically sensitive because the matrix that plays the role of a new direct transmission term, whose left-inverse needs to be found, contains small (but non-zero) values. This calls for a more genuine non-causal inverse model, which will be derived in the next section.

2.2 A Non-causal Inverse Model

Let us start by considering the measurement equation for a single output i. Using the simplified definitions,

$$z_i(k) = y_i(k + \alpha_i + 1), \quad \tilde{C}_i = C_i A^{\alpha_i + 1}, \quad \tilde{D}_i = C_i A^{\alpha_i} B \tag{11}$$

Equations (8) and (9) can be re-written as

$$x(k+1) = Ax(k) + Bu(k) \tag{12}$$

$$z_i(k) = \tilde{C}_i x(k) + \tilde{D}_i u(k) \tag{13}$$

Propagating (13) forward by $s-1$ time steps, and packaging the results produces

$$\mathbf{z}_i(k) = \tilde{O}_i x(k) + \tilde{T}_i \mathbf{u}(k) \tag{14}$$

where

$$\mathbf{z}_i(k) = \begin{bmatrix} z_i(k) \\ z_i(k+1) \\ \vdots \\ z_i(k+s-1) \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+s-1) \end{bmatrix} \tag{15}$$

$$\tilde{O}_i = \begin{bmatrix} \tilde{C}_i \\ \tilde{C}_i A \\ \vdots \\ \tilde{C}_i A^{s-1} \end{bmatrix}, \quad \tilde{T}_i = \begin{bmatrix} \tilde{D}_i & & & \\ \tilde{C}_i B & \tilde{D}_i & & \\ \vdots & \ddots & \ddots & \\ \tilde{C}_i A^{s-2} B & \cdots & \tilde{C}_i B & \tilde{D}_i \end{bmatrix} \tag{16}$$

Measurements from multiple locations can be combined. This action will improve the conditioning of the inverse problem. Mathematically, the improvement comes about because new non-redundant equations are added without increasing the number of unknowns in $\mathbf{u}(k)$. We can write Eq. (14) for each available sensor, $i = 1, 2, \ldots, q$, and the resultant equations can be combined into a single equation. Define

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{z}_1(k) \\ \mathbf{z}_2(k) \\ \vdots \\ \mathbf{z}_q(k) \end{bmatrix}, \quad \tilde{O} = \begin{bmatrix} \tilde{O}_1 \\ \tilde{O}_2 \\ \vdots \\ \tilde{O}_q \end{bmatrix}, \quad \tilde{T} = \begin{bmatrix} \tilde{T}_1 \\ \tilde{T}_2 \\ \vdots \\ \tilde{T}_q \end{bmatrix} \tag{17}$$

The counterpart of (14) for all available sensors is

$$\mathbf{z}(k) = \tilde{O}x(k) + \tilde{T}\mathbf{u}(k) \tag{18}$$

If there is a sufficient number of output sensors to make the inverse problem well-conditioned, $\mathbf{u}(k)$ can be solved from (18) as,

$$\mathbf{u}(k) = -\tilde{T}^{+}\tilde{O}x(k) + \tilde{T}^{+}\mathbf{z}(k) \tag{19}$$

Notice that the matrix $\tilde{T}$, whose left-inverse $\tilde{T}^{+}$ needs to be computed, involves not only $\tilde{D}_i$ but also the additional Markov parameters $\tilde{C}_i B$, $\tilde{C}_i AB$, ..., $\tilde{C}_i A^{s-2} B$ for all available sensors. The Moore-Penrose pseudo-inverse of $\tilde{T}$ computed via its singular value decomposition can be used for the left-inverse required in (19). The Markov parameters in (17) represent the propagation dynamics a number of steps after the source signal arrives at the sensor locations. To complete the derivation, we extract $u(k)$ from the first r rows of $\mathbf{u}(k)$,

$$u(k) = -\left[\tilde{T}^{+}\right]_r \tilde{O}x(k) + \left[\tilde{T}^{+}\right]_r \mathbf{z}(k) \tag{20}$$

where the notation $\left[\tilde{T}^{+}\right]_r$ is used to denote the first r rows of $\tilde{T}^{+}$. By including a sufficient number of additional Markov parameters beyond $\tilde{D}_i$, we have eliminated the single-term dependence on the expression of $u(k)$ in (13). Combining (20) with the state equation in (12) produces the following non-causal inverse model in state-space form,

$$x(k+1) = \bar{A}x(k) + \bar{B}\mathbf{z}(k) \tag{21}$$

$$u(k) = \bar{C}x(k) + \bar{D}\mathbf{z}(k) \tag{22}$$

where the matrices that define the non-causal inverse model are:

$$\bar{A} = A - B\left[\tilde{T}^{+}\right]_r \tilde{O} \tag{23}$$

$$\bar{B} = B\left[\tilde{T}^{+}\right]_r \tag{24}$$

$$\bar{C} = -\left[\tilde{T}^{+}\right]_r \tilde{O}\ , \qquad \bar{D} = \left[\tilde{T}^{+}\right]_r \tag{25}$$

In summary, the non-causal inverse model given in (21) and (22) has the following properties: (a) It can handle systems with no direct transmission term, (b) It allows for location-dependent time delays from source to sensors, (c) It does not depend on the inverse of a single term. The inverse model can be derived from a forward state-space model, including the state-space model in superstable form described below.

3 Superstable Model Representation

In the source signal propagation problem it has been found that the unit pulse response model is a very convenient model to develop. The unit pulse response model relates the output measurements to the input through a finite impulse response model,

$$y(k) = h_1 u(k-1) + h_2 u(k-2) + \cdots + h_p u(k-p) \tag{26}$$

where $h_i, i = 1, 2, \ldots, p$, are called the Markov parameters of the system,

$$h_1 = CB\,, \quad h_2 = CAB\,, \quad h_3 = CA^2B, \ldots \quad , \quad h_p = CA^{p-1}B \tag{27}$$

In an acoustic signal propagation problem, p is typically in the range of 512–1,024. A direct transmission term is usually not present. The Markov parameters, which define the coefficients of the pulse response model, can be estimated by the inverse Fast Fourier Transform (FFT) technique. The dynamic model is in unit pulse response form, yet the inverse models derived here are in state-space form. It is therefore necessary to convert the unit pulse response model to state-space form. This task can be accomplished by the Eigensystem Realization Algorithm (ERA), [9, 10]. However, for short duration processes, it is much more convenient to use an alternate representation which are the finite-time superstable models described in [8]. There are two forms of the superstable state-space representation:

The first form is suitable when the number of outputs is much fewer than the number of sources, $q \ll r$,

$$A_1 = \begin{bmatrix} 0 & I_{q\times q} & 0 & \cdots & 0 \\ 0 & 0 & I_{q\times q} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & I_{q\times q} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_{pq\times pq} \qquad B_1 = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_p \end{bmatrix}_{pq\times r} \tag{28}$$

$$C_1 = \begin{bmatrix} I_{q\times q} & 0 & 0 & \cdots & 0 \end{bmatrix}_{q\times pq} \tag{29}$$

The second form is suitable when the number of outputs is much larger than the number of sources, $q \gg r$,

$$A_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ I_{r\times r} & 0 & \cdots & 0 & 0 \\ 0 & I_{r\times r} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & I_{r\times r} & 0 \end{bmatrix}_{pr\times pr} \qquad B_2 = \begin{bmatrix} I_{r\times r} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{pq\times r} \tag{30}$$

$$C_2 = \left[h_1 \; h_2 \; h_3 \cdots h_p \right]_{q \times pr} \tag{31}$$

To model the propagation dynamics from a potentially large number of sources to a small number of sensors (e.g., a moving source where each point in space can be a potential source location), the first form is preferred because it results in a model of smaller state dimensions. On the other hand, to model the signal propagation from a small number of sources throughout a very large domain (where the number of output locations can be every point in the domain of interest), the second form is preferred.

4 Illustrative Examples

An HPC-derived acoustic model of an office and laboratory campus is used in this simulation. The model is derived by the following procedure. A 3D finite-difference time-domain (FDTD) computation is used to simulate the propagation of a sound source placed at the center of the campus model, Fig. 1. This simulation takes

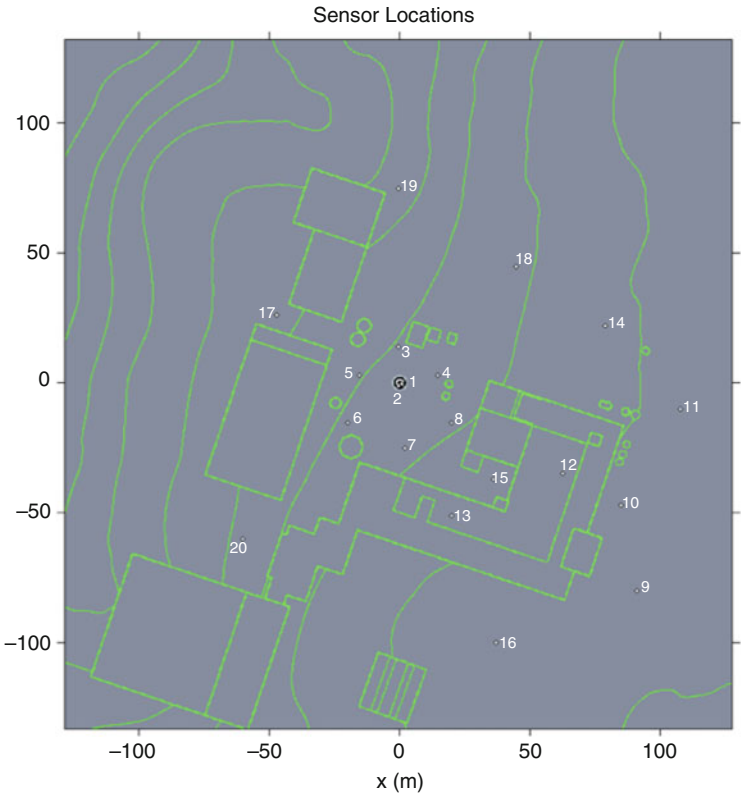

Fig. 1 A center source and various sensor locations located throughout the CRREL campus

approximately 15 h using 256 cores of a Cray XT3 with 2 GB of memory per core. The FDTD model has just under 2.7 billion cells, out of which 758 million output locations are selected to represent an output field 0.6-m above the ground surface and building roofs. From this simulation data, the inverse FFT method is used to compute 1,024 Markov parameters that describe the 1,024-sample long dynamics from the center source to the 758 million output locations. The sampling interval is selected to be 0.002115 s. The unit pulse response model, defined by these Markov parameters, is then converted into a state-space model via the second form of the superstable representation. Once the dynamic model is in state-space format, a non-causal inverse state-space model can be developed.

To test the validity of the inverse model, a test input is applied at the center source, and the outputs are recorded at a selected number of locations that are labeled 1 through 20 in Fig. 1. The inverse model is used to determine if the test input signal can be correctly recovered from the output at each of these locations. Typical results are reported here. Figure 2 shows the output at location 14, and the test input signal recovered from this output signal by the non-causal inverse model. Figure 3 shows the corresponding result if the output at location 10 is used instead. Because it is difficult to visually compare the recovered input signal to

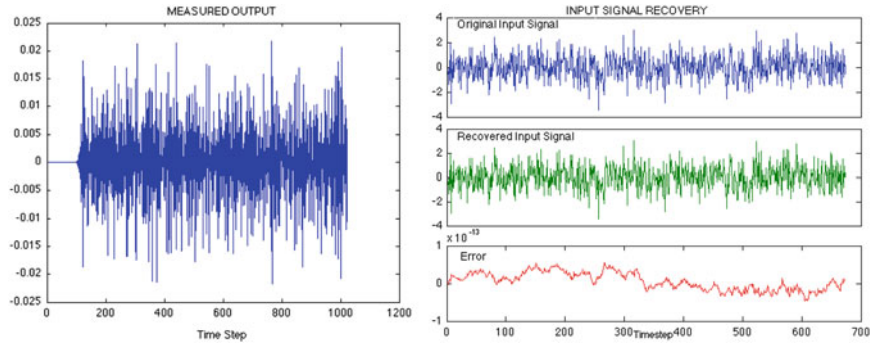

Fig. 2 Measured output at location 14 and recovered source signal by non-causal inverse model

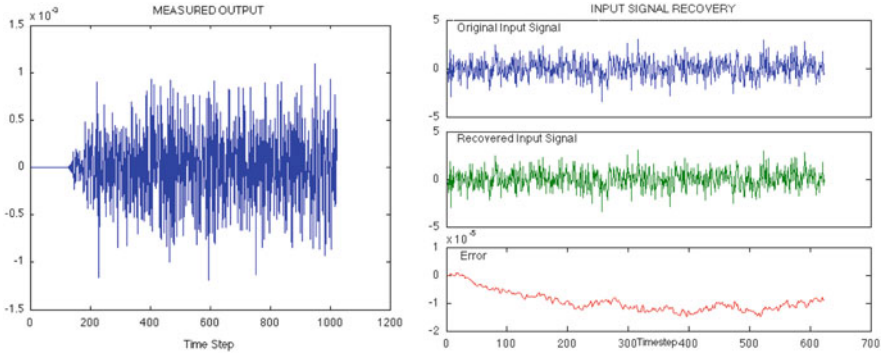

Fig. 3 Measured output at location 10 and recovered source signal by non-causal inverse model

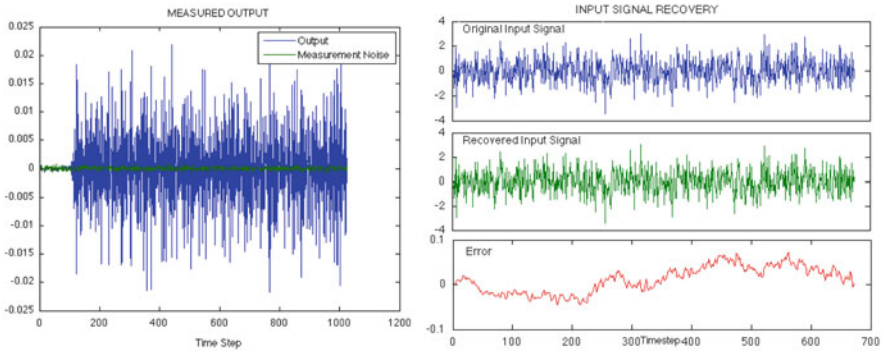

Fig. 4 Measured output at location 14 with noise and recovered source signal

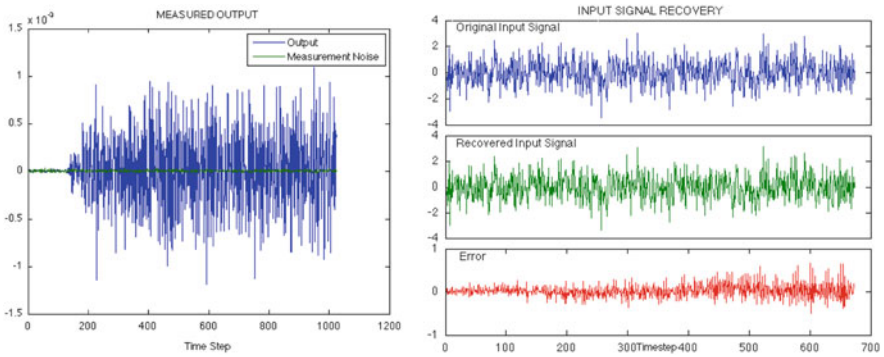

Fig. 5 Measured output at location 10 with noise and recovered source signal

the test input signal, the small difference between these two signals is also shown. Examination of these figures indicates that the source signal is recovered reasonably well. In a typical dynamical system, the high-frequency components of the input signal are attenuated by the dynamics of the model, thus one should not expect that these components can be recovered exactly. This explains the high-frequency error between the recovered and test input signals in these plots. In the pseudo-inverse computation of $\tilde{T}^+$ in (19) the "smaller" singular values are neglected. In so doing, the higher-frequency portion of the model corresponding to these singular values is not inverted. Counterparts of Figs. 2 and 3 when 5 % measurement noise is added to the output data are shown in Figs. 4 and 5 for the same two output locations. The results suggest a graceful degradation of the recovered source signal. Techniques to recover the source signal optimally in the presence of noise are treated in a subsequent paper.

5 Conclusions

A non-causal method for input signal recovery from output measurements has been derived. If a direct transmission term is present in the forward model, then the input at a certain time step immediately affects the output at the same time step. In this case, it is possible to derive an inverse model that is causal. Unfortunately, for most signal propagation problems, this is not the case due to the non-collocation of sources and sensors. Causal inverses also suffer from another fundamental issue. When the pole-zero excess of the original continuous-time dynamical system is 3 or more, the discrete-time transfer function contains at least one zero outside the unit circle if the sampling interval is sufficiently small. This fact causes the discrete-time causal inverse model to be unstable. A non-causal inverse can be used instead. Indeed, a non-causal inverse model can always be expected because the input at a certain time step influences the output at the subsequent time steps. It makes sense therefore to use measurements from subsequent time steps to recover the input at a previous time step. Simulation results confirm the validity of an inverse model derived from an HPC-model of an office and laboratory campus.

Acknowledgements This work is supported by a research grant from the US Army Corps of Engineers Cold Regions Research and Engineering Laboratory (CRREL) to Dartmouth College.

References

1. Anderson, T.S., Moran, M.L., Ketcham, S.A., Lacombe, J.: Tracked vehicle simulations and seismic wavefield synthesis in seismic sensor systems. Comput. Sci. Eng. **6**, 22–28 (2004)
2. Ketcham, S.A., Moran, M.L., Lacombe, J., Greenfield, R.J., Anderson, T.S.: Seismic source model for moving vehicles. IEEE Trans. Geosci. Remote Sens. **43**(2), 248–256 (2005)
3. Ketcham, S.A., Wilson, D.K., Cudney, H., Parker, M.: Spatial processing of urban acoustic wave fields from high-performance computations. In: DoD High Performance Computing Modernization Program Users Group Conference, Pittsburgh, pp. 289–295 (2007). ISBN:978-0-7695-3088-5, doi:10.1109/HPCMP–UGC.2007.68
4. Ketcham, S.A., Phan, M.Q., Cudney, H.H.: Reduced-order wave propagation modelling using the eigensystem realization algorithm. In: Bock, H.G., Phu, H.X., Rannacher, R., Schlöder, J.P. (eds.) Modeling, Simulation, and Optimization of Complex Process, pp. 183–193. Springer, Berlin/New York (2012)
5. Panomruttanarug, B., Longman, R.W.: Repetitive controller design using optimization in the frequency domain. In: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Providence (2004)
6. Longman, R.W.: On the theory and design of linear repetitive control systems. Eur. J. Control **16**(5), 447–496 (2010)
7. Longman, R.W., Peng, Y.-T., Kwon, T., Lus, H., Betti, R., Juang, J.-N.: Adaptive inverse iterative learning control. Adv. Astronaut. Sci. **114**, 115–134 (2003)
8. Phan, M.Q., Ketcham, S.A., Darling, R.S., Cudney, H.H.: Superstable models for short-duration large domain wave propagation. In: Bock, H.G., Phu, H.X., Rannacher, R.,

Schlöder, J.P. (eds.) Modeling, Simulation, and Optimization of Complex Process, pp. 257–269. Springer, Berlin/New York (2012)
9. Juang, J.-N., Pappa, R.S.: An eigensystem realization algorithm for modal parameter identification and model reduction. J. Guid. Control Dyn. **8**, 620–627 (1985)
10. Juang, J.-N.: Applied System Identification. Prentice-Hall, Upper Saddle River (2001)

Parallel-in-Space-and-Time Simulation of the Three-Dimensional, Unsteady Navier-Stokes Equations for Incompressible Flow

Roberto Croce, Daniel Ruprecht, and Rolf Krause

Abstract In this paper we combine the Parareal parallel-in-time method together with spatial parallelization and investigate this space-time parallel scheme by means of solving the three-dimensional incompressible Navier-Stokes equations. Parallelization of time stepping provides a new direction of parallelization and allows to employ additional cores to further speed up simulations after spatial parallelization has saturated. We report on numerical experiments performed on a Cray XE6, simulating a driven cavity flow with and without obstacles. Distributed memory parallelization is used in both space and time, featuring up to 2,048 cores in total. It is confirmed that the space-time-parallel method can provide speedup beyond the saturation of the spatial parallelization.

1 Introduction

Simulating three-dimensional flows by numerically solving the time-dependent Navier-Stokes equations leads to huge computational costs. In order to obtain a reasonable time-to-solution, massively parallel computer systems have to be utilized. This requires sufficient parallelism to be identifiable in the employed solution algorithms. Decomposition of the spatial computational domain is by now a standard technique and has proven to be extremely powerful. Nevertheless, for a fixed problem size, this approach can only push the time-to-solution down to some fixed threshold, below which the computation time for each subdomain becomes comparable to the communication time. While pure spatial parallelization can provide satisfactory runtime reduction, time-critical applications may require larger speedup and hence need additional directions of parallelism in the used numerical schemes.

One approach that has received increasing attention over recent years is parallelizing the time-stepping procedure typically used to solve time-dependent

R. Croce (✉) • D. Ruprecht • R. Krause
Institute of Computational Science, Via Giuseppe Buffi 13, CH-6906 Lugano, Switzerland
e-mail: roberto.croce@usi.ch; daniel.ruprecht@usi.ch; rolf.krause@usi.ch

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_2

problems. A popular algorithm for this is *Parareal*, introduced in [10] and comprehensively analyzed in [6]. Its performance has been investigated for a wide range of problems, see for example the references in [11, 12]. A first application to the 2D-Navier-Stokes equations, focussing on stability and accuracy without reporting runtimes, can be found in [5]. Some experiments with a combined Parareal/domain-decomposition parallelization for the two-dimensional Navier-Stokes equations have been conducted on up to 24 processors in [14, 15]. While they successfully established the general applicability of such a space-time parallel approach for the Navier-Stokes equations, the obtained speedups were ambiguous: Best speedups were achieved either with a pure time-parallel or a pure space-parallel approach, depending on the problem size.

In this paper we combine the Parareal-in-time and domain-decomposition-in-space techniques and investigate this space-time parallel scheme by means of solving a quasi-2D and a fully 3D driven cavity flow problem on a state-of-the-art HPC distributed memory architecture, using up to 2,048 cores. We demonstrate the capability of the approach to reduce time-to-solution below the saturation point of a pure spatial parallelization. Furthermore, we show that the addition of obstacles into the computational domain, leading to more turbulent flow, leads to slower convergence of Parareal. This is likely due to the reported stability issues for hyperbolic and convection-dominated problems, see [4, 12].

2 Physical Model and Its Discretization and Parallelization

The behavior of three-dimensional, incompressible Newtonian fluids is described by the incompressible Navier-Stokes equations. In dimensionless form the according momentum- and continuum equation read

$$\begin{aligned} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} &= \frac{1}{\mathrm{Re}} \Delta \mathbf{u} - \nabla p \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{1}$$

with $\mathbf{u} = (u, v, w)$ being the velocity field consisting of the Cartesian velocity-components, p being the pressure and Re the dimensionless Reynolds number.

The Navier-Stokes solver is based on the software-package NaSt3DGP [2, 8] and we further extended it by an MPI-based implementation of Parareal [10]. In NaSt3DGP, the unsteady 3D-Navier-Stokes equations are discretized via standard finite volume/finite differences using the Chorin-Temam [1, 13] projection method on a uniform Cartesian staggered mesh for robust pressure and velocity coupling. A first order forward Euler scheme is used for time discretization and as a building block for Parareal, see the description in Sect. 2.1.2. Second order central differences are used for the pressure gradient and diffusion. The convective terms are discretized with a second order TVD SMART [7] upwind scheme, which is basically a bounded QUICK [9] scheme. Furthermore, complex geometries are approximated

using a first order cell decomposition/enumeration technique, on which we can impose slip as well as no-slip boundary conditions. Finally, the Poisson equation for the pressure arising in the projection step is solved using a BiCGStab [16] iterative method.

2.1 Parallelization

Both the spatial as well as the temporal parallelization are implemented for distributed-memory machines using the MPI-library. The underlying algorithms are described in the following.

2.1.1 Parallelization in Space via Domain Decomposition

We uniformly decompose the discrete computational domain Ω_h into P subdomains by first computing all factorizations of P into three components, i.e. $P = P^x \cdot P^y \cdot P^z$, with $P^x, P^y, P^z \in \mathbb{N}$. Then we use our pre-computed factorizations of P as arguments for the following cost function C with respect to communication

$$C(P^x, P^y, P^z) = \frac{I}{P^x} \cdot \frac{J}{P^y} + \frac{J}{P^y} \cdot \frac{K}{P^z} + \frac{I}{P^x} \cdot \frac{K}{P^z} \tag{2}$$

with I, J and K as the total number of grid-cells in x-, y- and z-direction. Finally, we apply that factorization for the domain decomposition for which C is minimal, i.e. the space decomposition is generated in view of the overall surface area minimization of neighboring subdomains. Here, P is always identical to the number of processors $N_{\mathrm{p_{space}}}$, so that each processor handles one subdomain. Since the stencil is five grid-points large for the convective terms and three grid-points for the Poisson equation, each subdomain needs two ghost-cell rows for the velocities and one ghost-cell row for the pressure Poisson equation. Thus our domain decomposition method needs to communicate the velocities once at each time-step and the pressure once at each pressure Poisson iteration.

2.1.2 Parallelization in Time with Parareal

For a given time interval $[0, T]$, we introduce a coarse temporal mesh

$$0 = t_0 < t_1 < \ldots < t_{N_\mathrm{c}} = T \tag{3}$$

with a uniform time-step size $\Delta t = t_{i+1} - t_i$. Further, we introduce a fine time-step $\delta t < \Delta t$ and denote by N_f the total number of fine steps and by N_c the total number of coarse steps, that is

$$N_c \Delta t = N_f \delta t = T. \tag{4}$$

Also assume that the coarse time-step is a multiple of the fine, so that

$$\frac{\Delta t}{\delta t} =: N_r \in \mathbb{N}. \tag{5}$$

Parareal relies on the iterative use of two integration schemes: a fine propagator $\mathscr{F}_{\delta t}$ that is computationally expensive, and a coarse propagator $\mathscr{G}_{\Delta t}$ that is computationally cheap. We sketch the algorithm only very briefly here, for a more detailed description see for example [10].

Denote by $\mathscr{F}(\mathbf{y}, t_{n+1}, t_n)$, $\mathscr{G}(\mathbf{y}, t_{n+1}, t_n)$ the result of integrating from an initial value $\mathbf{y}$ at time t_n to a time t_{n+1}, using the fine or coarse scheme, respectively. Then, the basic iteration of Parareal reads

$$\mathbf{y}_{n+1}^{k+1} = \mathscr{G}_{\Delta t}(\mathbf{y}_n^{k+1}, t_{n+1}, t_n) + \mathscr{F}_{\delta t}(\mathbf{y}_n^k, t_{n+1}, t_n) - \mathscr{G}_{\Delta t}(\mathbf{y}_n^k, t_{n+1}, t_n) \tag{6}$$

with super-scripts referring to the iteration index and $\mathbf{y}_n$ corresponding to the approximation of the solution at time t_n. Iteration (6) converges to a solution

$$\mathbf{y}_{n+1} = \mathscr{F}_{\delta t}(\mathbf{y}_n, t_{n+1}, t_n), \tag{7}$$

that is a solution with the accuracy of the fine solver. Here, we always perform some prescribed number of iterations N_{it}. We use a forward Euler scheme for both $\mathscr{F}_{\delta t}$ and $\mathscr{G}_{\Delta t}$ and simply use a larger time-step for the coarse propagator. Experimenting with the combination of schemes of different and/or higher order is left for future work.

Once the values $\mathbf{y}_n^k$ in (6) from the previous iteration are known, the computationally expensive calculations of the values $\mathscr{F}_{\delta t}(\mathbf{y}_n^k, t_{n+1}, t_n)$ can be performed in parallel for multiple coarse intervals $[t_n, t_{n+1}]$. In the pure time-parallel case, the time-slices are distributed to $N_{p_{time}}$ cores assigned for the time-parallelization. Note that in the space-time parallel case, the time-slices are not handled by single cores but by multiple cores, each handling one subdomain at the specific time, see Sect. 2.1.3.

The theoretically obtainable speedup with Parareal is bounded by

$$s(N_p) \leq \frac{N_{p_{time}}}{N_{it}} \tag{8}$$

with $N_{p_{time}}$ denoting the number of processors in the temporal parallelization and N_{it} the number of iterations, see for example [11]. From (8) it follows that the maximum achievable parallel efficiency of the time parallelization is bounded by $1/N_{it}$. Parareal is hence considered as an additional direction of parallelization to be used when the spatial parallelization is saturated but a further reduction of time-to-solution is required or desirable. Some progress has recently been made deriving time-parallel schemes with less strict efficiency bounds [3, 11].

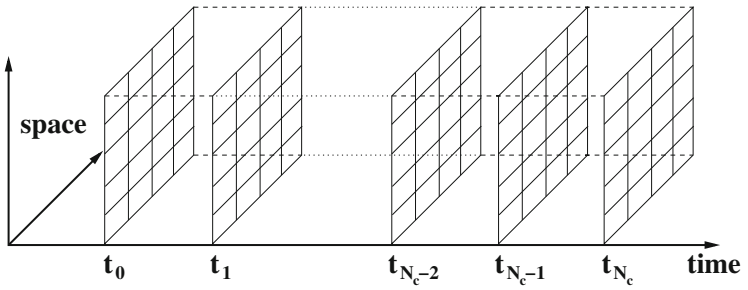

Fig. 1 Decomposition of the time interval $[0, T]$ into N_c time-slices. The spatial mesh at each point t_i is again decomposed into P subdomains, assigned to $N_{P_{space}}$ cores. Because the spatial parallelization does not need to communicate across time-slices, the cores from every spatial mesh are pooled into one MPI communicator. Also, in the time parallelization, only cores handling the same subdomain at different times have to communicate. Note that for readability the sketched spatial mesh is 2D, although the simulations use a fully 3D mesh

2.1.3 Combined Parallelization in Space and Time

In the combined space-time parallel approach as sketched in Fig. 1, each coarse time interval in (6) is assigned not to a single processor, but to one MPI communicator containing $N_{P_{space}}$ cores, each handling one subdomain of the corresponding time-slice. The total number of cores is hence

$$N_{P_{total}} = N_{P_{time}} \times N_{P_{space}}. \tag{9}$$

Note that the communication in time in (6) is local in the sense that each processor has only to communicate with the cores handling the same subdomain in adjacent time-slices. Also, the spatial parallelization is not communicating across time-slices, so that for the evaluation of $\mathscr{F}$ or $\mathscr{G}$ in (6), no communication between processors assigned to different points in time is required. We thus organize all available cores into two types of MPI communicators: (i) Spatial communicators collect all cores belonging to the solution at one fixed time-slice, but handling different subdomains. They correspond to the distributed representation of the solution at one fixed time-slice. There are $N_{P_{time}}$ spatial communicators and each contains $N_{P_{space}}$ cores. (ii) Time communicators collect all cores dealing with the same spatial subdomain, but at different time-slices. They are used to perform the iterative update in (6) of the local solution on a spatial subdomain. There are $N_{P_{space}}$ time communicators, each pooling $N_{P_{time}}$ cores. No special attention was paid to how different MPI tasks are assigned to cores. Because of the very different communication pattern of the space- and time-parallelization, this can presumably have a significant effect on the overall performance. More detailed investigation of the optimal placement of tasks is planned for future studies with the here presented code.

3 Numerical Examples

In the following, we investigate the performance of the space-time parallel approach for two numerical examples. The first is the classical driven-cavity problem in a quasi-2D setup. Figure 2 shows the flow in a xy-plane with $z = 0.05$ at times $t = 0.8$, $t = 8.0$ and $t = T = 80.0$. The second example is an extension where 49 obstacles (cubes) are inserted into the domain along the median plane, leading to fully 3D flow. Figure 3 sketches the obstacles and the flow at time $t = T = 24$. Both problems are posed on a 3D-domain with periodic boundary conditions in z-direction (note that x and z are the horizontal coordinates, while y is the vertical coordinate). Initially, velocity and pressure are set to zero. At the upper boundary, a tangential velocity $u_{\text{boundary}} = 1$ is prescribed, which generates a flow inside the domain as time progresses. No-slip boundary conditions are used at the bottom and in the two yz-boundary planes located at $x = 0$ and $x = 1$ as well as on the obstacles. The parameters for the two simulations are summarized in Table 1. To assess the temporal discretization error of $\mathscr{F}_{\delta t}$, the solution is compared to a reference solution computed with $\mathscr{F}_{\delta t/10}$, giving a maximum error of 1.2×10^{-5} for the full 3D flow with obstacles. That means that once the iteration of Parareal has reduced the maximum defect between the serial and parallel solution below this threshold, the time-parallel and time-serial solution are of comparable accuracy. We use this threshold also for the quasi-2D example, bearing in mind that the simpler structure of the flow in this case most likely renders the estimate too conservative. The code is run on a Cray XE6 at the Swiss National Supercomputing Centre, featuring a total of 1,496 nodes, each with two 16-core 2.1 GHz AMD Interlagos CPUs and 32 GB memory per node. Nodes are connected by a Gemini 3D torus interconnect and the theoretical peak performance is 402 TFlops.

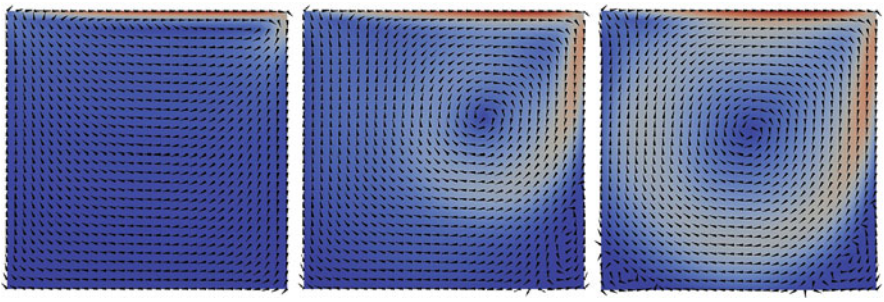

Fig. 2 Simulation 1: *Arrows* and grayscale plot for the range [0.0, 0.75] of the Euclidean norm of the quasi two-dimensional driven cavity flow field along the center plane at three points in time $t = 0.8$ and $t = 8.0$ and $t = T = 80.0$

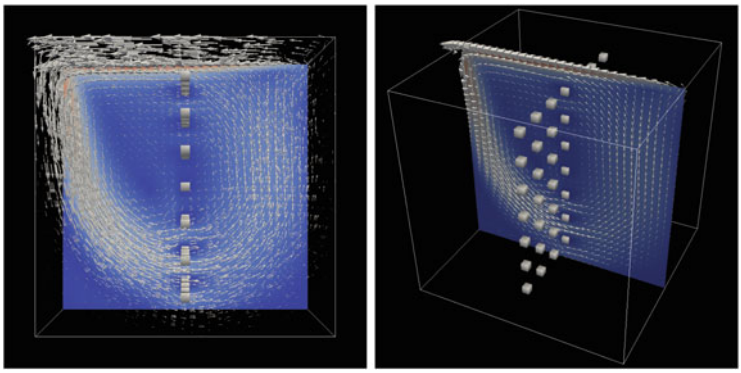

Fig. 3 Simulation 2: *Arrows* and grayscale plot for the range [0.0, 0.75] of the Euclidean norm of the fully three-dimensional driven cavity flow field with obstacles along the center plane at $t = T = 24.0$

Table 1 Simulation parameters for the quasi-2D driven cavity flow (Simulation 1) and the fully 3D driven cavity flow with obstacles (Simulation 2)

Sim. 1 :	Ω_h	$= [0,1] \times [0,1] \times [0,0.1]$	Sim. 2 :	Ω_h	$= [0,1] \times [0,1] \times [0,1]$
Sim. 1 :	$N_x \times N_y \times N_z$	$= 32 \times 32 \times 5$	Sim. 2 :	$N_x \times N_y \times N_z$	$= 32 \times 32 \times 32$
Sim. 1 :	T	$= 80$	Sim. 2 :	T	$= 24$
Both :	Δt	$= 0.01$	Both :	δt	$= 0.001$
Both :	Re	$= 1{,}000$	Both :	u_{boundary}	$= 1$
Sim. 1 :	N_{Pspace}	$= 1, 2, 4, 8$	Sim. 2 :	N_{Pspace}	$= 1, \ldots, 128$
Sim. 1 :	N_{Ptime}	$= 4, 8, 16$	Sim. 2 :	N_{Ptime}	$= 8, 16, 32$

3.1 Quasi-2D Driven Cavity Flow

Figure 4 shows the maximum difference between the time-parallel and the time-serial solution at the end of the simulation versus the number of iterations of Parareal. In all three cases, the error decreases exponentially with N_{it}. The threshold of 1.2×10^{-5} is reached after a single iteration, indicating that the performance of Parareal could probably be optimized by using a larger Δt. Figure 5 shows the total speedup provided by the time-serial scheme running $\mathscr{F}_{\delta t}$ with only space-parallelism (circles) as well as by the space-time parallel method for different values of $N_{p_{\text{time}}}$. All speedups are measured against the runtime of the time-serial solution run on a single core. The pure spatial parallelization reaches a maximum speedup of a little over 6 using 8 cores. For $N_{\text{it}} = 1$, the space-time parallel scheme reaches a speedup of 14 using 64 cores. This amounts to a speedup of roughly $14/6 \approx 2.33$ from Parareal alone. For $N_{\text{it}} = 2$ the speedup is down to 8, but still noticeably larger

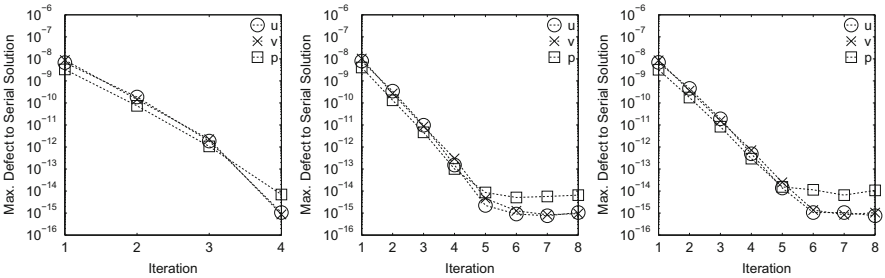

Fig. 4 Maximum difference to time-serial solution versus number of Parareal iterations for the Cartesian velocity (u, v) and pressure p of the quasi 2D driven cavity problem at time $t = 80.0$ for $N_{p_{\text{time}}} = 4$ (*left*), $N_{p_{\text{time}}} = 8$ (*middle*) and $N_{p_{\text{time}}} = 16$ (*right*)

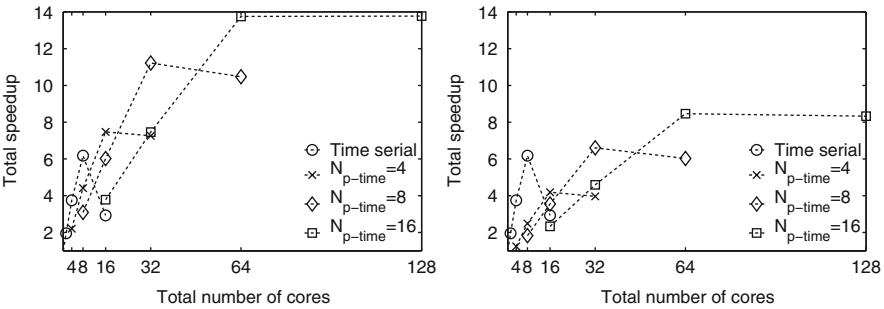

Fig. 5 Total speedup of the combined space-time parallelization for quasi 2D driven-cavity flow with $N_{\text{it}} = 1$ (*left*) and $N_{\text{it}} = 2$ (*right*) iterations

than the saturation point of the pure space-parallel method. Note that because of the limited efficiency of the time parallelization, the slopes of the space-time parallel scheme are lower for larger values of $N_{p_{\text{time}}}$.

3.2 Full 3D Driven Cavity Flow with Obstacles

Depending on the number of Parareal iterations for three different values of $N_{p_{\text{time}}}$ Fig. 6 shows the maximum difference between the time-parallel and the time-serial solution in terms of the 3D-Cartesian velocity (u, v, w) and pressure p. In general, as in the quasi-2D case, the error decays exponentially with the number of iterations, but now, particularly pronounced for $N_{p_{\text{time}}} = 32$, a small number of iterations has to be performed without large effect before the error starts to decrease. This is likely due to the increased turbulence caused by the obstacles, as it is known that Parareal exhibits instabilities for advection dominated problems or hyperbolic

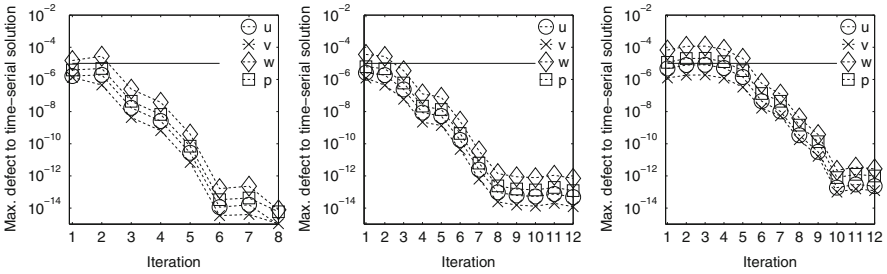

Fig. 6 Maximum difference to time-serial solution at end of simulation versus number of Parareal iterations for the 3D driven cavity flow with obstacles for $N_{p_{\text{time}}} = 8$ (*left*), $N_{p_{\text{time}}} = 16$ (*middle*), $N_{p_{\text{time}}} = 32$ (*right*). The *horizontal line* indicates an error level of 10^{-5}

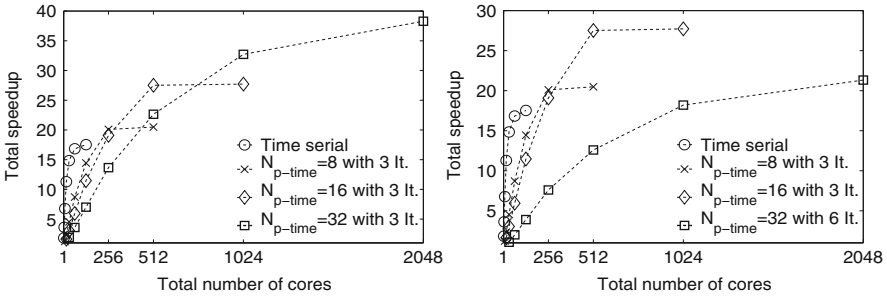

Fig. 7 Total speedup of the combined space-time parallelization for 3D cavity flow with obstacles for a fixed number of $N_{\text{it}} = 3$ iterations (*left*) and a number of iterations chosen to achieve a defect below 10^{-5} in all solution components (*right*). Note that the solutions in the left figure are not comparable in accuracy

problems [6, 12]. A more detailed analysis of the performance of Parareal for turbulent flow and larger Reynolds numbers is left for future work. Figure 7 shows the total speedup measured against the runtime of the solution running $\mathscr{F}_{\delta t}$ serially with $N_{\text{p}_{\text{space}}} = 1$. The time-serial-line (circles) shows the speedup for a pure spatial parallelization, which scales to $N_{\text{p}_{\text{space}}} = 16$ cores and then saturates at a speedup of about 18. Adding time-parallelism can significantly increase the total speedup, to about 20 for $N_{p_{\text{time}}} = 4$, about 27 for $N_{p_{\text{time}}} = 8$ and to almost 40 for $N_{p_{\text{time}}} = 16$ for a fixed number of $N_{\text{it}} = 3$ iterations (left figure). However, as can be seen from Fig. 6, the solution with $N_{\text{p}} = 32$ is significantly less accurate. The right figure shows the total speedup for a number of iterations adjusted so that the defect of Parareal in all cases is below 10^{-5} in all solution components (cf. Fig. 6). This illustrates that there is a sweet-spot in the number of concurrently treated time-slices: At some point the potential increase in speedup is offset by the additional iterations required. In the presented example, the solution with $N_{\text{p}} = 16$ is clearly more efficient than the one with $N_{\text{p}} = 32$.

4 Conclusions

A space-time parallel method, coupling Parareal with spatial domain decomposition, is presented and used to solve the three-dimensional, time-dependent, incompressible Navier-Stokes equations. Two setups are analyzed: A quasi-2D driven cavity example and an extended setup, where obstacles inside the domain lead to a fully 3D driven flow. The convergence of Parareal is investigated and speedups of the space-time parallel approach are compared to speedups from a pure space-parallel scheme. It is found that Parareal converges very rapidly for the quasi-2D case. It also converges in the 3D case, although for larger numbers of Parareal time-slices, convergence starts to stagnate for the first few iterations, likely because of the known stability issues of Parareal for advection dominated flows. Results are reported from runs on up to 128 nodes with a total of 2,048 cores on a Cray XE6, illustrating the feasibility of the approach for state-of-the-art HPC systems. The results clearly demonstrate the potential of time-parallelism as an additional direction of parallelization to provide additional speedup after a pure spatial parallelization reaches saturation. While the limited parallel efficiency of Parareal in its current form is a drawback, we expect the scalability properties of Parareal to direct future research towards modified schemes with relaxed efficiency bounds.

Acknowledgements This research is funded by the Swiss "High Performance and High Productivity Computing" initiative HP2C. Computational resources were provided by the Swiss National Supercomputing Centre CSCS.

References

1. Chorin, A.J.: Numerical solution of the Navier Stokes equations. Math. Comput. **22**(104), 745–762 (1968)
2. Croce, R., Engel, M., Griebel, M., Klitz, M.: NaSt3DGP – a Parallel 3D Flow Solver. http://wissrech.ins.uni-bonn.de/research/projects/NaSt3DGP/index.htm
3. Emmett, M., Minion, M.L.: Toward an efficient parallel in time method for partial differential equations. Commun. Appl. Math. Comput. Sci. **7**, 105–132 (2012)
4. Farhat, C., Chandesris, M.: Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. Int. J. Numer. Methods Eng. **58**, 1397–1434 (2005)
5. Fischer, P.F., Hecht, F., Maday, Y.: A parareal in time semi-implicit approximation of the Navier-Stokes equations. In: Kornhuber, R., et al. (eds.) Domain Decomposition Methods in Science and Engineering. LNCSE, vol. 40, pp. 433–440. Springer, Berlin (2005)
6. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. SIAM J. Sci. Comput. **29**(2), 556–578 (2007)
7. Gaskell, P., Lau, A.: Curvature-compensated convective transport: SMART a new boundedness-preserving transport algorithm. Int. J. Numer. Methods Fluids **8**, 617–641 (1988)
8. Griebel, M., Dornseifer, T., Neunhoeffer, T.: Numerical Simulation in Fluid Dynamics, a Practical Introduction. SIAM, Philadelphia (1998)

9. Leonard, B.: A stable and accurate convective modelling procedure based on quadratic upstream interpolation. Comput. Methods Appl. Mech. Eng. **19**, 59–98 (1979)
10. Lions, J.L., Maday, Y., Turinici, G.: A "parareal" in time discretization of PDE's. C. R. Acad. Sci. – Ser. I – Math. **332**, 661–668 (2001)
11. Minion, M.L.: A hybrid parareal spectral deferred corrections method. Commun. Appl. Math. Comput. Sci. **5**(2), 265–301 (2010)
12. Ruprecht, D., Krause, R.: Explicit parallel-in-time integration of a linear acoustic-advection system. Comput. Fluids **59**, 72–83 (2012)
13. Temam, R.: Sur l'approximation de la solution des equations de Navier-Stokes par la méthode des pas fractionnaires II. Arch. Ration. Mech. Anal. **33**, 377–385 (1969)
14. Trindade, J.M.F., Pereira, J.C.F.: Parallel-in-time simulation of the unsteady Navier-Stokes equations for incompressible flow. Int. J. Numer. Methods. Fluids **45**, 1123–1136 (2004)
15. Trindade, J.M.F., Pereira, J.C.F.: Parallel-in-time simulation of two-dimensional, unsteady, incompressible laminar flows. Numer. Heat Trans., Part B **50**, 25–40 (2006)
16. van der Vorst, H.: Bi-CGStab: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **13**, 631 (1992)

Mathematical Modeling of Emotional Body Language During Human Walking

Martin L. Felis, Katja Mombaur, and Alain Berthoz

Abstract The study of emotional facial expressions and of emotional body language is currently receiving a lot of attention in various research areas. In this research, we study implicit bodily expression of emotions during standard motions such as walking forwards. An underlying assumption of our work is that all human motion is optimal in some sense and that different emotions induce different objective functions, which result in different deformations of normal motion. We created a 2-D rigid-body model of a human for which we use its dynamics simulation in an optimal control context. This approach allows us to obtain different styles of motion by using different objective criteria. We present the model, the optimal control problem formulation and the direct multiple-shooting method that efficiently solves this problem. The results of this work form the foundation for further analysis of emotional motions using inverse optimal control methods.

1 Introduction

When actors portray characters on stage they use much more than only their voice and words to convey the feelings of their alter ego. It is an interplay of verbal and non-verbal expression that creates the impression of lively personalities instead of blank inanimate robots. But also when removing voice and facial expressions our perception is still able to read the emotional state.

Already Darwin [2] asked whether facial and bodily expressions of emotions are inherent or whether they are learned during lifetime. He also proposed that expressions which can be found in both man and animals, such as the *sneer*, are due to a common genetic ancestors. More recently Ekman [3] created the Facial

M.L. Felis (✉) • K. Mombaur
Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University,
INF 368, D-69120 Heidelberg, Germany
e-mail: martin.felis@iwr.uni-heidelberg.de; katja.mombaur@iwr.uni-heidelberg.de

A. Berthoz
Laboratoire de Physiologie de la Perception et de l'Action (LPPA), 11, place Marcelin Berthelot,
75231 Paris Cedex 05, France
e-mail: alain.berthoz@college-de-france.fr

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_3

Action Coding System (FACS), which is able to categorize and encode almost every anatomically possible facial expression.

But emotions are not only expressed in faces but also using body posture and movement. Already simple visualizations such as point light displays of a walking person suffices for us to be able to recognize emotions [1]. This is demonstrated by the computer program "BML Walker" described in [15]. It allows specification of physical properties such as sex, weight, and also emotional aspects nervous-relaxed and happy-sad via sliders. A point light display of a motion that fits to the chosen parameter is generated on-the-fly. Generation of emotional motions therefore seems to be possible. But how can we use this to gain insight into emotions?

In [13] kinematic analysis of emotional walking motions was done. They used motion capture data and used a non-linear blind-source separation method. It efficiently finds only a few source components that approximate high-dimensional emotional walking. This allows one to simulate different emotional styles on a kinematic skeleton. Furthermore these spatio-temporal primitives are specific for different emotions and indicate emotion-specific primitives in the human motor system.

Only little research of emotional body language has so far been done by using (rigid-body) dynamics. In [11] a learning method was used to create stylized motions which can be used as a physics-based animation system. It allows one to learn a parameter vector θ from motion-capture data, which can then be applied to different motions. The vector θ contains information such as elasticity parameters for shoe contact, muscles and tendons and also preferences for certain muscles. The method was used to learn emotional styles, but focus of this research is on generation of stylized animations and not on emotional body language itself.

With our research, we want to explore emotional body language on the level of rigid-body dynamics. One of the assumptions of our work is that human motions are optimal in some sense. Optimization has already been successfully used to create human like motions such as running [14] or platform diving [8]. We therefore use a model-based approach using a rigid-body model and optimal control methods. This allows us to look at emotional body language on a level of forces and torques that act in or on the body. One of the key questions we have is, whether it is possible to relate certain emotions to specific goals in the form of objective functions.

In this paper we want to show our preliminary results that will form the base for future work for that we want to use motion capture data and inverse optimal control methods. For this we want to follow the method presented in [12] which uses a bi-level approach to identify the objective function that is described as a linear combination of base criteria. Using this, the identification of the objective function is then formulated as a least-squares parameter estimation of the linear coefficients that tries to fit the optimally controlled motion to that of a recorded emotional motion which requires the solution of an optimal control problem in the lower loop.

This paper focuses on this lower level optimal control problem along with a selection of base criteria functions and their solutions.

2 Rigid-Body Model for Human Walking

The multi-body model we created consists of 12 bodies, including legs (which are split-up in thigh and shank), feet, upper- and lower-arms, head, and trunk. It is a 2-D model that moves in the sagittal plane and has 14 degrees of freedom (3 for the pelvis, 2 for each hip, knee, ankle, shoulder, and elbow and 1 for the head). During contact with the ground the model is subject to algebraic constraints. We use multiple distinct *Phases* for each set of contact constraints. Collisions of the foot with the ground are modeled as inelastic impacts.

The state of the body is described using generalized coordinates which are denoted by q. The variables $\dot{q}, \ddot{q}$, and τ are the generalized velocities, accelerations and joint torques. Our model is an *underactuated* system which means there are fewer actuators than degrees of freedom. In our case, the 3 degrees of freedom of the pelvis are not actuated, which results in $\tau_0(t) = \tau_1(t) = \tau_2(t) = 0$ at all times. All other degrees of freedom and also their corresponding joints are called *actuated*.

We use data from the biomechanics literature [10] for both the kinematic and inertial quantities so that our model matches that of an adult male person.

The damping effects of muscles and tendons are modeled using linear damper elements at the actuated joints. We use an identical damper constant for corresponding left and right joints which results in 6 damper constants. These constants are added as free parameters to the optimization problem.

An overview of the model is given in Fig. 1a and the possible types of contact for a single foot is shown in Fig. 1b.

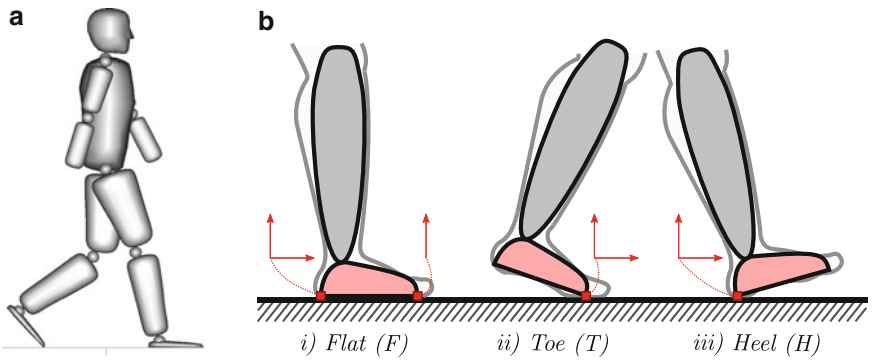

Fig. 1 Overview of the model structure and the contact constraints. (**a**) Overview of our model topology. The model has 12 segments and 14 degrees of freedom. (**b**) Contact model for ground foot. The *square markers* and *coordinated arrows* describe the contact points and contact normals along which the contact point translation is constrained. The constraint forces are applied on the foot segment

2.1 Phases

We are interested in a periodic gait, which means that the pose at the end of a double step is identical to that at the beginning. A symmetric periodic gait is a further simplification which assumes that a step with the right leg is the step of the left leg mirrored on the sagittal plane. This allows us to restrict ourselves to the optimization of a single step and posing appropriate periodicity constraints.

For each type of contact with the ground we have different constraint sets and therefore different ODEs for the model dynamics. Figure 2 shows the single step we optimize, the order of the individual model phases, and when collisions occur. In total we have four contact phases: Right Flat, Right Toes, Right Toes Left Heel, Right Toes Left Flat and two collision events: at touchdown of Left Heel and touchdown of the Left Toes.

The actual durations of each of the phases are not prescribed. Instead each they are determined by the optimization.

2.2 Contact Modeling

As a contact model we use a single segment flat foot. Different numbers of constraints are imposed on the model, depending on the type of contact: three constraints when the foot is flat on the ground, two constraints when either heel or toes are in contact, or no constraints when the foot is swinging (Fig. 1b).

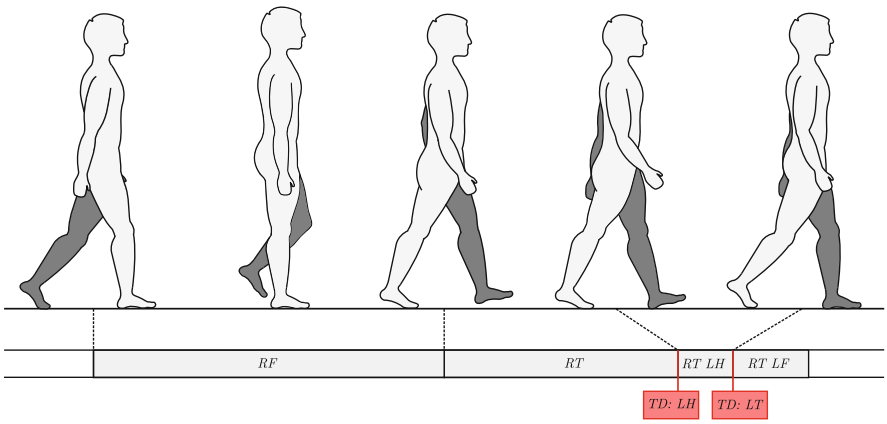

Fig. 2 Overview of the gait phases and the constraint sets *RF* Right Flat, *RT* Right Toes, *RT LH* Right Toes Left Heel, *RT LF* Right Toes Left Flat. Ground collisions occur at *TD LH* (touchdown Left Heel) and *TD LT* (touchdown Left Toes)

The constrained dynamics can be expressed as a differential algebraic equation with:

$$M(q)\ddot{q} = -C(q,\dot{q}) - G(q)^T\lambda + T\tau, \tag{1a}$$
$$g(q) = 0. \tag{1b}$$

where $M(q)$ is the joint-space inertia matrix, $C(q,\dot{q})$ the Coriolis and gravitational forces, and τ the joint actuation torques. The matrix T maps the applied torques onto the actuated joints. The matrix $G(q)$ is the contact Jacobian of the contact constraints $g(q)$ and λ are the contact forces. The constraint Jacobian $G(q)$ has full rank due to the choice of contact points and normals (Fig. 1b).

Using index reduction this is transformed into a linear system of the form:

$$\begin{pmatrix} M(q) & G^T(q) \\ G(q) & 0 \end{pmatrix} \begin{pmatrix} \ddot{q} \\ \lambda \end{pmatrix} = \begin{pmatrix} -C(q,\dot{q}) + T\tau \\ -\gamma(q,\dot{q}) \end{pmatrix} \tag{2}$$

where $\gamma(q,\dot{q})$ contains derivative terms of the algebraic contact equation (1b). The matrix on the left-hand side is regular and the system can therefore be solved to obtain $\ddot{q}$.

With the invariants $g_{pos}(t_0) = g(q(t_0)) = 0$ and $g_{vel}(t_0) = \frac{d}{dt}g(q(t_0)) = 0$, fulfilled at the beginning of the time horizon, the system (2) is mathematically equivalent to (1). When using this formulation one normally has to pay attention on drift of the algebraic constraint during numerical integration over larger time horizons. For our case the integration duration is usually less than a second so we do not need to account for this.

By setting $x(t) = (q(t), \dot{q}(t))^T$ and $u(t) = \tau(t)$ and due to the regularity of the matrix in (2) we can embed the contact dynamics (2) in an ordinary differential equation (ODE):

$$\dot{x} = f(t, x(t), u(t)). \tag{3}$$

The dimensions and equations for $G(q), \gamma(q,\dot{q})$, and λ change depending on the contact type which results in multiple distinct ODEs which each describe the dynamics for a specific set of constraints. The model equations for the different contact phases i are denoted by:

$$\dot{x} = f_i(t, x(t), u(t)). \tag{4}$$

This formulation allows us to describe the dynamics of the character as continuously differentiable functions for each set of constraints as required by our trajectory optimization method.

2.3 Ground Collision

We model the foot touchdown as an inelastic collision. Such a contact gain has two consequences: the set of constraints change, which results in a change of the model equation (4) and furthermore in a discontinuity of the generalized velocities $\dot{q}$.

If a contact gain is detected at time t_c we can compute the change of the velocities from right before the contact event $\dot{q}_c^-$ to after the collision $\dot{q}_c^+$ by solving

$$\begin{pmatrix} M(q) & G_{new}^T(q) \\ G_{new}(q) & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_c^+ \\ \Lambda \end{pmatrix} = \begin{pmatrix} M(q)\dot{q}_c^- \\ 0 \end{pmatrix}. \tag{5}$$

The matrix $G_{new}(q)$ is the constraint Jacobian for the new constraint set and Λ the impulse in cartesian coordinates that acts at the contact points of the new constraint set. Solving this equation ensures that the invariant $g_{vel}(t_c^+) = 0$ is fulfilled.

Analog to the model equations this equation can also be written as a *transition function*:

$$x(t_c^+) = h_i(x(t_c^-)) \tag{6}$$

for which i denotes the specific constraint set used for the constraint Jacobians.

On contact losses there is no discontinuity for the generalized velocities $\dot{q}$, therefore no special treatment is required. In this case the transition function (6) simplifies to $x(t^+) = x(t^-)$.

2.4 Dynamics Implementation

All modeling and computation is done using the RBDL – the Rigid Body Dynamics Library. It is a highly efficient C++ code that contains some essential rigid-body dynamics algorithms such as the Recursive Newton-Euler Algorithm (inverse dynamics), the Composite Rigid-Body Algorithm (computation of the joint-space inertia matrix), and the Articulated Body Algorithm (forward dynamics).

The library strongly builds on the concept of Spatial Algebra [4], which is a both concise and efficient notation for describing rigid-body kinematics and dynamics. Instead of expressing the motion of the individual bodies using two distinct sets of 3-D equations (one for translations and one for rotations) it combines the two equations in a uniform 6-D formulation. This results in less equations but also fewer lines of code.

RBDL uses by default the Eigen3 C++ template library [6] for linear algebra. It automatically makes use of high-performance SSE instructions that greatly speeds up evaluation of the 6-D expressions if supported by the CPU.

Computations for the joint-space inertia matrix $M(q)$ is done using the the Composite Rigid-Body Algorithm. The Recursive Newton-Euler Algorithm is used

to compute the coriolis and gravitational forces $C(q, \dot{q})$. The library also computes the contact Jacobians $G(q)$ and derivative terms $\gamma(q, \dot{q})$. RBDL then builds the linear system (2) and solves it using a Householder QR decomposition provided by the Eigen3 library.

The RBDL is freely available under the permissive *zlib*-License and can be obtained from [5].

3 Optimal Control Problem Formulation of a Human Gait

By using optimal control methods we can simultaneously optimize the motion $x(t) = [q(t), \dot{q}(t)]^T$, the controls $u(t) = \tau(t)$ and the parameters p, such as the spring damper constants and steplength and velocity. We have four model equations (*RF, RT, RT LH, RT LF*), which results in a four–phase optimal control problem.

3.1 General Formulation

The whole optimal control problem can be written as:

$$\min_{x(\cdot),u(\cdot),p,t_1,\ldots,t_4} \sum_{i=1}^{4} \int_{t_{i-1}}^{t_i} \Phi_{L_i}(t, x(t), u(t), p)dt + \sum_{i=1}^{4} \Phi_{M_i}(t_i, x(t_i)) \tag{7a}$$

subject to:

$$\dot{x}(t) = f_i(t, x(t), u(t), p), \quad t \in [t_{i-1}, t_i], \; i = 1, \ldots, 4, \tag{7b}$$
$$x(t_j^+) = h_j(x(t_j^-)), \quad j = 2, 3, \tag{7c}$$
$$g_i(t, x(t), u(t), p) \geq 0, \quad t \in [t_{i-1}, t_i], \; i = 1, \ldots, 4, \tag{7d}$$
$$r^{eq}(x(\hat{t}_0), \ldots, x(\hat{t}_k), p) = 0, \quad \hat{t}_0, \ldots \hat{t}_k \in [t_0, t_4], \tag{7e}$$
$$r^{ineq}(x(\hat{t}_0), \ldots, x(\hat{t}_k), p) \geq 0, \quad \hat{t}_0, \ldots \hat{t}_k \in [t_0, t_4]. \tag{7f}$$

The objective function (7a) used here is split up in two parts: the Lagrange terms Φ_{L_i} that are evaluated over the whole phase duration and the Mayer terms Φ_{M_i} that are evaluated at the end of each phase. The former can be used to e.g. minimize the torques applied whereas the latter can for example be used to minimize time. Equivalence of the two terms can be shown but their explicit use eases the formulation of certain optimization criteria. We used different objective criteria to achieve different walking styles:

- **Minimum Torques**: here we minimize the control effort formulated as $\Phi_{L_i} = ||Wu(t)||^2$ with a diagonal scaling matrix W that accounts for different strengths of joint actuation.

- **Minimum Time**: this optimization criterion does not need a Lagrangian term. Only a Mayer term at the last phase is needed which is $\Phi_{M_4}(t_4, x(t_4)) = t_4$.
- **Minimum Angular Amplitudes**: for this criterion we minimize the amplitudes of the actuated joints by setting $\Phi_{L_i} = ||q_{actuated}(t)||^2$.
- **Minimum Head Angular Velocity**: here we want to minimize the *global* head angular velocity, which we formulate as $\phi_{L_i} = (\dot{q}_{PelvisRotZ}(t) + \dot{q}_{HeadRotZ}(t))^2$.

Equations (7b) are the model equations from Sect. 2.2 for each contact phase. The transition functions in (7c) are used to describe the changes from one phase to another which we described in Sect. 2.3. General state and control bounds such as joint and torque limits are described by (7d). Posture conditions (e.g. foot positions), periodicity at given points in time and detection of phase switches are modeled by (7e) and (7f). The latter also includes a minimum step size of 0.4 m to ensure a sufficient walking distance.

3.2 Numerical Solution

To solve this problem we use a direct multiple shooting method which is implemented in the software package MUSCOD-II [9]. It discretizes the continuous formulation (7a)–(7f) for both controls and states by dividing the time horizon in M so-called multiple shooting intervals. The states are discretized as starting values s_j for initial value problems defined for each multiple shooting interval j. The controls are discretized by parameters r_j for simple base functions on each multiple-shooting interval, such as piecewise constant, piecewise linear or spline functions for each interval. To ensure that the resulting states represent a continuous solution additional continuity conditions

$$x(t_{j+1}; s_j, r_j) - s_{j+1} = 0$$

are formulated.

By doing so the functions $x(t)$ and $u(t)$ were replaced by their finite dimensional counterparts $s_0, \ldots, s_M$ and $r_0, \ldots, r_{M-1}$. Further discretization of the constraints and objective function leads to an nonlinear optimization problem of variables $y = [s_0, r_0, \ldots, r_{M-1}, s_M, p, t_1, \ldots, t_4]^T$ of the form:

$$\begin{aligned} &\min_y \ F(y) \\ &\text{subject to:} \\ &\quad g(y) \geq 0 \\ &\quad h(y) = 0 \end{aligned}$$

This problem is then solved by using a specially tailored sequential quadratic programming (SQP) method.

4 Results

For the control discretization we used spline functions and 10 multiple shooting intervals for the first phase (Right Flat) and 5 for each of the others (Right Toes, Right Toes Left Heel, and Right Toes Left Flat). The computation time for each motion is around 30 min on an Intel i7 920 processor.

A visualization of the computed gaits is presented in Fig. 3. The motions vary greatly in terms of speed and style. The motions are physically valid for the current model topology and look natural. There is a natural arm swing opposite to the leg swing.

While the motions "Minimum Torques" and "Minimum Head Angular Velocity" do look similar, there is a substantially stronger arm swing for the latter objective criteria. Arm motions counteract the rotational motions of the lower limbs and help to stabilize the upper body. In addition, arm motions can serve to damp out perturbations induced in the walking system by ground impacts. The two motions also differ in average velocity which is 0.99 m/s for "Minimum Torques" and 1.12 m/s for "Minimum Head Angular Velocity".

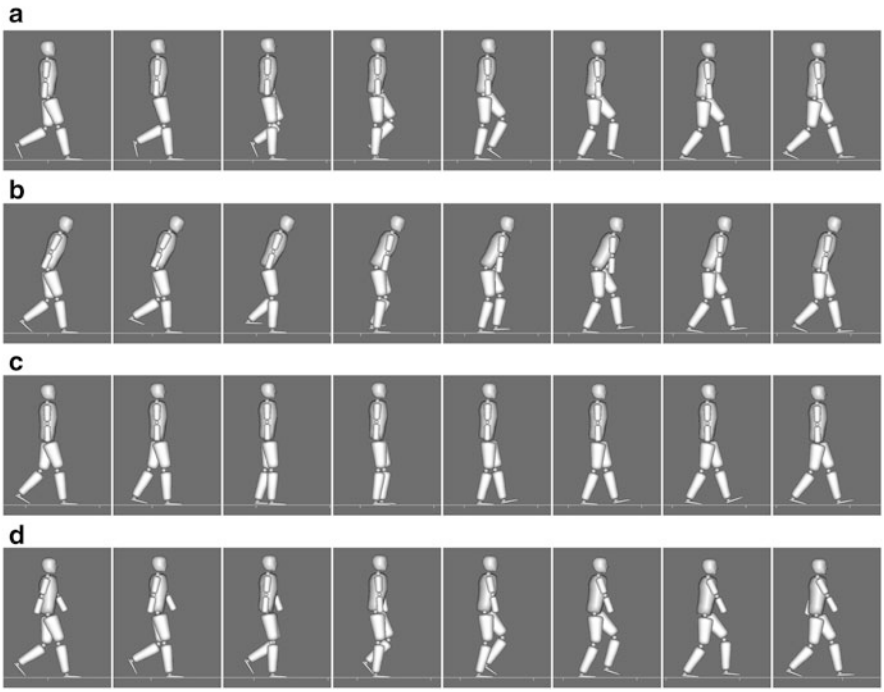

Fig. 3 Visualization of the optimized walking motions for the used objective criteria. (**a**) Minimum torques. (**b**) Minimum time. (**c**) Minimum angular amplitudes. (**d**) Minimum head angular velocity

The resulting motion for "Minimum Time" features a posture that is leaned forward as one would expect for a fast gait and it is also the motion with the shortest duration and also the highest average velocity of 1.6 m/s (duration: 0.3 s, step length: 0.48 m). The motion "Minimum Angular Amplitudes" has an average velocity of 1.45 m/s (duration: 0.44 s, step length: 0.64 m). Here, the overall posture is held upright and conveys a strong tenseness.

5 Conclusion and Outlook

The model and optimal control problem was presented together with various objective criteria and how the optimal control problem can be solved using a direct method. We computed optimized motions for four different objective criteria that resulted in four distinct walking styles. The motions themselves are highly coordinated and compliant. The method presented is well suited to generate off–line open–loop walking motions for a complex full–body human model.

The achieved results form the base of our further work to identify specific objective functions using an inverse optimal control problem formulation as successfully used in [12]. For this we would like to optimize gaits using a linear combination of multiple objective criteria, such as

- *Maximum average velocity*
- *Minimum/maximum energy*
- *Minimum/maximum acceleration*
- *Minimum/maximum jerk*
- *Minimum/maximum ground impacts*
- *Maximum angular amplitudes*
- *Minimum/maximum positive or negative work*
- *Maximum manipulability measure* [7].

These individual objective criteria will serve as base-functions for the identification of the objective function of emotional motions using inverse optimization. A first step for this would be a direct comparison of gaits computed by optimizations with recorded motions obtained by motion capture.

A more realistic model for the ground contact would be using spheres for the heel and toes of the foot or a single ellipsoid. This would allow continuous rolling curved surface instead of the discrete phases of the current flat foot. Another modeling extension would be to create a 3-D model and investigate turning or walking along curved paths. Additionally it would be an interesting question whether we could apply these objective functions for other types of motion such as grasping or other gestures.

Evaluating the results of the different objective criteria, the trajectories of the "Minimum Head Angular Velocity" solution seems to be the closest to a natural walking solution for a neutral emotional state. The importance of this criterion and

others in the context of neutral motions and emotionally modified motions is further evaluated in our current research.

Acknowledgements The authors gratefully acknowledge the financial support and the inspiring environment provided by the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences, funded by DFG (Deutsche Forschungsgemeinschaft). Furthermore we want to thank the Simulation and Optimization research group of the IWR at Heidelberg University for giving us the possibility to work with the software package MUSCOD-II.

References

1. Atkinson, A.P., Dittrich, W.H., Gemmell, A.J., Young, A.W.: Emotion perception from dynamic and static body expressions in point-light and full-light displays. Perception **33**(6), 717–746 (2004). http://www.perceptionweb.com/abstract.cgi?id=p5096
2. Darwin, C.: The Expression of the Emotions in Man and Animals. John Murray, London (1872)
3. Ekman, P., Friesen, W.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists, Palo Alto (1978)
4. Featherstone, R.: Rigid Body Dynamics Algorithms. Springer, New York (2008)
5. Felis, M.L.: RBDL – the Rigid Body Dynamics Library. http://rbdl.bitbucket.org (2012)
6. Guennebaud, G., Jacob, B., et al.: Eigen v3. http://eigen.tuxfamily.org (2010)
7. Klein, C.A., Blaho, B.E.: Dexterity measures for the design and control of kinematically redundant manipulators. Int. J. Robot. Res. **6**(2), 72–83 (1987). doi:10.1177/027836498700600206. http://ijr.sagepub.com/content/6/2/72.abstract
8. Koschorreck, J., Mombaur, K.: Optimization of sommersaults and twists in platform diving. Comput. Methods Biomech. Biomed. Eng. **12**(2-1), 157–159 (2009)
9. Leineweber, D., Bauer, I., Bock, H.G., Schlöder, J.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: theoretical aspects. Comput. Chem. Eng. **27**, 157–166 (2003)
10. de Leva, P.: Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. J. Biomech. **29**(9), 1223–30 (1996)
11. Liu, C.K., Hertzmann, A., Popović, Z.: Learning physics-based motion style with nonlinear inverse optimization. ACM Trans. Graph. **24**, 1071–1081 (2005). doi:10.1145/1073204.1073314
12. Mombaur, K., Truong, A., Laumond, J.P.: From human to humanoid locomotion inverse optimal control approach. Auton Robots **28**, 369–383 (2010). doi:10.1007/s10514-009-9170-7
13. Omlor, L., Giese, M.A.: Extraction of spatio-temporal primitives of emotional body expressions. Neurocomputing **70**, 1938–1942 (2007). doi:10.1016/j.neucom.2006.10.100
14. Schultz, G., Mombaur, K.: Modeling and optimal control of human-like running. IEEE/ASME Trans. Mechatron. **15**(5), 783–792 (2010). doi:10.1109/TMECH.2009.2035112
15. Troje, N.F.: Decomposing biological motion: a framework for analysis and synthesis of human gait patterns. J. Vis. **2**(5) (2002). doi:10.1167/2.5.2. http://www.journalofvision.org/content/2/5/2.abstract

On Quadratic Programming Based Iterative Learning Control for Systems with Actuator Saturation Constraints

Fei Gao and Richard W. Longman

Abstract When feedback control systems are given a commanded desired trajectory to perform, they produce a somewhat different trajectory. The concept of bandwidth is used to indicate what frequency components of the trajectory are executed reasonably well. Iterative Learning Control (ILC) iteratively changes the command, aiming to make the control system output match the desired output. The theory of linear ILC is reasonably well developed, but in hardware applications the nonlinear effects from hitting actuator saturation limits during the process of convergence of ILC could be detrimental to performance. Building on previous work by the authors and coworkers, this paper investigates the conversion of effective ILC laws into a quadratic cost optimization. And then it develops the modeling needed to impose actuator saturation constraints during the ILC learning process producing Quadratic Programming based ILC, or QP-ILC. The benefits and the need for ILC laws that acknowledge saturation constraints are investigated.

1 Introduction

Iterative learning control is a relatively new form of control theory that develops methods of iteratively adjusting the command to a feedback control system aiming to converge to that command that produces zero tracking error of a specific desired trajectory. References [1–3] develop various linear formulations. Reference [4] develops the supervector approach to mathematical modeling of ILC that is used here. There are some perhaps surprisingly strong mathematical convergence results for very general nonlinear systems, Refs. [5, 6], but they tend to use the simplest form of ILC that can have extremely bad transients [3] and may require that

F. Gao
Visiting Research Scholar, Columbia University, New York, NY 10027, USA

Doctoral Candidate, Tsinghua University, 100084 Beijing, China
e-mail: ambersro@gmail.com

R.W. Longman (✉)
Department of Mechanical Engineering, Columbia University, New York, NY 10027, USA
e-mail: rwl4@columbia.edu

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_4

the system being controlled has the property that the output is instantaneously changed by a step change in the input [6]. Reference [7] develops methods of numerical computation for implementation of various effective linear ILC laws to nonlinear systems. Our objective here is to generalize methods of ILC to handle the specific kind of nonlinearity presented by inequality constraints on the actuators. The work follows on from [8] including generalization to form well posed inverse ILC problems, and is related to [9].

The present paper considers three effective linear ILC laws, a Euclidean norm contraction mapping ILC law which we refer to as the P transpose ILC law [10], the partial isometry ILC law [11], and the ILC law based on a quadratic cost penalty function on the transients, or changes in the control action from iteration to iteration [12, 13]. Reference [14] shows how one can create a unified formulation of each of these control laws by appropriate choices of the weights in the quadratic cost control. Most discrete time systems that come from a continuous time system fed by a zero order hold have an unstable inverse which implies that the control action needed for zero tracking error is an unstable function of time step. This difficulty can be addresses by the methods of Refs. [15–17].

This paper considers linear systems subject to inequality limits on actuators. It combines these approaches to form a quadratic cost minimization problem, formulating the needed equations to represent actuator saturation limits, thus forming a well posed quadratic programming problem [18, 19]. From the point of view of quadratic programming, these problems are small and easily solved for the updates to the commands made each iteration.

2 Two Classes of ILC Problems with Constraints

Time optimal control problems usually involve control actions that are bang bang, i.e. on the actuator constraint boundary. Fuel optimal control can also have actuators at their limits. One form of ILC considers the equations from actuator to output. In this case, it is reasonable to consider the relatively simple situation of an ILC problem where this input is saturated. Figure 1 illustrates this situation, which is referred to later as Problem 1. We use $G(s) = [a/(s+a)][\omega_0^2/(s^2 + 2\zeta\omega_0^2)]$ with $\omega_0 = 37$ and $\zeta = 0.5$, sampling at $T = 1/100$ s in numerical examples. ILC makes use of stored data from the previous run, and hence must it be digital and must use sampled data. Here we use a zero order hold. Converted to state variable form, the continuous time and discrete time modes are

Fig. 1 Problem 1, where ILC adjusts input that is subject to inequality constraints

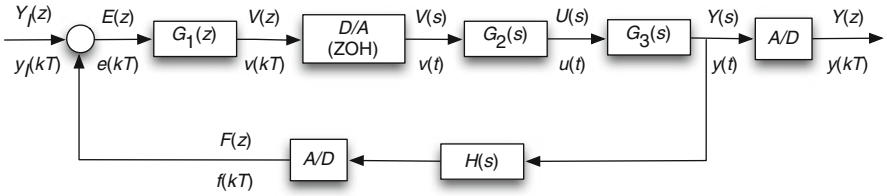

Fig. 2 Problem 2, a feedback control system with saturation limits on the actuator output

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \quad ; \quad y(t) = Cx(t) \\ x((k+1)T) &= Ax(kT) + Bu(kT) \quad ; \quad y(kT) = Cx(kT) \\ A &= e^{A_c T} \quad ; \quad B = A_c^{-1}(A - I)B_c \end{aligned} \tag{1}$$

The more common situation has the ILC adjusting the command to a digital feedback control system, as in Fig. 2. The $G_1(z)$ represents the digital feedback control law while $G_2(s)$ represents the dynamics of an actuator whose output feeds the plant equation $G_3(s)$. An $H(s)$ is included because applications such as robotics often benefit from the use of rate feedback. The mathematical formulation of ILC subject to actuator output saturation is developed in general for hard saturation limits on $u(t)$.

Numerical examples for Problem 2 use $G_1(z) = K_1 = 12{,}050$, a proportional controller whose computation is considered instantaneous and therefore it does not produce a time step delay. The actuator dynamics are represented by $G_2(s) = 1/(s + a)$ with $a = 41.8$ while $G_3(s) = 1/(s^2 + 4s)$, $H(s) = 1 + 0.1268s$, and $T = 1/100$ s.

3 Several Effective Linear ILC Laws

3.1 General ILC Supervector Formulation

Consider a discrete time input output state space model with its convolution sum solution

$$\begin{aligned} x((k+1)T) &= Ax(kT) + Bu(kT) \quad ; \quad y(kT) = Cx(kT) + \upsilon(kT) \\ y(kT) &= CA^k x(0) + \sum_{i=0}^{k-1} CA^{k-i-1} Bu(iT) + \upsilon(kT) \end{aligned} \tag{2}$$

The $\upsilon(kT)$ represents any disturbance that repeats every run, and it is represented by its equivalent effect on the output. A one time step delay from a change in the input to the resulting change in the output is assumed. The mathematics can easily be altered to account for a different delay. Hence, for control action starting at time step zero, the desired trajectory $y^*(kT)$ is defined starting with time step one, and continuing to time step p, with associated error $e(kT) = y^*(kT) - y(kT)$. Following [4] we define symbols with underbars to represent the history of the associated variables for any run (subscript j will be applied to indicate iteration or run number j). Based on the one time step delay through the system, the entries in $\underline{u}$ start with time step 0 and go to $p-1$, and for $\underline{y}, \underline{y}^*, \underline{e}, \underline{\upsilon}$ start with time step 1 and go to p. Then

$$\underline{y} = P\underline{u} + \bar{A}x(0) + \underline{\upsilon} \quad ; \quad \underline{e} = -P\underline{u} + (\underline{y}^* - \bar{A}x(0) - \underline{\upsilon})$$
$$P = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & \cdots & CB \end{bmatrix} \qquad \bar{A} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix} \tag{3}$$

3.2 General Linear ILC and Convergence

A general ILC law updates the input action according to

$$\underline{u}_{j+1} = \underline{u}_j + L\underline{e}_j \tag{4}$$

where L is a matrix of ILC gains. By taking the difference of the right hand equation in (3) for two successive iterations, one obtains the error convergence condition. By substituting the error equation in (3) into (4), one also obtains the control action convergence condition

$$\underline{e}_{j+1} = (I - PL)\underline{e}_j$$
$$\underline{u}_{j+1} = (I - LP)\underline{u}_j + L(\underline{y}^* - \bar{A}x(0) - \underline{\upsilon}) \quad ; \quad \Delta_{j+1}\underline{u} = (I - LP)\Delta_j\underline{u} \tag{5}$$

Defining $\Delta_j\underline{u} = \underline{u}_j - \underline{u}_\infty$ converts the second equation into the third. Convergence of the error requires that the spectral radius of $I - PL$ be less than unity, and a sufficient condition is that the maximum singular value is less than unity.

3.3 Euclidean Norm Contraction Mapping ILC Law (P Transpose Law)

This law picks $L = \phi P^T$ where ϕ is a positive gain to be chosen [10]. Write the singular value decomposition of P as $P = USV^T$. Note that P is lower triangular with nonzero elements on the diagonal, and therefore in theory is full rank with $S = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$ having all positive diagonal elements. It can however be badly ill conditioned, and this is accounted for below. Equations (5) can be rewritten for this law

$$U^T \underline{e}_{j+1} = (I - \phi S^2) U^T \underline{e}_j \quad ; \quad V^T \Delta_{j+1} \underline{u} = (I - \phi S^2) V^T \Delta_j \underline{u} \tag{6}$$

The Euclidean norms of $\underline{e}_j$ and $U^T \underline{e}_j$ are the same, and similarly for $\Delta_j \underline{u}$. Monotonic convergence is obtained for all initial error histories if and only if $1 - \phi\sigma_i^2$ is less than unity in magnitude for all i, or it converges if $0 < \phi < 2/\sigma_i^2$ for all i. If $1 - \phi\sigma_i^2$ is positive, then the corresponding component of the error will converge from the same side each update, and if it is negative this component of the error will alternate in sign from iteration to iteration.

3.4 Partial Isometry ILC Law

Set $L = \phi V U^T$ as in [11] and one obtains convergence to zero error as in Eq. (6) with $I - \phi S^2$ replaced by $I - \phi S$, with $0 < \phi < 2/\sigma_i$. This law learns faster for high frequency error components, but is somewhat less robust to model errors.

3.5 General Quadratic Cost ILC Law

Usually quadratic cost control implies a compromise between control effort and speed of decay of the transients. ILC wants zero error, but uses the quadratic cost compromise on the size of the control update from iteration to iteration, controlling the learning transients. The quadratic cost at iteration j that determines $\underline{u}_{j+1}$ minimizes

$$J_j = \underline{e}_{j+1}^T Q \underline{e}_{j+1} + \delta_{j+1} \underline{u}^T R \delta_{j+1} \underline{u} \quad ; \quad \delta_{j+1} \underline{u} = \underline{u}_{j+1} - \underline{u}_j \tag{7}$$

Normally, one asks that Q be positive semidefinite and R be positive definite. In this case, we want zero final tracking error so we want Q positive definite, and R need not be positive definite. Write the right hand equation in (3) for $j + 1$ and for j, and then (7) can be rewritten as

$$J_j = (\underline{e}_j - P\delta_{j+1}\underline{u})^T Q(\underline{e}_j - P\delta_{j+1}\underline{u}) + \delta_{j+1}\underline{u}^T R\delta_{j+1}\underline{u}$$
$$= \delta_{j+1}\underline{u}^T(P^T QP + R)\delta_{j+1}\underline{u} - 2\delta_{j+1}\underline{u}^T P^T Q\underline{e}_j + \underline{e}_j^T Q\underline{e}_j \tag{8}$$

Instead of R having to be positive definite, this quadratic cost law requires the Hessian $P^T QP + R$ to be positive definite, which actually allows for negative R. Setting the derivative with respect to $\delta_{j+1}\underline{u}$ to zero produces the ILC law

$$\underline{u}_{j+1} = \underline{u}_j + L\underline{e}_j = \underline{u}_j + (P^T QP + R)^{-1} P^T Q\underline{e}_j \tag{9}$$

3.6 Simple Quadratic Cost ILC

Often when applying quadratic cost control designs one does not have much guidance on how to pick the weight matrices, and one uses simple choices. Here we consider $Q = I$ because we want all error components to go to zero, and let $R = rI$ where r is a scalar gain. Then

$$L = V(rI + S^2)^{-1} SU^T$$
$$(U^T\underline{e}_{j+1}) = [I - S(rI + S^2)^{-1}S](U^T\underline{e}_j) \tag{10}$$
$$= diag\Big(1 - \frac{\sigma_i^2}{r + \sigma_i^2}\Big)(U^T\underline{e}_j) = diag\Big(\frac{r}{r + \sigma_i^2}\Big)(U^T\underline{e}_j)$$

In this case, each component of the error on the orthonormal basis vectors in U will converge monotonically without alternating sign when r is positive, but it is possible to have convergence for negative values provided $r > -\sigma_i^2$ for all i.

3.7 Defining Well Posed ILC Inverse Problems

Continuous time systems fed by a zero order hold can be converted to discrete time systems with identical output histories at sample times. For continuous time systems with pole excess of 3 or more and sufficiently fast sampling, this conversion introduces zeros outside the unit circle. This means that the inverse problem is unstable. References [15–17] address this problem, for example, by asking for zero error every other time step making a kind of generalized hold. We rewrite Eq. (3) as

$$\underline{e}_{D,j} = -P_D\underline{u}_j + \big(\underline{y}_D^* - \bar{A}_D x(0) - \underline{v}_D\big) \quad ; \quad \delta_{j+1}\underline{e}_D = -P_D\delta_{j+1}\underline{u} \tag{11}$$

Initially write the equation for all time steps at the faster sample rate used by the control input. Then delete whichever rows are associated with errors that are not to

be addresses. Note that this makes P a rectangular matrix. The General Quadratic Cost ILC Law generalizes to

$$\begin{aligned} J_j &= \underline{e}_{D,j+1}^T Q \underline{e}_{D,j+1} + \delta_{j+1} \underline{u}^T R \delta_{j+1} \underline{u} \\ \underline{u}_{j+1} &= \underline{u}_j + L_D \underline{e}_{D,j} \quad ; \quad L_D = (P_D^T Q P_D + R)^{-1} P_D^T Q \end{aligned} \tag{12}$$

Note that $P_D^T Q P_D$ is now positive semidefinite. The Euclidean Norm Contraction Mapping ILC law generalizes immediately to $L_D = \phi P_D^T$. The Partial Isometry Law needs the singular value decomposition of P_D denoted by $P_D = U \begin{bmatrix} S & 0 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T = U S V_1^T$ making $L_D = \phi V_1 U^T$.

4 Quadratic Cost Versions of These Effective ILC Laws

Equation (7) is the general version of quadratic cost ILC. By setting $Q = I$ and $R = rI$ the general version reduces to the Simple Quadratic Cost ILC law. However, the other two ILC law presented can also be produced from the general quadratic cost function. The Euclidean Norm Contraction Mapping Law $L = \phi P^T$ is produced when one sets $Q = \phi I$ and $R = I - \phi P^T P$ and substitutes into the L of Eq. (9). The partial isometry law is obtained when $R = VS(I - \phi S)V^T = P^T U(I - \phi S)V^T$ with the same $Q = \phi I$. We note that when substituted into L, one obtains $L = (VSV^T)^{-1} VSU^T \phi$. The inverse involved can be ill conditioned since S determines the condition number of P discussed above. The details of the computation of the updates using the quadratic cost formulation determine whether there is some additional difficulty when using this law in quadratic cost form.

One of the issues in the learning transients of ILC laws when there are inequality constraints is whether the iterations try to go beyond the actuator limit during the convergence process, and as a result perhaps the convergence process might fail. We note that the Simplified Quadratic Cost ILC Law in Eq. (10), when r is picked larger than zero, will have every component of the error projected onto the unit vectors of U converging to zero without changing sign. Thus if the initial trajectory starts with each of these components smaller than those needed for zero error, and the desired trajectory is feasible, i.e. it does not require actuator output beyond the actuator limits, then one expects convergence without difficulty using the learning law without regard to the inequality constraints. Using r that is negative seems counter intuitive, and furthermore how negative it can be is less than minus the smallest singular value which is often a very small number. When using the P transpose law, picking ϕ so that $0 < 1 - \phi\sigma_i^2 < 1$ for all i, will similarly ensure monotonic approach of the same sign to zero error for each of the error components on the unit vectors of U. Any i for which $-1 < 1 - \phi\sigma_i^2 < 0$ will alternate the sign of this component of the error each iteration. If the desired trajectory is feasible but goes near the actuator limit, the alternating sign of error components could easily

make the control law ask to go beyond the limit. For the partial isometry law, the corresponding conditions are $0 < 1 - \phi\sigma_i$ and $-1 < 1 - \phi\sigma_i < 0$.

To address the ill conditioning problem discussed above, and also convert to the P transpose and partial isometry laws to quadratic cost ILC form we pick weights in cost function (12) as follows. For P transpose, simply use $Q = \phi I$ and $R = I - \phi P_D^T P_D$. For the partial isometry law one again uses $Q = \phi I$, and the R matrix becomes more complicated

$$R = V \begin{bmatrix} S(I - \phi S) & 0 \\ 0 & I \end{bmatrix} V^T \tag{13}$$

Substitution into Eq. (12) produces the needed $L_D = \phi V_1 U^T$.

5 The Quadratic Programming Problem for Problem 1

Having produced quadratic cost versions of each of the above control laws, we are ready to impose inequality constraints on the actuator by formulating the update for each iteration as a quadratic programming (QP) problem:

$$\text{min: } J = \frac{1}{2} z^T H z + g^T z \quad \text{subject to: } A_1 z \le Z_1 \text{ and } A_2 z = Z_2 \tag{14}$$

Using the general quadratic cost ILC law, which can represent any of the ILC laws discussed above by choice of Q and R, at each iteration we seek to minimize J_j of Eq. (7). The equality constraint $A_2 z = Z_2$ could be used for the dynamics $\delta_{j+1}\underline{e} = -P\delta_{j+1}\underline{u}$, but we can simply substitute analytically to obtain the cost equation (8) with $H = \frac{1}{2}(R + P^T Q P)$ and $g^T = \underline{e}_j^T Q P$. In Problem 1, the input is subject to inequality constraints $\underline{u}_{min} \le \underline{u} \le \underline{u}_{max}$ which is to be interpreted component by component. These constraints are imposed with A_1, Z_1 using $\delta_{j+1}\underline{u} \le \underline{u}_{max} - \underline{u}_j$ and $-\delta_{j+1}\underline{u} \le -(\underline{u}_{min} - \underline{u}_j)$.

6 Formulating the Quadratic Programming Problem for Problem 2

6.1 Closed Loop Dynamics for Problem 2

Generation of the QP version of Problem 2 requires some effort to formulate the inequality constraints. We assume that the hardware has a hard constraint at its limit. This time the constraint is on the output of a continuous time transfer function that may represent the actuator, $u_{min} \le u(t) \le u_{max}$. For simplicity, we formulate the

problem asking to satisfy the constraints at the sample times $u_{min} \leq u(kT) \leq u_{max}$, and ignore any issues associated with violation of constraints between sample times. Referring to Fig. 2, we can write equations going around the loop. For the controller

$$\begin{aligned} e(kT) &= y_I(kT) - f(kT) \\ f(kT) &= y(kT) + K\dot{y}(kT) \\ x_d((k+1)T) &= A_d x_d(kT) + B_d e(kT) \\ v(kT) &= C_d x_d(kT) + D_d e(kT) \end{aligned} \tag{15}$$

We want to know not only the output $y(t)$ at least at the sample times, but also we want to monitor the value of $u(t)$ at the sample times, and for purposes of computing the feedback with $H(s) = 1 + Ks$ we want to know $\dot{y}(kT)$. The actuator and the plant equations in continuous time are

$$\begin{aligned} \dot{x}_a(t) &= A_{a,c}x_a(t) + B_{a,c}v(t) \quad ; \quad u(t) = C_a x_a(t) \\ \dot{x}_p(t) &= A_{p,c}x_p(t) + B_{p,c}u(t) = A_{p,c}x_p(t) + B_{p,c}C_a x_a(t) \quad ; \quad y(t) = C_p x_p(t) \\ \dot{y}(t) &= C_p\dot{x}_p(t) = C_p A_{p,c}x_p(t) + C_p B_{p,c}C_a x_a(t) \end{aligned} \tag{16}$$

One can combine these equations using a combined state $x_{ap} = \begin{bmatrix} x_a^T & x_p^T \end{bmatrix}^T$ and compute three output quantities

$$\begin{bmatrix} u(t) \\ y(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} C_a & 0 \\ 0 & C_p \\ C_p B_{p,c} C_a & C_p A_{p,c} \end{bmatrix} \begin{bmatrix} x_a(t) \\ x_p(t) \end{bmatrix} \tag{17}$$

The combined equations have input $v(t)$ coming from a zero order hold, and hence it can be converted to a difference equation giving these output quantities at the sample times without approximation. One can now combine these equations with (15) adding the controller state variable to those in (16) to form the closed loop state equations relating the command $y_{I,j}(kT)$ in iteration j to the resulting output $y_j(kT)$ and the actuator output $u_j(kT)$ which is subject to saturation

$$\begin{aligned} x_{CL,j}((k+1)T) &= A_{CL}x_{CL,j}(kT) + B_{CL}y_{I,j}(kT) \\ y_j(kT) &= C_{CL}x_{CL,j}(kT) \\ u_j(kT) &= C_u x_{CL,j}(kT) \end{aligned} \tag{18}$$

6.2 Dynamics When the Actuator Is Saturated

When the actuator is at a saturation limit u_{sat}, then when simulating, the feedback loop is only needed to determine when the actuator leaves its saturated value. The output and its derivative are given by

$$\begin{gathered} x_p((k+1)T)=A_p x_p(kT)+B_p u_{sat}\ ;\ A_p=exp(A_{p,c}T)\ ;\ B_p=A_{p,c}^{-1}(A_p-I)B_{p,c} \\ y(kT)=C_p x_p(kT)\quad ;\quad \dot{y}(kT)=C_p A_p x_p(kT) \end{gathered} \tag{19}$$

To determine in simulation when the actuator leaves saturation, one uses these values in Eqs. (15) to determine $v(kT)$, and feeds this into the discrete time version of the first row equations of (16) in order to monitor the value of $u(kT)$.

6.3 The QP Problem for Problem 2

Form two different P matrices, P_{CL} making use of system matrices A_{CL}, B_{CL}, C_{CL}, and the second P_u using A_{CL}, B_{CL}, C_u. For simplicity we ignore the repeating disturbance term so that

$$\begin{aligned} \underline{y}_j &= P_{CL}\underline{y}_{I,j}+\bar{A}_{CL}x_{CL}(0) \\ \underline{e}_j &= -P_{CL}\underline{y}_{I,j}+(\underline{y}^*-\bar{A}_{CL}x_{CL}(0)) \\ \underline{u}_j &= P_u\underline{y}_{I,j}+\bar{A}_u x_{CL}(0) \\ \delta_{j+1}\underline{u} &= P_u\delta_{j+1}\underline{y}_I \end{aligned} \tag{20}$$

where $\bar{A}_{CL}$ and $\bar{A}_u$ are formed as in Eq. (3) using A_{CL} and C_{CL} or C_u respectively. Then the QP problem at iteration j is to minimize the following quadratic cost subject to the following inequality constraints

$$\begin{gathered} J_j=\delta_{j+1}\underline{y}_I^T(P_{CL}^T Q P_{CL}+R)\delta_{j+1}\underline{y}_I-2\delta_{j+1}\underline{y}_I^T P_{CL}^T Q\underline{e}_j+\underline{e}_j^T Q\underline{e}_j \\ \underline{u}_{min}\le \underline{u}_j+P_u\delta_{j+1}\underline{y}_I\le \underline{u}_{max} \end{gathered} \tag{21}$$

7 Numerical Study of QP Based ILC

To create a desired trajectory that rides the actuator or input limit we pick $u(t)=sat\{0.5[1-\cos(2\pi t)]\}$, saturated at the value 0.7. Then we compute the output and use it as the desired trajectory. In some applications the inequality constraint

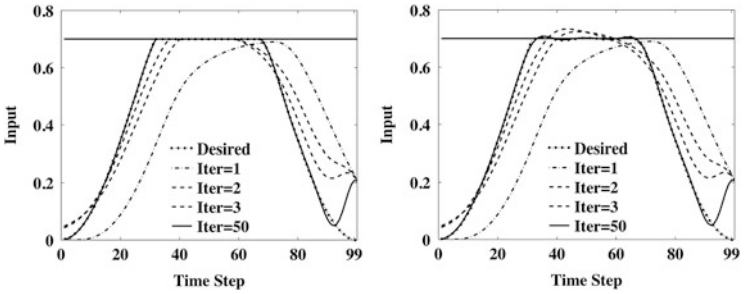

Fig. 3 Problem 1, P transpose ILC law with $\phi = 1$. *Left*: hardware imposes the limit. *Right*: hardware allows violation of limit

is a hard constraint that cannot be violated. For example, sometimes when a DC motor is used in a control system, it has a voltage limiter on it that imposes a hard constraint. Other times, the manufacturer lists a limit which can be based on such things as heating constraint in sustained operation. In this case, violating the constraint temporarily during the learning process can be acceptable. Here we examine the behavior of ILC laws in three cases: (1) When the law and the hardware are allowed to violate the limit temporarily. (2) When the law is allowed to ask for signals that violate the limit, but the hardware imposes the limit. (3) When the corresponding QP based ILC law incorporates the limits in the updates each iteration.

Figure 3 shows the convergence to the needed $u(kT)$ using a P transpose ILC law when the hardware does not allow violation of the constraint, and when it does. The root mean square (RMS) of the tracking error for each iteration is essentially identical, but the former learns slightly faster. It is interesting to note that although the mathematics indicates that all time steps converge to zero error, the last time step in these plots of the $u(kT)$ is converging very slowly. This phenomenon has now been studied in Ref. [20].

An interesting example was run using the P transpose ILC law with a gain of $\phi = 2.3$ which is above the stability limit of 2.2. As one might expect, the RMS error when violation was allowed went unstable. But with the hardware imposing the limit, convergence approaching zero error occurred, i.e. the hardware limit stabilized an unstable ILC.

Figure 4 illustrates the RMS error performance as ILC iterations progress for the partial isometry ILC law and the simple quadratic cost ILC law. The dashed curves in each case show the convergence when the ILC law could violate the constraint but the hardware imposed the constraint. The corresponding curves when the hardware allowed violation of the constraints looked very similar, except that the somewhat quick changes in slope in the figures disappear. The solid curves are the RMS errors for the corresponding QP algorithms that make use of the limits in the updates computed. Note that in both cases, the QP algorithm eventually outperforms the

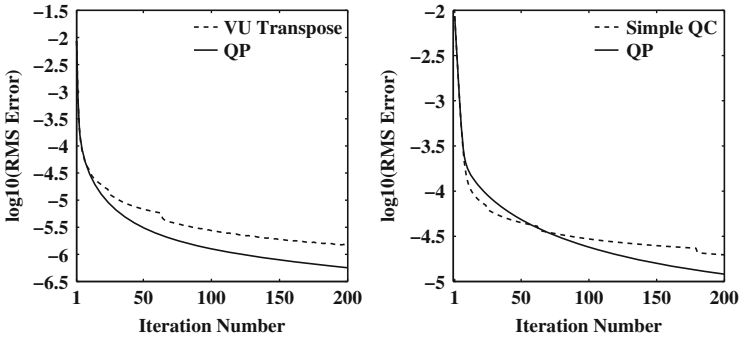

Fig. 4 Problem 2. *Left*: UV^T ILC with $\phi = 1$, *Right*: simple quadratic cost ILC, hardware imposes limit, $r = 1$. Versus QP

other algorithms. Note that the VU^T law reaches lower error levels since it converges linearly instead of quadratically in small errors.

References

1. Bien, Z., Xu, J.X. (eds.): Iterative learning control: analysis, design, integration and applications. Kluwer Academic, Boston (1998)
2. Moore, K., Xu, J.X. (guest eds.): Special issue on iterative learning control. Int. J. Control **73**(10) (2000)
3. Longman, R.W.: Iterative learning control and repetitive control for engineering practice. Int. J. Control **73**(10), 930–954 (2000)
4. Phan, M., Longman, R.W.: A mathematical theory of learning control for linear discrete multivariable systems. In: Proceedings of the AIAA/AAS Astrodynamics Conference, Minneapolis, pp. 740–746 (1988)
5. Longman, R.W., Chang, C.K., Phan, M.Q.: Discrete time learning control in nonlinear systems. In: A Collection of Technical Papers, AIAA/AAS Astrodynamics Specialist Conference, Hilton Head, pp. 501–511 (1992)
6. Xu, J.X., Tan, Y.: Linear and Nonlinear Iterative Learning Control. Springer, Berlin/New York (2003)
7. Longman, R.W., Mombaur, K.D.: Implementing linear iterative learning control laws in nonlinear systems. Adv. Astronaut. Sci. **130**, 303–324 (2008)
8. Longman, R.W., Mombaur, K.D., Panomruttanarug, B.: Designing iterative learning control subject to actuator limitations using QP methods. In: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Hawaii (2008)
9. Mishra, S., Topcu, U., Tomizuka, M.: Optimization-based constrained iterative learning control. IEEE Trans. Control Syst. Technol. **19**(6), 1613–1621 (2011)
10. Jang, H.S., Longman, R.W.: A new learning control law with monotonic decay of the tracking error norm. In: Proceedings of the Thirty-Second Annual Allerton Conference on Communication, Control, and Computing, Monticello, pp. 314–323 (1994)
11. Jang, H.S., Longman, R.W.: Design of digital learning controllers using a partial isometry. Adv. Astronaut. Sci. **93**, 137–152 (1996)

12. Phan, M.Q., Frueh, J.A.: System identification and learning control, chapter 15. In: Bien, Z., Xu, J.X. (eds.) Iterative Learning Control: Analysis, Design, Integration, and Applications, pp. 285–306. Kluwer Academic, Norwell (1998)
13. Owens, D.H., Amann, N.: Norm-optimal iterative learning control. Internal Report Series of the Centre for Systems and Control Engineering, University of Exeter (1994)
14. Bao, J., Longman, R.W.: Unification and robustification of iterative learning control laws. Adv. Astronaut. Sci. **136**, 727–745 (2010)
15. Li, Y., Longman, R.W.: Characterizing and addressing the instability of the control action in iterative learning control. Adv. Astronaut. Sci. **136**, 1967–1985 (2010)
16. Li, Y., Longman, R.W.: Addressing problems of instability in intersample error in iterative learning control. Adv. Astronaut. Sci. **129**, 1571–1591 (2008)
17. Li, T., Longman, R.W., Shi, Y.: Stabilizing intersample error in iterative learning control using multiple zero order holds each time step. Adv. Astronaut. Sci. **142**, 2965–2980 (2012)
18. 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**(4), 1040–1058 (1996)
19. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic, London (1981)
20. Gao, F., Longman, R.W.: Examining the learning rate in iterative learning control near the end of the desired trajectory. Adv. Astronaut. Sci. **148**, 2019–2037 (2013)

A Sparse Grid Based Generative Topographic Mapping for the Dimensionality Reduction of High-Dimensional Data

Michael Griebel and Alexander Hullmann

Abstract Most high-dimensional data exhibit some correlation such that data points are not distributed uniformly in the data space but lie approximately on a lower-dimensional manifold. A major problem in many data-mining applications is the detection of such a manifold from given data, if present at all. The generative topographic mapping (GTM) finds a lower-dimensional parameterization for the data and thus allows for nonlinear dimensionality reduction. We will show how a discretization based on sparse grids can be employed for the mapping between latent space and data space. This leads to efficient computations and avoids the 'curse of dimensionality' of the embedding dimension. We will use our modified, sparse grid based GTM for problems from dimensionality reduction and data classification.

1 Introduction

High-dimensional data often exhibit a correlation structure between the variables, which means that there are areas in the data space with little or no data points. A suitable low-dimensional projection of the data then allows a more compact description, a better visualization and a more efficient processing.

One approach to dimensionality reduction is to express the high-dimensional data in terms of latent variables. A well-known method is the Principal Component Analysis (PCA), which is based on the diagonalization of the data covariance matrix. However, the PCA is by construction a linear method. As such, it is not capable of modeling nonlinear lower-dimensional dependencies and sometimes may fail. A simple three-dimensional example, the so called 'Swiss roll', is given in Fig. 1. Here, the topological structure of the data is not preserved under the mapping into two dimensions and points originally far apart on the manifold are close-by in the two-dimensional projection.

This is why nonlinear methods are necessary. Some common approaches are multidimensional scaling (MDS), curvilinear component analysis (CCA), curvilin-

M. Griebel • A. Hullmann (✉)
Institute for Numerical Simulation, University of Bonn, Bonn, Germany
e-mail: griebel@ins.uni-bonn.de; hullmann@ins.uni-bonn.de

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_5

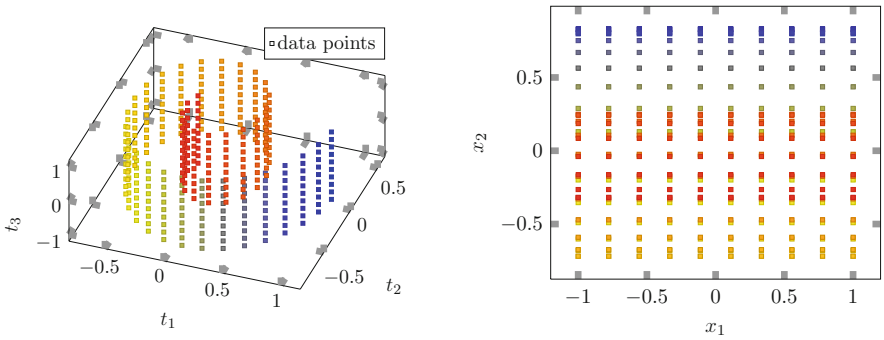

Fig. 1 The projection of the 'Swiss roll' data (*left*) onto the first two principal components results in a two-dimensional representation (*right*)

ear distance analysis (CDA), Laplacian eigenmaps (LE), locally linear embedding (LLE), Kohonen's self-organizing map (SOM), generative topographic mapping (GTM) and kernel PCA (KPCA), cf. [17]. Unfortunately, capturing nonlinearities comes at the price of a significant increase in computational complexity and with the problem of possibly finding only a locally optimal solution.

In this article we will focus on the generative topographic mapping (GTM) [4]. Usually, the latent space of the generative model is limited to two or three dimensions due to the 'curse of dimensionality'. It means that the cost complexity for the approximation to the solution of a problem grows exponentially with the dimension d, i.e. it is of the order $\mathcal{O}(h^{-d})$ with h being the one-dimensional mesh-width. Instead, we use sparse grids [6] for the discretization of the mapping between latent space and data space. Then, the number of degrees of freedom grows only by $\mathcal{O}(h^{-1}\,(\log h^{-1})^{d-1})$, which is a substantial improvement. This approach has also been followed for principal curves and manifolds in [10]. Of course, this saving in computational complexity comes at a cost, namely an additional logarithmic error term and a stronger smoothness assumption on the mapping. As a result, we get a sparse GTM (SGTM), which basically achieves the same level of accuracy with less degrees of freedom. In contrast to the conventional GTM, it can cope with higher-dimensional latent spaces.

This paper is organized as follows: In Sect. 2, we describe our generative model, which is based on a mapping between the lower-dimensional latent space and the data space. In Sect. 3, we present a method to find the mapping by minimizing a certain target functional, i.e. the regularized cross-entropy between the model and the given data. Then, in Sect. 4, we show how we can obtain the original GTM as well as the sparse GTM by special discretization choices. In Sect. 5, we apply the sparse GTM to a benchmark dataset from literature and a real-world classification problem. Some final remarks conclude this paper.

2 The Generative Model

In the following, we will describe a generative model, which is based on a low-dimensional parameterization.

We want to represent a d-dimensional density $p(\mathbf{t}) \geq 0, \mathbf{t} \in \mathbb{R}^d$, by a density that is intrinsically low-dimensional. To this end, we introduce a mapping

$$\mathbf{y} : [0,1]^L \to \mathbb{R}^d$$

with $L \ll d$ that connects the L-dimensional latent space $[0,1]^L$ and the data space $\mathbb{R}^d$. The generated density is then

$$q_{\mathbf{y},\beta}(\mathbf{t}) = \left(\frac{\beta}{2\pi}\right)^{d/2} \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2\right) \mathrm{d}\mathbf{x} \,. \tag{1}$$

It can be interpreted as the image of an L-dimensional uniform distribution under the mapping $\mathbf{y}$ with additional Gaussian noise, which is controlled by the parameter β, see Fig. 2 for an illustration. It is easy to see that $\int_{\mathbb{R}^d} q_{\mathbf{y},\beta}(\mathbf{t})\mathrm{d}\mathbf{t} = 1$, i.e. $q_{\mathbf{y},\beta}$ is a density in the d-dimensional data space.

The aim is now to choose a mapping $\mathbf{y}$ and an inverse variance $\beta \in \mathbb{R}_+$, such that the dissimilarity between $q_{\mathbf{y},\beta}$ and p is minimized. To be precise, for a given regularization term $\lambda S(\mathbf{y})$ and density $p(\mathbf{t})$, we want to minimize the regularized cross-entropy [16]

$$\begin{aligned} \mathscr{G}(\mathbf{y},\beta) &:= H(p, q_{\mathbf{y},\beta}) + \lambda S(\mathbf{y}) \\ &= -\int_{\mathbb{R}^d} p(\mathbf{t}) \log \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2\right) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} - \frac{d}{2}\log\frac{\beta}{2\pi} + \lambda S(\mathbf{y}) \end{aligned} \tag{2}$$

in $\mathbf{y}$ and β. For the remainder of this paper, we assume $S(\mathbf{y}) = \sum_{k=1}^{d} \|y_k\|_H^2$, where $\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), \ldots, y_d(\mathbf{x}))$ and $\|\cdot\|_H = (\cdot,\cdot)_H^{1/2}$ denotes a given norm or seminorm in a prescribed Hilbert space H. This naturally requires the components of the vector-valued function $\mathbf{y}$ to be an element of H. For an in-depth discussion of

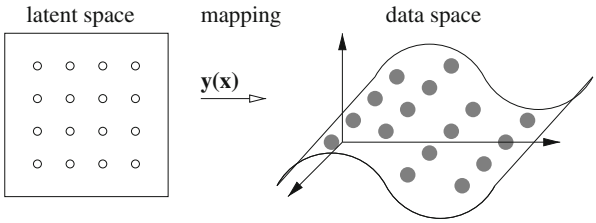

Fig. 2 The L-dimensional data space is mapped by $\mathbf{y}$ into the d-dimensional data space. There, the model assumes multivariate Gaussian noise with variance β^{-1}

the relation between regularization terms and associated function spaces, see [20]. A weak regularization with a too small λ or even $\lambda = 0$ leads to overfitting, i.e., the method models random noise instead of a meaningful underlying relationship between latent variables and the data set. A regularization that is too strong might prevent the method from discovering relevant features of the data. We recommend choosing the parameter λ for reconstruction and classification tasks by cross-validation techniques [7, 15].

3 Functional Minimization

Let us now show how the GTM functional $\mathscr{G}$ can be efficiently minimized even though it is nonlinear and nonconvex in $\mathbf{y}$ and β. It is important to note that the functional equals the logarithm of the partition function, and we can rearrange it to its free energy form for easier numerical treatment. First we define the posterior probabilities $R_{\mathbf{y},\beta} : \mathbb{R}^d \times [0,1]^L \to \mathbb{R}$ by

$$R_{\mathbf{y},\beta}(\mathbf{t},\mathbf{x}) := \frac{e^{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}')-\mathbf{t}\|^2} \mathrm{d}\mathbf{x}'} .$$

Next, we introduce the functional

$$\begin{aligned} \mathscr{K}(\psi,\mathbf{y},\beta) := &\int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t},\mathbf{x}) \log \psi(\mathbf{t},\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} \\ &+ \frac{\beta}{2} \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t},\mathbf{x}) \|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2 \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} - \frac{d}{2} \log \frac{\beta}{2\pi} + \lambda S(\mathbf{y}) . \end{aligned} \tag{3}$$

Here, for all $\mathbf{t} \in \mathbb{R}^d$ it must hold that $\psi(\mathbf{x},\mathbf{t})$ is a density in $\mathbf{x}$. Then, a lengthy, but otherwise simple calculation reveals that

$$\mathscr{K}(R_{\mathbf{y},\beta},\mathbf{y},\beta) = \mathscr{G}(\mathbf{y},\beta) \quad \text{for all} \quad \mathbf{y},\beta. \tag{4}$$

We now minimize $\mathscr{K}$ by successively minimizing with respect to its single parameters ψ, $\mathbf{y}$ and β. This is advantageous, because these subproblems are convex even though $\mathscr{G}$ is not.

The following three minimization steps have to be carried out in an outer iteration until we converge to a local minimum. Minimizing with respect to β yields

$$\arg\min_{\beta} \mathscr{K}(\psi,\mathbf{y},\beta) = \left(\frac{1}{d} \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t},\mathbf{x}) \|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2 \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} \right)^{-1} . \tag{5}$$

The posterior probabilities $R_{\mathbf{y},\beta}$ minimize $\mathscr{K}$ w.r.t. ψ, i.e.

$$\arg\min_{\psi} \mathscr{K}(\psi, \mathbf{y}, \beta) = R_{\mathbf{y},\beta} \ , \tag{6}$$

which is analogous to statistical physics, where the Boltzmann-distribution minimizes the free energy [18]. In combination with (4), this step can be understood as a projection back into the permissible search space since

$$\mathscr{K}(\arg\min_{\psi} \mathscr{K}(\psi, \mathbf{y}, \beta), \mathbf{y}, \beta) = \mathscr{K}(R_{\mathbf{y},\beta}, \mathbf{y}, \beta) = \mathscr{G}(\mathbf{y}, \beta) \ .$$

To minimize $\mathscr{K}$ in $\mathbf{y}$-direction, we need to solve the quadratic regression type problem

$$\arg\min_{\mathbf{y}} \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) \|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2 \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} + \frac{2\lambda}{\beta} S(\mathbf{y}) \ . \tag{7}$$

4 Discretization of the Model

We now discretize the mapping $\mathbf{y}$ by M basis functions $\phi_j : [0,1]^L \to \mathbb{R}, j = 1, \ldots, M$, and obtain $\mathbf{y}_M(\mathbf{x}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x})$ with the coefficient matrix $\mathbf{W} \in \mathbb{R}^{d \times M}$ and $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^T$. The minimization of the $\mathscr{K}$-functional in $\mathbf{y}$-direction (7) then amounts to solving d decoupled systems of linear equations for $r = 1, \ldots, d$

$$\mathbf{A}\mathbf{w}_r = \mathbf{z}_r \tag{8}$$

with $\mathbf{w}_r = ((\mathbf{W})_{r1}, \ldots, (\mathbf{W})_{rM})^T$, $\mathbf{A} \in \mathbb{R}^{M \times M}$ and $\mathbf{z}_r \in \mathbb{R}^M$. The entries of the matrix $\mathbf{A}$ and the vectors $\mathbf{z}_r$ can be computed for $i, j = 1, \ldots, M$ by

$$(\mathbf{A})_{ij} = \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) \phi_j(\mathbf{x}) \phi_i(\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} + \frac{2\lambda}{\beta} (\phi_i, \phi_j)_H \quad \text{and} \tag{9}$$

$$(\mathbf{z}_r)_i = \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) (\mathbf{t})_r \phi_i(\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} \ , \qquad r = 1, \ldots, d. \tag{10}$$

Note here that the derivation of our model in Sect. 2 started with the explicit knowledge of the continuous density $p(\mathbf{t})$. This is however in general not the case in most practical settings. There, rather an empirical density $p_N^{\text{emp}}(\mathbf{t})$ based on N data points $(\mathbf{t}_n)_{n=1}^N$ is given instead. Therefore, for the remainder of this paper, we furthermore replace the continuous density $p(\mathbf{t})$ by a sum of Dirac-delta-functions $p_N^{\text{emp}}(\mathbf{t}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{t}_n}(\mathbf{t})$. Then, the $\mathrm{d}\mathbf{t}$-integrals in (9) and (10) get replaced by sums, which corresponds to discretization by sampling.

4.1 Original GTM

Now, two further discretization steps can be taken. First, we choose L-variate Gaussians as the specific functions in the basis function vector $\phi : [0,1]^L \to \mathbb{R}^M$. Their centers lie on a uniform mesh in the L-dimensional latent space with mesh width h_1. Then $M = \mathcal{O}(h_1^{-L})$ and $h_1 = \mathcal{O}(M^{-\frac{1}{L}})$, respectively. Secondly, we choose a tensorized rectangle rule on a uniform mesh with width h_2 for the numerical quadrature of the d$\mathbf{x}$-integrals in (9) and (10), which results in $K = \mathcal{O}(h_2^{-L})$ quadrature points $(\mathbf{x}_i)_{m=1}^K$. This is equivalent to assuming a grid-based latent space distribution, as it is done in [4].

We obtain the resulting systems of linear equations (8), where now

$$(\mathbf{A})_{ij} = \frac{1}{NK} \sum_{n=1}^{N} \sum_{m=1}^{K} \psi(\mathbf{t}_n, \mathbf{x}_m) \phi_j(\mathbf{x}_m) \phi_i(\mathbf{x}_m) + \frac{2\lambda}{\beta} (\phi_i, \phi_j)_H \quad \text{and} \tag{11}$$

$$(\mathbf{z}_r)_i = \frac{1}{NK} \sum_{n=1}^{N} \sum_{m=1}^{K} \psi(\mathbf{t}_n, \mathbf{x}_m) (\mathbf{t}_n)_r \phi_i(\mathbf{x}_m) \,, \qquad r = 1, \ldots, d, \tag{12}$$

for $i, j = 1, \ldots, M$.

Note that in our successive minimization of $\mathcal{K}$, see Sect. 3, the minimization (6) with respect to ψ equals the E-Step and the minimization steps (5) and (7) with respect to β and $\mathbf{y}$ equal the M-Step of the well-known Expectation Maximization-algorithm [8]. In all steps, the discretized versions of $\mathbf{y}$ and the d$\mathbf{x}$-integrals now need to be employed. Altogether, we finally obtain the GTM [4], or, the other way around, we see that the original GTM is a special discretization of our generative model (1).

Note furthermore that the M degrees of freedom in the discretization and the K function evaluations for numerical quadrature have both an exponential dependence on the embedding dimension L. This severely limits the GTM to the cases $L \leq 3$. To overcome this issue, we will choose some other type of discretization of our generative model in the following.

4.2 Sparse GTM

We now suggest to use a sparse grid discretization [6] for the components of the mapping $\mathbf{y}$ instead of a uniform, full mesh. We denote the resulting numerical method as the sparse GTM. To explain our new approach in detail, let us first consider a one-dimensional level-wise sequence of conventional sets of piecewise linear basis functions on the interval $[0,1]$. There, the space V_l on level $l \geq 0$ contains $n_l = 2^l + 1$ hat functions $\phi_{l,i} : [0,1] \to \mathbb{R}$

$$\phi_{l,i}(x) = \max(1 - 2^l \, |x - x_{l,i}| \, , 0) \,,$$

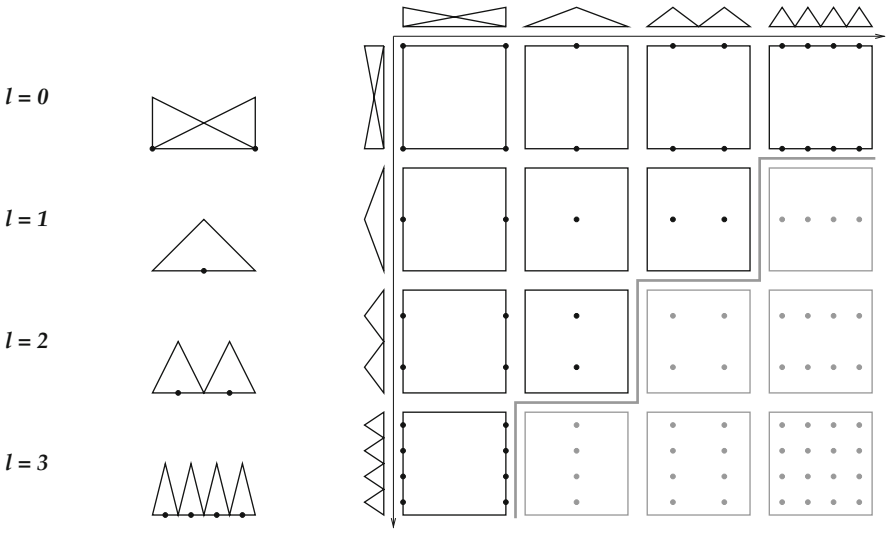

Fig. 3 The first four hierarchical surplus spaces of the one-dimensional hierarchical basis (*left*). Two-dimensional tensorization and the sparse subspace (*right*)

which are centered at the points of an equidistant mesh $x_{l,i} = 2^{-l}i, i = 0, \ldots, n_l - 1$. Next, we consider the hierarchical surplus spaces W_l, where $V_{l+1} = V_l \oplus W_{l+1}$, see also the left-hand side of Fig. 3. They can be easily constructed by

$$W_l = \text{span}\{\phi_{l,i} : i \in \xi_l\} \quad \text{with} \quad \xi_l := \begin{cases} \{0, 1\} & \text{for } l = 0 \\ \{i \text{ odd}, 1 \leq i \leq 2^l - 1\} & \text{else.} \end{cases}$$

With the multi-indices $\mathbf{l} = (l_1, \ldots, l_L) \in \mathbb{N}^L$, $\mathbf{i} = (i_1, \ldots, i_L) \in \mathbb{N}^L$, the d-variate functions $\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) = \phi_{l_1,i_1}(x_1) \cdots \phi_{l_L,i_L}(x_L)$ and the Cartesian products $\xi_{\mathbf{l}} := \times_{s=1}^{L} \xi_{l_s}$, we obtain L-dimensional spaces $W_{\mathbf{l}} = \text{span}\{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{i} \in \xi_{\mathbf{l}}\}$. Then,

$$V_J^{(\infty)} := \bigoplus_{|\mathbf{l}|_\infty \leq J} W_{\mathbf{l}} = \bigotimes_{s=1}^{L} \bigoplus_{l_s=0}^{J} W_{l_i} = \bigotimes_{s=1}^{L} V_J$$

resembles just a normal isotropic full grid (FG) space up to level J, while

$$V_J^{(1)} := \bigoplus_{|\mathbf{l}|_1 \leq J+d-1} W_{\mathbf{l}} \tag{13}$$

denotes the sparse grid (SG) space[1] on level J. The former has $M^{\mathrm{FG}} = \mathcal{O}(2^{LJ})$ degrees of freedom, while the latter has only $M^{\mathrm{SG}} = \mathcal{O}(J^{L-1}2^J)$ degrees of freedom. However, under the assumption of bounded mixed derivatives, both discretizations have essentially the same L_2-error convergence rate, see [6, 9] for further analysis and implementational issues. The use of this kind of discretization for every component of the vector-valued mapping $\mathbf{y}$, i.e. $\mathbf{y}^{\mathrm{SG}} = (y_1^{\mathrm{SG}}, \ldots, y_d^{\mathrm{SG}})$ with $y_r^{SG} \in V_J^{(1)}, r = 1, \ldots, d$, then yields a sparse GTM.

The corresponding L-dimensional integration problems (9) and (10) for setting up the associated systems of linear equations (8) are approximated by evaluation points $\mathbf{x}_m$ with fixed weights γ_m for $m = 1, \ldots, K$. Here, methods like Quasi Monte-Carlo or sparse grid quadrature [11] can be used. Then, K does not exhibit the 'curse of dimensionality' with respect to L.

We obtain the resulting systems of linear equations (8), where now

$$(\mathbf{A})_{\mathbf{l},\mathbf{i},\mathbf{k},\mathbf{j}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{K}\gamma_m\psi(\mathbf{t}_n,\mathbf{x}_m)\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}_m)\phi_{\mathbf{j},\mathbf{k}}(\mathbf{x}_m) + \frac{2\lambda}{\beta}(\phi_{\mathbf{l},\mathbf{i}},\phi_{\mathbf{k},\mathbf{j}})_H \quad \text{and} \tag{14}$$

$$(\mathbf{z}_r)_{\mathbf{l},\mathbf{i}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{K}\gamma_m\psi(\mathbf{t}_n,\mathbf{x}_m)(\mathbf{t}_n)_r\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}_m)\,, \qquad r = 1,\ldots,d, \tag{15}$$

with $|\mathbf{l}|_1, |\mathbf{k}|_1 \le J + d - 1, \mathbf{i} \in \xi_{\mathbf{l}}, \mathbf{j} \in \xi_{\mathbf{k}}$.

When we minimize the functional $\mathscr{K}$ in $\mathbf{y}$-direction the systems (8) have to be solved. As recommended in [4], we use a direct method. An LU factorization of the matrix $\mathbf{A}$ costs $\mathcal{O}\big((M^{SG})^3\big)$. Then, the forward and backward substitution steps for d different right-hand sides of (8) cost $\mathcal{O}\big(d \cdot (M^{SG})^2\big)$. For high-dimensional data sets with $d > M^{SG}$, these steps can be more relevant cost-wise than the initial factorization of $\mathbf{A}$.

It is also possible to solve the system (8) for each right-hand side by an iterative method. Then the costs are $\mathcal{O}(d \cdot \#it \cdot X)$, where $\#it$ denotes the number of necessary iteration steps and X is the cost of one matrix-vector multiplication. Typically, the unidirectional principle [2, 5] is used for the fast multiplication with sparse grid operator matrices, but this algorithm is not applicable here since the function ψ in (14) does not allow a product representation in $\mathbf{x}$. However, in contrast to the Original GTM from Sect. 4.1, our sparse GTM results in a somewhat sparse matrix $\mathbf{A}$. This can be exploited in the matrix vector multiplication of $\mathbf{A}$. Note that the regularization term $\frac{2\lambda}{\beta}(\cdot,\cdot)_H$ prevents the matrix $\mathbf{A}$ from being severely ill-conditioned. Here, however, keeping $\#it$ low and bounded independently of the discretization level J is a matter of preconditioning the matrix (14), which is nontrivial and future work. Since we presently cannot guarantee that the costs

[1] We can replace $|\mathbf{l}|_1$ by $|\mathbf{l}|_1 + |\{s : l_s = 0\}|$ in (13), which leads to a slightly different treatment of boundary functions, but has otherwise the same asymptotic properties, see [9].

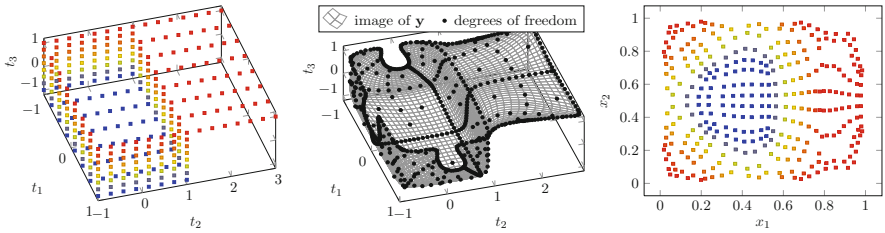

Fig. 4 The three-dimensional 'open box' (*left*), a sparse GTM fitted to this dataset (*middle*) and the two-dimensional projection of the data points (*right*)

$\mathcal{O}(d \cdot \#it \cdot X)$ are lower than $\mathcal{O}\left(d \cdot (M^{SG})^2\right)$ for the direct method in high-dimensions, we decided to stick with the LU factorization for now.

To demonstrate the nonlinear quality of the method, we apply the sparse GTM to the 'open box' benchmark dataset [17] in Fig. 4. We see a reasonable unfolding of the box in the two-dimensional embedding, which would not be possible with linear methods, like e.g. a conventional PCA.

5 Numerical Experiments

In this section, we will now present the results for the sparse GTM for some problems from dimensionality reduction and data classification.

5.1 Dimensionality Reduction

On the left-hand side of Fig. 5, we present a toy example with data points stemming from a wave-shaped manifold. Since we here have a sufficiently large amount of data points, we need no regularization term. We measure the GTM functional value $\mathscr{G}(\mathbf{y}, \beta)$, see (2), after 5 minimization cycles for $\mathscr{K}$, see (3). On the right-hand side, we see that the sparse GTM achieves about the same reduction in the GTM functional value with substantially less degrees of freedom than the GTM based on a full grid.

Next, we consider a real-world problem. Figure 6 shows a three-dimensional projection of a 12-dimensional data set. It consists of 1,000 data points with diagnostic measurements of oil flows along a multi-phase pipeline. The three different class types in the plot represent stratified, annular and homogeneous multi-phase configurations, compare [3] for further details. In [4], it was shown how a two-dimensional embedding of the data with the GTM gives an improved separation of the clusters compared to the embedding with the PCA. We now run this experiment with a sparse GTM with $L = 2$ and $L = 3$, discretization level

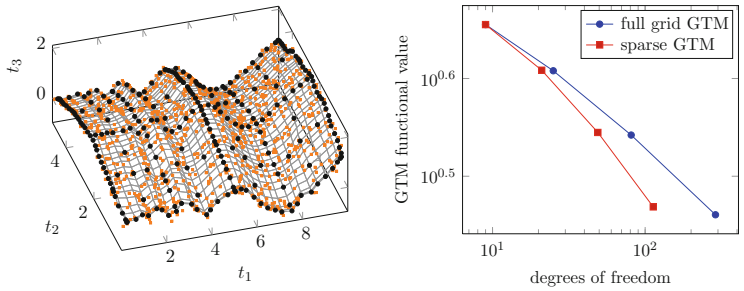

Fig. 5 Reduction in the GTM functional value with respect to the degrees of freedom per **y**-component for a GTM with a full grid discretization and the sparse GTM

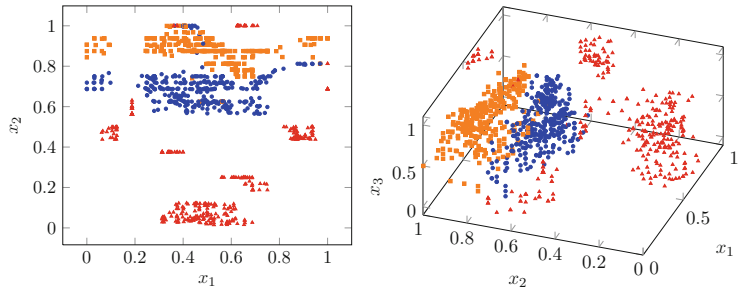

Fig. 6 Embedding of a 12-dimensional data set with three class labels by the sparse GTM in two dimensions (*left*) and three dimensions (*right*)

$J = 4$, H^1_{mix}-seminorm regularization and $\lambda = 4.0 \times 10^{-3}$. We see that the three-dimensional latent space offers an even more detailed picture of the data than the two-dimensional embedding and a slightly better separation of the different clusters.

5.2 Classification

We now use the sparse GTM for classification. To this end, we append a class variable $c_n \in \{-1, 1\}$ to the data points by

$$\mathbf{t}'_n := ((\mathbf{t}_n)_1, \ldots, (\mathbf{t}_n)_d, c_n)^T \quad \text{for} \quad n = 1, \ldots, N \ . \tag{16}$$

We first use the sparse GTM to fit the mapping $\mathbf{y}$ and the inverse variance β to these points. Then, we can classify new data points with help of the density $q_{\mathbf{y},\beta}$ of (1) by

$$c(\mathbf{t}) := \begin{cases} 1 & \text{if } q_{\mathbf{y},\beta}(t_1, \ldots, t_d, 1) \geq q_{\mathbf{y},\beta}(t_1, \ldots, t_d, -1) \\ -1 & \text{else.} \end{cases}$$

We apply this technique to 'Connectionist Bench (Sonar, Mines vs. Rocks)', a real-world data set from the UCI Machine Learning Repository [1]. It consists of approximately 200 measurements with 60 dimensions and two class labels.

In [12], this data was randomly split into two parts. One part was used to train various neuronal networks, the other one was used to measure the quality of the model. The best neuronal network achieved an average classification rate of 84.7 %.

We use our sparse GTM with latent space dimensions $L = 2$ and $L = 3$ and a regularization term based on the H^1_{mix}-seminorm. We achieve classification rates between 72.0 and 84.6 % already for $L = 2$, depending on the regularization parameter λ and the discretization level J. For $L = 3$, $J = 3$ and $\lambda = 1.0 \times 10^{-4}$, we even reach a classification rate of 85.6 %, which clearly shows the potential of our new approach.

6 Conclusions

We presented a generative model that can be used for dimensionality reduction and classification of high-dimensional data. For a certain choice of discretization involving uniform grids, we obtained the original generative topographic mapping from [4]. Using a sparse grid discretization for the mapping, we obtained our new sparse GTM. It gives about the same quality with less degrees of freedom. Moreover, it has the perspective to overcome complexity issues of the grid-like structures, which limit the conventional GTM to a low number of latent space dimensions. For example, in dimension $L = 4$ and discretization level $J = 5$ the sparse grid approach with index set $\{\mathbf{l} : |\mathbf{l}|_1 + |\{s : l_s = 0\}| \leq J + d - 1\}$ has 7,681 degrees of freedom, which is still treatable using a direct solver, whereas the full grid already has 1,185,921 degrees of freedom. For dimensions like $L = 10$ the situation is as follows: Full grids with $L = 10$ and $J = 4$ have $2.0 \cdot 10^{12}$ degrees of freedom ($5.8 \cdot 10^{11}$ inner functions and $1.4 \cdot 10^{12}$ boundary functions), which is clearly beyond the capabilities of current computers. Sparse grids have $1.1 \cdot 10^7$ degrees of freedom, of which only 2,001 are inner functions and 10,817,088 are boundary functions. Of course, this is still too much for a direct solver, but now only the number of boundary functions poses a bottleneck. Modified boundary functions with improved properties can be found in [9, 19], so there is some hope to treat higher dimensional latent spaces. Furthermore, note that the runtime complexity depends only linearly on the data space dimension d and the number of data points N. This makes the sparse GTM a suitable tool for high-dimensional data sets. For further experiments and results, cf. [13, 14].

References

1. Bache, K., Lichman, M.: UCI Machine Learning Repository. http://archive.ics.uci.edu/ml (2012)
2. Balder, R., Zenger, C.: The solution of multidimensional real Helmholtz equations on sparse grids. SIAM J. Sci. Comput. **17**, 631–646 (1996)
3. Bishop, C., James, G.: Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detect. Assoc. Equip. **327**(2–3), 580–593 (1993)
4. Bishop, C., Svensen, M., Williams, C.: GTM: the generative topographic mapping. Neural Comput. **10**(1), 215–234 (1998)
5. Bungartz, H.: Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung. Dissertation, Fakultät für Informatik, Technische Universität München (1992)
6. Bungartz, H., Griebel, M.: Sparse grids. Acta Numer. **13**, 1–123 (2004)
7. Craven, P., Wahba, G.: Smoothing noisy data with spline functions. Numer. Math. **31**(4), 377–403 (1978)
8. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. **39**, 1–38 (1977)
9. Feuersänger, C.: Sparse Grid Methods for Higher Dimensional Approximation. Südwestdeutscher Verlag für Hochschulschriften AG & Company KG, Saarbrücken (2010)
10. Feuersänger, C., Griebel, M.: Principal manifold learning by sparse grids. Computing **85**(4), 267–299 (2009)
11. Gerstner, T., Griebel, M.: Dimension–adaptive tensor–product quadrature. Computing, **71**(1), 65–87 (2003)
12. Gorman, R., Sejnowski, T.: Analysis of hidden units in a layered network trained to classify sonar targets. Neural Netw. **1**, 75 (1988)
13. Griebel, M., Hullmann, A.: Dimensionality reduction of high-dimensional data with a nonlinear principal component aligned generative topographic mapping. SIAM J. Sci. Comput. **36**(3), A1027–A1047 (2014)
14. Hullmann, A.: Schnelle varianten des generative topographic mapping. Diploma thesis, Institute for Numerical Simulation, University of Bonn (2009)
15. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence – Volume 2 (IJCAI'95), San Francisco, pp. 1137–1143. Morgan Kaufmann (1995)
16. Kullback, S.: Information Theory and Statistics. Wiley, New York (1959)
17. Lee, J., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, New York/London (2007)
18. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in Graphical Models, pp. 355–368. Kluwer Academic, Dordrecht/Boston (1998)
19. Pflüger, D., Peherstorfer, B., Bungartz, H.: Spatially adaptive sparse grids for high-dimensional data-driven problems. J. Complex. **26**(5), 508–522 (2010)
20. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT, Cambridge (2001)

Sparse Approximation Algorithms for High Dimensional Parametric Initial Value Problems

Markus Hansen, Claudia Schillings, and Christoph Schwab

Abstract We consider the efficient numerical approximation for parametric nonlinear systems of initial value Ordinary Differential Equations (ODEs) on Banach state spaces $\mathcal{S}$ over $\mathbb{R}$ or $\mathbb{C}$. We assume the right hand side depends *analytically* on a vector $y = (y_j)_{j\geq 1}$ of possibly countably many parameters, normalized such that $|y_j| \leq 1$. Such affine parameter dependence of the ODE arises, among others, in mass action models in computational biology and in stoichiometry with uncertain reaction rate constants. We review results by the authors on N-term approximation rates for the parametric solutions, i.e. summability theorems for coefficient sequences of generalized polynomial chaos (gpc) expansions of the parametric solutions $\{X(\cdot; y)\}_{y\in U}$ with respect to tensorized polynomial bases of $L^2(U)$. We give sufficient conditions on the ODEs for N-term truncations of these expansions to converge on the entire parameter space with efficiency (i.e. accuracy versus complexity) being independent of the number of parameters viz. the dimension of the parameter space U. We investigate a heuristic adaptive approach for computing sparse, approximate representations of the $\{X(t; y) : 0 \leq t \leq T\} \subset \mathcal{S}$. We increase efficiency by relating the accuracy of the adaptive initial value ODE solver to the estimated detail operator in the Smolyak formula. We also report tests which indicate that the proposed algorithms and the analyticity results hold for more general, nonaffine analytic dependence on parameters.

1 Introduction

Numerous systems in engineering and life- and in social sciences are modelled by initial value ordinary differential equations (ODEs). In particular, complex systems require state spaces $\mathcal{S}$ of high or even infinite dimension.

In recent years, in particular in connection with applications in life-sciences, climate-sciences but also in economics, particular attention has been paid to *initial value ODE models for systems with uncertainty*. We mention only stoichiometric

M. Hansen • C. Schillings • C. Schwab (✉)
Seminar for Applied Mathematics, ETH Zürich, 8092 Zürich, Switzerland
e-mail: markus.hansen@sam.math.ethz.ch; claudia.schillings@sam.math.ethz.ch; christoph.schwab@sam.math.ethz.ch

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_6

descriptions of biochemical reaction pathways with uncertain reaction rate constants, chemical reaction cascades with uncertain reaction rate constants, mass action models with uncertain reaction rates. In complex systems, the goal of computation is in *obtaining the system characteristics on the entire parameter space in one single numerical forward simulation*. Besides the efficient *numerical forward solution* of parametric initial value ODEs by combination of *adaptive parameter collocation approaches* with *adaptive numerical initial value solvers* such as [15,16] and the references there, additional problems consist in optimization resp. in optimal control of systems described by initial value ODEs.

Some form of *Sparsity* in the parametric dependence of the solution (resp. the control resp. the optimum) is necessary in order to allow for efficient approximations of the parametric solutions on the entire, possibly high-dimensional parameter space. Here, we present theoretical results from [17] on the sparsity of solutions of parametric ODEs and propose computational approaches which allow to exploit computationally the sparse parameter dependence of the solutions.

Unless stated otherwise, the state space $\mathcal{S}$ is assumed to be a separable, reflexive Banach space, and will be understood over the coefficient field $\mathbb{R}$; occasionally, however, we shall also work with the extension of $\mathcal{S}$ to the coefficient field $\mathbb{C}$. By $\mathbb{R}^{\mathbb{N}}$ and $\mathbb{C}^{\mathbb{N}}$, we denote the countable cartesian products of $\mathbb{R}$ and $\mathbb{C}$, respectively. Likewise, $U = (-1,1)^{\mathbb{N}}$ will denote the countable product of the open interval $(-1,1)$ and $\overline{U} = [-1,1]^{\mathbb{N}}$. We shall denote the state of the system by $X(t) \in \mathcal{S}$ for $t \in [0,T]$. The parameter dependence of X on the parameter sequence $y \in U$ is indicated by $X(t;y)$.

On the parameter domain U, we consider *high-dimensional, parametric, deterministic ODE initial value problems (ODE-IVP)*:
Given $x_0(y) \in \mathcal{S}$ and $T \in (0,\infty)$, find $X(t,x_0;y) : [t_0,T] \times \mathcal{S} \times U \to \mathcal{S}$ such that in $\mathcal{S}$

$$\frac{\mathrm{d}X}{\mathrm{d}t} = f(t,X;y)\,, \quad X(t_0;y) = x_0(y)\,, \quad t_0 \le t \le T\,, \quad \forall y \in U\,. \tag{1}$$

Here, $\mathcal{S}$ denotes the *state space* of the parametric model (1). We shall mostly be concerned with the case of initial value ordinary differential equations (ODEs), when $\mathcal{S} = \mathbb{R}^d$, with particular attention to the case of high or even infinite dimensional state spaces, i.e. $\mathbb{R}^d$ with large d, but [17] covers also the infinite dimensional case, when $\mathcal{S}$ *is a separable and reflexive Banach space*.

The parameter dependence of X on $y \in U$ is indicated by $X(t;y)$. We denote $\mathbb{N} = \{1,2,\ldots\}$ and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. We use standard multiindex notation: for a sequence $y = (y_j)_{j\ge1}$ of parameters and for a sequence $\nu \in \mathbb{N}_0^{\mathbb{N}}$ of nonnegative integers, we denote by $\mathfrak{F} = \{\nu \in \mathbb{N}_0^{\mathbb{N}} : |\nu| < \infty\}$. As any $\nu \in \mathfrak{F}$ has only finitely many nonzero entries, the definitions

$$\nu! = \prod_{j\in\mathbb{N}} \nu_j!\,, \quad |\nu| = \sum_{j\in\mathbb{N}} \nu_j\,, \quad \partial_y^\nu = \frac{\partial^{|\nu|}}{\partial y_1^{\nu_1} \partial y_2^{\nu_2} \cdots}$$

for multi-factorials, the length of a multi-index ν and for the partial derivative of order ν are well-defined for $\nu \in \mathfrak{F}$.

In practice efficient solution methods in the case where the number of parameters is large are of interest. In particular, it would be highly desirable to identify methods which are *dimensionally robust*, i.e. whose efficiency (meaning accuracy versus computational cost measured in terms of the total number of floating point operations to achieve this accuracy) is *provably robust with respect to the number of parameters* which requires consideration of (1) for *parameter sequences*. In [17] we showed, analogously to earlier results for linear, elliptic partial differential equations [8, 10, 11, 18] sparsity of the parametric solutions' dependence on y.

It is well-known and classical (e.g. [20, Chap. 13]) that for parametric right hand sides $f(t, X; y)$ which are Lipschitz continuous with respect to (t, X) and which depend *analytically* on the parameters y, the solution $X(t; y)$ in turn depends analytically on the parameter vector y. In [17], we extended the proof in [20] of this (classical) result to a possibly countable number of parameters with *quantitative bounds on the size of domains of analyticity*. This allows us to establish in [17] *best N-term convergence rates for parametric expansions of the solution $X(t; y)$* under a *sparsity hypothesis* on the vector field $f(t, X; y)$. The rates of best N-term approximation will be shown to be achievable with *N-term truncated Taylor expansions* of the solution $X(t, y)$ in the parameter space U which we prove to converge uniformly for all y belonging to the parameter domain U. The key mathematical principle behind these results is the fact that *sparsity in the input vector field $f(t, X; y)$ implies sparsity (in a sense to be made precise below) in the parametric solution's Taylor expansion*

$$X(t; y) = \sum_{\nu \in \mathfrak{F}} T_\nu(t) y^\nu \ , \quad T_\nu(t) := \frac{1}{\nu!} (\partial_y^\nu X(t; y))|_{y=0} \ , \quad t_0 \le t \le T \ , \quad y \in U \ . \tag{2}$$

In [17], similar results are also established for other polynomial expansions of the solution, such as Legendre or Chebyshev expansions.

The theoretical result on sparse parameter dependence in [17] opens the perspective of dimensionally robust, adaptive algorithms for the efficient solution of large systems of parametric ODE's on possibly infinitely dimensional parameter spaces. This requires to address the following issues: first, under the *(unrealistic) assumption of having available exact solutions of the ODE-IVP* (1) *for a single instance of the parameter vector $y \in U$ at unit cost*, concrete sequences of sparse, finite, *monotone* index sets $\mathcal{M}_N \subset \mathfrak{F}$ (to which we will also refer as "sparsity models") for at most N "active" Taylor coefficients $T_\nu(t)$, $\nu \in \mathcal{M}_N$, can be constructed such that the corresponding, finitely truncated parametric expansions

$$X_{\mathcal{M}_N}(t; y) = \sum_{\nu \in \mathcal{M}_N} T_\nu(t) y^\nu \tag{3}$$

realize the best N-term asymptotic convergence rate.

One particular class of sparsity models are *monotone index sets* $\Lambda \subset \mathfrak{F}$ which were introduced in [8] in the context of adaptive Taylor approximations of parametric elliptic partial differential equations. This notion is based on the following ordering of $\mathfrak{F}$: for any two indices $\mu, \nu \in \mathfrak{F}$, we say that $\mu \leq \nu$ if and only if $\mu_j \leq \nu_j$ for all $j \geq 1$. We will also say that $\mu < \nu$ if and only if $\mu \leq \nu$ and $\mu_j < \nu_j$ for at least one value of j.

Definition 1. A sequence $(a_\nu)_{\nu \in \mathfrak{F}}$ of nonnegative real numbers is said to be *monotone decreasing* if and only if for all $\mu, \nu \in \mathfrak{F}$

$$\mu \leq \nu \Rightarrow a_\nu \leq a_\mu \, .$$

A set $\emptyset \neq \Lambda \subset \mathfrak{F}$ is called *monotone* if and only if $\nu \in \Lambda$ and $\mu \leq \nu \Rightarrow \mu \in \Lambda$.

Once concrete, finite, monotone $\mathcal{M}_N$ sparsity models have been selected, the evaluation of the truncations (3) requires approximation of the expansion coefficients $T_\nu(t)$ in (2) for $\nu \in \mathcal{M}_N$. Naturally, the assumption of an exact solution of the ODE-IVP (1) for a single instance of the parameter vector y in $O(1)$ work and memory is not realistic. Thus to still achieve the rate of best N-term approximation also for the approximate partial sums

$$\tilde{X}_{\mathcal{M}_N}(t; y) = \sum_{\nu \in \mathcal{M}_N} \tilde{T}_\nu(t) y^\nu \, , \tag{4}$$

where $\tilde{T}_\nu(t) \in \mathcal{S}$ are the Taylor coefficients obtained with an approximate initial value ODE solver, the effort for computing the coefficients has to be balanced against the respective impact for approximating $X(t; y)$ by an initial value ODE solver.

In doing so, we obtain an approximate, adaptive numerical solution of the parametric ODE-IVP (1) to a prescribed accuracy ε *uniformly on the entire parameter domain* U. This ultimately enables us to approximately calculate all further relevant information about the parametric solution (e.g. statistical moments), again up to an arbitrary prescribed accuracy, by several classes of adaptive approximation algorithms based on Galerkin projection (see, e.g. [14]) or by sparse collocation as in [3,4,19] or by adaptive truncation (4) of the Taylor expansions (2) as in [8]. We let B denote a separable Banach space both over $\mathbb{R}$ as well as its complexification over $\mathbb{C}$ (i.e. an extension of B whose restriction to real valued elements coincides with the original space B). We shall need spaces of (differentiable) functions with values in B. We denote by $C(U; B) \equiv C^0(U; B)$ the space of functions from U into B which are, as B-valued functions, continuous on U (where U is equipped with the product topology). Moreover, for any $k \in \mathbb{N}$, we denote by $C^k([0, T]; B)$ the space of functions $f : [0, T] \to B$ whose k-th Fréchet derivative $\frac{\mathrm{d}^k f}{\mathrm{d}t^k}$ with respect to $t \in [0, T]$ belongs to $C^0([0, T]; B)$. These spaces $C^k([0, T]; B)$, equipped with the norms

$$\| f \|_{C^k([0,T];B)} := \max_{0 \leq j \leq k} \left\{ \| \tfrac{\mathrm{d}^j f}{\mathrm{d}t^j} f \|_{C^0([0,T];B)} \right\} \, , \quad k \in \mathbb{N} \, , \tag{5}$$

are themselves separable Banach spaces. Similar notations are used, if the interval $[0, T]$ is itself replaced by another Banach space $\mathcal{S}$. Then the derivatives $\frac{\mathrm{d}f}{\mathrm{d}x}$ have to be understood as Fréchet derivatives, i.e. $\frac{\mathrm{d}f}{\mathrm{d}x}$ is a mapping from $\mathcal{S}$ taking values in $\mathcal{L}(\mathcal{S}, B)$, the space of bounded linear operators from $\mathcal{S}$ into B. Finally, spaces of locally Lipschitz continuous functions will be defined below.

2 Parametric Initial Value ODEs

For a parameter sequence $y = (y_j)_{j\geq 1} \in U$ and a Banach state space $\mathcal{S}$, we assume given an initial state $x_0(y) \in \mathcal{S}$ and a parametric family of vector fields $f(t, X; y) : [0, T] \times \mathcal{S} \times U \mapsto \mathcal{S}$. Then we are interested in solving (1) numerically to a prescribed tolerance *uniformly for all values* $y \in U$.

As we think of applications to large mass-action models in computational chemistry and biology, attention will be in the following on the particular *case when the dependence of the vector field* f *in* (1) *on the parameter vector* $y \in U$ *is affine*, i.e. for every $t \in [0, T]$ and every $X \in \mathcal{S}$,

$$f(t, X; y) = f_0(t, X) + \sum_{j\geq 1} y_j f_j(t, X) , \quad 0 \leq t \leq T < \infty . \tag{6}$$

Here, we assume that each $f_j \in (f_j)_{j\geq 0}$ is continuous with respect to t and satisfies certain Lipschitz conditions with respect to X uniform in $t \in [0, T]$. For the non-parametric problem $\frac{dX}{dt} = g(t, X)$, $X(t_0) = x_0$, it is classical that the right-hand-side g being locally Lipschitz continuous, i.e. for every $X_0 \in \mathcal{S}$ there is a neighbourhood $U = U(X_0)$ such that

$$\forall X, X' \in U \quad \forall t \in [0, T] : \quad \|g(t, X) - g(t, X')\|_{\mathcal{S}} \leq L(X_0)\|X - X'\|_{\mathcal{S}} \tag{7}$$

for some constants $L(X_0)$, implies existence and uniqueness of local solutions, i.e. existence of unique solutions on some maximally extended subinterval $[0, \delta) \subset [0, T]$, see e.g. [12]. To obtain global, parametric solutions we imposed in [17] a local Lipschitz condition: for every $R > 0$, there exist constants $L(R) > 0$ such that for every $X, X' \in B_R = \{X \in \mathcal{S} : \|X\|_{\mathcal{S}} \leq R\}$ and for every $t \in [0, T]$ holds

$$\|g(t, X) - g(t, X')\|_{\mathcal{S}} \leq L(R)\|X - X'\|_{\mathcal{S}} ,$$

where

$$L(R) := \|g\|_{\ell\mathrm{Lip}(\mathcal{S},R)} = \sup_{t\in[0,T], X\neq X'\in B_R} \frac{\|g(t, X) - g(t, X')\|_{\mathcal{S}}}{\|X - X'\|_{\mathcal{S}}} < \infty .$$

A continuous function g belongs to $\ell\mathrm{Lip}(\mathcal{S})$, if $L(R) < \infty$ for all $R > 0$. The subclass $\ell\mathrm{Lip}_0(\mathcal{S})$ consists of all functions $g \in \ell\mathrm{Lip}(\mathcal{S})$ which additionally fulfill $g(t,0) = 0$ for all $t \in [0,T]$. Then $\ell\mathrm{Lip}_0(\mathcal{S})$ equipped with the increasing family of norms $\|\cdot\|_{\ell\mathrm{Lip}(\mathcal{S},\mathrm{R})}$ becomes a complete locally convex vector space. Our main assumption on (6) is $f_j \in \ell\mathrm{Lip}_0(\mathcal{S})$ for all j, i.e. for $j = 0, 1, 2, \ldots$ holds

$$L_j(R) = \sup_{t\in[0,T], X\neq X'\in B_R} \frac{\|f_j(t,X) - f_j(t,X')\|_{\mathcal{S}}}{\|X - X'\|_{\mathcal{S}}} < \infty\,, \quad f_j(t,0) = 0\,. \tag{8}$$

In order to prove results which are independent of the number of terms in the affine expansion (6), we shall further require *summability of the coefficient sequence* $(f_j)_{j\geq 1}$. Specifically, we assume the sequence of Lipschitz constants to be summable, i.e.

$$\forall R > 0: \quad \big(L_j(R)\big)_{j\geq 1} \in \ell^1(\mathbb{N})\,. \tag{9}$$

Under this assumption, the sum in (6) converges uniformly with respect to $y \in U$ and for all $(t,X) \in [0,T]\times\mathcal{S}$. In [17], we showed

Proposition 1. *Let the conditions (8) and (9) be satisfied. Then the sum in (6) converges absolutely and uniformly in U as a $\ell\mathrm{Lip}_0(\mathcal{S})$-valued mapping.*

Moreover, we may also consider $\mathcal{S}$ to be a complex Banach space: besides being of independent interest, the proofs in [17] used analytic continuations and complex variable techniques even for problems with real-valued parameter sequences $y \in U$. In [17] we showed

Theorem 1. *Assume (8) and (9). Moreover, suppose the initial condition $x_0 \in C(\mathcal{U},\mathcal{S})$ satisfies*

$$\sup_{z\in\mathcal{U}} \|x_0(z)\|_{\mathcal{S}} \leq (1-\kappa)r\,, \qquad r = Re^{-L(R)T/\kappa} \tag{10}$$

for some $R > 0$ and $0 < \kappa < 1$, where $\mathcal{U} = \{\zeta \in \mathbb{C} : |\zeta| < 1\}^{\mathbb{N}}$.

Then the IVP (1) (with $t_0 = 0$) admits a unique solution $X \in \mathcal{B}^1_{r,R} \subset C^1([0,T]; C(\mathcal{U};\mathcal{S}))$, where

$$\mathcal{B}^1_{r,R} = \big\{Y \in C^1([0,T]; C(\mathcal{U};\mathcal{S})) : \sup_{(t,z)\in[0,T]\times\mathcal{U}} \mathrm{e}^{-tL(R)/\kappa}\|Y(t,z)\|_{\mathcal{S}} \leq r\big\}$$

If, in addition, for some $k \in \mathbb{N}$

$$\forall j \geq 0: \quad f_j : [0,T]\times\mathcal{S} \longrightarrow \mathcal{S} \text{ is } k\text{-times continuously differentiable}\,, \tag{11}$$

then for every $z \in \mathcal{U}$ the unique solution $X(\cdot, x_0(z); z)$ of (1) belongs to $C^{k+1}([0,T];\mathcal{S})$.

Moreover, the solution $X(\cdot, x_0; z)$ depends continuously on the data x_0 and parameters z. If, additionally, the functions f_j are analytic as $C^k([0,T];\mathcal{S})$-valued mappings, then X is analytic on $\mathcal{U}$ as a $C^{k+1}([0,T];\mathcal{S})$-valued mapping.

3 Sparsity

It was shown in [17] that if the sequence f_j in (1) is sparse in the sense that if $(\|f_j\|_{\ell\mathrm{Lip}_0(\mathcal{S},\mathrm{R})})_{j\geq 1} \in \ell^p(\mathbb{N})$ for all $R > 0$ for some $0 < p < 1$, then the sequence $(T_\nu)_{\nu\in\mathfrak{F}}$ of Taylor coefficients of the solution is equally sparse.

Theorem 2. *Consider the parametric IVP ODE (1) for parameter vectors $y \in U = [-1,1]^{\mathbb{N}}$. If there exist real numbers $R > 0$ and $0 < \kappa < 1$ with the following properties:*

1. *In (1) the vector field f depends on the parameter vector y in the affine fashion (6) with the coefficient functions f_j satisfying for some $0 < p < 1$*

$$\left(\|f_j\|_{\ell\mathrm{Lip}_0(\mathcal{S},\mathrm{R})}\right)_{j\geq 1} \in \ell^p(\mathbb{N}) \qquad \text{and} \qquad \left(\overline{\rho}_j \|f_j\|_{\ell\mathrm{Lip}_0(\mathcal{S},\mathrm{R})}\right)_{j\geq 1} \in \ell^1(\mathbb{N}) \ , \tag{12}$$

   where the scaling vector $\overline{\rho}$ is given by $\overline{\rho}_j = \max\left(1, \frac{\delta}{4L_j(R)}\right)$ for some arbitrary fixed $\delta > 0$, and $L_j(R) := \|f_j\|_{\ell\mathrm{Lip}_0(\mathcal{S},\mathrm{R})}$.
2. *The initial data $x_0 \in C([0,T]\times U_{\overline{\rho}};\mathcal{S})$ satisfies*

$$\sup_{z\in U_{\overline{\rho}}} \|x_0(z)\| \leq (1-\kappa) R \mathrm{e}^{-TL(\overline{\rho},R)/\kappa} \ . \tag{13}$$

Then the Taylor expansion (2) of the parametric solution $X(t;y)$ of (1) is p-sparse in the following sense: for every $N \in \mathbb{N}$, there exists a finite, monotone set $\Lambda_N \subset \mathfrak{F}$ of indices $\nu \in \mathfrak{F}$ corresponding to N Taylor coefficients T_ν with largest norm in $C_{L(\overline{\rho},R)/\kappa}([0,T];\mathcal{S})$ such that it holds

$$\sup_{y\in U} \left\| X(\cdot;y) - \sum_{\nu\in\Lambda_N} T_\nu(t) y^\nu \right\|_{L(\overline{\rho},R)/\kappa,T,\mathcal{S}} \leq CN^{-r}, \quad r = \frac{1}{p} - 1 \tag{14}$$

and where

$$\sum_{\nu\in\Lambda_N} T_\nu(t) y^\nu \in \mathbb{P}_{\Lambda_N}(U; C^1([0,T];\mathcal{S})) \ . \tag{15}$$

For $0 < p \leq 1$ *as in* (12), $(T_\nu)_{\nu\in\mathfrak{F}} \in \ell^p\left(\mathfrak{F}; C^1([0,T];\mathcal{S})\right)$. *Finally, let* (11) *be satisfied for some* $k \geq 0$. *Denote by* $\Lambda_N^k \subset \mathfrak{F}$ *a finite,* monotone *set of* N *largest Taylor coefficients (measured in* $C^{k+1}_{L(\bar{\rho},R)/\kappa}([0,T];\mathcal{S})$*). Then*

$$\sup_{y\in U}\left\| X(\cdot;y) - \sum_{\nu\in\Lambda_N^k} T_\nu(t)y^\nu \right\|_{C^{k+1}_{L(\bar{\rho},R)/\kappa}([0,T];\mathcal{S})} \leq CN^{-r}, \quad r = \frac{1}{p} - 1\,. \tag{16}$$

In [17], also results analogous to Theorem 2 for N-term approximations with finite, monotone index sets for tensorized Legendre and Chebyshev systems are proved. The sparsity result Theorem 2 yields the *existence* of a family of sparse, N-term polynomial approximations of the parametric solutions $X(t;y)$. Apart from monotonicity its proof does not shed light on the structure resp. on the construction of concrete sets $\Lambda_N \subset \mathfrak{F}$ which would yield the proven convergence rate with, possibly, a suboptimal constant.

Unlike in the case of linear problems which was considered in [8], due to the strongly nonlinear nature of the problem (1), stable computation of Taylor coefficients $T_\nu(t)$ is, in general, not advisable (although in biological systems engineering schemes are developed for efficient computation of *sensitivities* $T_{e_j}(t)$).

We therefore consider *collocation approximations* of (1) using Smolyak type collocation operators which are unisolvent on finite, monotone sets (see, e.g. [9]). In order to exploit sparsity in polynomial expansions of the parametric solutions, as provided by Theorem 2, with collocation schemes, it is important that *for finite, monotone sets* $\Lambda \subset \mathfrak{F}$ *of "active" polynomial coefficients* we have available unisolvent, sparse polynomial interpolants.

4 Numerical Examples

In the present section, we present *heuristic adaptive algorithms* which attempt to iteratively localize a sequence $\{\Lambda_N\}_{N\in\mathbb{N}}$ of finite, monotone sets which, although possibly not optimal in the sense of best N-term approximation, will deliver the optimal rate for given summability of the parametric inputs. We place particular attention on high-dimensional parameter spaces, but also investigate the scaling of the proposed algorithms with respect to the dimension p of the state space $\mathcal{S}$ (always assumed here to be finite dimensional, i.e. $\mathcal{S} = \mathbb{R}^p$).

We emphasize that the examples which are presented here are illustrative model problems, and that the development of "industrial strength" numerical solvers for high-dimensional, parametric initial value ODEs is, currently, in its infancy; the present section is intended to give a first indication of scaling and performance of the proposed methods, and, in particular, also identifies specific directions for further algorithm development.

The proposed algorithm successively tries to identify the most profitable indices in a neighborhood of the finite, monotone set Λ_N of currently active indices in terms of error and work contribution. Following [1, 13], for a finite, montone subset $\Lambda \subset \mathfrak{F} = \mathbb{N}_0^d$ in the sense of Def. 1, to every $\nu \in \Lambda$ we associate its 'expected profit', defined by

$$g_\nu(\Lambda; \Xi) = \max_{t \in \Xi} \frac{\Delta E(\nu; \Lambda; t)}{\Delta W(\nu; \Lambda)}, \tag{17}$$

where $\Xi \subset [0, T]$ is a suitable, finite subset, $\Delta W(\nu; \Lambda) = \prod_{k=1}^d (m_{\nu_k} - m_{\nu_k - 1})$, $m_{-1} := 0$, denotes the work contribution associated with the number of interpolation points m_i and $\Delta E(\nu; \Lambda; t) = \sum_{j \in I} \| X(t; x_{j_1}^{\nu_1}, \ldots, x_{j_d}^{\nu_d}) - A(U; \tilde{\Lambda}) X(t; x_{j_1}^{\nu_1}, \ldots, x_{j_d}^{\nu_d}) \|_{\mathcal{S}} \in C^0([0, T])$ the error contribution with $\tilde{\Lambda} = \{\mu \in \Lambda : |\mu| < |\nu|\}$ and $I = \{j \in \mathbb{N}^d : j_l = 1, \ldots, m_{i_l}, l = 1, \ldots, d\}$. Note that the multiindices are assumed to be finite, i.e. $\mathfrak{F} = \mathbb{N}_0^d$, which fits into the theoretical framework discussed above, by setting the remaining entries equal to zero in the infinite-dimensional case.

In what follows, we consider for a given finite, monotone index set Λ_N the Smolyak interpolant (cf. [2])

$$\mathcal{A}(U; \Lambda_N) X(t; y) = \sum_{\nu \in \Lambda_N} (\Delta^\nu X(t; \cdot))(y),$$

with the increment $\Delta^\nu X(t; \cdot) = \bigotimes_{j=1}^d (U^{i_j} - U^{i_j - 1}) X(t; \cdot), \nu \in \Lambda_N$,

$$U^{-1} :\equiv 0, U^n(X(t; \cdot)) = \begin{cases} id, & \text{for } n = 0 \\ \sum_{j=1}^{m_n} X(t; x_j^n) l_j^n, & \text{for } n \geq 1 \end{cases},$$

l_j^n univariate Lagrange interpolation polynomials at

1. Clenshaw-Curtis abscissas x_j^n in $[-1, 1]$, i.e.
   $x_j^n = -\cos\left(\frac{\pi(j-1)}{m_n - 1}\right), j = 1, \ldots, m_n$ if $m_n > 1$ and $x_1^n = 0$ if $m_n = 1$
   with $m_0 = 1$ and $m_n = 2^{n-1} + 1$, for $n \geq 1$
2. Symmetrized Leja abscissas x_j^n, e.g.
   $x_1^n = 0$, if $j = 1$,
   $x_2^n = 1$, if $j = 2$,
   $x_3^n = -1$, if $j = 3$ and
   $x_j^n = \arg\max_{x \in [-1,1]} \prod_{k=1}^{j-1} |x - x_k^n|, j = 4, \ldots, m_n$, if j even,
   $x_j^n = -x_{j-1}^n, j = 4, \ldots, m_n$, if j odd,
   with $m_n = 2n + 1$, for $n \geq 0$ (cf. [5–7]).

In order to approximate the finite, monotone index set maximizing the profit of each index with respect to the indicator (17), we consider the following algorithm, due to [13].

```
1: function ASG
2:     d_a ← 2, ν ← (0,0), Λ ← ∅, I_active ← {ν}
3:     Compute Δ^ν(X(t;·)) and the error indicator g_ν
4:     while max_{ν∈I_active}(g_ν) > tol do
5:         Select ν from I_active with largest g_ν
6:         if ν = e_{d_a} then
7:             d_a ← d_a + 1, I_active ← I_active ∪ {e_{d_a}}
8:             Compute Δ^{e_{d_a}}(X(t;·)) and the error indicator g_{e_{d_a}}
9:         end if
10:        Λ ← Λ ∪ {ν}
11:        for j = 1,...,d_a do
12:            o ← ν + e_j
13:            if o − e_m ∈ Λ, ∀1 ≤ m ≤ d then
14:                I_active ← I_active ∪ {o}
15:                Compute Δ^o(X(t;·) and the error indicator g_o
16:            end if
17:        end for
18:    end while
19: end function
```

Due to the monotonicity requirement of the index set Λ, it holds $\{0\} \subset \Lambda$, so that the proposed algorithm starts with the initial index set $\Lambda_0 = \{0\}$. Then, all feasible neighbor indices are computed, so that the monotonicity of the set Λ is preserved and a percentage of new indices with the largest profit is added to the index set Λ. This procedure is repeated until the estimated profit of the remaining indices is smaller or equal than a given tolerance *tol*. Note that the dimension of the indices is also adaptively controlled, i.e. the dimension of the parameter space is iteratively enlarged according to the above results. The example concerns a parametric initial value problem of the following form.
Given $X(0, y) = x_0 \in \mathbb{R}$, $T = 1$, $U = [-1, 1]^d$, find $X(t; y) : [0, 1] \times U \to \mathbb{R}$ such that $X(0; y) = x_0 = 1 \in \mathbb{R}$

$$\frac{\mathrm{d}X}{\mathrm{d}t} = f(t, X; y) = (f_0 + \sum_{j=1}^{d} f_j y_j)^{\sigma} X\,, \qquad 0 \le t \le 1\,, \tag{18}$$

with $f_i = \left(\frac{1}{i+1}\right)^s$, $s > 1$, $i \in \mathbb{N}_0$.

In (18), the exponent $\sigma = \pm 1$ in all experiments which are considered below. We note that $\sigma = +1$ implies *affine dependence* of the right hand side in (18) on the parameter vector y, as was assumed in the theoretical setting in Sects. 2 and 3 above. Therefore, the parametric family of solutions admits, indeed, sparse representations with respect to the parameter vector y. *We emphasize, however, that due to the linear character of the ODE IVP* (18)*, this can also be verified directly from the explicit expression* (19). We note that (19) reveals that *for* $\sigma = +1$*, the parametric solution is a separable function of the parameters* y_j, so that very favourable approximation properties by Smolyak interpolation can be expected. On the other hand, for $\sigma = -1$, the exact solution is not separable with respect

to the parameters y_j. By direct analysis of the solution formula (19) ahead it can be verified that the parametric solution of (18) allows a representation as unconditionally convergent Taylor expansions with p-summable coefficients, even though for $\sigma = -1$ the dependence of $f(t, X; y)$ on the parameter vector y in (18) is not affine, and the abstract existence theory in [17, Sect. 2 & 3] is not applicable; the N-term approximation results in [17, Sect. 4], however, are applicable based on Taylor coefficient estimates obtained from (19) ahead (indicating, among others, that the theory in [17] could be generalized to certain types of nonaffine, analytic dependence of $f(t, X; y)$ on the parameter vector y). To study the potential of the sparse approximation with respect to the variable y, we will compare the resulting index sets with results based on the exact solution of (18) given by

$$X(t; y) = x_0 \exp(t(f_0 + \sum_{j=1}^{d} f_j y_j)^{\sigma}), \qquad 0 \le t \le 1, \quad \forall y \in U. \tag{19}$$

4.1 Separable Parametric ODE

First, we consider the case of separable solutions $\sigma = 1$ and restrict the discussion to the ten-dimensional case, i.e.

$$f(t, X; y) = \left(1 + \sum_{j=1}^{10} y_j \left(\frac{1}{j+1}\right)^s\right) X.$$

In the following figure, the adaptively constructed finite, monotone index sets based on Clenshaw-Curtis and Leja points as well as the corresponding error contributions, where the adaptivity indicator is chosen as

$$g_\nu(\Lambda; \Xi) = \max_{t \in \Xi} \frac{\Delta E(\nu; \Lambda; t)}{\Delta W(\nu; \Lambda)}, \qquad \Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$$

are shown. Results based on the exact solution using linear ansatz functions to discretize the time interval $[0, 1]$ ($\Delta t = 2^{-8}$) are compared to the numerical solution of the ODE (18) using MATLAB's ode45 (Runge-Kutta method 4(5) with variable time step and dense output, see [15, 16] for more details) with prescribed tolerance $eps = 2.22045 \cdot 10^{-14}$. In the case of Clenshaw-Curtis as well as of Leja points, it can be stated that both approaches lead to the same adaptively constructed index sets, so that the approximation of the solution (18) does not affect the approximation of the solution with respect to the parameter sequence y, see Fig. 1.

We further investigate this effect by comparing the resulting grids for the case $s = 4$ (cp. Fig. 2 using Clenshaw-Curtis points and Leja nodes).

Similar to the case $s = 2$, there is a perfect match between the two solutions in both cases, i.e. considering Clenshaw-Curtis as well as Leja points. Finally,

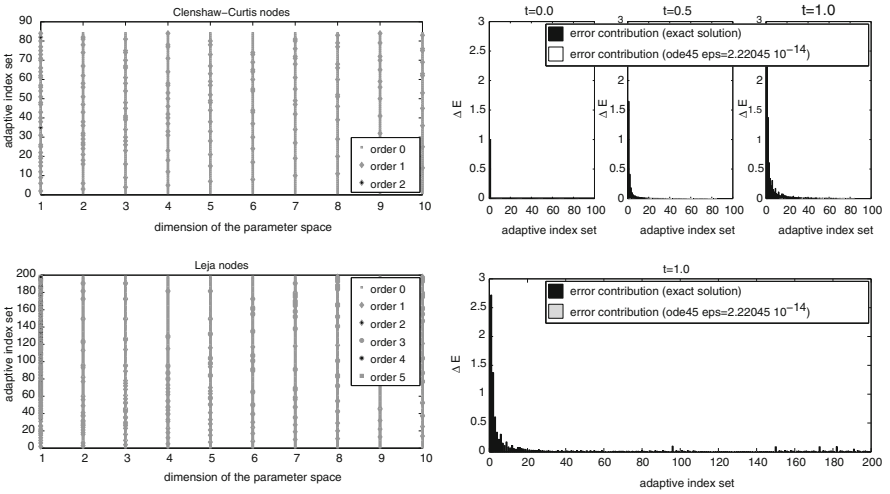

Fig. 1 Adaptively constructed index set Λ (*left*) and comparison of the error contribution of each index $\nu \in \Lambda$ exemplarily shown at time $t_0 = 0.0, t_5 = 0.5, t_{10} = 1.0$ based on the exact solution (*black*) and numerical solution (*gray*) ($\sigma = 1$, $s = 2$, $d = 10$, $tol = 10^{-4}$, exact: $\Delta t = 2^{-8}$, $\Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$, ode45: $eps = 2.22045 \cdot 10^{-14}$, Clenshaw-Curtis nodes (*above*), Leja nodes (*below*))

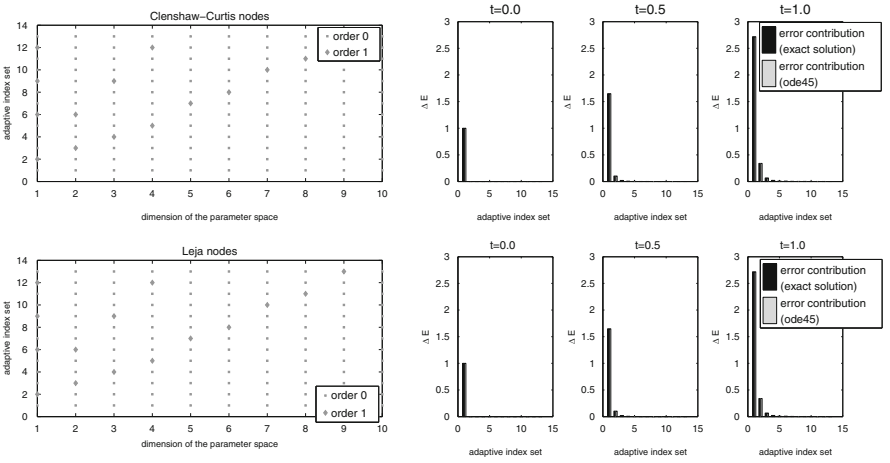

Fig. 2 Adaptively constructed index set Λ (*left*) and comparison of the error contribution of each index $\nu \in \Lambda$ exemplarily shown at time $t_0 = 0.0, t_5 = 0.5, t_{10} = 1.0$ based on the exact solution (*black*) and numerical solution (*gray*) ($\sigma = 1$, $s = 4$, $d = 10$, $tol = 10^{-4}$, exact: $\Delta t = 2^{-8}$, $\Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$, ode45: $eps = 2.22045 \cdot 10^{-14}$, Clenshaw-Curtis nodes (*above*), Leja nodes (*below*))

the sparsity of the solution $X(t; \cdot)$ with respect to y is explored by adapting the accuracy of the ODE solver according to the impact of the index on the solution. Therefore, we estimate the error contribution of a new feasible index ν for a given

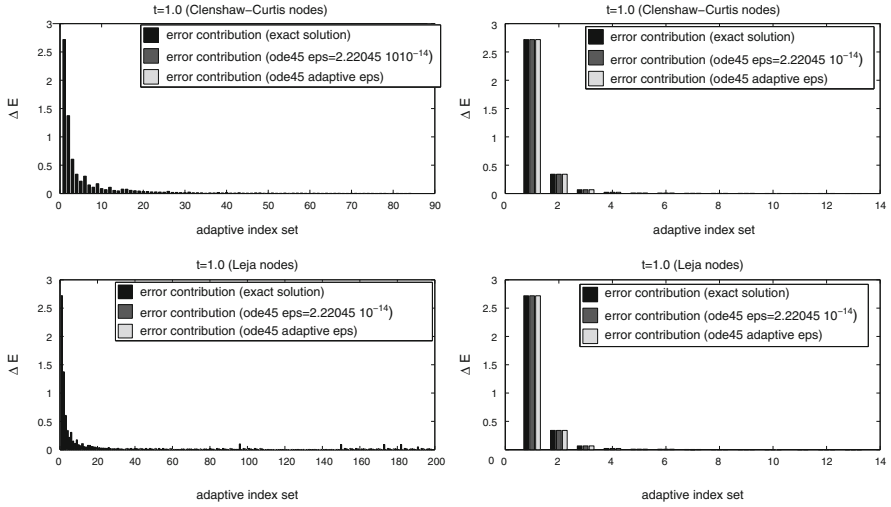

Fig. 3 Comparison of the error contribution of each index $\nu \in \Lambda$ for the case $s = 2$ (*left*) and $s = 4$ (*right*) exemplarily shown at time $t_{10} = 1.0$ based on the adaptive strategy controlling the tolerance of the ODE solver (*light gray*) with the exact solution (*black*) and numerical solution with fixed error tolerance (*gray*) ($\sigma = 1$, $d = 10$, $tol = 10^{-4}$, exact: $\Delta t = 2^{-8}$, $\Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$, ode45: $eps = 2.22045 \cdot 10^{-14}$, ode45: adaptive *eps*, Clenshaw-Curtis nodes (*above*), Leja nodes (*below*))

finite, monotone index set $\Lambda \subset \mathfrak{F}$ by the maximum of the error contributions of all predecessors

$$\Delta\tilde{E}(\nu;\Lambda) = \max_{\nu - e_j \in \Lambda, \forall j \in supp(\nu)} \max_{t \in \Xi} \Delta E(\nu - e_j;\Lambda;t)\,, \tag{20}$$

normalized by $\Delta\hat{E}(\nu;\Lambda) = \frac{\Delta\tilde{E}(\nu;\Lambda)}{\Delta E(0;\Lambda)}$, so that the error tolerance *eps* of the ODE solver is chosen as

$$eps_{adapt}(\nu) = \frac{eps}{\Delta\hat{E}(\nu;\Lambda)}\,. \tag{21}$$

The resulting adaptively constructed index sets for the case $s = 2$ as well as $s = 4$ are practically identical to the set obtained by using the exact solution. Figure 3 illustrates the error contribution based on the proposed adaptive strategy compared to the solutions depicted in Figs. 1 and 2. We can observe that the fully adaptive strategy yields the same accuracy as the reference solution while minimizing at the same time the work contribution for each index. The same effect can be explored using Leja points.

The speedup in terms of function evaluations required for the approximation of the ODE solution and in terms of computation time are summarized in the following table.

Table 1 Comparison of the number of function evaluations and computation time using the proposed adaptive strategy to control the error tolerance of the ODE solve and the non-adaptive approach using a fixed tolerance of $eps = 2.22045 \cdot 10^{-14}$ ($\sigma = 1$, $d = 10$, $tol = 10^{-4}$, Apple Mac Mini, 2.66 GHz Intel Core 2 Duo, 4 GB)

	Clenshaw-Curtis nodes		Leja nodes	
	# feval	cpu feval (s)	# feval	cpu feval (s)
$s = 2$				
Fixed tolerance	202,575	41.28	328,006	66.52
Adaptive strategy	95,399	20.84	138,089	29.504
$s = 4$				
Fixed tolerance	29,984	6.36	29,985	6.14
Adaptive strategy	14,348	3.20	14,350	3.13

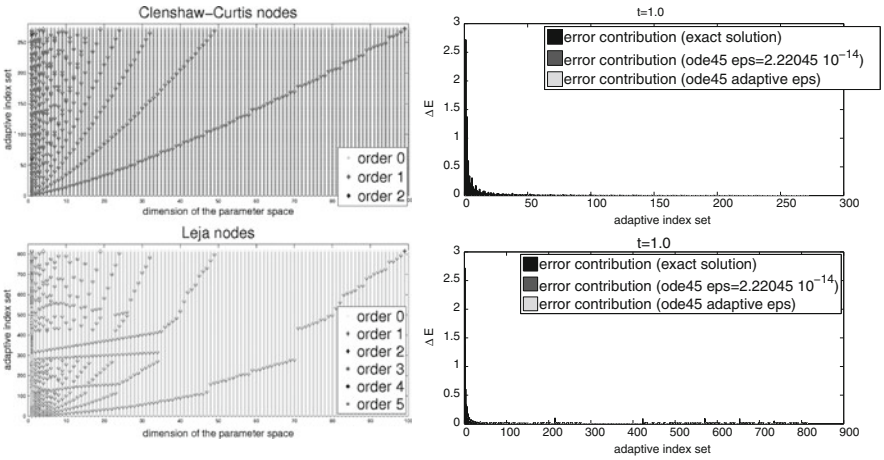

Fig. 4 Adaptively constructed index set Λ (*left*) and comparison of the error contribution of each index $\nu \in \Lambda$ exemplarily shown at time $t_{10} = 1.0$ based on the adaptive strategy controlling the tolerance of the ODE solver (*light gray*) with the exact solution (*black*) and numerical solution with fixed error tolerance (*gray*) ($\sigma = 1$, $d = 100$, $s = 2$, $tol = 10^{-4}$, exact: $\Delta t = 2^{-8}$, $\Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$, ode45: $eps = 2.22045 \cdot 10^{-14}$, ode45: adaptive *eps*, Clenshaw-Curtis nodes (*above*), Leja nodes (*below*))

To verify the efficiency of the proposed method in the high-dimensional case, the underlying problem is considered for $d = 100$. As in the previous example, a variation of the parameter s ($s = 2$, $s = 4$) as well as results based on the exact solution (19) and numerical solution of (18) using the Matlab ode45 solver with fixed and adaptive error tolerance considering Clenshaw-Curtis and Leja points are presented.

Figure 4 illustrate the case $s = 2$, where the grids based on the exact as well as on the non-adaptive and adaptive numerical solution of the ODE are identical. Increasing the parameter ($s = 4$) leads to the same finite, monotone index set as

Table 2 Comparison of the number of function evaluations and computation time using the proposed adaptive strategy to control the error tolerance of the ODE solver with the non-adaptive approach using a fixed tolerance of $eps = 2.22045 \cdot 10^{-14}$ ($\sigma = 1, d = 100, tol = 10^{-4}$, Apple Mac Mini, 2.66 GHz Intel Core 2 Duo, 4 GB)

	Clenshaw-Curtis nodes		Leja nodes	
	# feval	cpu feval (s)	# feval	cpu feval (s)
$s = 2$				
Fixed tolerance	3,591,145	830.50	4,089,927	935.77
Adaptive strategy	1,302,439	312.56	1,468,805	349.63
$s = 4$				
Fixed tolerance	29,984	9.86	29,985	9.58
Adaptive strategy	14,348	4.73	14,350	4.76

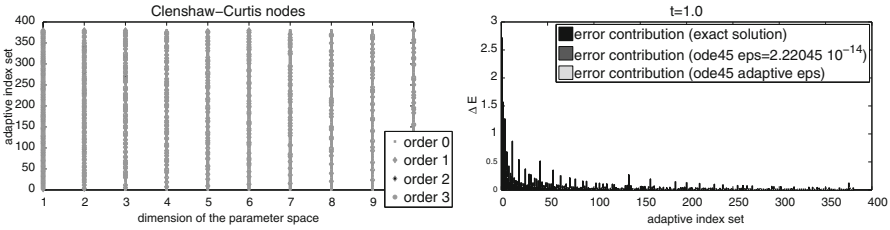

Fig. 5 Adaptively constructed index set Λ (*left*) and comparison of the error contribution of each index $\nu \in \Lambda$ exemplarily shown at time $t_{10} = 1.0$ based on the adaptive strategy controlling the tolerance of the ODE solver (*light gray*) with the exact solution (*black*) and numerical solution with fixed error tolerance (*gray*) ($\sigma = -1, d = 10, s = 2, tol = 10^{-4}$, exact: $\Delta t = 2^{-8}$, $\Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$, ode45: $eps = 2.22045 \cdot 10^{-14}$, ode45: adaptive *eps*, Clenshaw-Curtis nodes)

in the ten-dimensional case (see Fig. 3). The statistics for the case $d = 100$ are summarized in Table 2.

4.2 Non-separable ODE

We will now discuss the non-separable case, i.e. $\sigma = -1$ in (18). Numerical results are presented considering a variation of the parameter s in the ten-dimensional case, i.e. $d = 10$. The following figure shows the adaptively constructed index set and comparison of the error contribution of each index based on the adaptive strategy controlling the tolerance of the ODE solver with the exact solution and numerical solution with fixed error tolerance using Clenshaw-Curtis interpolation points (Fig. 5).

Comparing the results with the separable case (cf. Fig. 1), we can state that the number of indices of the adaptive sparse grid is enlarged approximately by a factor

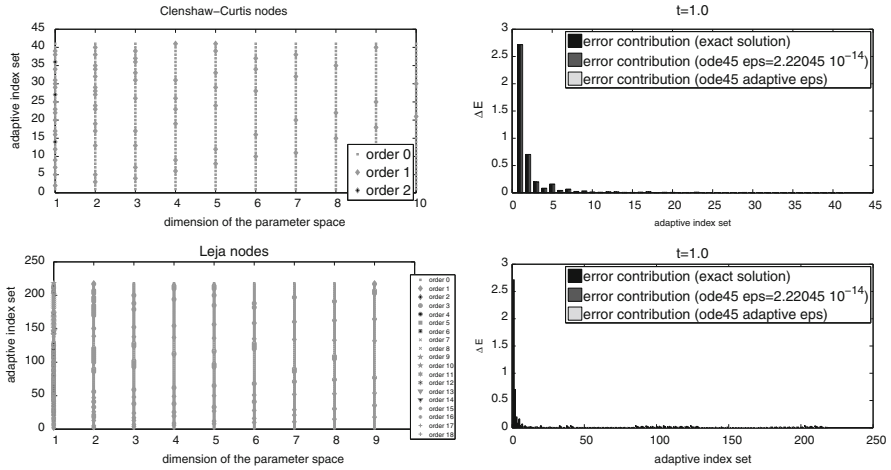

Fig. 6 Adaptively constructed index set Λ (*left*) and comparison of the error contribution of each index $\nu \in \Lambda$ exemplarily shown at time $t_{10} = 1.0$ based on the adaptive strategy controlling the tolerance of the ODE solver (*light gray*) with the exact solution (*black*) and numerical solution with fixed error tolerance (*gray*) ($\sigma = -1$, $d = 10$, $s = 3$, $tol = 10^{-4}$, exact: $\Delta t = 2^{-8}$, $\Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$, ode45: $eps = 2.22045 \cdot 10^{-14}$, ode45: adaptive eps, Clenshaw-Curtis nodes (*above*), Leja nodes (*below*))

5 to reach the given tolerance. In the case of Leja points, the prescribed tolerance cannot be reached in a reasonable computation time due to the higher number of indices (resulting from the linear growth of the interpolation nodes) and the related overhead caused by the search of new admissible indices. Therefore, we additionally present results for the case $s = 3$ in order investigate the influence of the choice of interpolation nodes in the non-separable case, see Fig. 6.

Increasing the sparsity with respect to the parameter y can be exploited by the algorithm in both cases, that means in the case of Clenshaw-Curtis and Leja points, cf. Fig. 7.

Similar to the separable case (cf. Tables 1 and 2), the proposed adaptive control of the accuracy of the ODE solver can significantly reduce the overall costs of the algorithm (see Table 3). Further, the computational effort needed to construct the finite, monotone index sets in the case of Clenshaw-Curtis and Leja nodes are comparable in the case $s = 4$ due to the low order of the grids.

4.3 ODE System

The separable as well as the non-separable case clearly demonstrate the speedup which can be gained by the fully adaptive strategy. The savings in CPU time become even more evident considering a high-dimensional ODE where the main

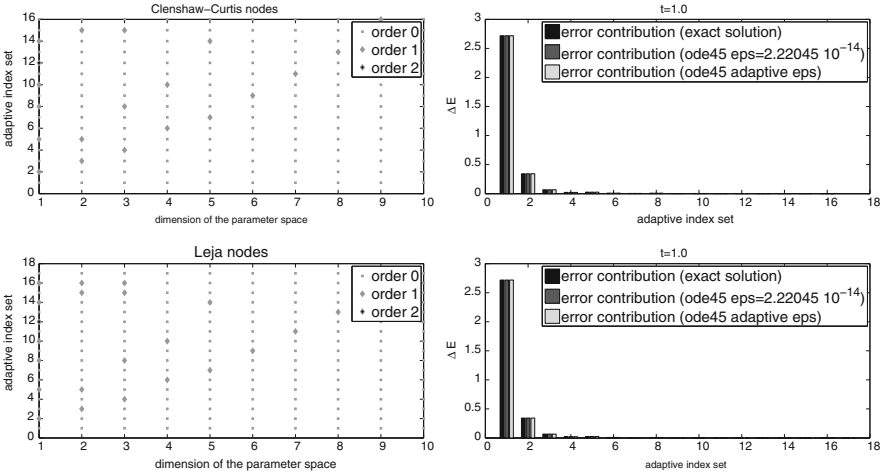

Fig. 7 Adaptively constructed index set Λ (*left*) and comparison of the error contribution of each index $\nu \in \Lambda$ exemplarily shown at time $t_{10} = 1.0$ based on the adaptive strategy controlling the tolerance of the ODE solver (*light gray*) with the exact solution (*black*) and numerical solution with fixed error tolerance (*gray*) ($\sigma = -1$, $d = 10$, $s = 4$, $tol = 10^{-4}$, exact: $\Delta t = 2^{-8}$, $\Xi = \{0, 0.1, 0.2, \ldots, 1.0\}$, ode45: $eps = 2.22045 \cdot 10^{-14}$, ode45: adaptive *eps*, Clenshaw-Curtis nodes (*above*), Leja nodes (*below*))

Table 3 Comparison of the number of function evaluations and computation time using the proposed adaptive strategy to control the error tolerance of the ODE solver with the non-adaptive approach using a fixed tolerance of $eps = 2.22045 \cdot 10^{-14}$ ($\sigma = -1$, $d = 10$, $tol = 10^{-4}$, Apple Mac Mini, 2.66 GHz Intel Core 2 Duo, 4 GB)

	Clenshaw-Curtis nodes		Leja nodes	
	# feval	cpu feval (s)	# feval	cpu feval (s)
$s = 2$				
Fixed tolerance	974,838	195.31	–	–
Adaptive strategy	528,549	108.97	–	–
$s = 3$				
Fixed tolerance	65,433	14.28	227,539	47.05
Adaptive strategy	33,378	7.54	88,696	19.26
$s = 4$				
Fixed tolerance	34,596	7.60	34,269	7.19
Adaptive strategy	15,846	3.33	15,779	3.38

part of the computational effort results from the numerical solution of the underlying differential equation. To investigate this point, we consider the following system of parametric ODEs given by $X(t; y) : [0, 1] \times U \to \mathbb{R}^p$, $T = 1$, $U = [-1, 1]^d$

$$\frac{\mathrm{d}X}{\mathrm{d}t} = A(y)X\,, \qquad X(0; y) = x_0 = 1 \in \mathbb{R}^p\,,\ 0 \le t \le 1\,,\ \forall y \in U \tag{22}$$

Table 4 Comparison of the number of function evaluations and computation time using the proposed adaptive strategy to control the error tolerance of the ODE solver with the non-adaptive approach using a fixed tolerance of $eps = 2.22045 \cdot 10^{-14}$ (Apple Mac Mini, 2.66 GHz Intel Core 2 Duo, 4 GB)

	Clenshaw-Curtis nodes			Leja nodes		
	# feval	cpu feval (s)	Total cpu time (s)	# feval	cpu feval (s)	Total cpu time (s)
$d = 10,\ p = 10$						
Fixed tolerance	223,554	96.28	470.53	223,142	98.21	477.26
Adaptive strategy	114,448	50.25	423.99	114,336	50.98	429.02
$d = 10,\ p = 100$						
Fixed tolerance	224,475	450.32	3,921.83	224,042	445.91	3,914.81
Adaptive strategy	114,819	227.41	3,698.53	114,703	228.42	3,677.16

with $A(y) = (a_{kl}(y)), k, l = 1, \ldots, p$ given by

$$a_{kl} = \begin{cases} 1 + \sum_{j=1}^{d} y_j \left(\frac{1}{j+1}\right)^{s_k}, & \text{if } k = p - l + 1 \\ 0, & \text{otherwise} \end{cases}$$

and $s_1 = 1.2,\ s_k = k, \forall k = 2, \ldots, p$. The error contribution of each index v is estimated by the maximum error contribution of the components $X_i, i = 1, \ldots p$ with $X = (X_1, \ldots, X_p)^T$. The results are summarized in Table 4.

The presented investigations show the potential of the adaptive error control of the ODE solver. In terms of function evaluations needed for the numerical solution of the underlying ODE and the corresponding cpu time, the adaptive approach is able to halve the computational effort by maintaining at the same time a comparable accuracy of the fully adaptive approximation in t and with respect to y, cp. Table 4.

Acknowledgements Research supported in part by the European Research Council (ERC) under the FP7 program AdG247277 and by the Eidgenössische Technische Hochschule (ETH) Zürich under the ETH-Fellowship Grant FEL-33 11-1.

References

1. Beck, J., Nobile, F., Tamellini, L., Tempone, R.: On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods. Math. Models Methods Appl. Sci. (M3AS) **22**(9), 1250023.1–1250023.33 (2012)
2. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. Adv. Comput. Math. **12**, 273–288 (2000). 10.1023/A:1018977404843
3. Bieri, M.: A sparse composite collocation finite element method for elliptic SPDEs. SIAM J. Numer. Anal. **49**, 2277–2301 (2011)

4. Bieri, M., Andreev, R., Schwab, C.: Sparse tensor discretization of elliptic SPDEs. SIAM J. Sci. Comput. **31**, 4281–4304 (2009/2010)
5. Caliari, M., Vianello, M., Bergamaschi, L.: Interpolating discrete advection-diffusion propagators at Leja sequences. J. Comput. Appl. Math. **172**, 79–99 (2004)
6. Calvi, J.-P., Phung Van, M.: On the Lebesgue constant of Leja sequences for the unit disk and its applications to multivariate interpolation. J. Approx. Theory **163**, 608–622 (2011)
7. Calvi, J.-P., Phung Van, M.: Lagrange Interpolation at Real Projections of Leja Sequences for the Unit Disk. Proceedings of the American Mathematical Society (2012). http://dx.doi.org/10.1090/S0002-9939-2012-11291-2
8. Chkifa, A., Cohen, A., DeVore, R., Schwab, Ch.: Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs. ESAIM: Math. Model. Numer. Anal. **47**(1), 253–280 (2013). doi:http://dx.doi.org/10.1051/m2an/2012027
9. Chkifa, A., Cohen, A., Schwab, Ch.: High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. Found. Comput. Math. **14**(4), 601–633 (2014). ISSN:1615-3375
10. Cohen, A., DeVore, R., Schwab, C.: Convergence rates of best n-term approximations for a class of elliptic SPDEs. J. Found. Comput. Math. **10**, 615–646 (2010)
11. Cohen, A., DeVore, R.A., Schwab, C.: Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs. Anal. Appl. **9**, 11–47 (2011)
12. Deimling, K.: Nonlinear Ordinary Differential Equations in Banach Spaces. Volume 596 of Springer Lecture Notes in Mathematics. Springer, New York (1977)
13. Gerstner, T., Griebel, M.: Dimension-adaptive tensor-product quadrature. Computing **71**, 65–87 (2003)
14. Gittelson, C.J.: Convergence rates of multilevel and sparse tensor approximations for a random elliptic PDE. SIAM J. Numer. Anal. **51**(4), 2426–2447 (2013). doi:10.1137/110826539
15. Hairer, E., Nørsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I: Nonstiff Problems. Volume 8 of Springer Series in Computational Mathematics. Springer, Berlin (1987)
16. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Volume 14 of Springer Series in Computational Mathematics, 2nd edn. Springer, Berlin (1996)
17. Hansen, M., Schwab, C.: Sparse adaptive approximation of high dimensional parametric initial value problems. Vietnam J. Math. 1–35 (2013). http://dx.doi.org/10.1007/s10013-013-0011-9
18. Hoang, V.H., Schwab, C.: Analytic regularity and polynomial approximation of stochastic, parametric elliptic multiscale PDEs. Anal. Appl. (Singapore) **11**(01), 1350001-1–1350001-50 (2013). doi:http://dx.doi.org/10.1142/S0219530513500012
19. Nobile, F., Tempone, R., Webster, C.G.: An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. SIAM J. Numer. Anal. **46**, 2411–2442 (2008)
20. Walter, W.: Ordinary Differential Equations. Volume 182 of Graduate Texts in Mathematics. Springer, New York (1998). Translated from the sixth German (1996) edition by Russell Thompson, Readings in Mathematics

Investigating Capturability in Dynamic Human Locomotion Using Multi-body Dynamics and Optimal Control

Khai-Long Ho Hoang, Katja Mombaur, and Sebastian I. Wolf

Abstract An important goal in the development of advanced prosthetic devices is to enhance the stability of prosthetic gait and eventually to augment safety. For this, a fundamental understanding of stability and stabilization mechanisms of human walking motions is crucial. The objective of this work is to evaluate if the stability of human walking can be linked to the concept of N-step Capturability developed in robotics. The main idea of this concept is represented by the Instantaneous Capture Point (ICP), which indicates the position on the ground where a two-legged walking system would have to place the next step in order to come to a complete stop and the respective N-Step Capture Regions where a stop after N steps can be reached. While a human walker does not precisely step onto the ICP since this would be inefficient, we hypothesize that the location of this point can be used to parameterize the location of the foot position. Experiments were performed in a gait lab to record the kinematic data of healthy human gait on level ground. The human body was modeled as a multi-body system composed of eight rigid bodies that represent the pelvis and three-segmented legs as well as the upper body merged into a single trunk segment. Using optimal control methods, joint angle trajectories of the multi-body model were generated that best fit the experimental data while considering the kinematic and physiological constraints of the human body. The trajectory of the ICP was computed using the kinematic and kinetic data of the multi-body model that derived from the solution of our optimal control problem. The results show that the ICP is directly approached by the swing foot during swing phase and suggest a correlation between the foot placement strategy in human walking and N-Step Capturability.

K.-L. Ho Hoang (✉) • K. Mombaur
Interdisciplinary Center for Scientific Computing, Optimization in Robotics and Biomechanics, Speyerer Str. 6, 69115 Heidelberg, Germany
e-mail: KhaiLong.HoHoang@iwr.uni-heidelberg.de; Katja.Mombaur@iwr.uni-heidelberg.de

S.I. Wolf
Motion Analysis Lab, Department of Orthopedic Surgery, Heidelberg University Clinics, Schlierbacher Landstr. 200a, 69118 Heidelberg, Germany
e-mail: Sebastian.Wolf@med.uni-heidelberg.de

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_7

1 Introduction

In the recent years highly versatile exo-prostheses have emerged enabling an amputee patient to master various difficult gait situations, e.g. walking at different gait speeds, ascending and descending slopes, and walking on uneven terrain. With micro-processor assisted prosthetic devices that make use of sensory information and passive-adaptive mechanical elements highly dynamic locomotion became possible for the patient and mobility is enhanced. Nevertheless, stable walking is still challenging for above-knee amputee patients due to the absence of active control possibilities over the knee and ankle joint of the prosthetic devices. Understanding the strategy of healthy humans in maintaining balance while performing dynamic locomotion is the key-task to develop more versatile and safe prosthetic components. Analytical stability criteria need to be established to exploit the entire potential given by controlled prosthetic devices.

Dynamic stability of prosthetic walking was analyzed in [1] where commercially available microprocessor-controlled knee joints were compared in a biomechanical analysis. Here, the risk of falling with transfemoral prostheses was investigated in everyday situations such as abruptly stopping and side-stepping, stepping on an obstacle and stumbling. However, a measure to quantify stability has not been formulated. Stability in terms of variability and symmetry of prosthetic gait was studied in [4, 9] where unstable walking was defined to be related to deviation from symmetric motion patterns. In our work, however, we assume that stable walking can also be achieved with asymmetric gait.

We try to understand stability in human walking by exploiting the concept of *Capturability* and perform dynamic simulation using optimal control. Our motivation is to find a measure to quantify stability and predict unstable gait situations. We regard the typical trajectory of walking motion as a sequence of initiated falling motions into the walking direction followed by well-timed catch motions performed by the leg that has just swung forward [13].

In Sect. 2 we will describe the motion capture experiments performed in the gait lab. The multi-body model that is used as a mathematical representation of the human body and the formulation of the equations of motion are introduced in Sect. 3 followed by a short review about the most commonly used stability criteria in biped walking analysis in Sect. 4. Section 5 contains the formulation of the optimal control problem to generate the human walking motion. The results of our investigation are summarized in Sect. 6 and discussed in Sect. 7.

2 Motion Capture Experiments for Human Walking

We performed experiments in the gait laboratory involving healthy subjects walking on level ground. Data from one subject (male, 30 years, 1.87 m, 86 kg) walking on level ground (walking speed: $v = 1.5$ m/s, step length $l_s = 0.83$ m) were further processed.

The walking motion of the subject was recorded using an image-based motion capture system which consists of 12 Vicon [14] infrared-based cameras recording the 3-dimensional trajectories of 48 retro-reflective markers attached to the subject's body. The images were recorded with a sample frequency of 120 Hz. The marker positions on the skin were chosen such that they can be easily identified, the relative movement between skin and bones was minimal, and the positions of the joint centers could easily by reconstructed from the markers.

The orientation and position of the major body segments in Euclidean space are reconstructed from the marker trajectories using the Plug-in Gait model [14], a marker-based representation of the joint centers of the human body. The body pose reconstruction provides us with the joint-angles for all degrees of freedom (DoF) of our model that we will introduce more detailed in Sect. 3.

3 Modeling the Human Body

Our investigation of balancing strategies in human walking concentrates on phenomena resulting from larger movements while smaller effects, such as deformations, can be omitted. This allows us to perform dynamic simulation using the methods of multi-body dynamics.

3.1 Multi-body Representation of the Human Body

The parts of the body that are relevant for walking motions are represented by rigid bodies that are connected to each other by ball joints allowing three rotational degrees of freedom in each joint. For the upper body we consider the head, arms and trunk combined in a single rigid body which, accordingly, is named the *HAT*-segment (*H*ead, *A*rms, *T*runk). The multi-body model of the human body is comprised of eight rigid bodies representing the segments HAT, pelvis, right and left thigh, right and left shank, and right and left foot. These segments are connected by seven joints representing the lumbo-sacral joint, the right and left hip, right and left knee, right and left ankle. Figure 1 shows the configuration of the 27 DoF multi-body model.

The 27 DoF of the model are composed of 3 translational and 3 rotational DoF for the pelvis segment which are relative to the global frame and 7×3 rotational DoF that specify the rotation of a more distal segment relative to the next proximal one with the pelvis being the most proximal segment.

The anthropometric data of the subject which will be set as model parameters in our multi-body model are obtained using regression equations by de Leva [3] that receive the subject's total mass (86 kg) and total height (1.87 m) as input. Table 1 lists all relevant parameters of the multi-body model where the CoM is assumed to

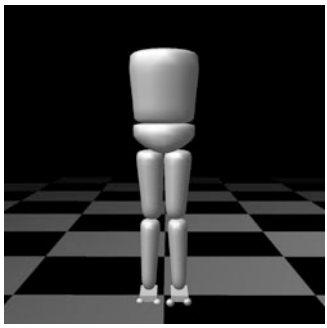

Fig. 1 The 27 DoF multi-body model of the human body is comprised of eight segments connected by seven ball joints

Table 1 Anthropometric data of the subject (male, 30 years, 1.87 m, 86 kg) according to regression equations by de Leva

Body part	Segment length (mm)	Segment mass (kg)	CoM position (mm)	Inertia radius (mm) (sagittal, transversal, longitudinal)		
Pelvis	156.5	9.61	95.7	96.3	86.2	91.9
Thigh	453.4	12.18	−185.7	149.2	149.2	67.6
Shank	466.2	3.72	−207.9	118.9	116.1	48.0
Foot	41.5[a]	1.37	−20.8[c]	71.2	67.9	34.4
	277.1[b]		122.4[d]			
HAT	491.6	22.22	249.4	168.4	144.6	144.8

[a] Height of the foot segment
[b] Length of the foot segment
[c] Relative to longitudinal axis of the foot segment
[d] Relative to sagittal axis of the foot segment

be located on the longitudinal axis of each segment and assigned relative to the next proximal joint position.

3.2 Equations of Motion

To describe the dynamics of the multi-body model equations of motion are formulated using the modeling tool RigidBodyDynamicsLibrary [6]. Since our model is formulated with generalized coordinates q, the equations of motion can be expressed as

$$M(q)\,\ddot{q} + N(q,\dot{q}) = \tau \tag{1}$$

where $M(q)$ is the symmetric positive definite 27×27-mass-matrix, $N(q,\dot{q})$ the 27×1-vector of generalized non-linear effects that contains the Coriolis, centrifugal and gravitational forces, and τ the 27×1-vector of generalized torques. We compute

$M(q)$ in a highly efficient way using the *Composite Rigid Body Algorithm* (CRBA) and $N(q, \dot{q})$ using the *Recursive Newton Euler Algorithm* (RNEA) [5].

3.3 Ground Contact Model

The contact of the model with the ground is modeled as a rigid three point contact at each foot where the three contact points are located at the *Calcaneus* (heel), the *medial Metatarsal Phalangeal Joint* (mMTPJ) and the *lateral Metatarsal Phalangeal Joint* (lMTPJ), see Fig. 2. As soon as contact occurs, the vertical position of the contact points that are in touch with the ground are restricted to the ground by algebraic equality constraints. Furthermore, the horizontal accelerations of the contact points are set to zero.

Using the ground contact model we receive the Index 1-DAE

$$M(q)\ddot{q} = \tau - N(q, \dot{q}) + G(q)^T \lambda \tag{2}$$

$$G(q)\ddot{q} = -\gamma(q, \dot{q}), \tag{3}$$

or in matrix notation

$$\begin{pmatrix} M & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} \ddot{q} \\ -\lambda \end{pmatrix} = \begin{pmatrix} \tau - N \\ -\gamma \end{pmatrix}. \tag{4}$$

where G are the point Jacobians of the contact points, λ the contact forces and γ the generalized acceleration independent part of the contact point accelerations [11].

In human walking, the occurrence of contact is subject to an impact and leads to discontinuities in the generalized velocities $\dot{q}$. We compensate these by introducing events of infinitesimal length where we apply the system of equations

$$\begin{pmatrix} M & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_+ \\ -\Lambda \end{pmatrix} = \begin{pmatrix} M\dot{q}_- \\ 0 \end{pmatrix}, \tag{5}$$

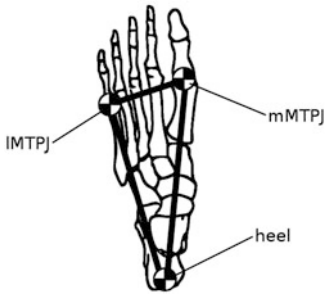

Fig. 2 The ground contact is modeled at the three contact points *Calcaneus* (heel), *medial Metatarsal Phalangeal Joint* (mMTPJ) and *lateral Metatarsal Phalangeal Joint* (lMTPJ)

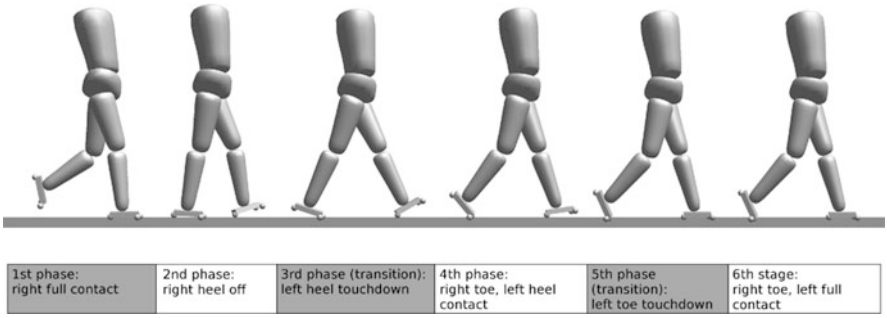

Fig. 3 The model stages according the typical human walking cycle

where Λ refers to an inelastic impulse, $\dot{q}_+$ denotes the generalized velocities right after the impact and $\dot{q}_-$ the generalized velocities right before the impact, respectively.

With every change in $G(q)$ that arises due to changing ground contact properties we formulate a separate model stage and create a new set of equations of motion (4). Figure 3 shows the human walking cycle and the resulting sequence of stages. Simulating the typical sequence of human gait with our three-point foot model leads to six different stages regarding the ground collision and lift-off of the contact points. This number of stages also includes transition stages that are needed to compensate impacts due to a touch-down of a contact point using (5).

4 Stability, Balance and Capturability

The investigation of balance and stability in human walking can lead to advances in the field of humanoid robotics motivated by the intention to find methods to efficiently control biped walking while avoiding the robot from falling down [18].

Considering strictly periodic motions, self-stable open-loop gait has been simulated exploiting Lyapunov's first method as a mathematical definition of stability [11]. Here, asymptotic stability is given for a T-periodic solution of a T-periodic non-autonomous system $\dot{x}(t) = f(t, x(t))$ with $f(t, \cdot) = f(t + T, \cdot)$ if all eigenvalues λ_i of the monodromy matrix $X(t + T) = \frac{\mathrm{d}x(t+T)}{\mathrm{d}x(t)}$ are inside the unit circle $|\lambda_i(X(t + T))| < 1$.

A very popular criterion to ensure balance of biped robots is used with the *Zero Moment Point* (ZMP) which is defined as a ground-reference point where the net moment generated from the ground reaction forces vanishes for the two axes that span the ground plane [17]. In humanoid robotics the dynamic feasibility of desired trajectories is usually ensured as long as the ZMP lies well within the borders of the *base of support* (BoS). Dynamic feasibility of the desired trajectory cannot be guaranteed if the ZMP lies on the edge of the BoS [15].

Another meaningful method which can also be used to quantify balance of a biped walking system keeps track of $\dot{H}_G$, the rate of change of angular momentum at the center of mass (CoM) of the system [7]. This method refers to the preservation of rotational stability and considers a biped walking system to be rotational stable if the external forces and moments sum up to a zero centroidal moment. According to fundamental principles of mechanics this leads to a minimization of $\dot{H}_G$. The ground-reference point *Zero Rate of Change of Angular Momentum* (ZRAM) indicates the position on the ground where $\min \dot{H}_G = GP \times R$ with G denoting the position of the center of mass, P the position of the center of pressure and R referring to the resultant ground reaction force (GRF).

Push recovery, i.e. enabling a humanoid robot to recover itself from a sudden arbitrary push to avoid falling down, led to the concepts of *N-Step Capturability* with the main idea represented by the *Capture Point* [16]. The Capture Point (CP) is defined as the ground-reference point indicating the position where a biped walking system would have to step on to come to a complete stop. Assuming that the CP can be reached instantaneously and neglecting the time that is needed to swing the foot forward leads to the *Instanteneous Capture Point* (ICP).

In order to find the position of the ICP we regard the human body as an inverted pendulum with variable rod length. The position of the inverted pendulum's point mass is equal to the CoM of our multi-body model while the pivot point of the inverted pendulum is located at the ground projection of the ankle joint $r_{ankle} = (x_{ankle}, y_{ankle}, z_{ankle})^T$ of the current stance leg. Considering the orbital energy of the inverted pendulum

$$E_{IP,x} = \frac{1}{2}\left(\frac{\dot{x}}{\omega_0 z}\right)^2 - \frac{1}{2}\left(\frac{x - x_{ankle}}{z}\right)^2 \qquad \text{and} \tag{6}$$

$$E_{IP,y} = \frac{1}{2}\left(\frac{\dot{y}}{\omega_0 z}\right)^2 - \frac{1}{2}\left(\frac{y - y_{ankle}}{z}\right)^2 \tag{7}$$

with $E_{IP,x} = E_{IP,y} \equiv 0$ we receive the location of the ICP:

$$r_{ICP} = P\left[r + \frac{\dot{r}}{\omega_0}\right] \qquad \text{where} \qquad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

projects r_{ICP} on the ground, r refers to the position of the CoM and $\omega_0 = \sqrt{\frac{g}{z_0}}$ is the reciprocal of the time constant of the inverted pendulum [8].

Since, as mentioned earlier, we regard human walking motion as a sequence of falling and catching, we hypothesize that walking can be described related to push recovery and exploiting the concept of N-Step Capturability. We try to find a correlation between the position of the ICP and the actual foot placement during human walking in order to quantify stability in human locomotion.

5 Optimal Control Problem

We intend to find joint angle trajectories $x(t)$ and joint torques $u(t)$ for our multi-body model that best fit the experimental data Φ_{MoCap} using *least-squares algorithms* (LSQ) while considering given constraints. Considering also the contact properties as discussed in Sect. 3.3) this leads us to a multi-stage optimal control problem of the following form:

$$\min_{x(\cdot),u(\cdot)} \quad \int_0^{t_f} \frac{1}{2} ||\Phi_{MoCap}(t) - \Phi_{Opt}(t, x(t))||_2^2 \, \mathrm{d}\, t \tag{8}$$

$$\text{subject to:} \qquad \dot{x}(t) = f_i(t,\, x(t),\, u(t)) \quad \text{or DAE} \tag{9}$$

$$x(\hat{t}_i^+) = h(x(\hat{t}_i^-)), \tag{10}$$

$$g_i(t, x(t), u(t)) \geq 0, \tag{11}$$

$$\text{for} \quad t \in [\hat{t}_{i-1}, \hat{t}_i],\ i = 1, \ldots, n_{ph},\ \hat{t}_0 = 0,\ \hat{t}_{n_{ph}} = t_f$$

$$r^{eq}(x(0), \ldots,\, x(t_f)) = 0, \tag{12}$$

$$r^{ineq}(x(0), \ldots,\, x(t_f)) \geq 0, \tag{13}$$

where we minimize the *objective function* (8) by modifying the *states* $x(t) = (q(t), \dot{q}(t))$ and the *controls* $u(t) = \tau(t)$. In (9) we find the right hand side of the set of equations of motion that are formulated separately for each of the n_{ph} model stages. With (10) stage transitions are modeled considering (5). The *path constraints* (11) define general limits to the states that are given by physiological constraints such as maximum joint angles. The equality constraints (12) and inequality constraints (13) ensure stage-wise general non-linear constraints such as contact point position at ground contact or positive contact forces.

This optimal control problem is solved using the *direct multiple-shooting method* that is implemented in the optimal control code MUSCOD-II [2,10,12]. This method uses a piecewise constant control discretization for the discretization of the optimal control problem. Furthermore, the original boundary value problem is transformed into a set of initial value problems with corresponding continuity and boundary conditions by the multiple shooting state parameterization.

With identical grids for the multiple-shooting state parameterization and the control discretization, this leads to a structured non-linear programming problem. This discretized problem can be efficiently solved with SQP algorithms adapted to the structure of the problem [10] as well as fast and reliable integration of the trajectories on the multiple-shooting intervals with regard to sensitivity information.

6 Results

We were able to simulate human gait fitting the model state variables to the experimental data in a least squares sense. Only the left step starting from *left toe off* until *right toe off* was simulated since we assume a healthy gait to be symmetric. The step is separated into the *left swing phase* (first 80 %) and the *initial double stance* (last 20 %). The trajectory of the simulated pelvis CoM in the global frame is compared to the measured trajectory in Fig. 4. The simulated and measured joint angle trajectories in the sagittal plane are shown in Fig. 5 for the left and the right leg, respectively.

The simulated and measured joint angle trajectories of both the right and the left leg as well as the simulated and measured displacement of the pelvis in the global frame show only small deviations. Larger deviations that occur in the double stance phase are due to the three-point foot model which does not precisely represent the complex kinematic behaviour of the human foot.

In the left and the middle plot of Fig. 6 the displacement in x and y-direction (in walking direction and to the left) of the ICP is depicted together with the displacement of the projection of the left and right ankle on the xy-plane for the whole step. The right plot shows the absolute distance from the ankle projections to the ICP. In x-direction the ICP trajectory is greater than the trajectory in both feet for the whole step with the position of the swing foot (*left*) converging to the ICP position during the swing phase. At the end of the swing phase when the left heel

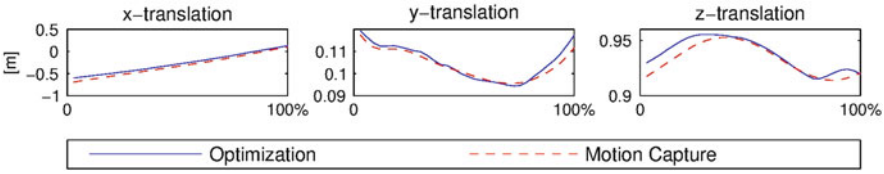

Fig. 4 Motion capture and simulation results for x,y,z-trajectories of the pelvis CoM

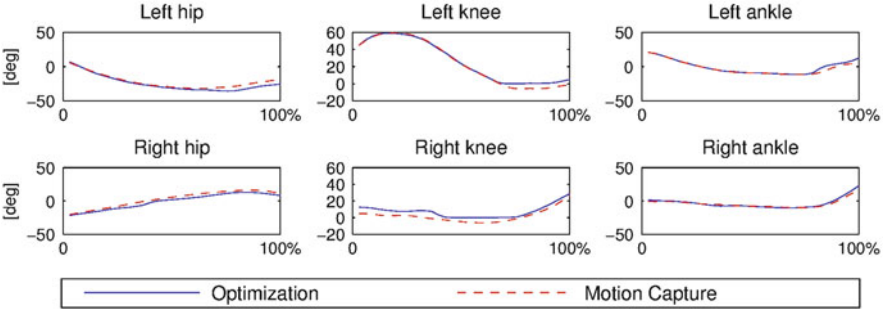

Fig. 5 Motion capture and simulation results for sagittal angles of right and left hip, knee and ankle

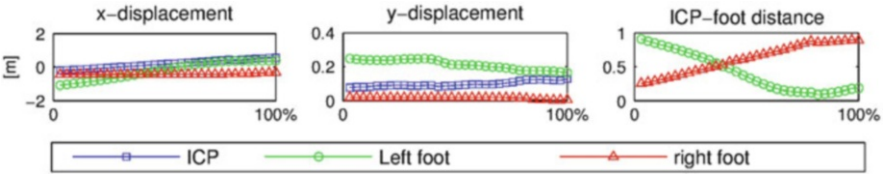

Fig. 6 The first two plots show the displacement of ICP and left and right ankle projection on the ground (xy-plane) in x and y-direction during the left step. The third plot shows the absolute distance from right and left foot to the ICP

gains ground contact, the x-position of the left ankle is 0.089 m less than the ICP x-position. In y-direction the ICP position is less than the left ankle position but greater than the position of the right ankle for the whole step with the swing foot position converging to the ICP position as close as 0.091 m. The absolute distance of the swing foot to the ICP is strictly monotonic decreasing during the swing phase indicating that the ICP is directly approached by the swing foot. The fact that the swing foot does not reach as far as the ICP can be interpreted as a foot placement strategy in human walking where the ability to come to a stop after each step is intentionally risked in order to initiate the next step.

7 Discussion

Our investigation was motivated by the need to understand the stability mechanisms of human walking in order to enhance stability in prosthetic devices. We hypothesize a correlation between the stability of human walking and the Instantaneous Capture Point. Although a human walker does not precisely step onto the ICP, we suggest that this point can be used to parameterize the location of the foot position. Using optimal control methods, joint angle trajectories of the multi-body model were generated that best fit the experimental data in a least-squares sense while considering the kinematic and physiological constraints of the human body. Our results on the actual foot placement at heel strike in relation to the current position of the Instantaneous Capture Point supported our assumption that human walking can be regarded as a sequence of falling and catching motions. The trajectory of the ICP was computed using the kinematic and kinetic data of the multi-body model that derived from the solution of our optimal control problem. The trajectory of the ground projection of the swing foot ankle was found to converge to the trajectory of the ICP. This suggests that the ICP trajectory can be used to compute the location of the foot position.

Acknowledgements The research project has been supported by the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp). I acknowledge the support of the Graduate Academy of the Heidelberg University in form of a Travel Grant, which enabled me to attend this conference.

References

1. Bellmann, M., Schmalz, T., Blumentritt, S.: Comparative biomechanical analysis of current microprocessor-controlled prosthetic knee joints. Arch. Phys. Med. Rehabil. **91**(4), 644–652 (2010)
2. Bock, H.G., Plitt, K.-J.: A multiple shooting algorithm for direct solution of optimal control problems. In: 9th IFAC World Congress, Budapest, pp. 242–247. International Federation of Automatic Control (1984)
3. de Leva, P.: Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. J. Biomech. **29**(9), 1223–1330 (1996)
4. Donker, S.F., Beek, P.J.: Interlimb coordination in prosthetic walking: effects of assymetry and walking velocity. Acta Psychol. **110**, 265–288 (2002)
5. Featherstone, R.: Rigid Body Dynamics Algorithms. Springer, New York (2007)
6. Felis, M.L.: RBDL – Rigid Body Dynamics Library. http://rbdl.bitbucket.org/. Cited 15 Apr 2012
7. Goswami, A., Kallem, V.: Rate of change of angular momentum and balance maintenance of biped robots. In: International Conference on Robotics and Automation, New Orleans (2004)
8. Koolen, T., de Boer, T., Rebula, J., Goswami, A., Pratt, J.E.: Capturability-based analysis and control of legged locomotion, Part 1: theory and application to three simple gait models. J. Biomech. **31**(9), 1094–1113 (2012)
9. Lamoth, C.J.C., Ainsworth, E., Polomski, W., Houdijk, H.: Variability and stability analysis of walking of transfemoral amputees. Med. Eng. Phys. (2010). doi:10.1016/j.medengphy.2010.07.001
10. Leineweber, D.B.: Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models. VDI-Fortschrittbericht, Reihe 3, No. 613, VDI-Verlag GmbH, Düsseldorf (1999)
11. Mombaur, K.: Using optimization to create self-stable human-like running. Robotica **27**, 321–330 (2009)
12. MUSCOD-II – a software package for numerical solution of optimal control problems involving differential-algebraic equations (DAE). IWR-SimOpt. http://www.iwr.uni-heidelberg.de/~agbock/RESEARCH/muscod.php. Cited 15 Apr 2012
13. Perry, J., Burnfield, J.M.: Gait Analysis – Normal and Pathological Function, 2nd edn. Slack Incorporated, Thorofare (2010)
14. Plug-in Gait: Vicon Plug-in Gait Product Guide (2010). Vicon Motion Systems Limited. http://www.vicon.com. Cited 15 Apr 2012
15. Pratt, J.E., Tedrake, R.: Velocity-Based Stability Margins for Fast Bipedal Walking. Fast Motions in Biomechanics and Robotics, Tsukuba (2006)
16. Pratt, J.E., Carff, J., Drakunov, S., Goswami, A.: Capture point: a step toward humanoid push recovery. In: 6th IEEE-RAS International Conference on Humanoid Robots, Genoa (2006)
17. Vukobratovic, M., Branislav, B.: Zero moment point – thirty five years of its life. Int. J. Humanoid Robot. **1**(1), 157–173 (2004)
18. Wieber, P.-B.: On the stability of walking systems. In: Proceedings of the International Workshop on Humanoid and Human Friendly Robotics, Tsukuba (2002)

High Performance Calculation of Magnetic Properties and Simulation of Nonequilibrium Phenomena in Nanofilms

Vitalii Yu. Kapitan and Konstantin V. Nefedev

Abstract Images of surface topography of ultrathin magnetic films have been used for Monte Carlo simulations in the frame of the ferromagnetic Ising model to study the hysteresis and thermal properties of nanomaterials. For high-performance calculations, a super-scalable parallel algorithm was used for finding the equilibrium configuration. The changing of the distribution of spins on the surface during the reversal of the magnetization and the dynamics of the nanodomain structure of thin magnetic films under the influence of a changing external magnetic field were investigated.

1 Introduction

Theoretical research and simulation of the physical properties of ultrathin ferromagnetic films due to the existence of fundamental problems of physics of magnetic phenomena are needed. Computer processing of experimental data and subsequent simulation of the surface of the magnet on the basis of these data allow for obtaining new information about the physical nature of ferromagnetism and ferromagnetic anisotropy and to visualize the processes of the reversal of magnetization in external fields. The physical properties of thin ferromagnetic films and the important points of view in their practical applications in microelectronics and computer technology were studied, as nanostructured soft magnetic thin films are currently the main material for manufacturing components of magnetic random access memory (MRAM) [1–6].

V.Yu. Kapitan (✉)
The School of Natural Sciences, Far Eastern Federal University, 8 Sukhanova St., Vladivostok 690950, Russia
e-mail: kapitan.vyu@dvfu.ru; kvy@live.ru

K.V. Nefedev
The School of Natural Sciences, Far Eastern Federal University, 8 Sukhanova St., Vladivostok 690950, Russia

Far Eastern Branch Russian Academy of Science, Institute of Applied Mathematics, 7 Radio St., Vladivostok 690041, Russia
e-mail: nefedev.kv@dvfu.ru

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_8

The problem of the existence of magnetic transitions in systems with long-range interaction between the particles has been discussed in terms of a random field [7–9]. In these papers, it has been shown that not only the long-range exchange interaction but also the usual short-range direct exchange interaction can lead to a state of a spin glass. In this regard, a comparison of the results of the analytical theory and numerical simulations, developed by the authors in [10–12], would be very useful and productive in terms of the development of our understanding of the processes of ordering and randomization.

The development of computing and supercomputer technology provides new classes of algorithms that can solve complex problems of numerical simulation and handle large and superlarge volumes of data [13]. Moreover, the level of sampling elements in a computer model of today is determined by the resolution of the scanning tunneling microscope (STM) or the atomic force microscope (AFM) [14]. The aim of this work is to develop a computer model and to create an application software for processing data obtained with an STM and an AFM, as well as to calculate the magnetic and structural properties of the quasi-nanocluster magnets and the simulation of magnetic hysteresis phenomena, with similar use of microscopic images for the simulation of the magnetic characteristics as considered in [15, 16].

2 The Model

A method for obtaining samples and experimental data was published in [17, 18]. The essence of the proposed method of computer image processing and subsequent Monte Carlo (MC) simulations is based on the fact that raster STM and AFM images were constructed by means of filling the three-dimensional space fcc lattice. The brightness of a pixel in the STM image is a function of the distance between the tip and the surface, so the image pixels were used to construct a magnet with a given number of atomic layers, the number of which was controlled by experimental methods. The selected algorithm is described in detail in [17]. Note that some aspects of the implementation of statistical Monte Carlo methods, including issues of their parallelization of simulation of magnetic phenomena and other problems of mathematical physics are presented in [19–25].

Model elements are located in the lattice sites-spins S_i, whose values have changed abruptly. Furthermore, a model lattice with a specified coordination number is constructed on the fcc lattice. In principle, the simulation can occur within any known model, such as the Ising model or the Heisenberg exchange integral of a certain value. We have used the Ising model, where each spin lattice model nanofilm interacts via direct exchange with its nearest neighbors (up to 12 neighbors). In the Metropolis algorithm, the value of the Boltzmann constant $k = 1$ and the value of the exchange integral $J = 1$ were given in dimensionless units. Dimensionless units are often used in numerical simulations to simplify the arithmetic operations with fractional numbers of very small magnitude (10^{-23}). The use of such units

can increase the efficiency of applied parallel algorithms for simulation and reduce the resource requirements of a computer system, such as RAM, and the size of sent messages between parallel streams that directly affect the performance of computing. In addition, it should be noted that the experimental determination of the Curie temperature or the blocking temperature of superparamagnetic particles to study monolayer nanostructures involves serious technical and engineering difficulties. Therefore, it is easy to recalculate the obtained qualitative characteristics of investigated nanosystems in the presence of experimental data on the Curie temperature and the spin of interacting magnetic ions and move to a quantitative predictive description of what is observed in numerical and physical experiment phenomena. The transition from these values to the experimental physical quantities can be carried out in accordance with expression (1):

$$J = \frac{3kT_c}{2zS}(S+1), \tag{1}$$

with z – the number of nearest neighbors, T_c – the Curie temperature, S – the spin of the ion.

3 A Parallel Metropolis Algorithm

The STM images processing algorithm is discussed in paper [18]. We formed the three-dimensional lattice of Ising spins from the data of BMP file. The number of rows in the array is the height of the studied image and the number of columns is the width of the picture, besides the "depth", is set equal to the number of layers of Co in the sample (based on experimental data). The parallelism of the algorithm is implemented by splitting the three-dimensional array of spins on the part (plane), the MPI library was used for their subsequent distribution, and accordingly, each of the planes was processed in a separate computation process. The selection of the spin for the coup and the execution of the MC simulation for it is produced in "checkerboard decomposition". This is done to solve the problem of boundary conditions, which are calculated for the selected spin of the plane (see Fig. 1). During the MC simulation with a part of the array in the frame of one thread, the initial values of the neighbors of the selected spin do not change, that is, in this case, for each step of temperature or field, half of the MC steps are performed for half of the system of spins (Fig. 1b) and then for the second part of the system of spins (Fig. 1c).

A parallel Metropolis algorithm is as follows:

1. In each process, we send the number of rows (planes) of the three-dimensional array of spins proportional to the number of processes (right braces in Fig. 2). We used an approach in which the maximum number of processes for the execution

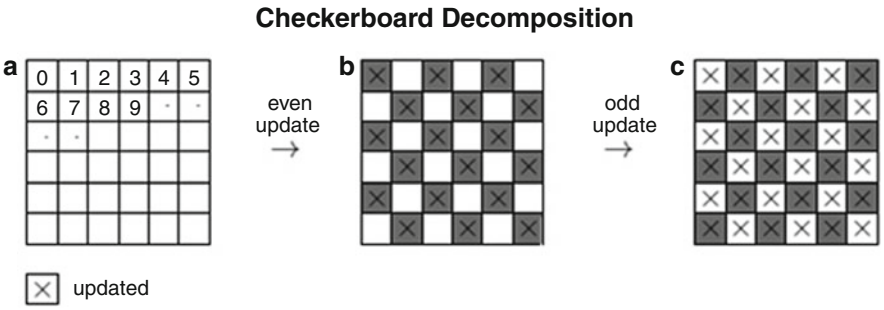

Fig. 1 Distribution of the matrix on the processes in a checkerboard pattern

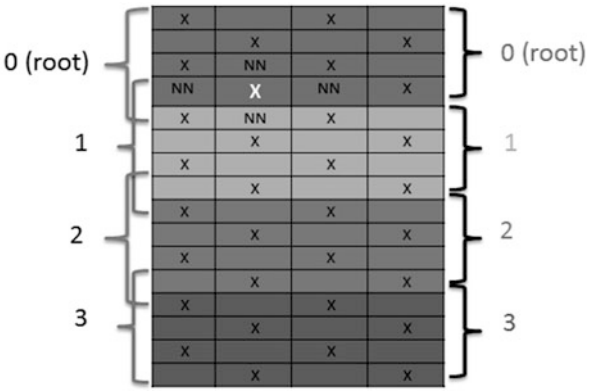

Fig. 2 Distribution of the rows of blocks to process

of the program is equal to one of the linear dimensions of the STM (AFM) image in pixels;

2. For each spin of the cobalt samples (see [17]), the highest possible number of nearest neighbors is $z = 12$. Therefore for the correct accounting of neighbors for spins located at the boundary of a block of rows in each process (Fig. 2, spin is marked by a white cross) we should take in account additional boundary rows. Nearest neighbors (NN), as seen in Fig. 2, are located in different processes (left braces in Fig. 2);
3. Distribution of the blocks of rows was done by using the in-line functions *MPI_Scatterv* and the gathering, respectively, by *MPI_Gatherv*.

The search for the equilibrium configuration is performed using a Monte Carlo method (Metropolis algorithm).

Hysteresis magnetic phenomena are explained as the effect of nonequilibrium. For the simulation of reversal magnetization processes in the monolayer samples, only surface points of the sample were used, and the number of MC steps was proportional to the number of lattice sites. The system of spins could not went into

an equilibrium state during the time of experiment. This leads to the phenomenon of magnetic hysteresis in the proposed model. The absence of an exact match of simulated and experimental data is due to the simplicity of the model.

For large systems of Ising spins, the movement toward equilibrium could be very slow, especially at low temperatures T. Therefore, in order to speed up obtaining the most probable configurations, the parallel computing scheme was used. The scalability of the algorithms is based on the independent scheme of calculations. Maximum scalability is determined by the pixel resolution of the STM (AFM) images. Splitting into two groups of processes by MPI command, *MPI_Comm_group*, could increase the efficiency of the calculation.

4 The Critical Concentration and Ferromagnetism

In [7], the authors have presented a method to calculate the critical concentration p_c required for a phase transition to ferromagnetism in the crystal lattices with different numbers of nearest neighbors. In this paper, the authors have determined the critical concentration of magnetic atoms for the transition to the ferromagnetic state for the monolayer and submonolayer samples of experimental data, which are given in [17, 18]. The critical concentration p_c for the transition to the ferromagnetic state at $T = 0$ was determined from the relation (2):

$$p_c = \frac{2}{z}. \tag{2}$$

Table 1 shows the critical concentration required for a phase transition to the ferromagnetic state, which implies that the sample of 1.5 ML at low temperatures should be in a state of the cluster ferromagnetism as $p < p_c$. That is confirmed by the data on the temperature dependence of magnetization for samples at 1.5, 2.0, 2.5, and 3.0 monolayer, shown in Fig. 3. It is clear that in our model, the number of nearest neighbors is a natural number for each selected node. However, the number of nearest neighbors varies depending on the vacancies, the presence or absence of which is determined in the computer processing of the STM images. It is known that in the mean-field model, the Curie temperature T_c is proportional to

Table 1 Calculated concentrations of atoms in comparison with the critical concentrations and Curie temperature for different samples

The number of monolayers	The average number of nearest neighbors $\bar{z}$	T_c	$p, at\%$	p_c
1.5 ML	3.615	3.00	0.38	0.55
2.0 ML	6.984	6.15	0.50	0.29
2.5 ML	8.177	7.80	0.63	0.24
3.0 ML	9.308	8.70	0.76	0.21

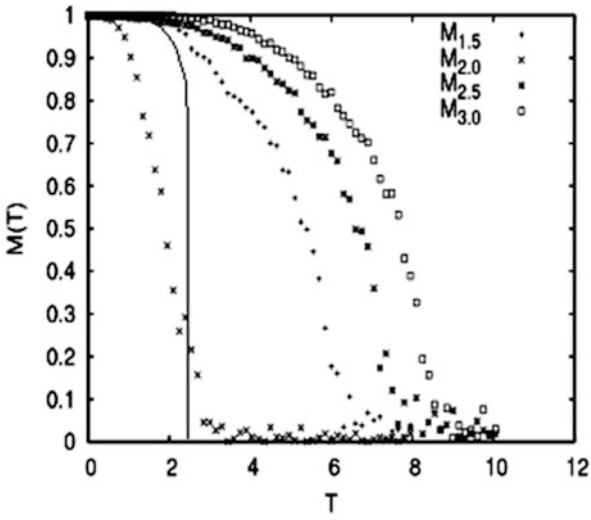

Fig. 3 Thermal destruction of the magnetization for surface nanostructures. The number of monolayers of the sample is shown in Table 1. The *solid line* indicates the temperature behavior of the magnetization in the solution of the Onsager

the number of nearest neighbors z in the lattice spin system. As seen from the table, the data obtained by numerical MC simulations confirm this dependence, for which the coefficient of proportionality is preserved, including substantially increasing the average number of neighbors.

5 Discussion of the Results of Numerical Simulations and Experimental Data

Reference [26] provides data about the magnetic properties of ultrathin films, which were epitaxially grown on Co single crystal Cu (111) using a magneto-optical Kerr effect. The magnetic behavior is compared with the samples that have the same structure but have different numbers of monolayers. The authors [26] attempted to establish a phenomenological law for the temperature dependence of magnetization for the number of monolayers of Cu coverage varying from 1 to 4. Moreover, a linear decrease in the magnetization was associated with the presence of islands and clusters, which show superparamagnetic behavior in a wide temperature range and cluster size. Similar results were obtained by Mossbauer microscopy and in [27, 28]. It is alleged that such superparamagnetic islands lead to the acceleration of the linear decrease in the magnetization with temperature, as opposed to the observed behavior of the Ising spins, which is discussed in [29]. As seen in Fig. 3 of this paper, the linear dependence of magnetization on the temperature of the samples with the number of monolayers of cobalt 2.0, 2.5, and 3.0 is not observed in the Ising model. Moreover, numerical simulation allows for visualizing the process of destruction of the magnetization on the surface of the model sample and in the surface layers.

The linearity of the behavior of the magnetization is observed only for the samples 1.5, but it was found that within the frame of model this sample did not show the superparamagnetic behavior of islands.

6 The Critical Field of Switching

The energy of each of the 2^N possible states of a system of N Ising spins interacting via direct exchange J, in an external magnetic field h, equals the sum of the energies of all pairwise interactions and the energy of interaction of the system with external magnetic field.

$$\mathscr{H} = -J \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} S_i S_j - h \sum_{i=1}^{N} S_i. \tag{3}$$

If a direct short-range exchange is used between the spins in the crystal lattice in which each node has z neighbors, then the second summation of the first term is carried out only on the nearest neighbors.

$$\mathscr{H} = -\sum_{i=1}^{N} S_i (\frac{J}{2} \sum_{j=1}^{z} S_j + h), \tag{4}$$

where 2 is introduced to compensate for double-counting pairwise interactions. In fact, the sum of external and internal fields is in parentheses.

In the partition function of the summation is over all the spins in the system

$$Z_N(h,T) = \sum_{s_1} \sum_{s_2} \cdots \sum_{s_n} \cdots \sum_{s_N} \exp[-\frac{\mathscr{H}}{kT}]. \tag{5}$$

The probability of one of the 2^N configurations determines the Gibbs factor (6)

$$P_k(h,T) = \frac{\exp[\frac{-\mathscr{H}}{kT}]}{\sum_{s_1} \sum_{s_2} \cdots \sum_{s_n} \cdots \sum_{s_N} \exp[\frac{-\mathscr{H}}{kT}]}. \tag{6}$$

If the system $(+J)$ spins, described by the Hamiltonian (3), and it is in an external magnetic field h at $T = 0$, then the system will be in the global energy minimum. This minimum corresponds to the magnetic state of complete ordering (ferromagnetism), and the probability of this event is equal to one, according to (5). The instantaneous change in the sign of the external field in the $-h$ should lead to an instantaneous change in the sign of the spin excess. This is due to decreasing of the Zeeman energy.

At finite temperature $T \neq 0$ and $T < T_c$, the sign of the external magnetic field changes from h to $-h$, which should also lead to a reversal of the spin excess. For an infinite number of spins N at finite temperature, there is an unlimited number of magnetic configurations with the same spin and the same excess energy, that is, an equal probability of realization. Changes in the sign of the field at $T \neq 0$ and $T < T_c$ should increase the probability of symmetric configurations with opposite value of the spin excess. Currently available methods for MC simulations, for example, the algorithm Metropolis-Hastings, to achieve equilibrium imply that the motion of the system in state space is similar to a Markov process, where the probability of each successive configuration depends on the previous implementation (6)

$$P(E_0) \to P(E_1, E_0) \to P(E_2, E_1) \to \cdots \to P(E_n, E_{n-1}). \tag{7}$$

The movement toward equilibrium in such an approach is carried out by successive reversal individual spins in accordance with rule (7).

The MC method for simulation of hysteresis phenomena in frame of the Ising model is used. A field of anisotropy h_a is a field of switching h_s in our model. It was introduced to save the sign of spin excess (8) in given direction in an external field

$$\mathscr{H} = -J \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} S_i S_j - h \sum_{i=1}^{N} S_i - h_a \sum_{i=1}^{N} |S_i|. \tag{8}$$

such that a magnetization reversal will occur at the equality of energy of the spin system in the external magnetic field and anisotropy energy.

To research the phenomenon of a magnetic hysteresis, the spin-flip probability was taken into account in the model, see Fig. 4. The transition to a state, which corresponds to the minimum energy, is possible only after overcoming a potential barrier.

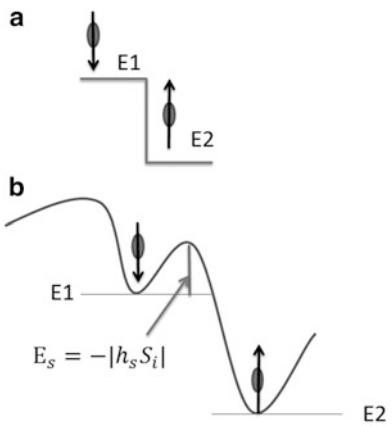

Fig. 4 Variants of the transition back to a state of minimum energy: (**a**) without the field of switching; (**b**) taking into account the field of switching

The Metropolis algorithm steps, taking into account the critical field of switching, are as follows

1. We calculate the interaction energy of the spin with its neighbors in the original position E_1 and in the new one E_2. The energy of the new position was compared with the energy of the old one;
2. The new position is accepted and becomes the initial for the next step, if $E_2 < E_1$. Otherwise calculate the probability of reversal p and generate a random number from the interval (0, 1) (9), where p is given by:

$$p = \begin{cases} 1, \ if E_2 < E_1, \\ e^{-\frac{\Delta E}{T}}, \ if E_2 > E_1. \end{cases} \tag{9}$$

3. If p is greater than this a random number, then the new position is accepted, otherwise it is rejected, and the old position remains the initial for a new attempt;
4. In addition to the spin-flip probability shown above, in the model was taken into account the spin-flip probability. It was calculated using the energy of the field of switching, which prevents the reversal (10):

$$p = e^{-\frac{\left|S_i h_S\right|}{T}}, \tag{10}$$

and then generate a random number from the interval (0,1);
5. If the flip probability is greater than this random number, then the MC-steps (1–3) are performed, else the orientation of the spin does not change and the algorithm moves to the next spin.

The critical field of switching of a magnetic particle (macrospins) is a field necessary to change the magnetization of the magnetic particles. The effective field of switching is introduced to account for the spin-orbit interaction, which leads to the well-known phenomenon of anisotropy in the macroscopic scale. In ultrafine materials, a scatter in the values of critical fields, so-called coercive spectrum, can be observed. In a computer model of epitaxial nanostructures in this approximation, all the spins interact, with some introduced to an average effective field, which supports the direction of spin. The reorientation of the spin occurs when the Zeeman energy (energy of the spin in an external field) and the energy of the spin in the effective field of switching are equal. The probability of switching is greater when the temperature is higher, and as the temperature increases, so does the probability of thermodynamic fluctuations, the probability of overcoming the energy barrier created by the effective field, and thus the probability of switching off the local energy minimum. In our model, we used the following values of the field of switching for the samples with different number of monolayers: 1.5 ML - 2 r. u., 2.0 ML - 8 r. u., 2.5 ML - 4 r. u., and 3.0 ML - 2 r. u. The different values of the field of switching were due to the need to provide qualitative agreement with the experiment.

7 Visualization of the Magnetization Reversal in an External Magnetic Field

The practical value of the proposed method also consists in the fact that the results of MC simulations of collective effects in nano-objects can be compared with experimental results using the parallel magneto-optical Kerr effect (PMOKE), which allows you to directly observe the magnetic state of the surface nanostructure shown in Fig. 5. The formation of the simulated PMOKE image is as follows: take into account only the surface atoms of the sample, and if the spin is up, then draw a white pixel, and if the spin is down, then draw a black pixel. The approach used for the simulation of images allows the visual to trace the magnetization reversal of the spins of the sample.

The magnetic state of the surface of the discontinuous film 2.5 ML cobalt in four of the most interesting points (Fig. 5a–d) of the hysteresis loop was obtained by computer processing of the STM image of Fig. 5e and subsequent numerical simulation based on the critical field of the magnetization reversal. From Fig. 5a–d, it can be seen and it is very logical that in an external magnetic field, the energy minimization of the Ising spin system is due to the decrease in the number of spins with the direction "antiparallel to the field". The corresponding images of the magnetization distribution are shown in Fig. 5b, c. Magnetization reversal process occurs primarily at the boundaries of clusters. In places where the spins have the

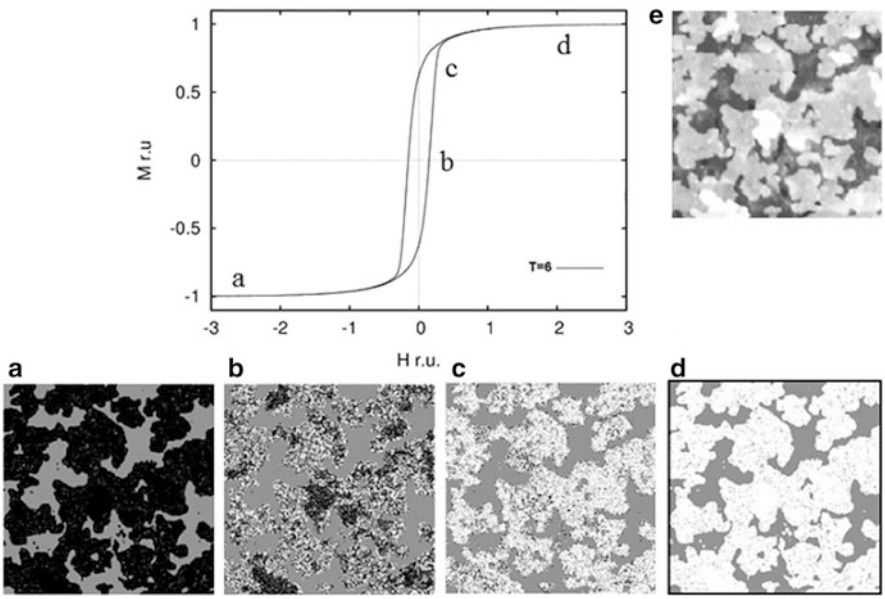

Fig. 5 Hysteresis loop for a sample of 2.5 ML; (**a–d**) Simulated PMOKE-images; (**e**) STM-image of a sample of 2.5 ML

smallest number of nearest neighbors reversal magnetization takes place more easy. Bright cluster boundaries in Fig. 5b and dark cluster boundaries in Fig. 5c indicate instability of spins in presence of thermodynamic fluctuations, despite the fact that the sample is at a temperature below T_c. The system proceeds to point "c" where there is great magnetization in the absence of an external field.

8 Conclusion

The three-dimensional array of spins at the designed lattice sites was divided into subarrays, whose number was divisible by the number of computational processes (nucleus). Access to elements of a subvector in the Metropolis algorithm was implemented in a staggered manner. This allows to avoid conflicts in the merge calculation operations.

Within the frame of the model, hysteretic magnetic phenomena are explained as the effect of nonequilibrium. It was shown that if the relaxation time is a lot more than the time of experiment, this could lead to the phenomenon of magnetic hysteresis.

Developed models, algorithms, and software have good scalability. The reason for this is to use the independent scheme of calculations. The simulation results and theoretical estimates for Co nanostructures are in qualitative agreement with the experiment for determining the concentrations of phase transitions in the ferromagnetic state.

This work was supported by Scientific Fund of Far Eastern Federal University (FEFU) #12 − 07 − 13000 − 18/13 and the state task of the Ministry of Education and Science of Russia #559.

References

1. Gallagher, W., Parkin, S.: Development of the magnetic tunnel junction MRAM at IBM: from first junctions to a 16-mb MRAM demonstrator chip. IBM J. Res. Dev. **50**(1), 5–23 (2006)
2. Tehrani, S., Chen, E., Durlam, M., DeHerrera, M., Slaughter, J., Shi, J., Kerszykowski, G.: High density submicron magnetoresistive random access memory. J. Appl. Phys. **85**(8), 5822–5827 (1999)
3. Tehrani, S., Engel, B., Slaughter, J., Chen, E., DeHerrera, M., Durlam, M., Naji, P., Whig, R., Janesky, J., Calder, J.: Recent developments in magnetic tunnel junction MRAM. IEEE Trans. Magn. **36**(5), 2752–2757 (2000)
4. Melo, L., Rodrigues, L., Freitas, P.: Novel spin-valve memory architecture. IEEE Trans. Magn. **33**(5), 3295–3297 (1997)
5. Boeve, H., Bruynseraede, C., Das, J., Dessein, K., Borghs, G., De Boeck, J., Sousa, R., Melo, L., Freitas, P.: Technology assessment for the implementation of magnetoresistive elements with semiconductor components in magnetic random access memory (MRAM) architectures. IEEE Trans. Magn. **35**(5), 2820–2825 (1999)

6. Daughton, J.: GMR applications. J. Magn. Magn. Mater. **192**(2), 334–342 (1999)
7. Nefedev, K., Savunov, M., Belokon, V.: Finite interaction range spin glass in the Ising model. Phys. Solid State **48**(9), 1746–1753 (2006)
8. Belokon, V., Nefedev, K.: Distribution function for random interaction fields in disordered magnets: spin and macrospin glass. J. Exp. Theor. Phys. **93**(1), 136–142 (2001)
9. Nefedev, K., Belokon, V.: Magnetic phase transitions in amorphous systems with competing exchange interactions. Phys. Solid State **44**(9), 1708–1710 (2002)
10. Ivanov, Y., Nefedev, K., Iljin, A., Pustovalov, E., Chebotkevich, L.: Magnetization reversal of nanodots with different magnetic anisotropy and magnetostatic energy. J. Phys.: Conf. Ser. **266**(1), 012117. IOP Publishing, 2011
11. Nefedev, K., Ivanov, Y., Peretyatko, A., Belokon, V.: Magnetic states of nanodot arrays. Physical and numerical experiments. Solid State Phenom. **168**, 325–328 (2011)
12. Nefedev, K., Ivanov, Y., Peretyatko, A.: Parallel algorithm for calculation of the nanodot magnetization. In: Methods and Tools of Parallel Programming Multicomputers, pp. 260–267. Springer, Berlin/Heidelberg (2011)
13. Nefedev, K., Kapitan, V.: Spin-glass-like behavior and concentration phase transitions in model of monolayer two-sublattice magnetics. Appl. Mech. Mater. **328**, 841–844 (2013)
14. Nefedev, K., Kapitan, V., Shevchenko, Y.: The inverse task for magnetic force microscopy data. Appl. Mech. Mater. **328**, 744–747 (2013)
15. Rudnev, V., Ustinov, A., Lukiyanchuk, I., Kharitonskii, P., Frolov, A., Morozova, V., Tkachenko, I., Adigamova, M.: Magnetic properties of plasma electrolytic iron-containing oxide coatings on aluminum and simulation of demagnetizing process. Solid State Phenom. **168**, 289–291 (2011)
16. Kharitonskii, P., Frolov, A., Rudnev, V., Ustinov, A., Lukiyanchuk, I., Morozova, V.: Magnetic properties of iron-containing coatings formed by plasma-electrolytic oxidation. Bull. Russ. Acad. Sci. Phys. **74**(10), 1404–1406 (2010)
17. Ivanov, Y., Ilin, A., Davydenko, A., Zotov, A.: Optimal Cu buffer layer thickness for growing epitaxial Co overlayers on Si (111) 7× 7. J. Appl. Phys. **110**(8), 083505 (2011)
18. Ivanov, Y., Nefedev, K., Ilin, A., Kapitan, V.: Ferromagnetism in epitaxial fcc Co films on Si (111) 7×7 with Cu buffer layer. Phys. Procedia **23**, 128–131 (2012)
19. Kovtanyuk, A., Botkin, N., Hoffmann, K.-H.: Numerical simulations of a coupled radiative–conductive heat transfer model using a modified Monte Carlo method. Int. J. Heat Mass Transf. **55**(4), 649–654 (2012)
20. Kovtanyuk, A., Prokhorov, I.: Tomography problem for the polarized-radiation transfer equation. J. Inverse Ill-Posed Probl. JIIP **14**(6), 609–620 (2006)
21. Kovtanyuk, A., Nefedev, K., Prokhorov, I.: Advanced computing method for solving of the polarized-radiation transfer equation. In: Methods and Tools of Parallel Programming Multicomputers, pp. 268–276. Springer, Berlin/Heidelberg (2011)
22. Nefedev, K., Belokon, V., Kapitan, V., Dyachenko, O.: Monte Carlo simulation of lattice systems with RKKY interaction. J. Phys. Conf.: Ser. **490**(1), 012163 (2014). IOP Publishing
23. Kovtanyuk, A., Prokhorov, I.: A boundary-value problem for the polarized-radiation transfer equation with Fresnel interface conditions for a layered medium. J. Comput. Appl. Math. **235**(8), 2006–2014 (2011)
24. Belokon, V., Kapitan, V., Dyachenko, O.: Concentration of magnetic transitions in dilute magnetic materials. J. Phys. Conf.: Ser. **490**(1), 012165 (2014). IOP Publishing
25. Kovtanyuk, A., Prokhorov, I.: Numerical solution of the inverse problem for the polarized-radiation transfer equation. Numer. Anal. Appl. **1**(1), 46–57 (2008)
26. Huang, F., Mankey, G., Willis, R.: Interfacial anisotropy and magnetic transition of cobalt films on Cu (111). J. Appl. Phys. **75**(10), 6406–6408 (1994)
27. Bayreuther, G.: Experiments on ferromagnetic surfaces and thin films. J. Magn. Magn. Mater. **38**(3), 273–286 (1983)

28. Mauri, D., Scholl, D., Siegmann, H., Kay, E.: Universal thermal stabilization of the magnetization in ultrathin ferromagnetic films. Phys. Rev. Lett. **62**(16), 1900–1903 (1989)
29. Kohlhepp, J., Elmers, H., Cordes, S., Gradmann, U.: Power laws of magnetization in ferromagnetic monolayers and the two-dimensional Ising model. Phys. Rev. B **45**(21), 12 287–12 291 (1992)

Inverse Problem of the Calculus of Variations for Second Order Differential Equations with Deviating Arguments

Galina Kurina

Abstract The paper is devoted to the solvability conditions for the inverse problem of the calculus of variations for second order differential equations with deviating arguments.

Also we are interested in explicit formulae for the functional of the inverse problem defined by the integral that differs from the standard one by that the required function has a retarded argument.

1 Introduction

We consider the following problem. Let a second order differential equation with deviating arguments be given. It is required to know whether there exists a functional defined by an integral for which this equation is a necessary condition for the extremum of the functional. If such a functional exists, then we must find it.

This problem is called the inverse problem of the calculus of variations.

For differential equations without deviating arguments, the inverse problem of the calculus of variations was considered in, e.g., [6].

The survey [2] is devoted to various approaches and results on the inverse problems of the calculus of variations.

If the one-parameter group of variational symmetries is known, then we can find a solution of Euler's equation of the second order in quadratures (see, e.g., [4]).

Differential equations with deviating arguments have numerous applications in automatic control theory, in the theory of self-oscillating systems, in the study of duct-burning problems in rocketry. They occur in problems of long-term forecasting in economics, in various biophysical problems, etc. The number of these applications is steadily increasing.

The reason for the occurrence of delays in variational problems in control theory is sometimes related to time delays incurred in signal transmission. However,

G. Kurina (✉)
Voronezh State University, Universitetskaya pl., 1, 394006 Voronezh, Russia

Voronezh Institute of Law and Economics, Leninskii pr., 119-A, 394042 Voronezh, Russia
e-mail: kurina@math.vsu.ru; 2gkurina@mail.ru

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_9

usually it is due to simplifying assumptions that reduce the action of intermediate transmitting and amplifying devices in the system to delays in the transmission of signals (see [1]).

The inverse problem of the calculus of variations for asymmetrical problems in the case of differential equations of second order with deviating arguments was considered in [7]. Some results concerning inverse problems in the symmetrical case were obtained in [5]. They will be discussed in Sect. 3.

The paper is organized as follows. In Sect. 2 we present the results devoted to solving inverse problems in the asymmetrical case. Section 3 deals with symmetrical case. Some illustrative examples are also given.

2 Asymmetrical Problems

We start with the results from [7] devoted to the so-called asymmetrical problems.

2.1 Necessary Condition for an Extremum of a Functional with Delay

Problem: find the extremum of the functional

$$J(y) = \int_a^b F\big(x, y(x), y(x-\theta), y'(x)\big)\, dx \tag{1}$$

under the given conditions

$$y(x) = \varphi(x), \quad a-\theta \le x \le a; \quad y(b) = y_b. \tag{2}$$

Admissible functions y are real valued and belong to W_2^1.

Here $y : [a,b] \mapsto \mathbb{R}$, $\theta \in (0, b-a)$ and y_b - are given numbers, $\varphi : [a-\theta, a] \mapsto \mathbb{R}$ and $F : [a,b] \times \mathbb{R}^3 \mapsto \mathbb{R}$ are given functions such that $\varphi \in C^1[a-\theta, a]$ and the function F is continuous and twice continuously differentiable with respect to all of its arguments.

We will use the notation

$$\tilde{F} = F(x+\theta, y(x+\theta), y(x), y'(x+\theta)),$$

$$\Phi = \begin{cases} F + \tilde{F}, & x \in [a, b-\theta], \\ F, & x \in (b-\theta, b]. \end{cases}$$

Theorem 1 ([3]). *Let an extremum of functional (1) under boundary conditions (2) be attained at y in the space W_2^1. Then y almost everywhere on $[a,b]$ satisfies the equation*

$$\Phi_{y(x)} - \frac{d}{dx}\Phi_{y'(x)} = 0. \tag{3}$$

Relation (3) is a generalization of Euler's equation to the case of a functional with retarded argument.

We can obtain the assertion of Theorem 1 (see [7]) by using well known control optimality conditions in the form of Pontryagin's maximum principle for problems with delay.

We obtain from (3) the analog of Euler's equation in the expanded form, namely,

$$\begin{gathered} F_{y(x)} + \tilde{F}_{y(x)} - F_{xy'(x)} - F_{y(x)y'(x)}y'(x) - F_{y(x-\theta)y'(x)}y'(x-\theta) - \\ -F_{y'(x)y'(x)}y''(x) = 0, \quad x \in [a, b-\theta], \end{gathered} \tag{4}$$

$$\begin{gathered} F_{y(x)} - F_{xy'(x)} - F_{y(x)y'(x)}y'(x) - F_{y(x-\theta)y'(x)}y'(x-\theta) - \\ -F_{y'(x)y'(x)}y''(x) = 0, \quad x \in (b-\theta, b]. \end{gathered} \tag{5}$$

2.2 The Inverse Problem of the Calculus of Variations

Next, let a second order differential equation with deviating arguments be given. Let us try to find a functional of the form (1) such that the given equation is a necessary condition for the extremum. In fact, we need to find a function F of four arguments defining the functional (1).

Thus, suppose that there is an equation depending on variables

$$x,\ y(x),\ y(x \pm \theta),\ y'(x),\ y'(x \pm \theta),\ y''(x),\ y''(x \pm \theta).$$

It follows from the necessary condition of extremum (4) and (5) for a functional of the form (1) that the given equation must be given by different formulas on two intervals $[a, b-\theta]$ and $(b-\theta, b]$. In addition, we also see that it must be linear with respect to

$$y'(x-\theta),\ y''(x)$$

and not containing

$$y''(x-\theta),\ y''(x+\theta).$$

Thus, in view of (4) and (5) the given equation has the following form:

$$Ay''(x) + B + Cy'(x-\theta) + \tilde{D} = 0, \quad x \in [a, b-\theta], \tag{6}$$

$$Ay''(x) + B + Cy'(x-\theta) = 0, \quad x \in (b-\theta, b], \tag{7}$$

where the functions A, B, C, D depend on x, $y(x), y(x-\theta)$, $y'(x)$; we assume that they are twice continuously differentiable. In fact, these smoothness assumptions can be weakened (see formulas below).

Let us refine the settings of the inverse problem. For given Eqs. (6) and (7) it is required to find a functional of the form (1) for which the left-hand side of Euler's equation coincides with the left-hand side of Eqs. (6) and (7) for all twice continuously differentiable functions $y(\cdot)$ satisfying conditions (2).

Theorem 2. *The inverse problem of the calculus of variations has a solution if and only if the functions in (6) and (7) satisfy the following conditions*

$$C_{y'(x)} - A_{y(x-\theta)} = 0, \quad x \in [a, b],$$

$$B_{y'(x)} - A_x - y'(x)A_{y(x)} = 0, \quad x \in [a, b],$$

$$D_{y'(x)} + C = 0, \quad x \in [a+\theta, b],$$

$$D_{y(x)} + \int_0^{y'(x)} C_{y(x)}dy'(x) - G_{y(x)y(x-\theta)} = 0, \quad x \in [a+\theta, b],$$

where

$$G = G(x, y(x), y(x-\theta)) = \int_0^{y(x)} \Big(B - \int_0^{y'(x)} B_{y'(x)}dy'(x) + E_x\Big)dy(x) + Q,$$

$$Q = \begin{cases} 0 & , \quad x \in (b-\theta, b], \\ \int\limits_0^{y(x)} \Big(\tilde{D} + \int\limits_0^{y'(x+\theta)} \tilde{C}dy'(x+\theta) - \tilde{G}_{y(x)}\Big)dy(x), & x \in [a, b-\theta], \end{cases}$$

$$E = E(x, y(x), y(x-\theta)) = \int_0^{y(x-\theta)} \Big(\int_0^{y'(x)} A_{y(x-\theta)}dy'(x) - C\Big)dy(x-\theta), x \in [a, b],$$

the function Q for $x \in [a, b-\theta]$ is determined successively on the intervals

$$(b-k\theta, b-(k-1)\theta], \ k = 2, \ldots : b-k\theta \geq a.$$

Further, the function F, which determines the solution of the form (1) of the inverse problem of the calculus of variations, can be expressed as follows:

$$F = -\int_0^{y'(x)} \Big(\int_0^{y'(x)} A\, dy'(x) \Big) dy'(x) + Ey'(x) + G, \quad x \in [a, b]. \tag{8}$$

2.3 Example 1

Let us solve the inverse problem of the calculus of variations for equations of the form

$$M(x, y(x), y(x-\theta), y'(x), y'(x-\theta), y''(x)) = -(2x^2 - \alpha y(x-\theta))y''(x) + \\ +24y(x) - 4xy'(x) + \alpha y'(x)y'(x-\theta) = 0, \quad x \in (b-\theta, b], \tag{9}$$

$$M(x, y(x), y(x-\theta), y'(x), y'(x-\theta), y''(x)) - \frac{\alpha}{2}(y'(x+\theta))^2 = 0, \quad x \in [a, b-\theta]. \tag{10}$$

Here α and $0 < \theta < b - a$ are parameters.

Comparing (9) and (10) with (7) and (6), respectively, we get

$$A = -2x^2 + \alpha y(x-\theta), \quad B = 24y(x) - 4xy'(x), \quad C = \alpha y'(x),$$

$$D = -\frac{\alpha}{2}(y'(x))^2.$$

It is easy to check that the solvability conditions obtained in Theorem 2 are valid in this case.

Using (8), we obtain the function

$$F = 12(y(x))^2 + x^2(y'(x))^2 - \frac{\alpha}{2} y(x-\theta)(y'(x))^2,$$

which determines the functional (1).

3 Symmetrical Problems

Further, we consider the so-called symmetrical problems where admissible functions y satisfy the following conditions:

$$y(x) = \begin{cases} \varphi(x), & x \in [a-\theta, a], \\ \psi(x), & x \in [b-\theta, b], \end{cases} \tag{11}$$

where φ and ψ are given continuously differentiable functions.

3.1 Euler's Equation

Problem: find the extremum of the functional

$$J(y) = \int_a^b F(x, y(x), y(x-\theta), y'(x), y'(x-\theta))dx \tag{12}$$

under the given conditions (11).

Here the function F is assumed to be twice continuously differentiable with respect to all arguments.

In contrast to Sect. 2, the integrand contains the derivative depending on a retarded argument and the given symmetrical conditions for a required function y are different from the previous asymmetrical conditions (2).

Set

$$\tilde{F} := F(x+\theta, y(x+\theta), y(x), y'(x+\theta), y(x)),$$
$$\Phi := F + \tilde{F}.$$

Theorem 3 ([3]). *If functional (12) under boundary conditions (11) attains on $y(x)$ an extremum in W_2^1 then $y(x)$ must satisfy the equation*

$$\Phi_{y(x)} - \frac{d}{dx}\Phi_{y'(x)} = 0 \tag{13}$$

almost everywhere on $[a, b-\theta]$.

Relation (13) is a generalization of Euler's equation to the case of functional (12).

From (13), we obtain the Euler's equation in the expanded form

$$\begin{gathered}\Phi_{y(x)} - \Phi_{xy'(x)} - \Phi_{y(x)y'(x)}y'(x) - \Phi_{y(x-\theta)y'(x)}y'(x-\theta) - \\ -\Phi_{y'(x)y'(x)}y''(x) - \Phi_{y'(x-\theta)y'(x)}y''(x-\theta) - \Phi_{y(x+\theta)y'(x)}y'(x+\theta) - \\ -\Phi_{y'(x+\theta)y'(x)}y''(x+\theta) = 0.\end{gathered} \tag{14}$$

3.2 The Inverse Problem of the Calculus of Variations

Taking into account the form of the Euler's equation (14), we will consider the following equation with deviating arguments

$$Ay''(x-\theta) + By''(x) + Cy''(x+\theta) + D = 0. \tag{15}$$

Here $D := D_1 + D_2$, the functions A and D_1 depend on

$$x, y(x), y(x-\theta), y'(x), y'(x-\theta),$$

the functions C and D_2 depend on

$$x+\theta, y(x+\theta), y(x), y'(x+\theta), y'(x),$$

and the function B depends on

$$x, y(x), y(x \pm \theta), y'(x), y'(x \pm \theta).$$

All functions in (15) are continuously differentiable with respect to the arguments. It is also assumed that admissible functions y satisfy (11).

Consider the inverse problem of the calculus of variations for Eq. (15). Namely, we seek for a functional of the form (12), for which Euler's equation coincides with Eq. (15).

The next result provides the necessary solvability conditions for the given inverse problem.

Theorem 4. *If the inverse problem of the calculus of variations for Eq. (15) has a solution in the form of the functional (12) with the function F, which is three times continuously differentiable, then the functions A, B, C, D satisfy the following conditions:*

$$A_{y'(x)} = B_{y'(x-\theta)}, \tag{16}$$

$$B_{y'(x+\theta)} = C_{y'(x)}, \tag{17}$$

$$\tilde{A} = C, \tag{18}$$

$$D_{y'(x)} = B_x + B_{y(x)}y'(x) + B_{y(x-\theta)}y'(x-\theta) + B_{y(x+\theta)}y'(x+\theta), \tag{19}$$

$$D_{y'(x+\theta)} = -\tilde{D}_{y'(x)} + 2(\tilde{A}_x + \tilde{A}_{y(x+\theta)})y'(x+\theta) + \tilde{A}_{y(x)}y'(x)), \tag{20}$$

$$\begin{gathered}\tfrac{d}{dx}(D_{y'(x+\theta)} - \tilde{D}_{y'(x)}) = 2((B_{y(x+\theta)} - \tilde{A}_{y(x)})y''(x) + \\ +(C_{y(x+\theta)} - \tilde{B}_{y(x)})y''(x+\theta) + D_{y(x+\theta)} - \tilde{D}_{y(x)}).\end{gathered} \tag{21}$$

Proof. If the inverse problem of the calculus of variations has a solution, then there is a function F such that Eq. (15) is the Euler's equation for the functional (12) with the integrand F. Taking into account (14), we obtain

$$A = -F_{y'(x-\theta)y'(x)}, \tag{22}$$

$$B = -\Phi_{y'(x)y'(x)}, \tag{23}$$

$$C = -\tilde{F}_{y'(x+\theta)y'(x)}, \tag{24}$$

$$D = \Phi_{y(x)} - \Phi_{xy'(x)} - \Phi_{y(x)y'(x)}y'(x) - \Phi_{y(x-\theta)y'(x)}y'(x-\theta) - \\ -\Phi_{y(x+\theta)y'(x)}y'(x+\theta). \tag{25}$$

The latter implies conditions (16)–(21) in Theorem 4. □

Remark 1. Necessary and sufficient solvability conditions for the inverse problem of the calculus of variations in the case of a system of differential-difference equations of second order under other assumptions on required functions are given in [5]. We note that in that paper there are some misprints.

Using the detailed structure of given Eq. (15), Theorem 4 provides conditions which are simpler than the conditions in [5]. Moreover, the conditions from [5] follow from our conditions in Theorem 4. In particular, the cumbersome condition (12) from [5] is equivalent for a one-dimensional function to the simpler conditions (16), (17), and (19) in Theorem 4.

Set

$$E := \int_0^{y'(x)} \Big(\int_0^{y'(x-\theta)} A_{y'(x)}dy'(x-\theta) + \int_0^{y'(x+\theta)} \tilde{A}_{y'(x)}dy'(x+\theta) - B \Big) dy'(x), \tag{26}$$

$$G := \int_0^{y(x)} \Big(D_1 + \int_0^{y'(x)} \Big(\int_0^{y'(x-\theta)} A_{y(x)}dy'(x-\theta) - E_{y(x)} \Big) dy'(x) - \\ \int_0^{y'(x-\theta)} A_x dy'(x-\theta) + E_x - \Big(\int_0^{y'(x-\theta)} A_{y(x)}dy'(x-\theta) - E_{y(x)} \Big) y'(x) - \\ -\Big(\int_0^{y'(x-\theta)} A_{y(x-\theta)}dy'(x-\theta) - E_{y(x-\theta)} \Big) y'(x-\theta) \Big) dy(x). \tag{27}$$

Theorem 5. *If the coefficients of (15) satisfy (16)–(18) and*

$$B_{y(x+\theta)} = 0 \quad \textit{under} \quad y'(x+\theta) = 0, \quad G_{y'(x)} = 0, \quad G_{y'(x-\theta)} = 0, \\ D_2 = \tilde{G}_{y(x)} + \int_0^{y'(x+\theta)} \Big(-\int_0^{y'(x)} \tilde{A}_{y(x)}dy'(x) + \tilde{E}_{y(x)} + \tilde{A}_x + \\ + \tilde{A}_{y(x+\theta)} + \tilde{A}_{y(x)}y'(x) \Big) dy'(x+\theta) + \int_0^{y'(x+\theta)} \tilde{A}_{y(x+\theta)}dy'(x+\theta)y'(x+\theta), \tag{28}$$

then the function F determining the solution (12) of the inverse problem of the calculus of variations can be expressed as follows:

$$F = \int_0^{y'(x)} \Big(E - \int_0^{y'(x-\theta)} A dy'(x-\theta)\Big) dy'(x) + G. \tag{29}$$

Proof. Formulae (26), (27), and (29) are obtained taking into account relations (22)–(25). Using the given conditions, we obtain the assertion of this theorem by immediate transformations. □

Remark 2. With the help of the explicit formula (29) we can easily find a solution of the inverse problem of the calculus of variations. For general nonlinear problems, the formula (2) for the solution from [5] is not always practically applicable.

3.3 Example 2

Consider the following equation

$$\begin{gathered} -y'(x-\theta)y''(x-\theta) - (2x^2 - y(x-\theta) + y'(x+\theta))y''(x) - \\ -y'(x)y''(x+\theta) + 24y(x) - (4x - y'(x-\theta))y'(x) - \tfrac{1}{2}(y'(x+\theta))^2 = 0. \end{gathered} \tag{30}$$

Let us solve the corresponding inverse problem of the calculus of variations. Comparing (30) with (15), we get

$$\begin{gathered} A = -y'(x-\theta), \quad B = -2x^2 + y(x-\theta) - y'(x+\theta), \quad C = -y'(x), \\ D_1 = 24y(x) - (4x - y'(x-\theta))y'(x), \quad D_2 = -\tfrac{1}{2}(y'(x+\theta))^2. \end{gathered}$$

It is easy to verify that the solvability conditions obtained in Theorems 4 and 5 are valid in this case.

Therefore, by (26) and (27), we get

$$E = (2x^2 - y(x-\theta))y'(x), \quad G = 12y^2(x).$$

Finally, using (29), we obtain the function

$$F = (x^2 - \frac{1}{2}y(x-\theta))(y'(x))^2 + \frac{1}{2}y'(x)(y'(x-\theta))^2 + 12y^2(x),$$

which defines the required functional (12). It is not difficult to verify directly that (30) is really the Euler's equation for functional (12) with the obtained function F.

3.4 Example 3 (The Linear Case)

Let us consider the linear equation with deviating arguments from [5]

$$\sum_{l=0}^{2}(A_l(x)y^{(l)}(x-\theta)+B_l(x)y^{(l)}(x)+C_l(x)y^{(l)}(x+\theta))+p(x)=0, \qquad (31)$$

where $A_l, B_l, C_l, p \in C^2$. The solution of the inverse problem of the calculus of variations for this equation is presented in [5]. We note that in that paper there is the misprint in the formula for the solution, namely, the number 2 must be stand before the term $u^T(t)\,p(t)$.

Comparing (31) with (15) we have

$$A=A_2(x),\quad B=B_2(x),\quad C=C_2(x),\quad D_1=A_0(x)y(x-\theta)+B_0(x)y(x)+p(x)+$$
$$+A_1(x)y'(x-\theta)+B_1(x)y'(x),\quad D_2=C_0(x)y(x+\theta)+C_1(x)y'(x+\theta).$$

From Theorem 4 we obtain conditions (20) in [5]. The conditions from Theorem 5 give the additional condition for coefficients in (31)

$$A_1(x)=A_2'(x),\quad C_0(x)=A_0(x+\theta),\quad A_2'(x+\theta)=C_1(x). \qquad (32)$$

In view of (26), (27), (19), and (29) we have

$$F=-\frac{1}{2}B_2(x)(y'(x))^2-A_2(x)y'(x-\theta)y'(x)+(A_0(x)y(x-\theta)+p(x))y(x)+$$
$$+\frac{1}{2}B_0(x)(y(x))^2.$$

The last expression coincides with the integrand in [5] if we take into account the conditions in Theorem 5 and correct the indicated misprint in [5].

In order to obtain the solution of the inverse problem of the calculus of variations in the form from [5] without additional conditions (32), we must take in (29)

$$E=\frac{1}{2}(y(x)(B_1(x)-B_2'(x))-2B_2(x)y'(x)+(C_1(x-\theta)-C_2'(x-\theta))y(x-\theta)),$$

$$G=\frac{1}{2}(y(x)(A_0(x)+C_0(x-\theta))y(x-\theta)+y(x)(A_1(x)-A_2'(x))y'(x-\theta)+$$
$$+B_0(x)(y(x))^2+2y(x)p(x)).$$

References

1. El'sgol'ts, L.E.: Qualitative Methods in Mathematical Analysis. Gosudarstvennoe izdatel'stvo tehniko–teoreticheskoi literatury, Moscow (1955) [in Russian]
2. Fillipov, V.M., Savchin, V.M., Shorohov, S.G.: Variational Principles for Non-Potential Operators, vol. 40. VINITI, Moscow (1992) [in Russian]
3. Kamenskii, G.: On boundary value problems connected with variational problems for nonlocal functionals. Funct. Differ. Equ. **12**, 245–270 (2005)
4. Olver, P.J.: Applications of Lie Groups to Differential Equations. Springer, Berlin/Heidelberg/New York/Tokio (1986)
5. Popov, A.M.: Inverse problem of the calculus of variations for systems of differential-difference equations of second order. Math. Notes **72**, 687–691 (2002)
6. Zadorozhnii, V.G.: Methods of Variational Analysis. Institut kompyuternyh issledovanii, Moscow/Izhevsk (2006) [in Russian]
7. Zadorozhnii, V.G., Kurina, G. A.: Inverse problem of the variational calculus for differential equations of second order with deviating argument. Math. Notes **90**, 218–226 (2011)

State-Space Model and Kalman Filter Gain Identification by a Superspace Method

Ping Lin, Minh Q. Phan, and Stephen A. Ketcham

Abstract This paper describes a superspace method to identify a state-space model and an associated Kalman filter gain from input-output data. Superstate vectors are simply vectors containing input-output measurements, and used directly for the identification. The superstate space is unusual in that the state portion of the Kalman filter becomes completely independent of both the system dynamics and the input and output noise statistics. The system dynamics is entirely carried by the measurement portion of the superstate Kalman filter model. When model reduction is applied, the system dynamics returns to the state portion of the state-space model.

1 Introduction

Finding a state-space model of a system and an associated optimal observer/Kalman filter gain in the presence of unknown process and measurement noises is a well-known and important system identification problem. The identification problem is nonlinear because it involves the product of the system matrix and the state, both of which are unknown. A method known as Observer/Kalman filter identification (OKID) bypassed the need to determine the system states by working through an intermediate set of parameters called Observer/Kalman filter Markov parameters [1]. These parameters are related to input-output data linearly, and can thus be found by a simple linear least-squares solution. The nonlinear step is handled by the Eigensystem Realization Algorithm (ERA), which offers an analytical (exact) solution to the problem of recovering a state-space model representation from the identified Markov parameters [2]. Under appropriate conditions, OKID identifies a state-space model of the system and an associated Kalman filter gain that is optimal with respect to the unknown process and measurement noises embedded in the input-output data. There are numerous extensions of the observer-based method

P. Lin • M.Q. Phan (✉)
Thayer School of Engineering, Dartmouth College, Hanover, NH 03755, USA
e-mail: ping.lin@dartmouth.edu; minh.q.phan@dartmouth.edu

S.A. Ketcham
Cold Regions Research and Engineering Laboratory (CRREL), Hanover, NH 03755, USA
e-mail: stephen.a.ketcham@erdc.usace.army.mil

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_10

[3–5], such as residual whitening [6], and methods that are derived from interaction matrices [7].

Another important class of methods that solved this problem is known as "subspace methods" [8, 9]. These methods put the emphasis on the recovery of the system states from input-output measurements by various oblique projection techniques. A fundamental attraction of this approach is that once the states are known, the identification of the state-space model becomes linear. These methods are also capable of recovering the optimal Kalman filter gain when the input-output measurements are corrupted by unknown process and measurement noises. There are numerous variations of the subspace technique [10–12]. The method used in this paper for comparison purpose is N4SID [13].

This paper describes a recently developed class of methods which can be referred to as "superspace methods". A superstate vector is made up of input and output measurements. These superstate vectors are treated as the states of the system, and used directly in the identification of the state-space model and an associated Kalman filter gain. The superspace method bypasses the need to recover the states of the system as required in a subspace method. It also sidesteps the need to work through the Markov parameters as in OKID-based methods. In this paper, we further show that in the space of the superstates, the system matrices that define the state portion of the Kalman filter are made up entirely of 1's and 0's. They do not need to be identified, and even more interestingly, they are independent of the system dynamics and the process and measurement noise statistics. All system dynamics are carried in the output measurement portion of the Kalman filter. When model reduction is applied, the dynamics of the system returns to the state portion of the model as one would expect in a state-space model. The superspace idea has also been recently applied successfully to the bilinear system identification problem [14].

2 Mathematical Formulation

The state-space identification problem has many related or equivalent forms. We will use one that is both common in literature and convenient for stating our algorithm.

2.1 Problem Statement

Suppose a set of input-output measurements, $\{u_0, u_1, \cdots, u_s\}$, $\{y_0, y_1, \cdots, y_s\}$, of an m-input, q-output system corrupted by unknown process and measurement noises, is given. The objective of the problem is to find a state-space representation $\{A, B, C, D\}$ and a steady-state Kalman filter gain K such that the input-output measurements are related to each other by the following innovation form of the state-space model,

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + Ke_k \tag{1}$$

$$y_k = C\hat{x}_k + Du_k + e_k \tag{2}$$

where $\hat{x}_k$ denotes the (unknown) Kalman filter state, K is the (unknown) corresponding steady-state Kalman filter gain, and e_k is the (unknown) Kalman filter residual that is unique for the given system, input-output measurements, and noises in the system, $e_k = y_k - \hat{y}_k = y_k - (C\hat{x}_k + Du_k)$. The innovation form is derived from the conventional process form,

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + w_k \tag{3}$$

$$y_k = C\hat{x}_k + Du_k + n_k \tag{4}$$

The process noise w_k and measurement noises n_k are assumed to be independent, white, zero-mean, and Gaussian. The Kalman filter gain K is a function of the system state-space model and the covariances of the process and measurement noises. In the identification problem, only input-output measurements are known. The noise covariances are unknown.

2.2 A Superstate Vector Definition

A superstate vector z_k is defined from input-output data as follows,

$$z_k = \begin{bmatrix} v_{k-p} \\ \vdots \\ v_{k-2} \\ v_{k-1} \end{bmatrix} \qquad v_k = \begin{bmatrix} u_k \\ y_k \end{bmatrix} \tag{5}$$

From the given input-output measurements, these superstates can be easily created, and used in the subsequent superspace identification method.

3 A Superspace Identification Algorithm

A superspace identification algorithm is summarized below.

- Choose a value for p such that $(m + q)p$ defines the dimension of the superstate vector, which is typically larger than the true minimum state dimension of the system being identified.
- Form the following matrix Z from the available input-output data,

$$Z = \begin{bmatrix} v_0 & v_1 & \cdots & v_{s-p} & v_{s-p+1} \\ v_1 & v_2 & \cdots & v_{s-p+1} & v_{s-p+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{p-2} & v_{p-1} & \cdots & v_{s-2} & v_{s-1} \\ v_{p-1} & v_p & \cdots & v_{s-1} & v_s \end{bmatrix} \tag{6}$$

Define Z_0 as Z with its last column removed, and Z_1 as Z with its first column removed. Furthermore, define

$$U_p = \begin{bmatrix} u_p & u_{p+1} & \cdots & u_s \end{bmatrix} \qquad Y_p = \begin{bmatrix} y_p & y_{p+1} & \cdots & y_s \end{bmatrix} \qquad V_p = \begin{bmatrix} U_p \\ \hline Y_p \end{bmatrix} \tag{7}$$

- Solve for $\bar{A}^*$, $\bar{B}^*$, C^*, and D^* by least-squares from

$$Z_1 = \bar{A}^* Z_0 + \bar{B}^* V_p \tag{8}$$

$$Y_p = C^* Z_0 + D^* U_p + E_p^* \tag{9}$$

It turns out that there is no need to solve for $\bar{A}^*$, $\bar{B}^*$ from (8), as they are simply matrices made up of 0's and 1's,

$$\bar{A}^* = \left[\begin{array}{c|c} 0_{(p-1)b\times b} & I_{(p-1)b\times(p-1)b} \\ \hline 0_{b\times b} & 0_{b\times(p-1)b} \end{array}\right] \qquad \bar{B}^* = \left[\begin{array}{c} 0_{(p-1)b\times b} \\ \hline I_{b\times b} \end{array}\right] \tag{10}$$

where $b = m + q$. A key point to observe here is that $\bar{A}^*$, $\bar{B}^*$ are completely independent of the system dynamics and the process and measurement noise statistics. Information about the system is completely contained in C^* and D^*, which can be solved by least-squares,

$$\begin{bmatrix} C^* & | & D^* \end{bmatrix} = Y_p \left[\begin{array}{c} Z_0 \\ \hline U_p \end{array}\right]^\dagger \tag{11}$$

The † denotes the Moore-Penrose pseudo-inverse. Z_1 is not needed in (11), but will be used later in establishing the optimality of the algorithm.

- Lastly, a representation of $\{A, B, C, D, K\}$ denoted by $\{A^*, B^*, C^*, D^*, K^*\}$ can be recovered from $\bar{A}^*$, $\bar{B}^*$, C^*, D^* based on the following relationship,

$$\bar{A}^* = A^* - K^* C^* \qquad \bar{B}^* = \begin{bmatrix} B^* - K^* D^* & | & K^* \end{bmatrix} \tag{12}$$

Recall that C^* and D^* are obtained from (11), $\bar{A}^*$ and $\bar{B}^*$ are given in (10). The matrix $\bar{B}^*$, which is defined in (10), has two partitions according to (12): the left partition is $B^* - K^* D^*$ of dimensions $pb \times m$, the right partition is K^* of dimensions $pb \times q$. Because $\bar{B}^*$ is made up of entirely 0's and 1's

and known beforehand, these partitions are known. B^* can be recovered from the first partition of $\bar{B}^*$ because K^* are D^* known. Similarly, A^* can be recovered from $\bar{A}^*$ because K^* and C^* are known. Having obtained a full set of $\{A^*, B^*, C^*, D^*, K^*\}$, standard model reduction techniques can be applied to reduce the dimension of $\{A^*, B^*, C^*, D^*, K^*\}$ to the correct minimum state dimension of the system being identified.

4 Optimality of Superspace Identification

We now establish the optimality of the combination $\{A^*, B^*, C^*, D^*, K^*\}$ by proving that Markov parameters of the combination $\{A^*, B^*, C^*, D^*, K^*\}$ match the Markov parameters of the optimal Kalman filter given in (1) and (2). First, we need to eliminate e_k from the state portion of the Kalman filter by solving for e_k in (2) and substituting it into (1) to produce

$$\hat{x}_{k+1} = \bar{A}\hat{x}_k + \bar{B}v_k \tag{13}$$

$$y_k = C\hat{x}_k + Du_k + e_k \tag{14}$$

where $\bar{A} = A - KC$, $\bar{B} = [B - KD \mid K]$, and v_k is defined in (5). Define

$$\hat{X}_p = \begin{bmatrix} \hat{x}_p \cdots \hat{x}_s \end{bmatrix} \qquad \hat{X}_{p+1} = \begin{bmatrix} \hat{x}_{p+1} \cdots \hat{x}_{s+1} \end{bmatrix} \qquad E_p = \begin{bmatrix} \hat{e}_p \cdots \hat{e}_s \end{bmatrix} \tag{15}$$

Equations (13) and (14) can be written for all available time steps,

$$\hat{X}_{p+1} = \bar{A}\hat{X}_p + \bar{B}V_p \tag{16}$$

$$Y_p = C\hat{X}_p + DU_p + E_p \tag{17}$$

where $V_p = \begin{bmatrix} v_p \; v_{p+1} \cdots v_s \end{bmatrix}$. Next, we express $\hat{X}_p$ and $\hat{X}_{p+1}$ in terms of input and output measurements. As long as p is sufficiently large such that $\bar{A}^p \approx 0$, then by repeated substitution, the Kalman filter state can be expressed in terms of input and output measurements, and when packaged together, can be put in the form,

$$\begin{bmatrix} \hat{x}_p \; \hat{x}_{p+1} \cdots \hat{x}_{s+1} \end{bmatrix} = \begin{bmatrix} \bar{A}^{p-1}\bar{B} \cdots \bar{A}\bar{B} \; \bar{B} \end{bmatrix} \begin{bmatrix} v_0 & v_1 & \cdots & v_{s-p+1} \\ v_1 & v_2 & \cdots & v_{s-p+2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{p-1} & v_p & \cdots & v_s \end{bmatrix} \tag{18}$$

Define $\mathscr{C}_p = \begin{bmatrix} \bar{A}^{p-1}\bar{B}, & \ldots, & \bar{A}\bar{B}, & \bar{B} \end{bmatrix}$. It follows from (18) that

$$\hat{X}_p = \mathscr{C}_p Z_0 \qquad \hat{X}_{p+1} = \mathscr{C}_p Z_1 \tag{19}$$

where Z_1 and Z_2 are the two partitions of Z as previously defined in (6). Substituting (19) into (16) produces

$$\mathscr{C}_p Z_1 = \bar{A}\mathscr{C}_p Z_0 + \bar{B}V_p \tag{20}$$

$$Y_p = C\mathscr{C}_p Z_0 + DU_p + E_p \tag{21}$$

This is the relation that the ideal optimal system matrices need to satisfy with the given input-output measurements. Notice that E_p here is made of the innovation sequence $\{e_k\}$. In the superspace algorithm, we impose (8) and (9). To relate (20) and (21) to (8) and (9), premultiplying (8) by $\mathscr{C}_p$ produces

$$\mathscr{C}_p Z_1 = \mathscr{C}_p \bar{A}^* Z_0 + \mathscr{C}_p \bar{B}^* V_p \tag{22}$$

$$Y_p = C^* Z_0 + D^* U_p + E_p^* \tag{23}$$

In the superspace identification algorithm, C^* and D^* are solved from (11) by least-squares. This step ensures that the residual E_p^* is minimized and orthogonal to the input and output data. These are the conditions that the optimal Kalman filter residual must satisfy, hence $E_p^* = E_p$. Furthermore, if the input-output data set is sufficiently rich such that the matrix formed by Z_0 and V_p is full rank, we also have

$$\bar{A}\mathscr{C}_p = \mathscr{C}_p \bar{A}^* \tag{24}$$

$$\bar{B} = \mathscr{C}_p \bar{B}^* \tag{25}$$

$$C\mathscr{C}_p = C^* \tag{26}$$

$$D = D^* \tag{27}$$

As long as p is sufficiently large such that $\bar{A}^p \approx 0$, our choices of $\bar{A}^*$ and $\bar{B}^*$ in (10) indeed satisfy (24) and (25). Furthermore, it can be shown that the Markov parameters of the identified Kalman filter match the Markov parameters of the optimal Kalman filter. For example, the first Markov parameter can be shown to match,

$$C^*\bar{B}^* = (C\mathscr{C}_p)\bar{B}^* = C(\mathscr{C}_p \bar{B}^*) = C\,\bar{B} \tag{28}$$

Similarly, the second Markov parameter can also be shown to match,

$$C^*\bar{A}^*\bar{B}^* = C(\mathscr{C}_p \bar{A}^*)\bar{B}^* = C(\bar{A}\mathscr{C}_p)\bar{B}^* = C\,\bar{A}(\mathscr{C}_p \bar{B}^*) = C\,\bar{A}\bar{B} \tag{29}$$

and so on. The two sets of system matrices have the same Markov parameters. This result establishes the optimality of the identified Kalman filter.

We discuss briefly how the superspace method is different from the subspace method. Although there are several variants of the subspace identification method, the well-known N4SID is used for the present discussion. The N4SID method first estimates the optimal states by an oblique projection, then estimates all the system state-space matrices by least-squares. The superspace method bypasses the state estimation step by forming the superstates, and uses them directly in the identification. The least-squares calculation is used to obtain the coefficients of the measurement equations. The state portion of the Kalman filter are made up of 1's and 0's, and they do not need to be computed. More fundamentally, in deriving the subspace method, the process form of the state-space model is used to establish the input-output relationship, whereas in the superspace method the innovation form which involves the Kalman filter is used. The subspace method is thus process form oriented, whereas the superspace method is innovation form oriented. A consequence of using the innovation form is that the superspace identification method simultaneously recovers the steady-state Kalman filter gain together with the system state-space model, whereas in the subspace method, the Kalman filter gain is typically computed as an additional step.

5 Illustrative Examples

This section demonstrates the effectiveness of the superspace identification method using both simulated and experimental data. Comparison of the results to those obtained with the subspace N4SID algorithm will also be provided.

5.1 A Simulated System

The following system is driven by random input excitation, and the resultant output is recorded for system identification,

$$A = \begin{bmatrix} -0.35 & -0.5 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 0.5 \end{bmatrix} \qquad D = 0$$

In order to show the convergent behavior, the input-output data record is deliberately chosen to be large (2^{16} samples). The input and measurement noise covariances are $Q = 0.025, R = 0.025$ corresponding to the input and measurement noises shown in Figs. 1 and 2. Using $p = 8$ in the identification, the final model order is reduced to 2. Figure 3 shows the Kalman filter Markov parameters, $C\,(A - KC)^k\,B$ and $C\,(A - KC)^k\,K$, constructed from the identified state-space model and the identified Kalman filter gain by both methods. These Markov parameters are compared to those of the optimal Kalman filter whose gain is computed from perfect

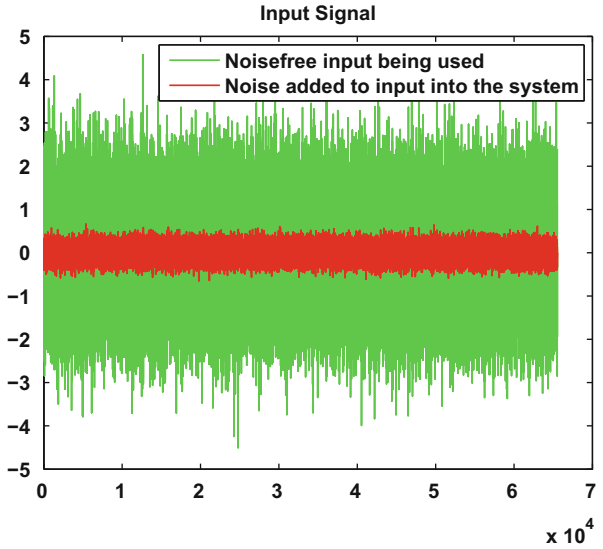

Fig. 1 Input data with added noise

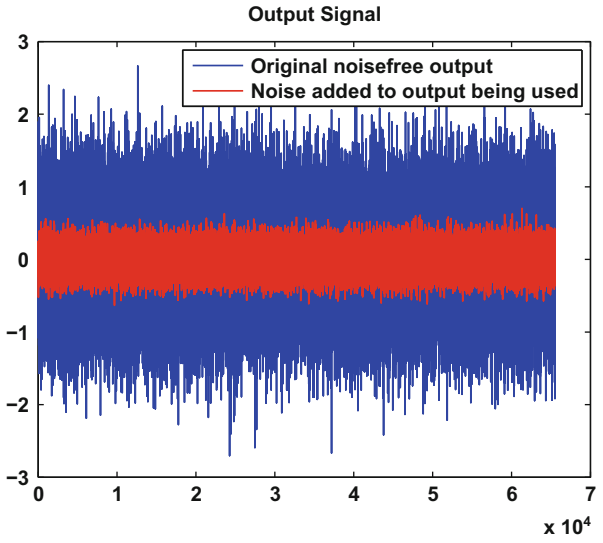

Fig. 2 Output data with added noise

knowledge of the system model and the input and measurement noise covariances. Figure 4 shows the residuals of the identified Kalman filters and the state-state models by both methods (superspace and N4SID) matching the residuals of the

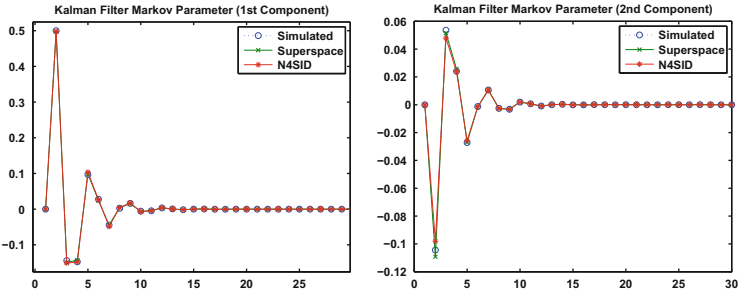

Fig. 3 Kalman filter Markov parameters by two identification methods (superspace and N4SID) compared to the Kalman filter Markov parameters computed from perfect knowledge of the system model and noise statistics

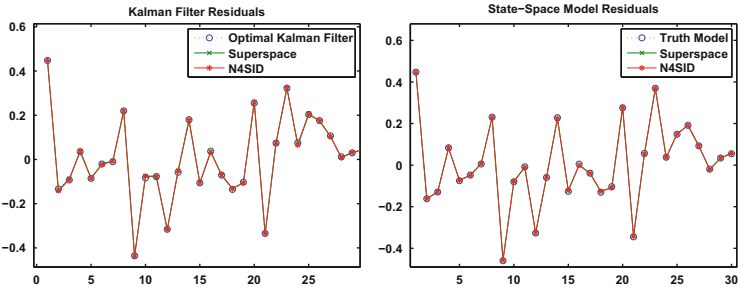

Fig. 4 Comparison of identified filter residuals to optimal Kalman residual (*left*), and identified state-space model residuals to truth model residual (*right*)

optimal filter and of the truth model point-wise. The results confirm that both methods produce optimal identification results as expected.

5.2 CD Player Arm

A set of experimental data of a CD player arm is used in this example [15]. The system has two inputs and two outputs. The input-output data record used for identification is shown in Figs. 5 and 6 (2,048 samples). Using $p = 6$ in the identification, the final system order is reduced to 12 before the identification results are compared. Figure 7 shows a comparison of the identified Kalman filter outputs to the measured outputs. Figure 8 shows a comparison of the identified state-space model outputs to the measured outputs. Overall, both methods appear to capture the dynamics of the mechanism relatively well with this set of input-output data.

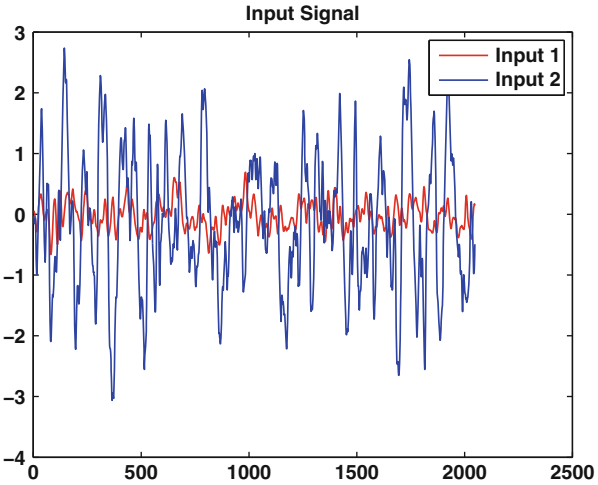

Fig. 5 Input signals of CD player arm

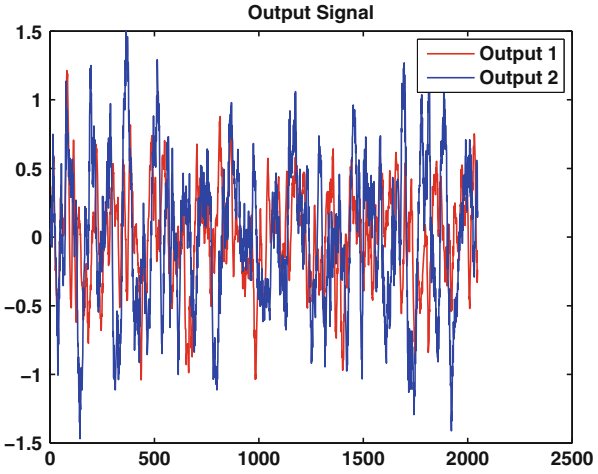

Fig. 6 Output signals of CD player arm

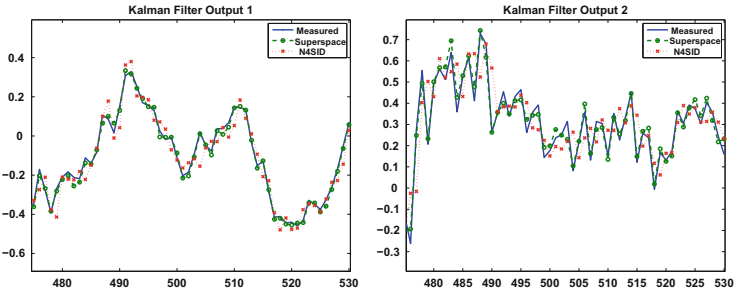

Fig. 7 Comparison of Kalman filter outputs to measured outputs by two identification methods (superspace and N4SID)

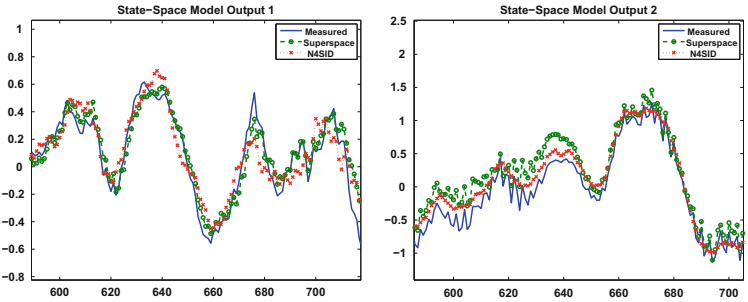

Fig. 8 Comparison of state-space model outputs to measured outputs by two identification methods (superspace and N4SID)

6 Conclusions

A superspace method for identification of a system state-space model and its associated Kalman filter gain has been formulated. It is found that in the space of the superstates, which are vectors of input and output measurements, the matrices that define the state portion of the Kalman filter are made up entirely of 0's and 1's. These matrices are known in advance, and do not need to be identified. Because they are known in advance, they are completely independent of the actual system dynamics and the noise statistics. This is a highly intriguing and very counter-intuitive result. Moreover, in the superstate space, the system dynamics are contained the measurement equation, not the state portion, of the Kalman filter. When model reduction is applied, the actual system dynamics returns to the state portion of the reduced-order model as one would expect in a state-space model. Optimality of the proposed superspace identification method is also established in theory and confirmed in numerical simulation. The Kalman filter identified from input-output measurements by the superspace technique is found to match the optimal Kalman filter derived from perfect knowledge of the system and perfect knowledge of the noise statistics, both in their Markov parameters and their output

residuals. When applied to experimental data of a CD arm mechanism, the method produces excellent results when compared to an established subspace identification method.

Acknowledgements This research is supported by a STTR Phase II contract from the US Army Corps of Engineers Cold Regions Research and Engineering Laboratory (CRREL) to Dartmouth College and Sound Innovations, Inc.

References

1. Juang, J.-N., Phan, M.Q., Horta, L.G., Longman, R.W.: Identification of observer/Kalman filter Markov parameters – theory and experiments. J. Guid. Control Dyn. **16**(2), 320–329 (1993)
2. Juang, J.-N., Pappa, R.S.: An eigensystem realization algorithm for modal parameter identification and model reduction. J. Guid. Control Dyn. **8**, 620–627 (1985)
3. Phan, M.Q., Juang, J.-N., Longman, R.W.: Identification of linear multivariable systems by identification of observers with assigned real eigenvalues. J. Astronaut. Sci. **15**(1), 88–95 (1992)
4. Phan, M.Q., Horta, L.G., Juang, J.-N., Longman, R.W.: Linear system identification via an asymptotically stable observer. J. Optim. Theory Appl. **79**(1), 59–86 (1993)
5. Phan, M.Q., Horta, L.G., Juang, J.-N., Longman, R.W.: Improvement of observer/Kalman filter identification (OKID) by residual whitening. J. Vib. Acoust. **117**, 232–238 (1995)
6. Juang, J.-N.: Applied System Identification. Prentice-Hall, Upper Saddle River (2001)
7. Phan, M.Q.: Interaction matrices in system identification and control. In: Proceedings of the 15th Yale Workshop on Adaptive and Learning Systems, New Haven (2011)
8. Van Overchee, P., De Moor, B.: Subspace Identification for Linear Systems. Kluwer Academic, Boston (1996)
9. Qin, S.J.: An Overview of Subspace Identification. Comput. Chem. Eng. **30**(10–12), 1502–1513 (2006)
10. Verhaegen, M., Dewilde P.: Subspace model identification part 1: the output error state-space model identification class of algorithms. Int. J. Control **56**(5), 1187–1210 (1992)
11. Van Overchee, P., De Moor, B.: A unifying theorem for three subspace system identification algorithms. Automatica **31**(12), 1853–1864 (1995)
12. Jansson, M., Wahlberg, B.: A linear regression approach to state-space subspace system identification. Signal Process. **52**, 103–129 (1996)
13. Van Overchee, P., De Moor, B.: N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica **30**(1), 75–93 (1994)
14. Phan, M.Q., Čelik, H.: A superspace method for discrete-time bilinear model identification by interaction matrices. J. Astronaut. Sci. **59**(1–2), 433–452 (2012)
15. De Moor, B., De Gersem, P., De Schutter, B., Favoreel, W.: DAISY: a database for identification of systems. Journal A **38**(3), 4–5 (1997)

Stiff Order Conditions for Exponential Runge–Kutta Methods of Order Five

Vu Thai Luan and Alexander Ostermann

Abstract Exponential Runge–Kutta methods are tailored for the time discretization of semilinear stiff problems. The actual construction of high-order methods relies on the knowledge of the order conditions, which are available in the literature up to order four. In this short note, we show how the order conditions for methods up to order five are derived; the extension to arbitrary orders will be published elsewhere. Our approach is adapted to stiff problems and allows us to prove high-order convergence results for variable step size implementations, independently of the stiffness of the problem.

1 Introduction

In this paper, we derive the stiff order conditions for exponential Runge–Kutta methods up to order five. These conditions are important for constructing high-order time discretization schemes for semilinear problems

$$u'(t) = Au(t) + g(u(t)), \quad u(t_0) = u_0, \tag{1}$$

where A has a large norm or is even an unbounded operator. The nonlinearity g, on the other hand, is supposed to be nonstiff with a moderate Lipschitz constant in a strip along the exact solution. Abstract parabolic evolution equations and their spatial discretizations are typical examples of such problems.

Exponential integrators have shown to be very competitive for stiff problems, see [1, 4, 9]. They treat the linear part of problem (1) exactly and the nonlinearity in an explicit way. A recent overview of such integrators and their implementation was given in [7]. The class of exponential Runge–Kutta methods was first considered by Friedli [2] who also derived the nonstiff order conditions. For stiff problems, the methods were analyzed in [5]. In that paper, the stiff order conditions for methods up to order four were derived.

V.T. Luan (✉) • A. Ostermann
Institut für Mathematik, Universität Innsbruck, Technikerstr. 13, A–6020 Innsbruck, Austria
e-mail: vu.thai-luan@uibk.ac.at; alexander.ostermann@uibk.ac.at

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_11

Motivated by the fact that exponential Runge–Kutta methods can be viewed as small perturbations of the exponential Euler method, we present here a new and simple approach to derive the stiff order conditions. Instead of inserting the exact solution into the numerical scheme and working with defects, as it was done in [5, 8], we analyze the local error in a direct way. For this purpose, we reformulate the scheme as a perturbation of the exponential Euler method and carry out a perturbation analysis. This allows us to generalize the order four conditions that were given in [5] to methods up to order five. The error analysis is performed in the framework of strongly continuous semigroups [11] which covers parabolic problems and their spatial discretizations. The work is inspired by our recent paper [10], where exponential Rosenbrock methods were constructed up to order five.

The paper is organized as follows. In Sect. 2, we introduce a reformulation of exponential Runge–Kutta methods which turns out to be advantageous for the analysis. Our abstract framework is given in Sect. 3. The new stiff order conditions are derived in Sect. 4. Section 5 is devoted to the convergence analysis. The main results are given in Table 1 and Theorem 1.

Table 1 Stiff order conditions for explicit exponential Runge–Kutta methods up to order 5. The variables Z, J, K, L denote arbitrary square matrices, and B an arbitrary bilinear mapping of appropriate dimensions. The functions $\psi_{k,l}$ are defined in (21)

No.	Order condition	Order
1	$\sum_{i=1}^{s} b_i(Z) = \varphi_1(Z)$	1
2	$\sum_{i=2}^{s} b_i(Z)c_i = \varphi_2(Z)$	2
3	$\sum_{j=1}^{i-1} a_{ij}(Z) = c_i\varphi_1(c_i Z), \quad i = 2,\dots,s$	2
4	$\sum_{i=2}^{s} b_i(Z)\frac{c_i^2}{2!} = \varphi_3(Z)$	3
5	$\sum_{i=2}^{s} b_i(Z)J\psi_{2,i}(Z) = 0$	3
6	$\sum_{i=2}^{s} b_i(Z)\frac{c_i^3}{3!} = \varphi_4(Z)$	4
7	$\sum_{i=2}^{s} b_i(Z)J\psi_{3,i}(Z) = 0$	4
8	$\sum_{i=2}^{s} b_i(Z)J\sum_{j=2}^{i-1} a_{ij}(Z)J\psi_{2,j}(Z) = 0$	4
9	$\sum_{i=2}^{s} b_i(Z)c_i K\psi_{2,i}(Z) = 0$	4
10	$\sum_{i=2}^{s} b_i(Z)\frac{c_i^4}{4!} = \varphi_5(Z)$	5
11	$\sum_{i=2}^{s} b_i(Z)J\psi_{4,i}(Z) = 0$	5
12	$\sum_{i=2}^{s} b_i(Z)J\sum_{j=2}^{i-1} a_{ij}(Z)J\psi_{3,j}(Z) = 0$	5
13	$\sum_{i=2}^{s} b_i(Z)J\sum_{j=2}^{i-1} a_{ij}(Z)J\sum_{k=2}^{j-1} a_{jk}(Z)J\psi_{2,k}(Z) = 0$	5
14	$\sum_{i=2}^{s} b_i(Z)J\sum_{j=2}^{i-1} a_{ij}(Z)c_j K\psi_{2,j}(Z) = 0$	5
15	$\sum_{i=2}^{s} b_i(Z)c_i K\psi_{3,i}(Z) = 0$	5
16	$\sum_{i=2}^{s} b_i(Z)c_i K\sum_{j=2}^{i-1} a_{ij}(Z)J\psi_{2,j}(Z) = 0$	5
17	$\sum_{i=2}^{s} b_i(Z)B\big(\psi_{2,i}(Z),\ \psi_{2,i}(Z)\big) = 0$	5
18	$\sum_{i=2}^{s} b_i(Z)c_i^2 L\psi_{2,i}(Z) = 0$	5

2 Reformulation of Exponential Runge–Kutta Methods

In order to solve (1) numerically, we consider a class of explicit one-step methods, the so-called explicit exponential Runge–Kutta methods

$$U_{ni} = \mathrm{e}^{c_i h_n A} u_n + h_n \sum_{j=1}^{i-1} a_{ij}(h_n A) g(U_{nj}), \quad 1 \le i \le s, \tag{2a}$$

$$u_{n+1} = \mathrm{e}^{h_n A} u_n + h_n \sum_{i=1}^{s} b_i(h_n A) g(U_{ni}). \tag{2b}$$

The stages U_{ni} are approximations to $u(t_n + c_i h_n)$, the numerical solution u_{n+1} approximates the true solution at time t_{n+1} and $h_n = t_{n+1} - t_n$ denotes the step size. The coefficients $a_{ij}(h_n A)$ and $b_i(h_n A)$ are usually chosen as linear combinations of the entire functions $\varphi_k(c_i h_n A)$ and $\varphi_k(h_n A)$, respectively. These functions are given by

$$\varphi_0(z) = \mathrm{e}^z, \quad \varphi_k(z) = \int_0^1 \mathrm{e}^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!}\, \mathrm{d}\theta, \quad k \ge 1 \tag{3}$$

and thus satisfy the recurrence relation

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad k \ge 0. \tag{4}$$

It turns out that the equilibria of (1) are preserved if the coefficients a_{ij} and b_i of the method fulfill the following simplifying assumptions (see [5])

$$\sum_{i=1}^{s} b_i(h_n A) = \varphi_1(h_n A), \qquad \sum_{j=1}^{i-1} a_{ij}(h_n A) = c_i \varphi_1(c_i h_n A), \quad 1 \le i \le s. \tag{5}$$

The latter implies in particular that $c_1 = 0$. Without further mention, we will assume throughout the paper that (5) is satisfied.

Following an idea of [6, 12], we now express the vector $g(U_{ni})$ as

$$g(U_{ni}) = g(u_n) + D_{ni}, \quad 1 \le i \le s \tag{6}$$

and rewrite (2) in terms of D_{ni}. Since $c_1 = 0$, we consequently have $U_{n1} = u_n$ and $D_{n1} = 0$. The method (2) then takes the equivalent form

$$U_{ni} = u_n + c_i h_n \varphi_1(c_i h_n A) F(u_n) + h_n \sum_{j=2}^{i-1} a_{ij}(h_n A) D_{nj}, \quad 1 \le i \le s, \tag{7a}$$

$$u_{n+1} = u_n + h_n \varphi_1(h_n A) F(u_n) + h_n \sum_{i=2}^{s} b_i(h_n A) D_{ni} \tag{7b}$$

with $F(u) = Au + g(u)$.

Since the vectors D_{ni} are small in norm, in general, exponential Runge–Kutta methods can be interpreted as small perturbations of the exponential Euler scheme

$$u_{n+1} = u_n + h_n \varphi_1(h_n A) F(u_n).$$

The reformulated scheme (7) can be implemented more efficiently than (2), and it offers advantages in the error analysis, see below.

3 Analytic Framework

For the error analysis of (7), we work in an abstract framework of strongly continuous semigroups on a Banach space X with norm $\|\cdot\|$. Background information on semigroups can be found in the monograph [11].

Throughout the paper we consider the following assumptions.

Assumption 1. The linear operator A is the infinitesimal generator of a strongly continuous semigroup e^{tA} on X.

This implies (see [11, Thm. 2.2]) that there exist constants M and ω such that

$$\|\mathrm{e}^{tA}\|_{X \leftarrow X} \le M \mathrm{e}^{\omega t}, \quad t \ge 0. \tag{8}$$

Under the above assumption, the expressions $\varphi_k(h_n A)$ and consequently the coefficients $a_{ij}(h_n A)$ and $b_i(h_n A)$ of the method are bounded operators, see (3). This property is crucial in our proofs.

For high-order convergence results, we require the following regularity assumption.

Assumption 2. We suppose that (1) possesses a sufficiently smooth solution $u : [0, T] \to X$ with derivatives in X and that $g : X \to X$ is sufficiently often Fréchet differentiable in a strip along the exact solution. All occurring derivatives are assumed to be uniformly bounded.

Assumption 2 implies that g is locally Lipschitz in a strip along the exact solution. It is well known that semilinear reaction-diffusion-advection equations can be put into this abstract framework, see [3].

4 A New Approach to Construct the Stiff Order Conditions

In this section, we present a new approach to derive the stiff order conditions for exponential Runge–Kutta methods. It is the well-known that the exponential Euler method

$$u_{n+1} = u_n + h_n\,\varphi_1(h_n A)F(u_n) \tag{9}$$

has order one. In view of (7b), exponential Runge–Kutta methods can be considered as small perturbations of (9). This observation motivates us to investigate the vectors D_{ni} in order to get a higher-order method.

Let $\tilde{u}_n$ denote the exact solution of (1) at time t_n, i.e., $\tilde{u}_n = u(t_n)$. In order to study the local error of scheme (7), we consider one step with initial value $\tilde{u}_n$, i.e.

$$\hat{U}_{ni} = \tilde{u}_n + c_i h_n \varphi_1(c_i h_n A)F(\tilde{u}_n) + h_n \sum_{j=2}^{i-1} a_{ij}(h_n A)\hat{D}_{nj}, \tag{10a}$$

$$\hat{u}_{n+1} = \tilde{u}_n + h_n \varphi_1(h_n A)F(\tilde{u}_n) + h_n \sum_{i=2}^{s} b_i(h_n A)\hat{D}_{ni} \tag{10b}$$

with

$$\hat{D}_{ni} = g(\hat{U}_{ni}) - g(\tilde{u}_n), \quad \hat{U}_{ni} \approx u(t_n + c_i h_n). \tag{11}$$

Let $\tilde{u}_n^{(k)}$ denote the k-th derivative of the exact solution $u(t)$ of (1), evaluated at time t_n. For $k = 1, 2$ we use the corresponding notations $\tilde{u}_n', \tilde{u}_n''$ for simplicity. We further denote the k-th derivative of $g(u)$ with respect to u by $g^{(k)}(u)$.

4.1 Taylor Expansion of the Exact and the Numerical Solution

On the one hand, expressing the exact solution of (1) at time t_{n+1} by the variation-of-constants formula

$$\tilde{u}_{n+1} = u(t_{n+1}) = \mathrm{e}^{h_n A}\tilde{u}_n + h_n \int_0^1 \mathrm{e}^{(1-\theta)h_n A} g(u(t_n + \theta h_n))\,\mathrm{d}\theta \tag{12}$$

and then expanding $g(u(t_n + \theta h_n))$ in a Taylor series at $\tilde{u}_n$ gives

$$\begin{aligned}\tilde{u}_{n+1} = \tilde{u}_n &+ h_n \varphi_1(h_n A)F(\tilde{u}_n) \\ &+ \sum_{q=1}^{k} h_n^{q+1} \int_0^1 \mathrm{e}^{(1-\theta)h_n A} \frac{\theta^q}{q!} g^{(q)}(\tilde{u}_n)(\underbrace{V, \ldots, V}_{q \text{ times}})\,\mathrm{d}\theta + \mathscr{R}_k\end{aligned} \tag{13}$$

with $V = \frac{1}{\theta h_n}\big(u(t_n + \theta h_n) - u(t_n)\big)$ and the remainder

$$\mathscr{R}_k = h_n^{k+2} \int_0^1 \mathrm{e}^{(1-\theta)h_n A} \int_0^1 \frac{\theta^{k+1}(1-s)^k}{k!} g^{(k+1)}(\tilde{u}_n + s\theta h_n V)(\underbrace{V, \dots, V}_{k+1 \text{ times}})\, \mathrm{d}s\, \mathrm{d}\theta.$$

It is easy to see that $\|\mathscr{R}_k\| \le C h_n^{k+2}$ where the constant C only depends on values that are uniformly bounded by Assumptions 1 and 2. From now on, we will use the Landau notation for such remainder terms. Thus, we will write $\mathscr{R}_k = \mathscr{O}(h_n^{k+2})$.

Expanding $u(t_n + \theta h_n)$ in a Taylor series at t_n gives

$$V = \sum_{r=1}^{m} \frac{(\theta h_n)^{r-1}}{r!} \tilde{u}_n^{(r)} + \mathscr{O}(h_n^m).$$

Inserting these expressions into (13) for $k = 4$, using (3) and the symmetry of the multilinear mappings in (13), we obtain

$$\begin{aligned} \tilde{u}_{n+1} = \tilde{u}_n &+ h_n \varphi_1(h_n A) F(\tilde{u}_n) + h_n^2 \varphi_2(h_n A)\mathbf{L} + h_n^3 \varphi_3(h_n A)\mathbf{M} \\ &+ h_n^4 \varphi_4(h_n A)\mathbf{N} + h_n^5 \varphi_5(h_n A)\mathbf{P} + \mathscr{O}(h_n^6) \end{aligned} \tag{14}$$

with

$$\begin{aligned} \mathbf{L} &= g'(\tilde{u}_n)\tilde{u}_n', \quad \mathbf{M} = g'(\tilde{u}_n)\tilde{u}_n'' + g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n'), \\ \mathbf{N} &= g'(\tilde{u}_n)\tilde{u}_n^{(3)} + 3g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n'') + g^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n'), \\ \mathbf{P} &= g'(\tilde{u}_n)\tilde{u}_n^{(4)} + 3g''(\tilde{u}_n)(\tilde{u}_n'', \tilde{u}_n'') + 4g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n^{(3)}) \\ &\quad + 6g^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n'') + g^{(4)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n', \tilde{u}_n'). \end{aligned} \tag{15}$$

On the other hand, expanding $\hat{D}_{ni}$ in (11) in a Taylor series at $\tilde{u}_n$, we obtain

$$\hat{D}_{ni} = \sum_{q=1}^{k} \frac{h_n^q}{q!} g^{(q)}(\tilde{u}_n)(\underbrace{V_i, \dots, V_i}_{q \text{ times}}) + \mathscr{O}(h_n^{k+1}) \tag{16}$$

with

$$V_i = \frac{1}{h_n}\big(\hat{U}_{ni} - \tilde{u}_n\big) = c_i \varphi_1(c_i h_n A) F(\tilde{u}_n) + \sum_{j=2}^{i-1} a_{ij}(h_n A)\hat{D}_{nj}. \tag{17}$$

Inserting (16) into (10b), we get

$$\begin{aligned} \hat{u}_{n+1} = \tilde{u}_n &+ h_n \varphi_1(h_n A) F(\tilde{u}_n) \\ &+ \sum_{i=2}^{s} b_i(h_n A) \sum_{q=1}^{k} \frac{h_n^{q+1}}{q!} g^{(q)}(\tilde{u}_n)(\underbrace{V_i, \dots, V_i}_{q \text{ times}}) + \mathscr{O}(h_n^{k+2}). \end{aligned} \tag{18}$$

In order to construct methods of order 5 we set $k = 4$ and compute V_i.

Lemma 1. *Under Assumptions 1 and 2, we have*

$$\varphi_1(c_i h_n A) F(\tilde{u}_n) = \tilde{u}_n' + \frac{c_i h_n}{2!}\mathbf{X}_i + \frac{c_i^2 h_n^2}{3!}\mathbf{Y}_i + \frac{c_i^3 h_n^3}{4!}\mathbf{Z}_i + \mathscr{O}(h_n^4) \tag{19}$$

with

$$\begin{aligned} \mathbf{X}_i &= \tilde{u}_n'' - 2!\varphi_2(c_i h_n A)\,\mathbf{L}, \quad \mathbf{Y}_i = \tilde{u}_n^{(3)} - 3!\varphi_3(c_i h_n A)\,\mathbf{M}, \\ \mathbf{Z}_i &= \tilde{u}_n^{(4)} - 4!\varphi_4(c_i h_n A)\,\mathbf{N}. \end{aligned} \tag{20}$$

Proof. It is easy to see from (1) that $Au^{(k)}(t) = u^{(k+1)}(t) - \frac{\mathrm{d}^k}{\mathrm{d}t^k} g(u(t))$. Thus $Au^{(k)}(t)$ is bounded for all k. Evaluating it at $t = t_n$ for $k = 1, 2, 3$ by using the chain rule, we obtain expressions for $A\tilde{u}_n'$, $A\tilde{u}_n''$, and $A\tilde{u}_n^{(3)}$. Using $F(\tilde{u}_n) = \tilde{u}_n'$ and employing the recurrence relation $\varphi_k(h_n A) = \frac{1}{k!} + h_n A\varphi_{k+1}(h_n A)$, we get

$$\varphi_1(c_i h_n A) F(\tilde{u}_n) = \tilde{u}_n' + \frac{c_i h_n}{2!}\mathbf{X}_i + \frac{c_i^2 h_n^2}{3!}\mathbf{Y}_i + \frac{c_i^3 h_n^3}{4!}\mathbf{Z}_i + h_n^4 c_i^4 \varphi_5(c_i h_n A) A\tilde{u}_n^{(4)}.$$

□

In the subsequent analysis, we use the abbreviations $a_{ij} = a_{ij}(h_n A)$, $b_i = b_i(h_n A)$, and

$$\psi_{j,i} = \psi_{j,i}(h_n A) = \sum_{k=2}^{i-1} a_{ik}(h_n A)\frac{c_k^{j-1}}{(j-1)!} - c_i^j \varphi_j(c_i h_n A). \tag{21}$$

Lemma 2. *Under Assumptions 1 and 2, the following holds*

$$\begin{aligned} V_i = {} & c_i \tilde{u}_n' + h_n\Big(\frac{c_i^2}{2!}\tilde{u}_n'' + \psi_{2,i}\,\mathbf{L}\Big) \\ & + h_n^2\Big(\frac{c_i^3}{3!}\tilde{u}_n^{(3)} + \psi_{3,i}\,\mathbf{M} + \sum_{j=2}^{i-1} a_{ij} g'(\tilde{u}_n)\psi_{2,j}\,\mathbf{L}\Big) + h_n^3\Big(\frac{c_i^4}{4!}\tilde{u}_n^{(4)} + \psi_{4,i}\,\mathbf{N} \\ & + \sum_{j=2}^{i-1} a_{ij} g'(\tilde{u}_n)\psi_{3,j}\,\mathbf{M} + \sum_{j=2}^{i-1} a_{ij} g'(\tilde{u}_n) \sum_{k=2}^{j-1} a_{jk} g'(\tilde{u}_n)\psi_{2,k}\,\mathbf{L} \\ & + \sum_{j=2}^{i-1} a_{ij} c_j\, g''(\tilde{u}_n)(\tilde{u}_n', \psi_{2,j}\,\mathbf{L})\Big) + \mathscr{O}(h_n^4). \end{aligned} \tag{22}$$

Proof. Using (16) and (17) repeatedly, one obtains the following representations

$$\hat{D}_{nj} = h_n g'(\tilde{u}_n)V_j + \frac{h_n^2}{2!}g''(\tilde{u}_n)(V_j, V_j) + \frac{h_n^3}{3!}g^{(3)}(\tilde{u}_n)(V_j, V_j, V_j) + \mathscr{O}(h_n^4),$$

$$V_j = c_j\varphi_1(c_j h_n A)F(\tilde{u}_n) + \sum_{k=2}^{j-1} a_{jk}\Big(h_n g'(\tilde{u}_n)V_k + \frac{h_n^2}{2!}g''(\tilde{u}_n)(V_k, V_k)\Big) + \mathscr{O}(h_n^3),$$

$$V_k = c_k\varphi_1(c_k h_n A)F(\tilde{u}_n) + h_n\sum_{l=2}^{k-1} a_{kl}g'(\tilde{u}_n)V_l + \mathscr{O}(h_n^2) \quad \text{and}$$

$$V_l = c_l\varphi_1(c_l h_n A)F(\tilde{u}_n) + \mathscr{O}(h_n).$$

Applying Lemma 1 to the first terms of V_j, V_k, V_l and then sequentially inserting V_l into V_k, V_k into V_j, and V_j into $\hat{D}_{nj}$, we obtain the full expression of $\hat{D}_{nj}$ with the remainder $\mathscr{O}(h_n^4)$. Substituting this into (17), employing Lemma 1 once more and combining all obtained terms we get (22). □

The following result follows immediately from Lemma 2.

Lemma 3. *Under Assumptions 1 and 2 we have*

$$g^{(4)}(\tilde{u}_n)(V_i, V_i, V_i, V_i) = c_i^4 g^{(4)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n', \tilde{u}_n') + \mathscr{O}(h_n),$$

$$g^{(3)}(\tilde{u}_n)(V_i, V_i, V_i) = c_i^3 g^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n') + h_n\Big(\tfrac{3}{2}c_i^4 g^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \tilde{u}_n'')$$
$$+ 3c_i^2 g^{(3)}(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n', \psi_{2,i}\,\mathbf{L})\Big) + \mathscr{O}(h_n^2),$$

$$g''(\tilde{u}_n)(V_i, V_i) = c_i^2 g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n') + h_n\Big(c_i^3 g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n'')$$
$$+ 2c_i g''(\tilde{u}_n)(\tilde{u}_n', \psi_{2,i}\,\mathbf{L})\Big) + h_n^2\Big(\frac{c_i^4}{3}g''(\tilde{u}_n)(\tilde{u}_n', \tilde{u}_n^{(3)}) + 2c_i g''(\tilde{u}_n)(\tilde{u}_n', \psi_{3,i}\,\mathbf{M})$$
$$+ 2c_i g''(\tilde{u}_n)\big(\tilde{u}_n', \sum_{j=2}^{i-1} a_{ij}g'(\tilde{u}_n)\psi_{2,j}\,g'(\tilde{u}_n)\tilde{u}_n'\big) + \frac{c_i^4}{4}g''(\tilde{u}_n)(\tilde{u}_n'', \tilde{u}_n'')$$
$$+ c_i^2 g''(\tilde{u}_n)(\tilde{u}_n'', \psi_{2,i}\,\mathbf{L}) + g''(\tilde{u}_n)(\psi_{2,i}\,\mathbf{L}, \psi_{2,i}\,\mathbf{L})\Big) + \mathscr{O}(h_n^3). \qquad \square$$

Employing the results of Lemma 3, we get the expansion of the numerical solution

$$\hat{u}_{n+1} = \tilde{u}_n + h_n\varphi_1(h_n A)F(\tilde{u}_n) + h_n^2\big(\sum_{i=2}^{s} b_i c_i\big)\mathbf{L} + h_n^3\Big(\sum_{i=2}^{s} b_i\frac{c_i^2}{2!}\Big)\mathbf{M}$$
$$+ h_n^4\Big(\sum_{i=2}^{s} b_i\frac{c_i^3}{3!}\Big)\mathbf{N} + h_n^5\Big(\sum_{i=2}^{s} b_i\frac{c_i^4}{4!}\Big)\mathbf{P} + \mathbf{R} + \mathscr{O}(h_n^6) \tag{23}$$

with **L**, **M**, **N** and **P** as in (15), and the remaining terms

$$
\begin{aligned}
\mathbf{R} = {} & h_n^3 \sum_{i=2}^{s} b_i g'(\tilde{u}_n)\psi_{2,i}\,\mathbf{L} + h_n^4 \sum_{i=2}^{s} b_i g'(\tilde{u}_n)\psi_{3,i}\,\mathbf{M} \\
& + h_n^4 \sum_{i=2}^{s} b_i g'(\tilde{u}_n) \sum_{j=2}^{i-1} a_{ij} g'(\tilde{u}_n)\psi_{2,j}\,\mathbf{L} + h_n^4 \sum_{i=2}^{s} b_i c_i g''(\tilde{u}_n)\big(\tilde{u}_n', \psi_{2,i}\,\mathbf{L}\big) \\
& + h_n^5 \sum_{i=2}^{s} b_i g'(\tilde{u}_n)\psi_{4,i}\,\mathbf{N} + h_n^5 \sum_{i=2}^{s} b_i g'(\tilde{u}_n) \sum_{j=2}^{i-1} a_{ij} g'(\tilde{u}_n)\psi_{3,j}\,\mathbf{M} \\
& + h_n^5 \sum_{i=2}^{s} b_i g'(\tilde{u}_n) \sum_{j=2}^{i-1} a_{ij} g'(\tilde{u}_n) \sum_{k=2}^{j-1} a_{jk} g'(\tilde{u}_n)\psi_{2,k}\,\mathbf{L} \\
& + h_n^5 \sum_{i=2}^{s} b_i g'(\tilde{u}_n) \sum_{j=2}^{i-1} a_{ij} c_j g''(\tilde{u}_n)\big(\tilde{u}_n', \psi_{2,j}\,\mathbf{L}\big) + h_n^5 \sum_{i=2}^{s} b_i c_i g''(\tilde{u}_n)\big(\tilde{u}_n', \psi_{3,i}\,\mathbf{M}\big) \\
& + h_n^5 \sum_{i=2}^{s} b_i c_i g''(\tilde{u}_n)\big(\tilde{u}_n', \sum_{j=2}^{i-1} a_{ij} g'(\tilde{u}_n)\psi_{2,j}\,\mathbf{L}\big) + h_n^5 \sum_{i=2}^{s} \frac{b_i}{2!} g''(\tilde{u}_n)\big(\psi_{2,i}\,\mathbf{L}, \psi_{2,i}\,\mathbf{L}\big) \\
& + h_n^5 \sum_{i=2}^{s} b_i \frac{c_i^2}{2!} g''(\tilde{u}_n)\big(\tilde{u}_n'', \psi_{2,i}\,\mathbf{L}\big) + h_n^5 \sum_{i=2}^{s} b_i \frac{c_i^2}{2!} g^{(3)}(\tilde{u}_n)\big(\tilde{u}_n', \tilde{u}_n', \psi_{2,i}\,\mathbf{L}\big).
\end{aligned}
$$

4.2 Local Error and Derivation of Stiff Order Conditions

Now we are ready to study the order conditions. Let $\tilde{e}_{n+1} = \hat{u}_{n+1} - \tilde{u}_{n+1}$ denote the local error, i.e., the difference between the numerical solution $\hat{u}_{n+1}$ after one step starting from $\tilde{u}_n$ and the corresponding exact solution of (1) at t_{n+1}, and let

$$
\psi_j(h_n A) = \sum_{i=2}^{s} b_i(h_n A) \frac{c_i^{j-1}}{(j-1)!} - \varphi_j(h_n A), \quad j \geq 2.
$$

Subtracting (14) from (23) gives

$$
\begin{aligned}
\tilde{e}_{n+1} = {} & h_n^2 \psi_2(h_n A)\,\mathbf{L} + h_n^3 \psi_3(h_n A)\,\mathbf{M} + h_n^4 \psi_4(h_n A)\,\mathbf{N} \\
& + h_n^5 \psi_5(h_n A)\,\mathbf{P} + \mathbf{R} + \mathcal{O}(h_n^6).
\end{aligned} \tag{24}
$$

The stiff order conditions can easily be identified from (24). They are summarized in Table 1. Note that the last two terms in $\mathbf{R}$ give rise to the same order condition, which is labeled 18 in Table 1.

The first nine conditions in Table 1 are the same as in [5]. Note that the method satisfies $c_1 = 0$ and $\psi_{j,1} = 0$ for all j. Therefore, all sums in Table 1 with the very exception of the first one start with the lower index 2.

5 Convergence Analysis

With the above local error analysis at hand, we are now ready to prove convergence.

Theorem 1. *Let the initial value problem (1) satisfy Assumptions 1 and 2. Consider for its numerical solution an explicit exponential Runge–Kutta method (7) that fulfills the order conditions of Table 1 up to order p for some $2 \le p \le 5$. Then, the numerical solution u_n satisfies the error bound*

$$\|u_n - u(t_n)\| \le C \sum_{i=0}^{n-1} h_i^{p+1}, \tag{25}$$

uniformly on $t_0 \le t_n \le T$. In particular, the constant C can be chosen independently of the step size sequence h_i in $[t_0, T]$.

Proof. The proof is quite standard. It only remains to verify that the numerical scheme (7) is stable. For this, let v_k and w_k denote two approximations to $u(t_k)$ at time t_k. Performing $n - k$ steps ($n > k$) gives

$$v_n = \mathrm{e}^{(h_{n-1}+\ldots+h_k)A} v_k + \sum_{m=k}^{n-1} h_m \mathrm{e}^{(h_{n-1}+\ldots+h_{m+1})A} \sum_{i=1}^{s} b_i(h_m A) g(V_{mi})$$

and a similar expression for w_n. Using the Lipschitz condition of g and the stability estimate (8) on the semigroup shows the bound

$$\|v_n - w_n\| \le \tilde{C} \Big(\|v_k - w_k\| + \sum_{m=k}^{n-1} h_m \|v_m - w_m\| \Big)$$

with a constant $\tilde{C}$ that can be chosen uniformly in n and k for $t_0 \le t_k \le t_n \le T$. The application of a standard Gronwall inequality thus proves stability.

We now make use of the fact that the global error $u_n - u(t_n)$ can be estimated by the sum of the propagated local errors $\hat{u}_k - \tilde{u}_k$, $k = 1, \ldots, n$. Due to the stability of the error propagation, we obtain at once (25). □

A discussion of the solvability of the order conditions given in Table 1, sample methods and numerical experiments will be published elsewhere.

Acknowledgements This work was supported by the FWF doctoral program 'Computational Interdisciplinary Modelling' W1227. The work of the first author was partially supported by the Tiroler Wissenschaftsfond grant UNI-0404/1284.

References

1. Cox, S., Matthews, P.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**, 430–455 (2002)
2. Friedli, A.: Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme. In: Burlirsch, R., Grigorieff, R., Schrödinger, J. (eds.) Numerical Treatment of Differential Equations. Lecture Notes in Mathematics, vol. 631, pp. 35–50. Springer, Berlin (1978)
3. Henry, D.: Geometric Theory of Semilinear Parabolic Equations. Lecture Notes in Mathematics, vol. 840. Springer, Berlin/Heidelberg (1981)
4. Hochbruck, M., Lubich, C., Selhofer, H.: Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput. **19**, 1552–1574 (1998)
5. Hochbruck, M., Ostermann, A.: Explicit exponential Runge–Kutta methods for semilinear parabolic problems. SIAM J. Numer. Anal. **43**, 1069–1090 (2005)
6. Hochbruck, M., Ostermann, A.: Explicit integrators of Rosenbrock-type. Oberwolfach Rep. **3**, 1107–1110 (2006)
7. Hochbruck, M., Ostermann, A.: Exponential integrators. Acta Numer. **19**, 209–286 (2010)
8. Hochbruck, M., Ostermann, A., Schweitzer, J.: Exponential Rosenbrock-type methods. SIAM J. Numer. Anal. **47**, 786–803 (2009)
9. Kassam, A.-K., Trefethen, L.N.: Fourth-order time stepping for stiff PDEs. SIAM J. Sci. Comput. **26**, 1214–1233 (2005)
10. Luan, V.T., Ostermann, A.: Exponential Rosenbrock methods of order five – construction, analysis and numerical comparisons. J. Comput. Appl. Math. **255**, 417–431 (2014)
11. Pazy, A.: Semigroups of Linear Operators and Applications to Partial Differential Equations. Springer, New York (1983)
12. Tokman, M.: Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. J. Comput. Phys. **213**, 748–776 (2006)

A Reduced-Order Strategy for Solving Inverse Bayesian Shape Identification Problems in Physiological Flows

Andrea Manzoni, Toni Lassila, Alfio Quarteroni, and Gianluigi Rozza

Abstract A reduced-order strategy based on the reduced basis (RB) method is developed for the efficient numerical solution of statistical inverse problems governed by PDEs in domains of varying shape. Usual discretization techniques are infeasible in this context, due to the prohibitive cost entailed by the repeated evaluation of PDEs and related output quantities of interest. A suitable reduced-order model is introduced to reduce computational costs and complexity. Furthermore, when dealing with inverse identification of shape features, a reduced shape representation allows to tackle the geometrical complexity. We address both challenges by considering a reduced framework built upon the RB method for parametrized PDEs and a parametric radial basis functions approach for shape representation. We present some results dealing with blood flows modelled by Navier-Stokes equations.

1 Introduction

In a parametrized context, given a mathematical model of a system the forward problem consists in evaluating some outputs of interest (depending on the PDE solution) for specified parameter inputs. Whenever some parameters are uncertain,

A. Manzoni • G. Rozza (✉)
MATHICSE-CMCS Modelling and Scientific Computing, École Polytechnique Fédérale de Lausanne, Switzerland

Now at SISSA, MathLab, International school for Advanced Studies, Trieste, Italy
e-mail: andrea.manzoni@epfl.ch; gianluigi.rozza@epfl.ch; grozza@sissa.it

T. Lassila
MATHICSE-CMCS Modelling and Scientific Computing, École Polytechnique Fédérale de Lausanne, Switzerland
e-mail: toni.lassila@epfl.ch

A. Quarteroni
MATHICSE-CMCS Modelling and Scientific Computing, École Polytechnique Fédérale de Lausanne, Switzerland

Dipartimento di Matematica, MOX Modeling and Scientific Computing, Politecnico di Milano, Italy
e-mail: alfio.quarteroni@epfl.ch

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_12

we aim at inferring their values (and/or distributions) from indirect observations by solving an inverse problem: given an observed output, can we deduce the value of the parameters that resulted in this output? Parameter identification can be performed in two ways, either in a deterministic or in a statistical framework. In the former case, we solve an optimization problem by minimizing (in the least-square sense) the *discrepancy* between the output quantities predicted by the PDE model and observations: this leads to a single-point estimate in the parameter space, provided the optimization problem is feasible. In the latter case, we quantify the relative likelihood of the parameters, which are consistent with the observed output. Following a Bayesian approach, this results in the posterior probability density function, which includes information both on prior knowledge on parameters distribution and on the model used to compute the PDE-based outputs. Inverse problems governed by PDEs entail several computational challenges for current discretization techniques, such as the finite element method. When the parameters to be identified are related with the shape of the domain, the problem is even more complicated. In this framework, computational costs arise from three distinct sources: (i) numerical approximation of the state system (usually a nonlinear system of PDEs); (ii) handling domains of arbitrary shapes; (iii) sampling high-dimensional parameter spaces or performing numerical optimization procedures.

In this paper, we address these challenges by developing a reduced framework based on both *state* and *parameter* reduction, in order to devise a low-dimensional, computationally inexpensive but accurate model that predicts outputs of a high-fidelity, computationally expensive model.

The reduction in state is obtained through a reduced basis (RB) approximation [7]: thanks to a suitable offline/online stratagem, *online* PDE evaluations for any value of input parameters are completely independent of the expensive *offline* computation and storage of the basis functions. On the other hand, when input parameters are related to geometrical features, we rely on low-dimensional but flexible shape parametrizations, able to represent wide families of complex shapes by means of a handful of input parameters.

2 Inverse Problems Governed by PDEs

We introduce a compact description of general inverse problems governed by parametrized PDEs. We denote by $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$ the finite-dimensional vector of parameters to be identified, and consider an input-output map $\boldsymbol{\mu} \mapsto \boldsymbol{y}(\boldsymbol{\mu})$ from parameters to observations that is given by two discretized PDEs (taken here linear for notational simplicity):

$$\begin{aligned} \text{State equation:} \quad & A_N(\boldsymbol{\mu})\boldsymbol{u}_N(\boldsymbol{\mu}) = \boldsymbol{f}_N(\boldsymbol{\mu}) \\ \text{Observation equation:} \quad & \boldsymbol{y}_N(\boldsymbol{\mu}) = C_N(\boldsymbol{\mu})\boldsymbol{u}_N(\boldsymbol{\mu}) \end{aligned} \tag{1}$$

State variables and observed outputs are denoted by $\boldsymbol{u}_N \in \mathbb{R}^N$ and $\boldsymbol{y}_N \in \mathbb{R}^M$, respectively. By subscript N we signify the dimension of the state space, $\dim(\boldsymbol{u}_N) = N$, which in the case of Finite Element (FE) discretizations is typically very large, whereas the dimension of the parameter space, $\dim(\boldsymbol{\mu}) = P$, and of the observation space, $\dim(\boldsymbol{y}_N) = M$ can be different and typically $P, M \ll N$. In our case, $\boldsymbol{\mu}$ is related to the shape of the domain $\Omega = \Omega(\boldsymbol{\mu})$ where the state problem is posed.

Whereas the forward problem is to evaluate $\boldsymbol{y}(\boldsymbol{\mu})$ given $\boldsymbol{\mu}$, the inverse problem can be formulated as follows [3,5]: given an observation $\boldsymbol{y}^* = \boldsymbol{y} + \boldsymbol{\varepsilon}$ with (additive) noise $\boldsymbol{\varepsilon}$, find the parameter $\boldsymbol{\mu}^*$ that satisfies $\mathbf{y}^* = C_N(\boldsymbol{\mu}^*)\boldsymbol{u}_N(\boldsymbol{\mu}^*)$. This problem is often ill-posed in one of three basic ways: (i) the solution $\boldsymbol{\mu}^*$ does not exist (e.g. due to $M > P$ and the presence of noise); (ii) the solution $\boldsymbol{\mu}^*$ is not unique (e.g. due to $M < P$ or data degeneracy); or (iii) the solution $\boldsymbol{\mu}^*$ does not depend continuously on $\boldsymbol{y}^*$. An example of an inverse problem that is ill-posed in the third sense is the *Calderón problem* of determining the conductivity field inside an object based on the observation of a Dirichlet-to-Neumann or Neumann-to-Dirichlet map on a subsection of the boundary.

2.1 A Deterministic Approach

In order to treat ill-posed inverse problems, the classical approaches [1] are largely based on solving regularized least-squares (RLS) problems of the type:

$$\boldsymbol{\mu}^*_{\mathrm{RLS}} = \arg\min_{\mu \in \mathbb{R}^P} \quad \tfrac{1}{2}\|\boldsymbol{y}^* - \boldsymbol{y}_N(\boldsymbol{\mu})\|_2^2 + \tfrac{\alpha}{2}\|\boldsymbol{\mu} - \boldsymbol{\mu}_{\mathrm{prior}}\|_r^2. \tag{2}$$

The first term minimizes the discrepancy between the observation $\boldsymbol{y}^*$ and the model prediction $\boldsymbol{y}_N(\boldsymbol{\mu})$ given by (1). The second term convexifies the problem and assures a unique estimator $\boldsymbol{\mu}^*_{\mathrm{RLS}}$ is recovered. This approach is also sometimes called *variational data assimilation*. The choice of the norm $\|\boldsymbol{\mu}\|_r := \sqrt{\boldsymbol{\mu}^T R \boldsymbol{\mu}}$, the regularization parameter $\alpha > 0$, and the prior value $\boldsymbol{\mu}_{\mathrm{prior}}$ play an important role in the quality of the estimator $\boldsymbol{\mu}^*_{\mathrm{RLS}}$.

2.2 A Bayesian Approach

Under the assumption of independent and identically distributed (i.i.d.) noise, $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I)$, and Gaussian parameter distribution, $\mu \sim \mathcal{N}(\bar{\mu}, \Sigma)$, it is easy to show that the maximum a posteriori (MAP) estimator

$$\boldsymbol{\mu}^*_{\mathrm{MAP}} := \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathbb{R}^P} \quad \pi_{\boldsymbol{\mu}\,|\,\boldsymbol{y}^*}(\boldsymbol{\mu} \mid \boldsymbol{y}^*) \tag{3}$$

obtained by maximizing the conditional probability density function

$$\pi_{\boldsymbol{\mu} \mid \boldsymbol{y}^*}(\boldsymbol{\mu} \mid \boldsymbol{y}^*) \sim \exp\left(-\tfrac{1}{2}\|\boldsymbol{y}^* - \boldsymbol{y}_N(\boldsymbol{\mu})\|_2^2 - \tfrac{\sigma^2}{2}(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^T \Sigma^{-1}(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})\right)$$

coincides with the Tykhonov-regularized least-squares estimator, that is to say $\boldsymbol{\mu}^*_{\text{RLS}} = \boldsymbol{\mu}^*_{\text{MAP}}$, as long as we choose $\boldsymbol{\mu}_{\text{prior}} = \bar{\boldsymbol{\mu}}$, $\alpha = \sigma^2$, and $R = \Sigma^{-1}$ in (2). In fact, the estimator given by (3) is an example of a wider class of statistical estimators called *Bayesian estimators*. The benefit of using statistical methods for solving inverse problems is that one is able to characterize the variance of the prediction $\boldsymbol{\mu}^*$ due to measurement and model errors more precisely than from the single-point estimates obtained by solving (2).

Bayesian estimators are a subset of statistical estimators that are widely used to solve ill-posed inverse problems. The basic principle of Bayesian inference is that the conditional distribution of the unknown parameters $\boldsymbol{\mu}$ given an observation $\boldsymbol{y}^*$ can be approximated by

$$\pi_{\boldsymbol{\mu}|\boldsymbol{y}^*}(\boldsymbol{\mu}|\boldsymbol{y}^*) = \frac{\pi_{\boldsymbol{y}^*|\boldsymbol{\mu}}(\boldsymbol{y}^*|\boldsymbol{\mu})\pi_{\boldsymbol{\mu}}(\boldsymbol{\mu})}{\pi_{\boldsymbol{y}}(\boldsymbol{y}^*)} \approx \frac{\pi_{\boldsymbol{y}^*|\boldsymbol{\mu}}(\boldsymbol{y}^*|\boldsymbol{\mu})\pi_{\boldsymbol{\mu},\text{prior}}(\boldsymbol{\mu})}{\pi_{\boldsymbol{y}}(\boldsymbol{y}^*)} =: \pi_{\boldsymbol{\mu},\text{post}}(\boldsymbol{\mu}|\boldsymbol{y}^*)$$

obtained by using the Bayes' formula on a prior distribution $\pi_{\boldsymbol{\mu},\text{prior}}(\boldsymbol{\mu})$ for the unknown parameters. The prior $\pi_{\boldsymbol{\mu},\text{prior}}$ encapsulates our prior knowledge (structure, regularity, locality, etc.) about the distribution of the uncertain parameters and should be carefully selected based on problem-specific considerations – we do not treat this point in this work since selecting an informative prior is a challenging problem all by itself. The conditional distribution $\pi_{\boldsymbol{y}^*|\boldsymbol{\mu}}$, which in the case of additive noise can be expressed as

$$\pi_{\boldsymbol{y}^* \mid \boldsymbol{\mu}}(\boldsymbol{y}^* \mid \boldsymbol{\mu}) = \pi_{\text{noise}}(\boldsymbol{y}^* - \boldsymbol{y}_N(\boldsymbol{\mu})),$$

is called the *likelihood function*. The posterior *distribution* $\pi_{\boldsymbol{\mu},\text{post}}$ can then be used to compute various estimators for $\boldsymbol{\mu}^*$ and to provide conditional statistics such as covariances for these estimators. The advantage of the Bayesian approach compared to more classical methods is that a prior that carries sufficient information about the true underlying structure of the parameters often provides more meaningful estimates and regularizes the inverse problem in a more natural way than relying on abstract regularization terms, as in (2), that might not have any interpretation.

Statistical methods used to solve an inverse problem can be computationally much more expensive than the deterministic approach due to the necessity of performing sampling in high-dimensional spaces in order to compute sample statistics [3,5]. This cost is exacerbated by the fact that each evaluation requires the solution of the forward problem in the form of a (potentially large-scale) discrete PDE. To this end we introduce a reduced order model to speed up the computations entailed by statistical inversion.

3 Computational and Geometrical Reduction

We now present a brief description of the two main blocks on which the reduced order model relies: reduced basis method for parametrized PDEs and radial basis functions for low-dimensional shape parametrization. Further methodological aspects and details and can be found e.g. in [6].

The reduced basis method provides an efficient way to compute an approximation $\mathbf{u}_n(\boldsymbol{\mu})$ of the solution $\mathbf{u}_N(\boldsymbol{\mu})$ (as well as an approximation $\boldsymbol{y}_n(\boldsymbol{\mu})$ of the output $\boldsymbol{y}_N(\boldsymbol{\mu})$) through a Galerkin projection onto a reduced subspace made up of well-chosen full-order solutions (also called *snapshots*), corresponding to a set of parameter values $S_n = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^n\}$ selected by means of a *greedy* algorithm [7]. Let us denote by $Z_n \in \mathbb{R}^{N\times n}$ the matrix

$$Z_n = [\mathbf{u}_1(\boldsymbol{\mu}) \mid \ldots \mid \mathbf{u}_n(\boldsymbol{\mu})] \tag{4}$$

obtained by aligning the snapshot vectors (a Gram-Schmidt orthonormalization procedure has to be considered after each basis is added to the reduced space, but for the sake of simplicity we consider the same notation). We denote by $n \ll N$ the dimension of the *reduced* state space. Then, the reduced-order solution is given by a linear combination $Z_n\mathbf{u}_n(\boldsymbol{\mu})$ of the snapshots, being $\boldsymbol{u}_n \in \mathbb{R}^n$ the solution of the following problem:

$$\begin{aligned} \text{State equation:} \quad & A_n(\boldsymbol{\mu})\boldsymbol{u}_n(\boldsymbol{\mu}) = \boldsymbol{f}_n(\boldsymbol{\mu}) \\ \text{Observation equation:} \quad & \boldsymbol{y}_n(\boldsymbol{\mu}) = C_n(\boldsymbol{\mu})\boldsymbol{u}_n(\boldsymbol{\mu}), \end{aligned} \tag{5}$$

where

$$A_n(\boldsymbol{\mu}) = Z_n^T A_N(\boldsymbol{\mu}) Z_n, \qquad \mathbf{f}_n = Z_n^T \mathbf{f}_N, \qquad C_n = C_N Z_n.$$

To get very fast input/output evaluations, RB methods rely on the assumption of affine parametric dependence in $A_N(\boldsymbol{\mu})$ and $\boldsymbol{f}_N(\boldsymbol{\mu})$, i.e. on the possibility to express $A_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_A} \Theta_A(\boldsymbol{\mu}) A_N^q$ and $\boldsymbol{f}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f(\boldsymbol{\mu}) \boldsymbol{f}_N^q$, so that the expensive $\boldsymbol{\mu}$-independent quantities can be evaluated and stored just once. This is a property inherited by the PDE model, which can be eventually recovered at the discretization stage [7].

Once the reduced model is built in the offline stage, it can be exploited at the online stage to speed up the solution of the optimization problem (2) in the deterministic case or (3) in the Bayesian case. The corresponding reduced-order version of the former reads as follows:

$$\min_{\mu\in\mathbb{R}^P} \quad \tfrac{1}{2}\|\boldsymbol{y}^* - \boldsymbol{y}_n(\boldsymbol{\mu})\|_2^2 + \tfrac{\alpha}{2}\|\boldsymbol{\mu} - \boldsymbol{\mu}_{\text{prior}}\|_r^2, \tag{6}$$

whereas in the case of a statistical inverse problem we obtain:

$$\boldsymbol{\mu}^*_{\text{MAP}} := \underset{\boldsymbol{\mu}\in\mathbb{R}^P}{\text{argmax}} \quad \pi_{\boldsymbol{\mu},\text{post}}(\boldsymbol{\mu} \mid \boldsymbol{y}^*) \tag{7}$$

being

$$\pi_{\boldsymbol{\mu},\text{post}}(\boldsymbol{\mu}|\boldsymbol{y}^*) = \frac{\pi_{\text{noise}}(\boldsymbol{y}^* - \boldsymbol{y}_n(\boldsymbol{\mu}))\pi_{\boldsymbol{\mu},\text{prior}}(\boldsymbol{\mu})}{\pi_{\boldsymbol{y}}(\boldsymbol{y}^*)}.$$

In this way, state reduction allows to speed up both numerical optimization schemes or sampling algorithms required e.g. to compute statistical estimates based on the posterior distribution.

Concerning parameter space reduction, here we consider a low-dimensional parametrization based on Radial Basis Functions (RBF), an interpolatory technique which allows to define shape deformations through a set of control points (which can be freely chosen, according to the family of deformations to be described), i.e. a linear combination of affine and radial, nonaffine terms; see e.g. [6] for more insights. In this way, parameter space reduction is afforded by selecting only a small set of $P \approx \mathcal{O}(10)$ control points at a preceding stage – state reduction through the RB method is built for a problem where shape parametrization has already been performed. A RB paradigm for simultaneous state and parameter reduction has been introduced in [5] in order to tackle the case of distributed parametric fields (instead of parameter vectors), and represents a possible extension of our current framework.

4 Application and Results

We now apply the reduced framework of the previous section to the solution of an inverse problem arising in modeling of blood flows. Since a strong mutual interaction exists between haemodynamic factors and vessels geometry, improving the understanding of the interplay between flows and geometries may be useful not only for the sake of design of better prosthetic devices [4], but also to characterize pathological risks, such as in the case of narrowing or thickening of an arterial vessel [3]. Typical portions of cardiovascular network where lesions and pathologies may develop are made up by curved vessels and bifurcations; an important segment where vessel diseases are often clinically observed is the human carotid artery [2,6], which supplies blood to the head.[1]

[1]The common carotid artery (CCA) bifurcates in the lower neck into two branches, the internal and the external carotid arteries (ICA and ECA, respectively). Stenoses, that is the narrowing of the inner portion of an artery, manifest quite often in the ICA.

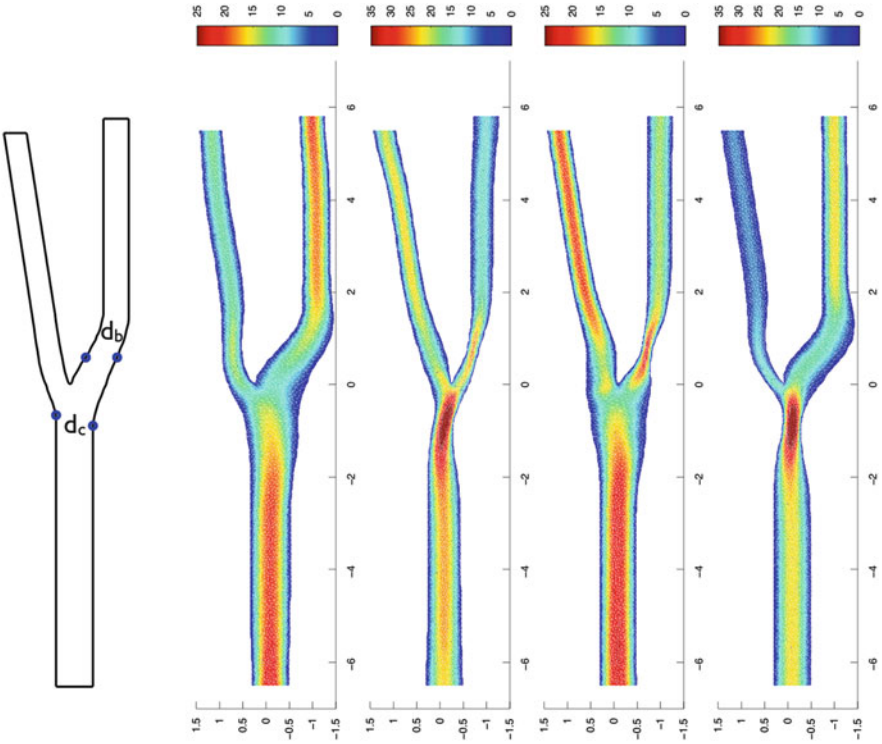

Fig. 1 *Left*: shape representation of a stenosed carotid artery bifurcation through RBF parametrization. *Right*: velocity profiles (cm/s) in four different carotid bifurcations parametrized with respect to the diameters $d_c = d_c(\mu_1, \mu_2)$ of the CCA at the bifurcation and $d_b = d_b(\mu_3, \mu_4)$ of the mid-sinus level of the ICA

Let us consider a steady, incompressible Navier-Stokes model to describe blood flows in a two-dimensional carotid bifurcation (see Fig. 1):

$$\begin{cases} -\nu \Delta \boldsymbol{v} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + \nabla p = \mathbf{f} & \text{in } \Omega(\mu) \\ \nabla \cdot \boldsymbol{v} = 0 & \text{in } \Omega(\mu) \\ \boldsymbol{v} = \boldsymbol{v}_{in} & \text{on } \Gamma_{in} \\ \boldsymbol{v} = \mathbf{0} & \text{on } \Gamma_{w} \\ -p \cdot \mathbf{n} + \nu \dfrac{\partial \boldsymbol{v}}{\partial \mathbf{n}} = \mathbf{0} & \text{on } \Gamma_{out} \end{cases} \tag{8}$$

being $(\boldsymbol{v}, p)$ the velocity and the pressure of the fluid, respectively, and $\nu > 0$ its kinematic viscosity. In view of studying computationally expensive inverse problems, which entail the repeated simulation of these flow equations, we cannot afford at the moment the solution of PDE models involving more complex features,

such as flow unsteadiness and arterial wall deformability – computational costs would be too prohibitive.

In this context, a typical forward problem is the evaluation of flow indices related with geometry variation that assess/measure the occlusion risk. Typical examples are given by vorticity, shear rates, wall shear stresses. On the other hand, we might be interested in recovering some geometrical features by observing some physical index related to flow variables. In particular, the inverse problem we want to solve is the following: is it possible to identify the entity of the occlusions (i.e. the diameters d_c of the CCA at the bifurcation and d_b of the mid-sinus level of the ICA, respectively) from the observation of the mean pressure drop

$$y(\boldsymbol{\mu}) = \int_{\Gamma_{in}} p(\boldsymbol{\mu})\, d\Gamma - \int_{\Gamma_{out}} p(\boldsymbol{\mu})\, d\Gamma$$

between the internal carotid outflow Γ_{out} and the inflow Γ_{in}?

To exploit the reduced framework presented in Sect. 3, we represent local shape deformations through a RBF parametrization built over $p = 4$ control points (represented as the bullets in Fig. 1), located in one of the branches and close to the bifurcation. In this case, Gaussian RBFs have been used in order to describe local but moderate deformations representing possible stenoses, being $\boldsymbol{\mu} \in \mathcal{D} = [-0.25, 0.25]^4$ the vector of the displacements of the control point in the horizontal direction; see [6] for more details.

By applying the RB method to the parametrized Navier-Stokes problem (8) we reduce the dimension of the state space from $N \approx 26{,}000$ ($\mathbb{P}_2/\mathbb{P}_1$ FE discretization) to $n = 45$. Four examples of computed RB solutions are reported in Fig. 1. We remark the strong sensitivity of the flow with respect to varying diameters $d_c = d_c(\mu_1, \mu_2)$ of the CCA at the bifurcation and $d_b = d_b(\mu_3, \mu_4)$ of the mid-sinus level of the ICA, respectively. See e.g. [3, 6] for more insights on RB methodology for nonlinear Navier-Stokes equations.

Thus, we can take advantage of both the deterministic and the Bayesian framework to solve this inverse identification problem, by considering surrogate measurements of the mean pressure drop.

In the first case, we demonstrate the solution of the deterministic inverse problem for two different observed values of the pressure drop, $s^* = -1{,}400$ and $s^* = -2{,}200$, by assuming 5 % relative additive noise in the measurements. The results of the inverse identification problem are given in Fig. 2 for 100 realization of random noise in both cases: each point in the graph corresponds to the recovered diameters (d_c, d_b) given a noisy observation. We observe that in the case $s^* = -1{,}400$ recovered values of the diameters are more smeared out, since locally the pressure drop surface is almost flat, but result is close in values to the considered observation.

Thus, in the former case $s^* = -1{,}400$ the inverse problem is worse conditioned than in the latter $s^* = -2{,}200$, where the recovered values (d_c, d_b) lie in a smaller region of the space. However, the solution of a single optimization problem is more feasible in the former case compared to the latter: solving 100 optimization

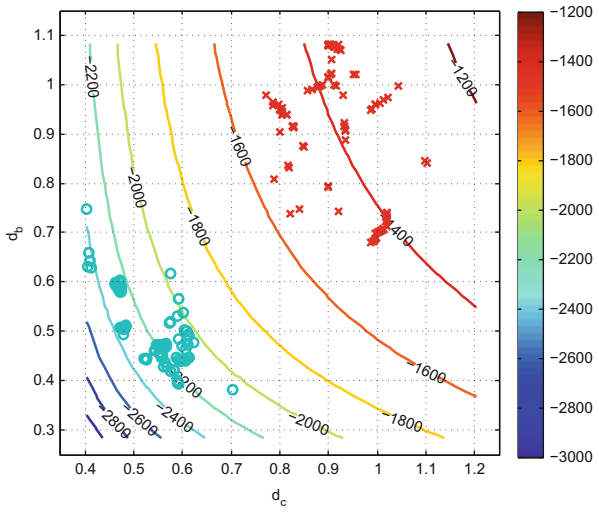

Fig. 2 Results of the deterministic inverse problems for $s^* = -1{,}400$ (in *red*) and $s^* = -2{,}200$ (in *green*). Isocontours of the pressure drop (RB Navier-Stokes problem)

problems took about 14 h in the former and about 25.6 h in the latter case, respectively. We remark that solving 100 inverse problems of this type through a full-order discretization technique would have been infeasible on a standard workstation. Thus, even in presence of small noises, the result of a deterministic inverse problem may be very sensitive – just when one diameter is known, the second one can be recovered. This is due to the fact that several geometrical configurations – in terms of diameters (d_c, d_b) – may correspond to the same output observation.

Following instead the Bayesian approach, we are able to characterize a *set* of configurations, rather than a single configuration: this is done by providing the *joint* probability distribution function for the (uncertain) diameters (d_c, d_b) encapsulating the noise related to measurements, as discussed in Sect. 2.2. Let us denote by $\mathbf{d} = (d_c, d_b)^T \in \mathbb{R}^2$ the vector of the two diameters and assume that the prior distribution is $\pi_{\mathbf{d},\text{prior}}(\boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{d}_M, \Sigma_M)$, being $\mathbf{d}_M \in \mathbb{R}^2$ the (prior) mean and $\Sigma_M \in \mathbb{R}^{2\times 2}$ the (prior) covariance matrix, encapsulating a possible prior knowledge on the diameters distribution (e.g. from observations of previous shape configurations). By supposing that also the measurements of the pressure drop are expressed by i.i.d. Gaussian variables, such that $\pi_{\text{noise}}(\boldsymbol{y}^* - \boldsymbol{y}_n(\mathbf{d})) \sim \mathcal{N}(0, \sigma^2)$, we can compute the explicit form of the posterior probability density $\pi_{\mathbf{d},\text{post}}(\mathbf{d}|\boldsymbol{y}^*)$. Thus, provided some preliminary information on plausible values of the diameters, the observation of a (large) sample of outputs allows to characterize a set of plausible configurations as the ones maximizing the posterior probability density $\pi_{\mathbf{d},\text{post}}(\mathbf{d}|\boldsymbol{y}^*)$. In particular, we consider two different realizations of prior normal distributions, obtained by choosing the mean $\mathbf{d}_M = (0.803, 0.684)^T$ as given by the diameters corresponding to the reference carotid configuration, and

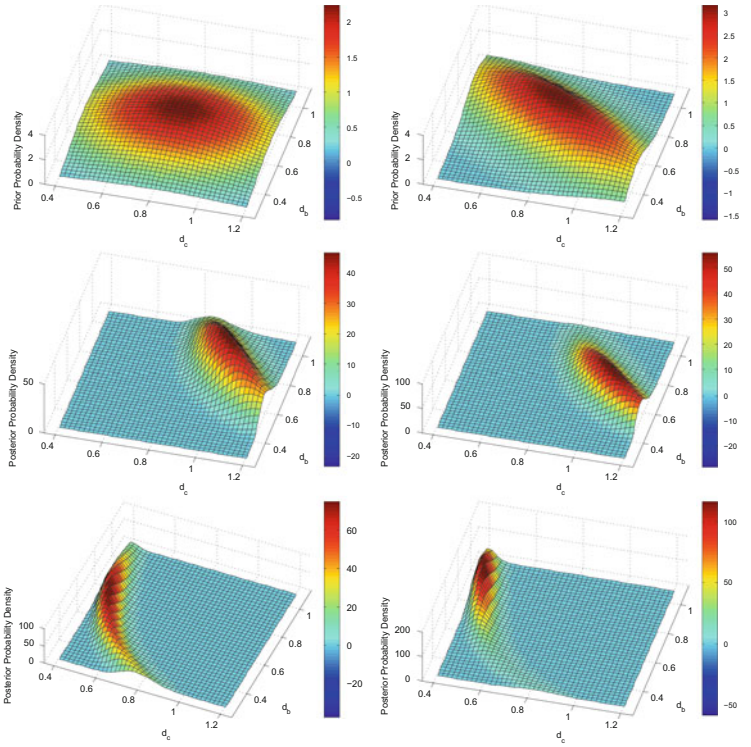

Fig. 3 *Top*: two different choices of the prior distribution on diameters $\mathbf{d} = (d_c, d_b)^T$; *left*: $\pi_{\mathbf{d},\text{prior}} \sim \mathcal{N}(\mathbf{d}_M, \Sigma_{M_1})$, *right*: $\pi_{\mathbf{d},\text{prior}} \sim \mathcal{N}(\mathbf{d}_M, \Sigma_{M_2})$. *Center* and *bottom*: results of the Bayesian inverse problems (*left*: $\pi_{\mathbf{d},\text{prior}} \sim \mathcal{N}(\mathbf{d}_M, \Sigma_{M_1})$, *right*: $\pi_{\mathbf{d},\text{prior}} \sim \mathcal{N}(\mathbf{d}_M, \Sigma_{M_2})$) and observed pressure drop $s^* = -1{,}400$ (*second row*) and $s^* = -2{,}200$ (*third row*)

$$\Sigma_{M_1} = \begin{bmatrix} 0.025 & 0 \\ 0 & 0.0125 \end{bmatrix}, \qquad \Sigma_{M_2} = \begin{bmatrix} 0.025 & -0.0125 \\ -0.0125 & 0.0125 \end{bmatrix},$$

i.e., we assume that the two diameters are a priori independent (Σ_{M_1} case) or correlated (Σ_{M_2} case), respectively. The two prior distributions, as well as the resulting posterior distribution obtained for two different observed values $s^* = -1{,}400$ and $s^* = -2{,}200$ of the pressure drop are reported in Fig. 3. In the case at hand, we do not rely on the Metropolis-Hastings algorithm for the evaluation of the posterior distribution, since its expression can be computed explicitly. Thus, by computing a sample of 1,600 values of pressure drops on a uniform 40×40 grid on the (d_c, d_b) space, we obtain the posterior densities $\pi_{\mathbf{d},\text{post}}(\mathbf{d}|\boldsymbol{y}^*)$ represented in Fig. 3 in about 0.1 h, since any online evaluation of the reduced Navier-Stokes problem takes about 2.5 s.

Acknowledgements This work was partially funded by the European Research Council Advanced Grant "Mathcard, Mathematical Modelling and Simulation of the Cardiovascular System" (Project ERC-2008-AdG 227058), and by the Swiss National Science Foundation (Projects 122136 and 135444).

References

1. Kaipio, J., Somersalo, E.: Statistical and Computational Inverse Problems. Applied Mathematical Sciences, vol. 160. Springer, New York (2005)
2. Kolachalama, V., Bressloff, N., Nair, P.: Mining data from hemodynamic simulations via Bayesian emulation. Biomed. Eng. Online **6**(1), 47 (2007)
3. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: A reduced computational and geometrical framework for inverse problems in haemodynamics. Technical report MATHICSE 12.2011. **29**(7), 741–776 (2013). http://mathicse.epfl.ch/
4. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: Boundary control and shape optimization for the robust design of bypass anastomoses under uncertainty. ESAIM: Math. Mod. Numer. Anal. **47**(4), 1107–1131 (2013). http://dx.doi.org/10.1051/m2an/2012059
5. Lieberman, C., Willcox, K., Ghattas, O.: Parameter and state model reduction for large-scale statistical inverse problems. SIAM J. Sci. Comput. **32**(5), 2523–2542 (2010)
6. Manzoni, A., Quarteroni, A., Rozza, G.: Model reduction techniques for fast blood flow simulation in parametrized geometries. Int. J. Numer. Methods Biomed. Eng. **28**(6–7), 604–625 (2012)
7. Quarteroni, A., Rozza, G., Manzoni, A.: Certified reduced basis approximation for parametrized partial differential equations in industrial applications. J. Math. Ind. **1**, 3 (2011)

A Mathematical Study of Sprinting on Artificial Legs

Katja Mombaur

Abstract In 2008, the remarkable performance of the double amputee sprinter Oscar Pistorius initiated a discussion of whether his running prostheses might give him an advantage over able-bodied sprinters. He uses carbon fiber Cheetah devices by Össur that have spring-like properties; and the assumption was that the high passive torques and the lower moments of inertia of the prosthetic lower legs more than compensate for the absence of active ankle torques. The purpose of our research is to use mathematical models and optimal control techniques to better understand the underlying mechanics and control of sprinting on prostheses and to bring new insights into the continuing discussion. We established rigid multibody system models for the hybrid dynamics of able-bodied as well as double amputee sprinters. In the present study, we use models in the sagittal plane with 9 bodies and 11 degrees of freedom. In the able-bodied case, there are torque actuators at all eight internal joints; in the double amputee case, the actuators at the ankles are replaced by linear spring damper elements, but the other six actuators remain. Running motions for this model are generated by solving a multiphase optimal control problem with discontinuities and periodicity constraints, using an efficient direct multiple shooting approach.

1 Introduction

Before the Olympic Games 2008 in Beijing a big discussion started about whether the double amputee sprinter Oscar Pistorius, who came remarkably close to Olympic standards in the 400 m race, should be allowed to participate. In the original study presented by Brüggemann [3] and co-workers the claim was that the carbon fiber Cheetah devices (Össur) that Pistorius uses actually give him an advantage over able-bodied sprinters and that the spring-like properties of the prostheses and the lower moment of inertia of the lower leg more than compensate for the absence of an active ankle torque.

K. Mombaur (✉)
Interdisciplinary Center for Scientific Computing, University of Heidelberg, Speyerer Str. 6, 69115 Heidelberg, Germany
e-mail: kmombaur@uni-hd.de; katja.mombaur@iwr.uni-heidelberg.de

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_13

The debate between biomechanical experts is still not settled, as can be seen in the point-counterpoint article [8]. While one party (Weygand and Bundle) claims to have detected an indication of advantages of the amputee sprinter, the other party (Kram et al.) disputes that measurements from one single world class double amputee do not allow generalization of any findings. Also comparisons with single amputees may not help because it is assumed that persons with one healthy and one prosthetic leg will have to rely on completely different running strategies than double amputees. Some of the arguments (against Pistorius) in the above literature are based on very simple mass-spring models. While such models of the SLIP type have already been very useful for studying some gait characteristics [1, 6], we consider them misleading in this case because they do not give insights into the energetics and performance limits of the running systems.

But why is there such a controversy? Why is it not just considered as a great achievement that it is possible to provide specific running prostheses that allow amputees to reach able-bodied performance ranges – a goal that it still unreached in the case of (more versatile) prostheses for everyday walking motions? Or formulated simply: why don't they just let him run? As Burkett et al. [4] point out, there are several ethical issues to be considered:

- In a running competition the nature of the movement of athletes should clearly be running and not any form of bouncing, hopping etc. Studies of the knee angle range of Pistorius let this appear questionable.
- A permission for the springy carbon devices might open the field for other technologies, resulting e.g. in able-bodied athletes competing with giant springs below their feet and completing the race in a few bounces. Where to set the boundary between allowed and forbidden devices?
- In order to guarantee justice in sports, it is important to keep competitions at the performance level and not shift them to a technology level (as is the case in formula 1 races), which would put even greater burdens on athletes from poorer countries.

Later, Oscar Pistorius was allowed to compete (since no advantage could yet be proven) and has participated in the World Championships in South Korea in 2011 as well as the 2012 Olympics in London, which was a great success for him. But the discussion continues, and it seems necessary to apply alternative techniques to obtain new insights.[1]

The goal of this paper is to address the problem by means of Scientific Computing techniques. In order to evaluate double amputee sprinting, we are using detailed multibody system models of the human body (with healthy as well as

[1] Remark: Very sadly, at the time of completing the final version of this article (Spring 2013), 1 year after the HPSC conference, Pistorius is not competing any more since he is facing trial for murder of his girlfriend, R. Steenkamp. His personal future is unclear. However, the questions discussed in this article is still very relevant, since there are also other equally talented double amputee sprinters, such as Alan Oliveira from Brazil.

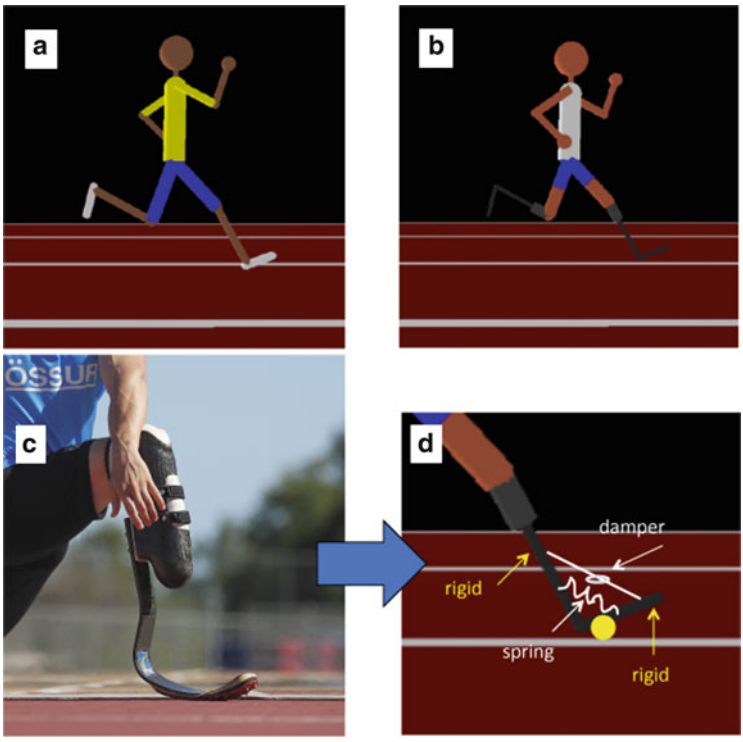

Fig. 1 *Top*: Stick figures of the able-bodied (**a**) and the double amputee (**b**) sprinting models. *Bottom*: Össur cheetah carbon fiber prostheses (**c**, courtesy of R. Saemunsson, Össur) and corresponding spring-damper model (**d**)

prosthetic legs (Fig. 1)) and optimal control techniques in order to generate natural dynamic running motions for both systems.

In Sect. 2, we present the mathematical models of running motions of able-bodied and amputee running that we established and used for the study. Section 3 describes the formulation and solution of optimal control problems to determine optimal running motions. In Sect. 4 we show numerical optimization results for both models. Section 5 contains a discussion and perspectives for future research in this area.

2 Modeling Running Motions With and Without Prostheses

In this study we use two different models, one of a double transtibial amputee sprinter, and – for reference purposes – a model of an able-bodied athlete with comparable figure, i.e. with the same geometric and inertial parameters for the

non-affected segments. To describe human running motions, we use a multibody system model in the sagittal plane with 9 segments: trunk, arms, upper legs, lower legs and feet. The model has 11 degrees of freedom (DOF) in flight: 3 global DOF associated with the position and orientation of the trunk and 8 internal DOF related to internal joint angles. In the case of the able-bodied sprinter, all 8 internal joints (shoulders, hips, knees and ankles) are equipped with torque actuators summarizing the action of all related muscles at these joints. For the double amputee sprinter, the two actuators at the ankles are replaced by linear spring damper elements, but the other six actuators remain.

The modeled subject has an overall weight of 83.3 kg and a height of 1.85 m corresponding to the data of Oscar Pistorius. For the anthropometric geometry and inertia data, we use an extrapolation of the de Leva data [5] to the desired height and weight, as well as data for the prostheses and the stump given in [3].

The model describes human-like forefoot running, i.e. there is no flat foot ground contact but only point-like contact with the ball of foot, which is assumed to be rigid and non-sliding. Detailed muscle dynamics are not included in the model yet, but we consider doing so at a later stage of this research.

From a mathematical perspective, models of running motions take the form of a periodic hybrid dynamical system. They consist of a sequence of alternating flight phases and single-leg contact phases. We consider here only running motions that are periodic and symmetric, i.e. right and left steps are identical. Each phase of the motion (flight phase and single-leg contact phase) is described by its own set of ordinary differential or differential-algebraic equations, as described below. Between phases, there may be discontinuities in the state variables, namely the velocities, e.g. at touchdown of the foot on the floor, which is assumed to be completely inelastic. These assumptions allow us to reduce the model of the periodic running motion to the model of a single step with a subsequent leg shift and periodicity constraints (see Fig. 2). The total time of the step T as well as the individual phase times are free variables of the model.

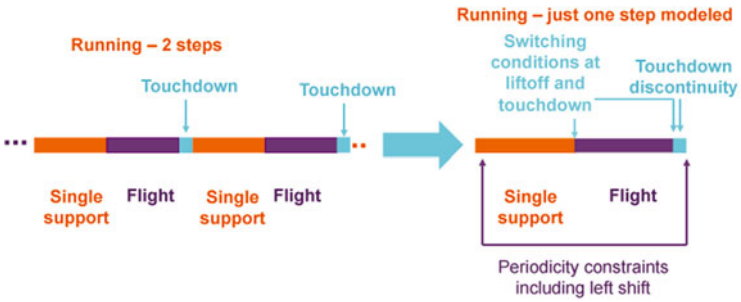

Fig. 2 The sequence of periodic running steps can be reduced to the model of one step with leg shift and periodicity constraints for the optimal control problem formulation

The motion during flight phase is described by a set of ordinary differential equations:

$$M(q, p)\ddot{q} + N(q, \dot{q}, p)\dot{q} = F(q, \dot{q}, p, \mathscr{M}), \tag{1}$$

which can also be written as a first-order system with position q and velocity $v = \dot{q}$ as state variables. M is the mass matrix containing the system's inertial properties, and N the vector of nonlinear effects, such as Coriolis, gyroscopic and centrifugal forces. F is the vector of all external forces, such as gravity, muscle torques $\mathscr{M}$, drag, etc. Multibody models of such complexity cannot be derived by hand but have to be established using automatic model generator software. We have used the tool HuMAnS by Wieber [15] to generate the terms M and N of the model.

During the single-leg contact phase, the number of DOF is reduced by two because the ball of one foot is in non-sliding contact with the ground. We keep the same number of coordinates and describe the reduction by a constraint of the form $g(q) = 0$, which results in a description by an index 3 differential algebraic equation (DAE). Index reduction finally results in an index 1 DAE system with invariants:

$$\dot{q} = v \tag{2}$$

$$\dot{v} = a \tag{3}$$

$$\begin{pmatrix} M & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} = \begin{pmatrix} -N + F \\ -\gamma \end{pmatrix} \tag{4}$$

$$g_{pos} = g(q(t), p) = 0 \tag{5}$$

$$g_{vel} = G(q(t), p) \cdot \dot{q}(t) = 0. \tag{6}$$

Position p and velocity v are again differential state variables, and acceleration $a := \ddot{q}$ and Lagrange multipliers λ form the algebraic state variables. G denotes the Jacobian of the position constraints $G = (\partial g/\partial q)$, and γ the corresponding Hessian $\gamma = ((\partial G/\partial q)\,\dot{q})\,\dot{q}$. Equations (5) and (6) describe the invariant manifolds on position and velocity level (resulting from index reduction) that the solution must satisfy. In the optimization, we take into account the unilateral nature of the ground contact constraint (i.e. ground cannot pull but only push against the foot) by formulating an inequality constraint on the Lagrange multiplier associated with the normal contact force.

Phase switches between flight and contact phase do not take place at given time points but depend on the position variables of the human as expressed in the corresponding switching functions:

$$s(q(\tau_s), v(\tau_s), p) = 0. \tag{7}$$

Touch-down occurs when the foot gets down to the height of the ground, and lift-off takes place when the vertical contact force (represented by the negative of the respective Lagrange multiplier in Eq. (4)) becomes zero.

The discontinuities of the velocities at touchdown (resulting from the fact that the velocity of the foot contact point is instantly set to zero at inelastic contact and that this shock wave propagates through the whole body) can be computed as

$$\begin{pmatrix} M & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} v_+ \\ \Lambda \end{pmatrix} = \begin{pmatrix} M v_- \\ 0 \end{pmatrix} \tag{8}$$

using the same matrices as above. Periodicity constraints are imposed in the model on all velocity variables v and a reduced set of position variables q_{red} eliminating the coordinate describing the forward running direction of the robot, after formulation of the leg shift.

3 Computing Optimal Running Motions

Optimal running motions can be generated by solving a multi-phase optimal control problem with discontinuities:

$$\min_{x(\cdot),u(\cdot),\tau} \quad \sum_{j=1}^{n_{ph}} \left(\int_{\tau_{j-1}}^{\tau_j} \phi_j(x(t),u(t))\, dt + \Phi_j(\tau_j, x(\tau_j)) \right) \tag{9}$$

$$\text{s. t.} \quad \dot{x}(t) = f_j(t, x(t), u(t)) \quad \text{for} \quad t \in [\tau_{j-1}, \tau_j], \quad j = 1, \ldots, n_{ph}, \ \tau_0 = 0, \tau_{n_{ph}} = T \tag{10}$$

$$x(\tau_j^+) = x(\tau_j^-) + J(\tau_j^-) \quad \text{for} \quad j = 1, \ldots, n_{ph} \tag{11}$$

$$g_j(t, x(t), u(t)) \geq 0 \quad \text{for} \quad t \in [\tau_{j-1}, \tau_j], \quad j = 1, \ldots, n_{ph} \tag{12}$$

$$r_{eq}(x(0), \ldots, x(T)) = 0 \tag{13}$$

$$r_{ineq}(x(0), \ldots, x(T)) \geq 0 \tag{14}$$

In this formulation, $x(t)$ represents the vector of state variables (summarizing position and velocity variables) and $u(t)$ is the vector of control variables (here joint torques $\mathscr{M}_i$ produced by the muscles). For one step, the number of phases $n_{ph} = 2$. τ is the vector of free phase switching times with total step time $T = \tau_{n_{ph}}$. Equation (9) describes the objective function in a general form and is further discussed below. Equations (10) and (11) are placeholders for the hybrid dynamic model of the running motion discussed in the previous section. In addition, there are continuous inequality constraints of form (12), including lower and upper bounds on all variables, but also more complex relations between several variables, and coupled and decoupled point-wise equality and inequality constraints (13) and (14), such as start and end point constraints, phase switching conditions or periodicity constraints.

For the solution of these multi-phase optimal control problems, we use the powerful optimal control code MUSCOD developed at IWR Heidelberg [2, 9, 10]. This code can be applied to mechanical DAEs of the above form, as described in [13]. MUSCOD uses a direct method (also called first-discretize-then-optimize approach) for control discretization, and multiple shooting for state parameterization. The result of these two discretization steps for which identical grids are chosen is a nonlinear programming problem (NLP) of large dimension. It is solved by a specially tailored sequential quadratic programming (SQP) method that uses condensing in the solution of each QP subproblem.

4 Numerical Results

We now present the numerical results for the optimal control problem solutions for both the able-bodied runner model and the double amputee model. In both cases, an objective function minimizing the integral over the weighted sum of all joint torques squared has been applied

$$\min_{x(\cdot),u(\cdot),T} \quad \int_0^T \sum_{i=1,\ldots,8} (w_i u_i^2)\, dt \tag{15}$$

where w_i are weight factors taking into account the respective maximum torque at each joint. Note that $u_3 \equiv u_6 \equiv 0$ in the double amputee case, since there is no active torque in the ankle. Our experience has shown that this objective function, with and without additionally minimizing the variation of torques, creates very natural motions for a variety of specified motion tasks [7, 12, 14]. The task set in this case is a periodic running motion at a given average speed of $9\,\mathrm{km/h} = 32.4\,\mathrm{m/s}$ which is about the top speed in 400 m races. This speed is imposed as a constraint of type (13) in the optimal control problem. Humans are highly redundant systems which are capable of performing motions in an infinite number of ways, some of them more natural, and some of them quite awkward (some not very serious but convincing demonstration of the latter case are given in the Monty Python sketch "The ministry of silly walks"). The above objective function (15) serves to select from all possible periodic motions at that speed the one that is characterized by minimum effort in that particular measure and that also is perceived as a natural way of performing the motions.

Figure 3 shows animation sequences of the optimized solutions for both models.

Figure 4 contains the histories of all torque variables for the able-bodied and the amputee running solution. The meaning of the eight torque plots is explained in the figures. As mentioned before, the active torque in the ankles is zero for the running motion with prostheses. It is remarkable that all other six torques are much smaller in the double amputee case than in the able-bodies case (corresponding torque plots in the upper and lower part of the figure have the same scale). It seems that much

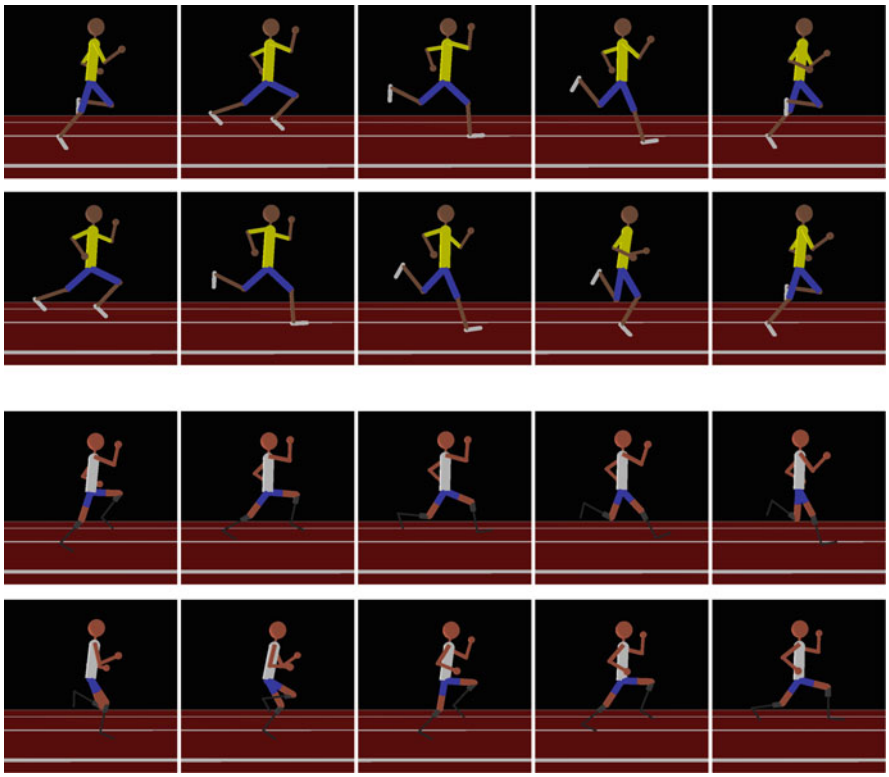

Fig. 3 Animation sequences of optimized able-bodied and double amputee running

of the work actively done in the able-bodied case can be compensated by passive action of the spring element, such that much less active work is required.

This impression of Fig. 4 is supported by the computational results for different criteria given in Fig. 5. For comparison purposes, we have computed the following effort-related criteria:

- The integral over the absolute values of the control variables (= torques) $\int_0^T |u_i| dt$
- The integral over the squared of the control values $\int_0^T u_i^2 dt$
- The mechanical work $\int_0^T (u_i \dot{\phi}_i) dt$, where $\dot{\phi}_i$ denotes the angular velocity at joint i
- The absolute mechanical work $\int_0^T |u_i \dot{\phi}_i| dt$.

All integrals are approximately computed in terms of Riemann integrals using information at multiple shooting and control grid points. In the tables below, values are given for each joint separately, as well as the sums over all joints $\sum_i$, except for the case of the mechanical work where a sum would be meaningless since positive and negative values would computationally cancel out each other, while in reality there is no gain of energy due to negative mechanical work in a joint. All values are much lower in the double amputee case.

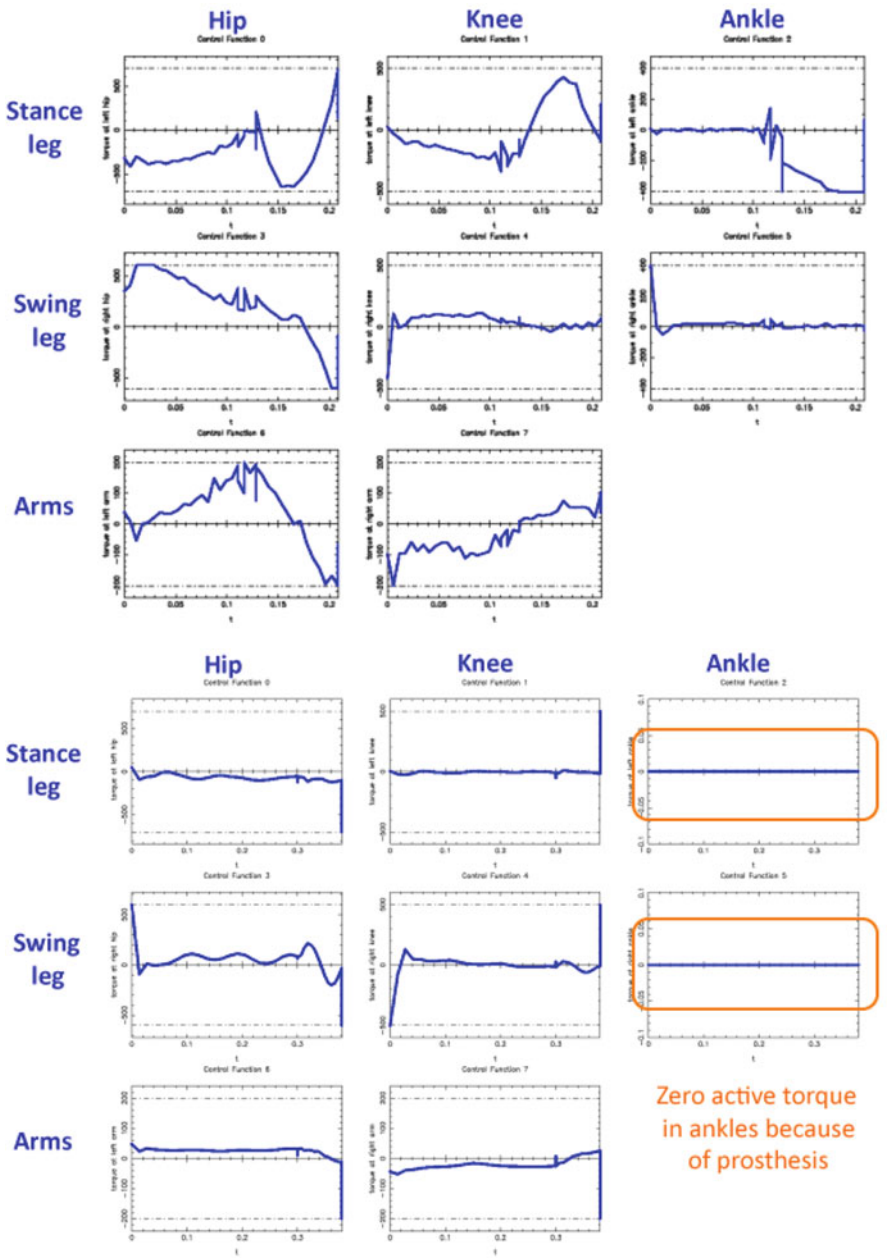

Fig. 4 Comparison of torque variables for able-bodied (*top*) and double amputee running (*bottom*)

Integral over absolute values of controls

	healthy	amputee
0	70.66	26.67
1	36.50	2.452
2	29.72	0.0
3	59.01	36.61
4	12.82	17.25
5	6.81	0.0
6	12.51	10.02
7	15.12	9.58
sum	243.15	102.59

Integral over squared control values

	healthy	amputee
0	27226.6	2164.8
1	9184.1	31.54
2	9265.8	0.0
3	16874.0	8136.9
4	2429.8	3981.9
5	1487.6	0.0
6	896.9	286.1
7	1217.3	277.6
sum	68582.4	14878.9

Mechanical Work

	healthy	amputee
0	24995	1933
1	8646	20
2	8157	0
3	15972	2110
4	-88	721
5	-538	0
6	770	267
7	1064	262
sum	---	---

Absolute mechanical work

	healthy	amputee
0	25332	2054
1	8710	25
2	8349	0
3	16004	3640
4	843	1011
5	568	0
6	786	267
7	1069	262
sum	61665	7259

Fig. 5 Comparison of four different criteria for able-bodied and double amputee running

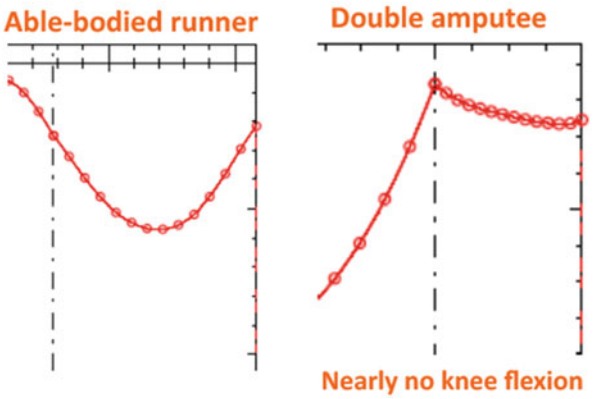

Fig. 6 Comparison of knee flexion angles for optimal solutions of the two models (the plots show the relative knee angles as functions of time, and the *dashed line marks* the start of the contact phase while the final *solid lines marks* lift-off)

Another difference that has previously been observed in the experiments [3] is that the knee angle flexion is much smaller in running motions with prostheses than in able-bodied running, leading sometimes to the claim that the resulting motion can not be classified as running any more. This observation of the smaller knee angle-variation is confirmed by the computational results (see Fig. 6). The resulting running style of the double amputee model is clearly different from the one of the able-bodied model, but it still looks like what a layman would define as running (compare Fig. 3). We find it hard to decide at this point where the line between running and other types of motions should be drawn.

5 Discussion and Perspectives

This paper presents first steps towards an analysis of prosthetic running motions using dynamical models and optimal control techniques. The ultimate goal of this research is to address the question if springy prosthetic devices can present

an advantage or disadvantage with respect to able-bodied running in the case of 400 m races. We are still quite far from answering this question, but the computations presented in this paper were meant to demonstrate that mathematical optimal control can be a useful tool to investigate motions, and in particular those types of motions for which no sufficient basis of experimental motion capture data is available. It could be shown in this paper that indeed the double amputee running motions require much less effort than the able-bodied motions in terms of several different measures, i.e. it is more efficient.

However, it is important to point out the obvious fact that the goal of a track race is not to run with the least effort, but to be the fastest at the end. While in a marathon race the more efficient runner will have a severe advantage at the end, it is undisputed that in a 100 m race the goal is to produce as much power as possible in the very short time of the race. For 400 m it is not so clear, since runners tend to fatigue towards the end, and efficiency may play a role, but certainly it is not the only deciding factor. Together with cooperation partners from biomechanics we will perform an extensive literature study about the right performance criterion for the different races (100, 200 and 400 m) and how to properly formulate this criterion based on the dynamical model. We are also investigating the question about the objective functions applied by able-bodied and amputee sprinters during these races. Mathematically, this results in the solution of an inverse optimal control problem based on motion capture measurements of the respective sprinters, as started in [11] with a three-dimensional 25 DOF model.

In addition, we are working on several modeling questions in order to have a more realistic representation of the problem. Is a mechanical model using torque inputs sufficient to perform the study, or do models of muscle fatigue, metabolic energy consumption etc. have to be taken into account? Is there a way to relate these quantities to joint torques? Another concern is how realistic (individual) torque limits for able-bodied and amputee sprinters can be determined from measurements and taken into account for the computations. A realistic model also would have to consider the fact that not only the missing links and joints are affected (clearly Pistorius has no active ankle torques and different properties of the lower legs), but also the adjacent segments and joints are affected, since muscles responsible for the knee motion which have their insertion points in the lower leg are impaired due to the amputation. Due to two-joint muscles, this may even affect joints that are further away.

Last, but not least, it remains to be determined which race situations should be considered for a fair comparison. So far only cyclic running at uniform (top) speed has been considered in this study as well as in the other studies mentioned in the introduction. However, 400 m races also include other phases, such as a propulsive start from a forward crouched, half-lying position, an acceleration phase and also a deceleration phase after the finish line (which does not have to be optimal but at least feasible without a fall). While it may be comparatively easy to optimize the design of such a passive device as a carbon fiber spring for one particular mode of operation, i.e. a running motion at one selected speed, it seems quite impossible to tune it simultaneously for the different challenges of the different phases of motion.

Pictures of Pistorius in the starting blocks with his extremely long lower prosthetic legs suggest that he has a disadvantage at least in this phase of the race, if not in others, too. Our future computations will consider multiple set point optimization for the different phases of the race.

Acknowledgements Financial support by the German Excellence Initiative within the third funding line is gratefully acknowledged. We thank the Simulation and Optimization group of H. G. Bock at the University of Heidelberg for providing the optimal control code MUSCOD.

References

1. Blickhan, R.: The spring mass model for running and hopping. J. Biomech. **22**, 1217–1227 (1989)
2. Bock, H.G., Plitt, K.-J.: A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the 9th IFAC World Congress, Budapest, pp. 242–247. International Federation of Automatic Control (1984)
3. Brüggemann, G.-P., Arampatzis, A., Emrich, F., Potthast, W.: Biomechanics of double transtibial amputee sprinting using dedicated sprinting prostheses. Sports Technol. **1**(4–5), 220–227 (2008)
4. Burkett, B., McNamee, M., Potthast, W.: Shifting boundaries in sports technology and disability: equal rights or unfair advantage in the case of Oscar Pistorius? Disabil. Soc. **26**(5), 643–654 (2011)
5. de Leva, P.: Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. J. Biomech. **9**, 1223–1230 (1996)
6. Geyer, H., Seyfarth, A., Blickhan, R.: Compliant leg behaviour explains basic dynamics of walking and running. Proc. R. Soc. Lond. B **273**, 2861–2876 (2006)
7. Koschorreck, J., Mombaur, K.: Modeling and optimal control of human platform diving with somersaults and twists. Optim. Eng. **12**(4), 29–56 (2011)
8. Kram, R., Grabowski, A., McGowan, C., Brown, M., McDermott, W., Beale, M., Herr, H., Weygand, P., Bundle, M.: Point – counterpoint: artificial legs do/do not make artificially fast running speeds possible. J. Appl. Physiol. **108**(4), 1012–1014 (2010)
9. Leineweber, D.B., Bauer, I., Bock, H.G., Schlöder, J.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization – part I: theoretical aspects. Comput. Chem. Eng. **27**, 157–166 (2003)
10. Leineweber, D.B., Schäfer, A., Bock, H.G., Schlöder, J.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization – part II: software aspects and applications. Comput. Chem. Eng. **27**, 167–174 (2003)
11. Mombaur, K., Olivier, A.H., Cretual, A.: Forward and inverse optimal control of bipedal running. In: Modeling, Simulation and Optimization of Bipedal Walking. COSMOS, vol. 18, pp. 165–179. Springer, Berlin, Heidelberg (2013)
12. Mombaur, K., Scheint, M., Sobotka, M.: Optimal control and design of legged robots with compliance. at-Automatisierungstechnik **57**(7), 349–359 (2009)
13. Mombaur, K.D., Bock, H.G., Schlöder, J.P., Longman, R.W.: Open-loop stable solution of periodic optimal control problems in robotics. ZAMM – J. Appl.Math. Mech. **85**(7), 499–515 (2005)
14. Schultz, G., Mombaur, K.: Modeling and optimal control of human-like running. IEEE/ASME Trans. Mech. **15**(5), 783–792 (2010)
15. Wieber, P.-B.: Humans toolbox. http://www.inrialpes.fr/bipop/software/humans/ (2007)

Hilbert Space Treatment of Optimal Control Problems with Infinite Horizon

Sabine Pickenhain

Abstract We consider a class of infinite horizon optimal control problems as optimization problems in Hilbert spaces. For typical applications it is demonstrated that the state and control variables belong to a Weighted Sobolev – and Lebesgue space, respectively. In this setting Pontryagin's Maximum Principle as necessary condition for a strong local minimum is shown. The obtained maximum principle includes transversality conditions as well.

1 Introduction

This paper is devoted to the theory of Pontryagin's Maximum Principle derived to infinite horizon optimal control problems. This special class of problems arises in the theory of economic growth and in processes where the time T is an exponentially distributed random variable, see [1, 11]. Known results about the validity of Pontryagin's Maximum Principle use an approximation approach, [1, 2, 6]. Halkin in [6] remarks: "The first tendency, when one considers optimization problems with infinite horizon, is to assume that all results which are known for the finite horizon case can be carried to the infinite horizon case by replacing evaluations of quantities at the terminal time with evaluations of the limit of the same quantities as the time tends to infinity." The limitation of this approach is underscored in the cited paper of Halkin [6]. In the present paper we propose a completely different approach. We show that in typical applications the state and control variable belongs to a Weighted Sobolev- and Lebesgue space respectively. If the problem is formulated in the Hilbert spaces $W_2^1(\mathbb{R}^+, \mu)$ for the state and $L_2(\mathbb{R}^+, \mu)$ for the control, it can be treated by Hilbert space methods. Making appropriate assumptions on the growth of the data of the problem, we can prove Pontryagin's Maximum Principle as a separation theorem in Hilbert spaces. In contradiction to the first approach the obtained maximum principle includes also transversality conditions and the existence of optimal solutions can be guaranteed, see [9].

S. Pickenhain (✉)
Brandenburg University of Technology, Cottbus, Germany
e-mail: sabine.pickenhain@tu-cottbus.de; sabine@math.tu-cottbus.de

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_14

2 Applications

(a) Optimal economic growth: The earliest consideration of an economic optimal control problem on an unbounded interval goes back to Ramsey [12]. In a recent version, the problem can be formulated as follows, see ([3], p. 6 ff.):

$$J(K,Z,C) = \int_0^\infty e^{-\varrho\, t}\, U(\,C(t)\,)\, dt \longrightarrow \text{Max !} \tag{1}$$

$$F(\,K(t)\,) = Z(t) + C(t)\,, \tag{2}$$

$$\dot{K}(t) = Z(t) - \mu\, K(t)\,, \tag{3}$$

$$K(0) = k_0\,. \tag{4}$$

Here the production function F and the utility function U are given while the investment resp. consumption rates Z and C and the capital stock K are optimization variables. Under certain assumptions on the data, it can be shown that there exists a constant capital level $\overline{k}$ such that, "for any nonnegative value of ϱ the optimal trajectory over an infinite time horizon exists and converges toward $\overline{k}$, and this is true for any initial state k_0" ([3], p. 8). From this property it is clear that the function K cannot belong to any usual Sobolev space but to a weighted space as introduced below.

(b) Production-inventory model: This model has been presented in ([13], pp. 154 ff.):

$$J(I,P) = \int_0^\infty e^{-\varrho\, t} \left(\frac{h}{2} \left(I(t) - \hat{I}(t) \right)^2 + \frac{c}{2} \left(P(t) - \hat{P}(t) \right)^2 \right) dt \longrightarrow \text{Min !} \tag{5}$$

$$\dot{I}(t) = P(t) - S(t)\,,\ I(0) = I_0\,. \tag{6}$$

Here $\hat{I}$ and $\hat{P}$ are given goal levels for inventory and production, S is the given sales rate, h and c are given positive coefficients, and the actual inventory and production rates I and P are optimization variables. Again, the optimal trajectory of the problem belongs to a weighted Sobolev space. Since the objective in this problem is similar to the norm in the weighted space $W_2^1(\mathbb{R}_+, \mu)$ with $\mu(t) = e^{-\varrho\, t}$, it seems to be very natural to choose $W_2^1(\mathbb{R}_+, \mu)$ as the state space. We mention that here and in the preceding example, the integrals have to be understood in the Lebesgue sense.

3 Problem Formulation

The considered applications belong to the following general class of infinite horizon optimal control problems:

$$(\mathrm{P})_{\infty}^{L}: \qquad J_{\infty}(x,u) = \int_0^{\infty} r(t,x(t),u(t))\,\mu(t)\,dt \longrightarrow \mathrm{Min}\ ! \tag{7}$$

$$(x,u) \in W_2^1(\mathbb{R}_+,\mu) \times L_2(\mathbb{R}_+,\mu)\,, \tag{8}$$

$$\dot{x}(t) = f(t,x(t),u(t)) \quad \text{a. e. on } \mathbb{R}_+\,,\ x(0) = x_0\,, \tag{9}$$

$$u(t) \in \mathrm{U} \subset \mathbb{R} \quad \text{a. e. on } \mathbb{R}_+ \tag{10}$$

The integral in (7) is the Lebesgue integral. The set of all *admissible pairs*, denoted by $\mathscr{A}_L$, consists of all processes satisfying (8)–(10) and make the Lebesgue integral in (7) finite. The function μ is a density function in the sense explained below. The weighted spaces $W_2^1(\mathbb{R}_+,\mu)$ and $L_2(\mathbb{R}_+,\mu)$ will be defined in the next session.

4 Weighted Lebesgue and Sobolev Spaces

Let us write $[\,0\,,\,\infty\,) = \mathbb{R}_+$. We denote by $M(\mathbb{R}_+)$, $L_p(\mathbb{R}_+)$ and $C^0(\mathbb{R}_+)$ the spaces of all functions $x\colon \mathbb{R}_+ \to \mathbb{R}$ which are Lebesgue measurable, in the pth power Lebesgue integrable or continuous, respectively, see ([4], p. 146 and pp. 285, [5], pp. 228). The Sobolev space $W_p^1(\mathbb{R}_+)$ is defined then as the space of all functions $x\colon \mathbb{R}_+ \to \mathbb{R}$, that belong to $L_p(\mathbb{R}_+)$ and admit distributional derivatives $\dot{x}$ ([14], p. 49) belonging to $L_p(\mathbb{R}_+)$ as well. A continuous function $\mu\colon \mathbb{R}_+ \to \mathbb{R}$ with positive values is called a *weight function*. A weight function is a *density function* iff it is Lebesgue integrable over $\mathbb{R}_+$, $\int_0^{\infty} \mu(t)\,dt < \infty$ holds, (see [8], p. 18). By means of a weight function $\mu \in C^0(\mathbb{R}_+)$, we define for any $1 \le p < \infty$ the *weighted Lebesgue space*

$$L_p(\mathbb{R}_+\,,\mu) = \left\{ x \in M(\mathbb{R}_+) \;\middle|\; \|\,x\,\|_{L_p(\mathbb{R}_+\,,\mu)} = \Big(\int_0^{\infty} |\,x(t)\,|^p\,\mu(t)\,dt \Big)^{1/p} < \infty \right\}.$$

For $x \in L_p(\mathbb{R}_+\,,\mu)$ we can define its distributional derivative $\dot{x}$, ([14], p. 46), and we are led to the *weighted Sobolev space* of those $L_p(\mathbb{R}_+\,,\mu)$ functions having a distributional derivative in $L_p(\mathbb{R}_+\,,\mu)$:

$$W_p^1(\mathbb{R}_+,\mu) = \left\{ x \in M(\mathbb{R}_+) \;\middle|\; x \in L_p(\mathbb{R}_+,\mu)\,,\ \dot{x} \in L_p(\mathbb{R}_+,\mu) \right\}$$

(see [8], p. 11 f.). Equipped with the norm

$$\| x \|_{W_p^1(\mathbb{R}_+,\mu)} \;=\; \| x \|_{L_p(\mathbb{R}_+,\mu)} \;+\; \| \dot{x} \|_{L_p(\mathbb{R}_+,\mu)} ,$$

$W_p^1(\mathbb{R}_+,\mu)$ becomes a Banach space (this can be confirmed analogously to ([8], p. 19). Any linear, continuous functional $\varphi : L_p(\mathbb{R}_+,\mu) \to \mathbb{R}$ can be represented by a function $\varphi \in L_q(\mathbb{R}_+,\mu)$ with $p^{-1}+q^{-1}=1$ if $1<p<\infty$ and $q=\infty$ if $p=1$:

$$\langle\, \varphi\,,\, x\,\rangle \;=\; \int_0^\infty \varphi(t)\, x(t)\, \mu(t)\, dt \quad \forall\, x \in L_p(\mathbb{R}_+,\mu)\,. \tag{11}$$

We may apply ([5], p. 287), since the measure generated by the density function μ is σ-finite on $\mathbb{R}_+$. For $p=2$ the spaces $L_2(\mathbb{R}_+\,,\mu)$ and $W_2^1(\mathbb{R}_+,\mu)$ become separable Hilbert spaces, see [8]. and (11) is the scalar product $L_2(\mathbb{R}_+\,,\mu)$.

5 Main Result and Conclusions

We prove Pontryagin's Maximum Principle for a model problem with one state, linear state equation with respect to state and control, and objective which is convex with respect to the control:

$$(\mathrm{P})_\infty^L: \qquad J_\infty(x,u) \;=\; \int_0^\infty r(t,x(t),u(t))\,\mu(t)\,dt \longrightarrow \text{Min !} \tag{12}$$

$$(x,u) \;\in\; W_2^1(\mathbb{R}_+,\mu)\times L_2(\mathbb{R}_+,\mu)\,, \tag{13}$$

$$\dot{x}(t) \;=\; A(t)x(t) + B(t)u(t) \;\text{ a. e. on } \mathbb{R}_+\;,\; x(0)=x_0\,, \tag{14}$$

$$u(t) \;\in\; \mathrm{U}\subset\mathbb{R} \;\text{ a. e. on } \mathbb{R}_+\,. \tag{15}$$

Let the following assumptions be satisfied for $(\mathrm{P})_\infty^L$.

- **V1**: $r, A, B \in C^1$ with respect to all its arguments, $\mu(t)$ is a density.
- **V2**: $r(t,\xi,\cdot)$ is convex on U for all $(t,\xi)\in\mathbb{R}_+\times\mathbb{R}$, U is convex and compact.
- **V3**: For all $(\zeta,w)\in L_2(\mathbb{R}_+,\mu)\times L_\infty(\mathbb{R}_+)$, $(\zeta(t),w(t))\in W(t)$, with

  $$W(t) := \{(\xi,v)\in \mathrm{R}^2 \,|\, |\xi - x^*(t)|\mu(t) < \varepsilon_0,\; v\in U\},\; t\in\mathbb{R}_+,\; \varepsilon_0>0,$$

  let

  $$r_v(\cdot,\zeta(\cdot),w(\cdot))\in L_2(\mathbb{R}_+,\mu) \quad \text{ and } \; r_\xi(\cdot,\zeta(\cdot),w(\cdot))\in L_2(\mathbb{R}_+,\mu)$$

- **V4**: $A, B\in L_\infty(\mathbb{R}_+)$ and $\exists T>0$ with $\int\limits_T^t 2A(s)ds < -\ln(\mu(t))\forall t\geq T$.

In the case of infinite horizon optimal control problems we can find several optimality criteria, which are adapted either to problems with improper Lebesque integrals J_{lim}:

$$J_{lim} := \lim_{T\to\infty} L\text{-}\int_0^T f(t)\,dt$$

see [3], or to problems $(P)^L_\infty$ with Lebesgue integrals, see [11].

Our considerations are focused on *global* optimality in the sense of Lebesgue integrals:

Definition 1. Let processes $(x,u),(x^*,u^*) \in \mathscr{A}_L$ be given. Then the pair $(x^*,u^*) \in \mathscr{A}_L$ is called *globally* optimal for $(P)^L_\infty$ (**criterion L1**), if for any pair $(x,u) \in \mathscr{A}_L$ holds

$$\int_0^\infty r(t,x(t),u(t))\mu(t)\,dt - \int_0^\infty r(t,x^*(t),u^*(t))\mu(t)\,dt \geq 0. \tag{16}$$

The maximum principle for $(P)^L_\infty$ reads as follows:

Theorem 1. *Let assumptions **V1–V4** be satisfied and $(x^*,u^*) \in \mathscr{A}_L$ be an optimal solution of $(P)^L_\infty$ in the sense of criterion **L1**. Then there are multipliers (λ_0, y), with*

$$\lambda_0 = 1 \tag{N}$$

$$y \in W_2^1(\mathbb{R}^+,\mu^{-1}), \quad \lim_{T\to\infty} y(T) = 0 \tag{T}$$

$$H(t,x^*(t),u^*(t),y(t),\lambda_0) = \max_{v\in U} H(t,x^*(t),v,y(t),\lambda_0) \quad \text{a.e. on } \mathbb{R}^+ \tag{M}$$

$$\dot{y}(t) = -H_\xi(t,x^*(t),u^*(t),y(t),\lambda_0) \quad \text{a.e. on } \mathbb{R}^+, \tag{K}$$

where $H : \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the Pontryagin function,

$$H(t,\xi,v,\eta,\lambda_0) = -\lambda_0 r(t,\xi,v)\mu(t) + \eta f(t,\xi,v).$$

Proof. **Step 1:**

Lemma 1 (Variational Lemma). *Let $(x,u) \in \mathscr{A}_L$ and $(x^*,u^*) \in \mathscr{A}_L$ optimal in the sense of criterion **L1**, then for all $\zeta := x - x^* \in W_2^1(\mathbb{R}^+,\mu)$, $\|\zeta\| \leq 1$, $w := u - u^* \in L_2(\mathbb{R}^+,\mu)$ the first order one-sided variation of the objective is nonnegative,*

$$0 \le \delta^+ J_\infty((x^*,u^*),(\zeta,w)) := \lim_{\varepsilon\to 0+0}\left\{\frac{J_\infty(x_\varepsilon,u_\epsilon)-J_\infty(x^*,u^*)}{\varepsilon}\right\}$$

$$\le \int_0^\infty \{r_\xi(t,x^*(t),u^*(t))\zeta(t)+[r(t,x^*(t),u(t))-r(t,x^*(t),u^*(t))]\}\mu(t)\,dt. \quad (17)$$

Proof. Let $x_\varepsilon(t) := x^*(t)+\varepsilon(x-x^*)(t)$, $u_\epsilon(t) := u^*(t)+\varepsilon(u-u^*)(t)$. Then for sufficiently small ε it holds $J_\infty(x_\varepsilon,u_\varepsilon) < \infty$, since

$$\begin{aligned} J_\infty(x_\varepsilon,u_\epsilon)-J_\infty(x^*,u^*) &= \int_0^\infty \{r(t,x_\varepsilon(t),u_\varepsilon(t))-r(t,x^*(t),u^*(t))\}\mu(t)dt \\ &= \varepsilon\int_0^\infty \{r_\xi(t,\tilde\xi_\varepsilon(t),\tilde u_\varepsilon(t))(x-x^*)(t) \\ &\quad + r_v(t,\tilde\xi_\varepsilon(t),\tilde u_\varepsilon(t))(u-u^*)(t)\}\mu(t)dt \end{aligned}$$

by the Mean Value Theorem with $(\tilde\xi_\varepsilon(t),\tilde u_\varepsilon(t)) \in W(t)$, since

$$|\tilde\xi_\varepsilon(t)-x^*(t)|\mu(t) \le \varepsilon|x(t)-x^*(t)|\mu(t) \le c\,\varepsilon\|x-x^*\|^2_{W_2^1(\mathbb{R}_+,\mu)} \le \varepsilon_0$$

and $\tilde u_\varepsilon(t) \in U$. Applying Schwarz inequality we find

$$\begin{aligned} J_\infty(x_\epsilon,u_\varepsilon)-J_\infty(x^*,u^*) &\le \varepsilon\left(\int_0^\infty r_\xi^2(t,\tilde\xi_\varepsilon(t),\tilde u_\varepsilon(t))\mu(t)dt\right)^{\frac12}\|x-x^*\|_{W_2^1(\mathbb{R}_+,\mu)} \\ &\quad + \varepsilon\left(\int_0^\infty r_v^2(t,\tilde\xi_\varepsilon(t),\tilde u_\varepsilon(t))\mu(t)dt\right)^{\frac12}\|u-u^*\|_{L_2^1(\mathbb{R}_+,\mu)} < \infty \end{aligned} \quad (18)$$

due to assumption **V3**. Calculating $\delta^+ J_\infty((x^*,u^*),(\zeta,w))$ we obtain

$$\begin{aligned} \delta^+ J_\infty((x^*,u^*),(\zeta,w)) &= \lim_{\varepsilon\to 0+0}\left\{\frac{J_\infty(x_\varepsilon,u_\epsilon)-J_\infty(x^*,u^*)}{\varepsilon}\right\} \\ &= \int_0^\infty \{r_\xi(t,x^*(t),u^*(t))\zeta(t)+r_v(t,x^*(t),u^*(t))w(t)\}\mu(t)dt \ge 0 \end{aligned} \quad (19)$$

Assumption **V3** is used again to ensure that the limiting passage $\varepsilon \to 0+0$ can be done under the integral sign, since the Lebesque integral converges uniformly with respect to the parameter ε. Because of the convexity of r, assumption **V1**, it follows

$$\int_0^\infty \{r_\xi(t, x^*(t), u^*(t))\zeta(t) + [r(t, x^*(t), u(t)) - r(t, x^*(t), u^*(t))]\}\mu(t)dt$$

$$\geq \delta^+ J_\infty((x^*, u^*), (\zeta, w)) \geq 0. \tag{20}$$

Step 2: Let

$$a_1(u) := \int_0^\infty [r(t, x^*(t), u(t)) - r(t, x^*(t), u^*(t))]\mu(t)\, dt, \tag{21}$$

$$b_1(\zeta) := \int_0^\infty r_\xi(t, x^*(t), u^*(t))\zeta(t)\mu(t)\, dt, \tag{22}$$

$$a_2(w(t)) := -B(t)(w(t)), \quad w(t) = u(t) - u^*(t), \tag{23}$$

$$b_2(\zeta(t)) := \dot{\zeta}(t) - A(t)\zeta(t). \tag{24}$$

Lemma 2. *(1)* $a_1 : L_2(\mathbb{R}^+, \mu) \to \mathbb{R}$ *is convex and weakly lower semicontinuous on* $\mathbf{U}$*, where the set* $\mathbf{U}$ *is defined by*

$$\mathbf{U} := \{u \in L_\infty(\mathbb{R}_+) \,|\, u(t) \in U \text{ a.e. on } \mathbb{R}_+\}. \tag{25}$$

(2) $b_1 : W_2^1(\mathbb{R}^+, \mu) \to \mathbb{R}$ *is linear and continuous.*

Proof. (1) Remark that $\mathbf{U}$ is a convex and bounded subset of $L_2(\mathbb{R}^+, \mu)$, since U is convex and compact and μ is a density function,

$$\|u\|^2_{L_2(\mathbb{R}^+,\mu)} = \int_0^\infty u^2(t)\mu(t)dt \leq \max_{t\in\mathbb{R}_+} |u^2(t)| \int_0^\infty \mu(t)dt \leq c < \infty.$$

Assume $\{u^n\} \rightharpoonup \tilde{u}$ in $L_2(\mathbb{R}^+, \mu)$. Then

$$a_1(u^n) - a_1(\tilde{u}) \geq \int_0^\infty r_v(t, x^*(t), \tilde{u}(t))(u^n(t) - \tilde{u}(t))\mu(t)dt. \tag{26}$$

Since $r_v(\cdot, x^*(\cdot), \tilde{u}(\cdot)) \in L_2(\mathbb{R}^+, \mu)$ due to assumption **V3**, L with

$$L(u^n(t) - \tilde{u}(t)) := \int_0^\infty r_v(t, x^*(t), \tilde{u}(t))(u^n(t) - \tilde{u}(t))\mu(t)dt$$

is a linear and bounded functional in $L_2(\mathbb{R}^+,\mu)$ thus

$$\liminf_{u^n \rightharpoonup \tilde{u}} [a_1(u^n) - a_1(\tilde{u})] \geq 0.$$

(2) The linearity of b_1 is obvious and b_1 is bounded, since

$$|b_1(\zeta)| = \Big| \int_0^\infty r_\xi(t, x^*(t), u^*(t))\zeta(t)\mu(t)\,dt \Big| \leq c\|\zeta\|_{L_2(\mathbb{R}^+,\mu)} \leq c\|\zeta\|_{W_2^1(\mathbb{R}^+,\mu)} \tag{27}$$

due to Schwarz inequality.

Lemma 3. *(1) $a_2 : L_2(\mathbb{R}^+,\mu) \to L_2(\mathbb{R}^+,\mu)$ is linear and continuous on* $\mathbf{W}$*, where the convex set* $\mathbf{W}$ *is*

$$\mathbf{W} := \{w(t) = u(t) - u^*(t) | u \in L_2(\mathbb{R}^+,\mu),\ u(t), u^*(t) \in U\ \}.$$

(2) $b_2 : W_2^1(\mathbb{R}^+,\mu) \to L_2(\mathbb{R}^+,\mu)$ is linear and continuous.

Proof. (1) It is again obvious that a_2 is linear. a_2 is continuous since

$$\|a_2(w)\|^2_{L_2(\mathbb{R}^+,\mu)} \leq \int_0^\infty \Big(\sup_{t\in\mathbb{R}_+} B^2(t) \Big) [u(t) - u^*(t)]^2 \mu(t)\,dt \leq c\|w\|^2_{L_2(\mathbb{R}^+,\mu)}, \tag{28}$$

if assumption **V4** is satisfied.

(2) b_2 is bounded since

$$\|b_2\|^2_{L_2(\mathbb{R}^+,\mu)} = \int_0^\infty \{\dot{\zeta}(t) - A(t)\zeta(t)\}^2 \mu(t)\,dt \leq c\|\zeta\|^2_{W_2^1(\mathbb{R}^+,\mu)} \tag{29}$$

assuming $\sup\limits_{t\in\mathbb{R}_+} A^2(t) < \infty$ according to **V4**.

Now the set of variations M is defined by

$$\begin{aligned} M := \big\{ (z_0, z) \in \mathbb{R} \times L_2(\mathbb{R}^+,\mu) \mid\ & z_0 \geq a_1(u) + b_1(\zeta), \\ & z(t) = a_2(u(t)) + b_2(\zeta(t)), \\ & \zeta \in W_2^1(\mathbb{R}^+,\mu),\ \zeta(0) = 0,\ \|\zeta\| \leq 1 \\ & u \in L_2(\mathbb{R}^+,\mu),\ u(t) \in U \big\}. \end{aligned} \tag{30}$$

It follows immediately from Lemmas 2 and 3 that M is a convex set in $\mathbb{R} \times L_2(\mathbb{R}^+,\mu)$. Its closure in $\mathbb{R} \times L_2(\mathbb{R}^+,\mu)$ is denoted by $\overline{M}$.

Step 3: ($\overline{M}$ dos not coincide with the whole space $\mathbb{R} \times L_2(\mathbb{R}^+,\mu)$)

Lemma 4. *The pair* $(-\alpha, 0) \in \mathbb{R} \times L_2(\mathbb{R}^+, \mu)$ *does not belong to* $\overline{M}$ *for* $\alpha > 0$.

Proof. Let $(\tilde{z}_0, 0) \in \overline{M}$, then there is a sequence $\{(z_0^N, z^N)\}$, $(z_0^N, z^N) \in M$ with:

$$z_0^N \geq a_1(u^N) + b_1(\zeta^N),\ \{z_0^N\} \to \tilde{z}_0 \in \mathbb{R}, \tag{31}$$

$$z^N = a_2(u^N) + b_2(\zeta^N),\ \{z^N\} \to 0 \in L_2(\mathbb{R}^+, \mu). \tag{32}$$

The corresponding sequences

$$\{\zeta^N\},\ \zeta^N \in W_2^1(\mathbb{R}^+, \mu), \zeta^N(0) = 0,\ \|\zeta^N\| \leq 1, \tag{33}$$

$$\{u^N\},\ u^N \in L_2(\mathbb{R}^+, \mu), u^N(t) \in U \tag{34}$$

have the following properties:

A: There is a subsequence of $\{\zeta^N\}$, denoted again by $\{\zeta^N\}$ such that

$$\{\zeta^N\} \rightharpoonup \tilde{\zeta},\ \tilde{\zeta}(0) = 0, \|\tilde{\zeta}\| \leq 1.$$

B: There is a subsequence of $\{u^N\}$, denoted again by $\{u^N\}$, such that

$$\{u^N\} \rightharpoonup \tilde{u} \in L_2(\mathbb{R}^+, \mu),\ \tilde{u}(t) \in U \text{ a.e.}$$

This follows from the theorem of Banach/Alaoglu, in this special case see ([9], p. 84 ff.)

Applying Lemmas 2 and 3, we can pass to the limit in (32) and obtain

$$a_2(\tilde{w}) + b_2(\tilde{\zeta}) = 0, \tilde{\zeta}(0) = 0,\ \tilde{\zeta} \in W_2^1(\mathbb{R}^+, \mu),\ \|\tilde{\zeta}\| \leq 1, \tag{35}$$

$$\tilde{w} := \tilde{u} - u^*, \tilde{u}(t) \in U,\ \tilde{u} \in L_2(\mathbb{R}^+, \mu). \tag{36}$$

We conclude that $(\tilde{x}, \tilde{u})$, with $\tilde{x} := x^* + \tilde{\zeta}$, $\tilde{u} := u^* + \tilde{w}$, is admissible for $(P)_\infty^L$ and therefore we obtain by Lemma 1

$$\lim_{N \to \infty} z_0^N = \tilde{z}_0 \geq a_1(\tilde{u}) + b_1(\tilde{\zeta}) \geq 0.$$

Step 4: We apply the Hahn Banach separation theorem, to separate the closed set $\overline{M}$ and the compact set $\{(-\alpha, 0)\} \in \mathbb{R} \times L_2(\mathbb{R}^+, \mu)$, $\alpha > 0$, which are disjoint sets in $\mathbb{R} \times L_2(\mathbb{R}^+, \mu)$ according to Lemma 4.

We obtain the existence of functions $q_\alpha^* \in \mathbb{R} \times L_2(\mathbb{R}^+, \mu)$ such that for all $(z_0, z) \in M$ the following inequalities hold:

$$\langle q_\alpha^*, (-\alpha, 0) \rangle \leq \gamma_{1,\alpha} < \gamma_{2,\alpha} \leq \langle q_\alpha^*, (z_0, z) \rangle . \tag{37}$$

With $q_\alpha^* = (\lambda_\alpha, y_\alpha)$ inequalities (37) read as follows:

$$-\lambda_\alpha \alpha < \lambda_\alpha z_0 + y_\alpha z \quad \forall (z_0, z) \in M. \tag{38}$$

Of cource $(z_0 = 0, z = 0)$ belongs to M, thus we obtain $\lambda_\alpha > 0$ from (38) for each $\alpha > 0$.

Step 5: (The canonical equation)
Let now $u = u^*$, then we obtain from (38) by division by λ_α and $\tilde{y}_\alpha := \frac{y_\alpha}{\lambda_\alpha}$

$$-\alpha < b_1(\zeta) + \langle \tilde{y}_\alpha, b_2(\zeta) \rangle \quad \forall \zeta \in W_2^1(\mathbb{R}^+, \mu),\ \|\zeta\| \le 1,\ \zeta(0) = 0. \tag{39}$$

For $\alpha = \frac{1}{n}$, $n \in \mathrm{N}$, we conclude because of the linearity of b_1 and b_2 with $\tilde{y}_n := \tilde{y}_\alpha$,

$$|\langle \tilde{y}_n, b_2(\zeta) \rangle| \le c\|\zeta\| \quad \forall \zeta \in W_2^1(\mathbb{R}^+, \mu),\ \zeta(0) = 0. \tag{40}$$

The sequence $\{\tilde{y}_n\}$ is bounded on

$$\mathbf{Z} := \{z \in L_2(\mathbb{R}^+, \mu) | z = b_2(\zeta), \zeta \in W_2^1(\mathbb{R}^+, \mu),\ \zeta(0) = 0\}. \tag{41}$$

We show that $\mathbf{Z}$ is dense in $L_2(\mathbb{R}^+, \mu)$.

Lemma 5. *The set*

$$\mathbf{Z_1} := \{z \in L_2(\mathbb{R}^+, \mu) \cap C(\mathbb{R}^+) | z = b_2(\zeta), \zeta \in W_2^1(\mathbb{R}^+, \mu),\ \zeta(0) = 0\} \tag{42}$$

is dense in $L_2(\mathbb{R}^+, \mu) \cap C(\mathbb{R}^+)$.

Proof. Let $z \in L_2(\mathbb{R}^+, \mu) \cap C(\mathbb{R}^+)$ be arbitrary and $\hat{\zeta} \in W_2^1[0, T]$ be the unique solution of the equation

$$\zeta(t) = \int_0^t \{A(s)\zeta(s) + z(s)\} ds,\ t \le T, \tag{43}$$

see ([7], p.60) with $\hat{\zeta}(T) =: \zeta_T$. We extend this solution continuously to (T, ∞) by the solution of

$$\zeta(t) = \zeta_T + \int_T^t \{A(s)\zeta(s)\} ds. \tag{44}$$

This solution has the representation $\hat{\zeta}(t) = \zeta_T exp\{\int_T^t A(s) ds\}$, $t \ge T$. Due to assumption **V4** it follows

$$\int_T^\infty \hat{\xi}^2(t)\mu(t)dt < \infty. \tag{45}$$

We conclude from (45) that the generated solution $\hat{\xi}$ has the properties $\hat{\xi} \in W_2^1(\mathbb{R}^+, \mu)$, $\hat{\xi}(0) = 0$. With $\hat{z}_T = b_2(\hat{\xi})$ we have

$$\int_0^\infty \big(z(t) - \hat{z}_T(t)\big)^2 \mu(t)dt \le \int_T^\infty z^2(t)\mu(t)dt \tag{46}$$

which tends to 0 for $T \to \infty$. Thus for $z \in L_2(\mathbb{R}^+, \mu) \cap C(\mathbb{R}^+)$ there exists $\hat{\xi} \in W_2^1(\mathbb{R}^+, \mu)$, $\hat{\xi}(0) = 0$, $z_T = b_2(\xi)$ and $\|z - \hat{z}_T\|_{L_2(\mathbb{R}^+,\mu)} < \varepsilon$ for sufficiently large T. Since $L_2(\mathbb{R}^+, \mu) \cap C(\mathbb{R}^+)$ is dense in $L_2(\mathbb{R}^+, \mu)$ the proof is complete.

Using the previous Lemma the sequence $\{\tilde{y}_n\}$ is bounded on $L_2(\mathbb{R}^+, \mu)$ and by the theorem of Banach/Alaoglu it possesses a weak convergent subsequence, again denoted by $\{\tilde{y}_n\}$ converging to $y_0 \in L_2(\mathbb{R}^+, \mu)$. Passing to the limit, we obtain from (39)

$$b_1(\zeta) + \langle y_0, b_2(\zeta)\rangle = 0 \quad \forall \zeta \in W_2^1(\mathbb{R}^+, \mu),\ \zeta(0) = 0. \tag{47}$$

Equation (47) has the following form

$$\begin{aligned} 0 &= \int_0^\infty \{r_\xi(t, x^*(t), u^*(t))\zeta(t) + y_0(t)[\dot{\zeta}(t) - A(t)\zeta(t)]\}\mu(t)dt \\ &= \int_0^\infty \{[y_0(t)\mu(t)]\dot{\zeta}(t) + [r_\xi(t, x^*(t), u^*(t))\mu(t) - y_0(t)\mu(t)A(t)]\zeta(t)\}dt \qquad (48) \\ &\qquad\qquad \forall \zeta \in W_2^1(\mathbb{R}^+, \mu),\ \zeta(0) = 0. \end{aligned}$$

Equation (48) is a variational equation which shows by definition that $y_0\mu$ has a generalized derivative ([14], p. 49), with

$$[y_0(t)\mu(t)]' = [r_\xi(t, x^*(t), u^*(t))\mu(t) - y_0(t)\mu(t)A(t)]. \tag{49}$$

Using the definition of the Pontryagin function we arrive with $y := y_0\mu$ at the canonical equation

$$\dot{y}(t) = -H_\xi(t, x^*(t), u^*(t), y(t), 1) \quad \text{a.e. on } \mathbb{R}^+. \tag{50}$$

Step 6: (Transversality condition) The adjoint function y belongs to $W_2^1(\mathbb{R}^+, \mu^{-1})$ since

$$\int_0^\infty y^2(t)\mu^{-1}(t)dt = \int_0^\infty y_0^2(t)\mu^2(t)\mu^{-1}(t)dt = \int_0^\infty y_0^2(t)\mu(t)dt < \infty, \tag{51}$$

$$\int_0^\infty \dot{y}_0^2(t)(\mu(t))^{-1}dt = \int_0^\infty [r_\xi(t, x^*(t), u^*(t)) - y_0(t)A(t)]^2\mu(t)dt \tag{52}$$

$$\leq 2\Big(\int_0^\infty \{r_\xi^2(t, x^*(t), u^*(t)) + y_0^2(t)A^2(t)\}\mu(t)dt \Big) < \infty$$

Lemma 6. *If $x \in W_2^1(\mathbb{R}^+, \mu)$, $y \in W_2^1(\mathbb{R}^+, \mu^{-1})$ then*

$$\lim_{T\to\infty} y(T)x(T) = 0 \quad \textit{and} \quad \lim_{T\to\infty} y(T) = 0. \tag{53}$$

Proof. Since $x \in W_2^1(\mathbb{R}^+, \mu)$, $y \in W_2^1(\mathbb{R}^+, \mu^{-1})$, we have

$$\int_0^\infty |x(t)y(t)|dt \leq \|x\|_{L_2(\mathbb{R}^+,\mu)}\|y\|_{L_2(\mathbb{R}^+,\mu^{-1})} < \infty, \tag{54}$$

$$\int_0^\infty |\frac{d}{dt}(x(t)y(t))|dt \leq \|\dot{x}\|_{L_2(\mathbb{R}^+,\mu)}\|y\|_{L_2(\mathbb{R}^+,\mu^{-1})} \tag{55}$$

$$+ \|x\|_{L_2(\mathbb{R}^+,\mu)}\|\dot{y}\|_{L_2(\mathbb{R}^+,\mu^{-1})} < \infty$$

Both inequalities imply $xy \in W_1^1(\mathbb{R}^+)$ and $\lim_{T\to\infty} y(T)x(T) = 0$, see [10]. Since μ is a density function, we can apply the last equation with $x = 1$ and obtain $\lim_{T\to\infty} y(T) = 0$.

Step 7: (Maximum condition) Let now $x = x^*$ then we obtain from (38)

$$-\frac{1}{n} < a_1(u) + \langle \tilde{y}_n, a_2(u)\rangle \qquad \forall u \in \mathbf{U},\ n \in \mathrm{N}. \tag{56}$$

Passing to the limit and using Lemmas 2 and 3 we obtain

$$0 \leq a_1(u) + \langle \tilde{y}_0, a_2(u)\rangle \qquad \forall u \in \mathbf{U}. \tag{57}$$

The last inequality is the integrated Maximum condition,

$$\int_0^\infty H(t, x^*(t), u^*(t), y(t), \lambda_0)dt \geq \int_0^\infty H(t, x^*(t), u(t), y(t), \lambda_0)dt \quad \forall u \in \mathbf{U}. \quad (58)$$

Finally the pointwise condition (M) follows from standard techniques, see [7] and the proof is complete. □

6 Conclusions

For the class of problems considered we obtained a maximum principle in normal form ($\lambda_0 = 1$) including transversality conditions. Additionally, we proved that the adjoint variable y belongs to a weighted Sobolev space too, $y \in W_2^1(\mathbb{R}^+, \mu^{-1})$. Using a similar scheme of proof the result can be extended to more general problem settings, i.e. to problems with vector valued states and controls and state equations which are linear with respect to the control only. The Hilbert space approach opens the door for spectral methods and Ritz type approximations of the problem.

References

1. Aseev, S.M., Kryazhimskii, A.V., Tarasyev, A.M.: The pontryagin maximum principle and transversality conditions for a class of optimal control problems with infinite time horizons. Proc. Steklov Inst. Math. **233**, 64–80 (2001)
2. Aseev, S.M., Veliov, V.M.: Maximum principle for infinite-horizon optimal control problems with dominating discount. DCDIS: Dyn. Contin. Discret. Impuls. Syst. Ser. B: Appl. Algorithms **19**(1–2), 43–63 (2012)
3. Carlson, D.A., Haurie, A.B., Leizarowitz, A.: Infinite Horizon Optimal Control. Springer, New York/Berlin/Heidelberg (1991)
4. Dunford, N., Schwartz, J.T.: Linear Operators. Part I: General Theory. Wiley-Interscience, New York, etc. (1988)
5. Elstrodt, J.: Maß und Integrationstheorie. Springer, Berlin (1996)
6. Halkin, H.: Necessary conditions for optimal control problems with infinite horizons. Econometrica **42**, 267–272 (1979)
7. Ioffe, A.D., Tichomirow, V.M.: Theorie der Extremalaufgaben. VEB Deutscher Verlag der Wissenschaften, Berlin (1979)
8. Kufner, A.: Weighted Sobolev Spaces. Wiley, Chichester, etc. (1985)
9. Lykina, V.: Beiträge zur Theorie der Optimalsteuerungsprobleme mit unendlichem Zeithorizont. Dissertation. BTU Cottbus (2010)
10. Magill, M.J.P.: Pricing infinite horizon programs. J. Math. Anal. Appl. **88**, 398–421 (1982)
11. Pickenhain, S.: On adequate transversality conditions for infinite horizon optimal control problems – a famous example of Halkin. In: Crespo Cuaresma, J., Palokangas, T., Tarasyev, A. (eds.) Dynamic Systems, Economic Growth, and the Environment. Dynamic Modeling and Econometrics in Economics and Finance, vol. 12, pp. 3–22. Springer, Berlin etc. (2010).

12. Ramsey, F.P.: A mathematical theory of savings. Econ. J. **152**(38), 543–559 (1928)
13. Sethi, S.P., Thompson, G.L.: Optimal Control Theory. Applications to Management Science and Economics, 2nd edn. Kluwer, Boston/Dordrecht/London (1985)
14. Yosida, K.: Functional Analysis. Springer, New York (1974)

Optimum Operation of a Beer Filtration Process

Cesar de Prada, Smaranda Cristea, Rogelio Mazaeda, and Luis G. Palacín

Abstract This paper deals with the optimum operation of a beer filtration process that uses membranes for this task. Due to fouling, the operation requires cleaning, which damages the membranes, and creates a discontinuous operation. The optimal economic operation can be defined in terms of minimizing the number of chemical cleanings, as well as the use of energy, when processing a certain amount of beer in a given time. The problem is hybrid in nature, due to the discontinuities created by the cleanings. The corresponding optimization problem is formulated in the framework of predictive control but integrating the economic operation as target of the controller and different time scales. Also, instead of using binary variables for representing the discontinuities, the problem employs a sequential approach, embedding them in the dynamic simulation of the process model combined with a control parameterization that allows computing the solution in terms of the continuous variables that represent its degrees of freedom. Results of the optimal operation are presented.

1 Introduction

In breweries, beer is filtered in order to remove different types of impurities helping obtaining in this way the desired color and taste. Today, this operation is performed in many factories using special membranes that allow pass the beer and retain the impurities. The process follows the schematic displayed in Fig. 1: from a pressurized tank of filtered beer, this is pumped to a set of membranes where the separation takes place. One fraction goes through the membrane as permeate while the rest, the retentate, returns to the recycled beer tank, circulating again in closed loop until all beer has been processed.

The membranes considered in this paper are of hollow fiber type, like the ones in Fig. 2: a bunch of fibers each one being a long pipe with microscopic holes in its

C. de Prada (✉) • S. Cristea • R. Mazaeda • L.G. Palacín
Department of Systems Engineering and Automatic Control, University of Valladolid, Valladolid, Spain
e-mail: prada@autom.uva.es; smaranda@autom.uva.es; rogelio@cta.uva.es; palacin@cta.uva.es

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_15

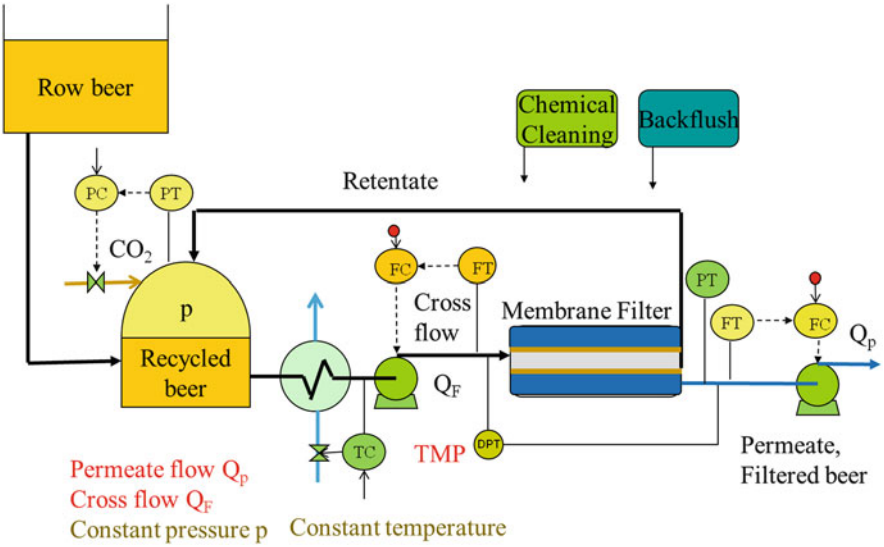

Fig. 1 Schematic of the basic control system of the beer filtration process

surface that allow permeating the beer but not the impurities. The beer flows along the fibers flowing out partially as permeate, while the non-filtered portion leaves the fiber by the other end. From the point of view of the beer filtration, these are of two kinds: those that do not cross the membrane and form a cake layer in the inside, acting as an additional filter, and those that get trapped in the membrane pores obstructing them partially.

The operation of the plant is usually performed using the control system represented in Fig. 1. A pressure control loop maintains the pressure in the recycled beer tank injecting CO_2 gas to avoid foam formation, while two flow control loops fix the values of permeate and cross flow and another loop helps maintaining the recycled beer temperature using a heat exchanger. In addition, there are measurements of the permeate side pressure and the trans-membrane pressure. The cake layer grows as time passes due to new deposits of impurities and, as a result, the trans-membrane pressure (TMP) that is required to maintain the flow of filtered beer increases. When this pressure has risen up to a certain value (TMP_{max}), the filter needs to be cleaned to restore normal operation, which is done with the so called backflush cleanings (BF). These remove the cake layer, but not all the impurities attached to the membrane pores, so that, when the process is re-started again, the initial trans-membrane pressure required to maintain the permeate flow is higher than in the previous cycle, and the time required to reach the TMP_{max} decreases, as can be seen in Fig. 3. After several cycles, this time t is too short and a deeper cleaning is needed, the so called chemical cleaning (*CIP*) that restores the membrane to its original state, but producing damage so shortening its life

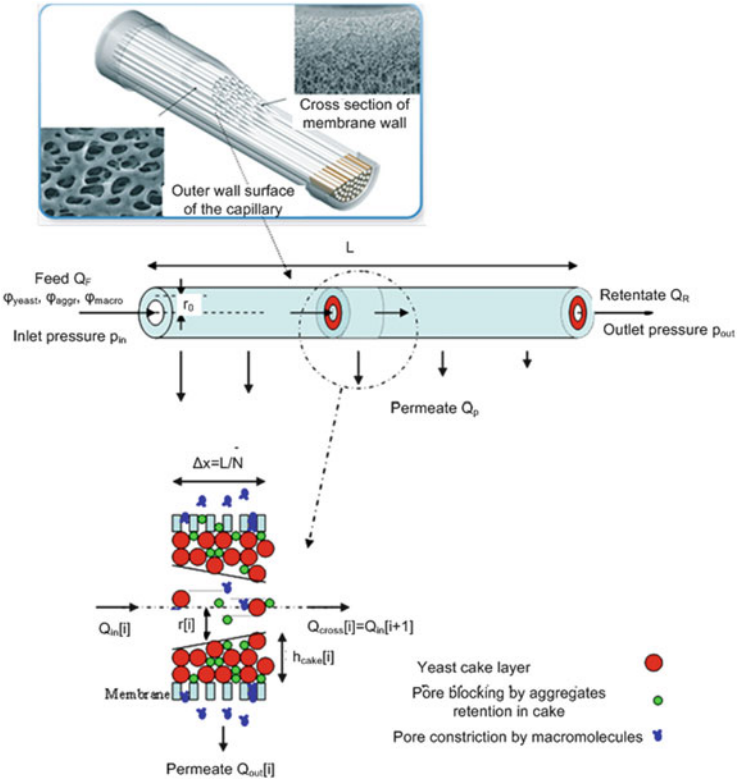

Fig. 2 Schematic of a membrane, fiber and a section with the internal cake layer and different types of impurities

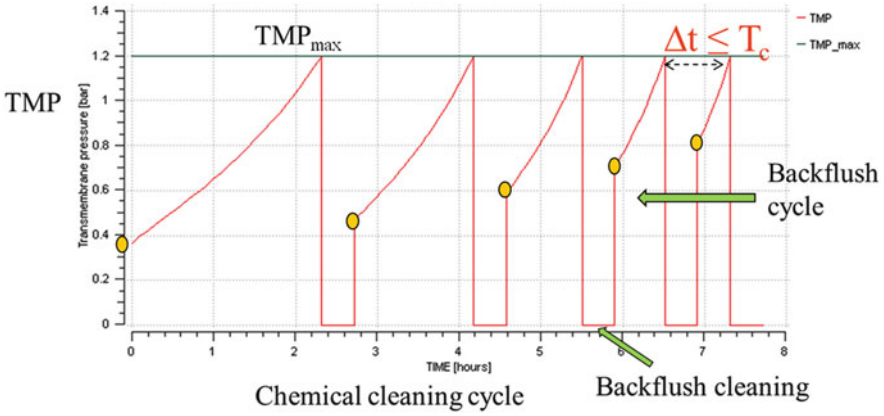

Fig. 3 Time evolution of the trans-membrane pressure between two chemical cleanings, including several backflush cleanings

The usual target is to filter a certain amount of beer to be delivered to the clients by a specified time. The beer to be filtered is stored in a raw beer tank and is supplied to the recycled beer tank to be processed. Besides fulfilling this target, there is a clear interest in performing the task in the best possible way, both from economic and process points of view. The economics involve the energy pumping costs and the cleaning costs, while the process operation requires fulfilling a set of constraints and enlarge the membrane life, minimizing the number of chemical cleanings needed to process a certain amount of beer, because membranes are quite expensive.

The optimal plant operation involves decisions at different levels, in particular, the following values have to be chosen: the number of chemical cleanings (*CIPs*), the number of backflushes per *CIP*, the value of the cross and permeate flow set points (Q_F, Q_P) and the maximum trans-membrane pressure (TMP_{max}) between backflushes that allow processing the specified amount of beer by the required final time. Notice that besides different levels, there are also discrete and continuous variables implicated. The natural approach is, then, to formulate a set of hierarchical optimization problems like in [3], where a three layer optimization is proposed, with the two upper ones solving MIP problems to schedule production, providing the number of cleaning cycles, and the one below them and above the control layer using NLP optimization to determine the best TMP_{max} and flows. Nevertheless, this approach involves a lot of computation and it is not suitable for real-time application.

This paper presents an alternative that merges economic optimization and control and uses especial parameterizations to solve the problem using a small number of NLP problems in a single layer. This provides an efficient way of solving the problem and shows a way of dealing with mix-integer dynamic optimization problems that can be applied in other contexts as well. The paper is organized as follows: after this introduction, the optimization problem is formulated in Sect. 2, then, Sect. 3 shows results obtained with a particular beer assignment and gives implementation directions. The paper ends with some conclusions and references.

2 Optimizing Control

The main elements of the proposed approach are synthesized next. The first one is integration of economic optimization and dynamic control in the framework of Model Predictive Control (MPC), what is called optimizing control, [4, 6, 8]. The approach takes decisions about the manipulated process variables at a certain time instant according to the solution of a dynamic optimization problem, where the cost function is directly the economic aim one tries to optimize. The solution is applied to the process and the next sampling time the problem is repeated, updated with the new information collected from the process. The second element recognizes the fact that the behavior of the process between two chemical cleanings (*CIP*) is uniform and that not all the decision variables mentioned in the different time scales involved are fully independent. In fact, given the cross and permeate flow and a TMP_{max},

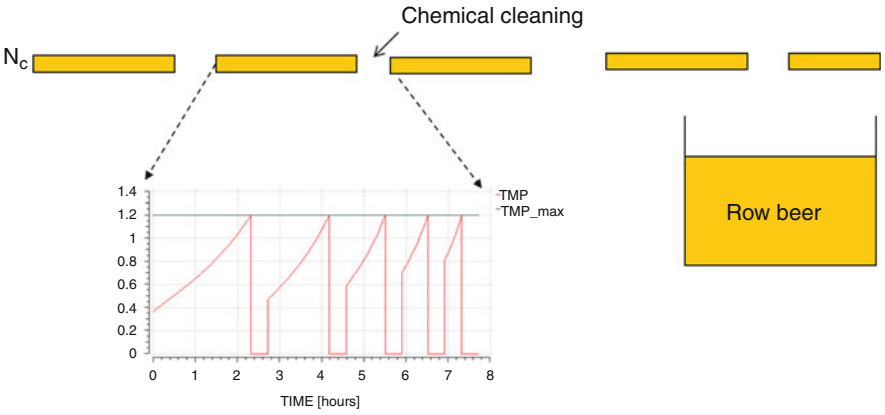

Fig. 4 Sequence of N_C chemical cleaning cycles and detail of one of them

and assuming that a minimum time between backflushes, T_c, has been fixed, the number of backflushes between two chemical cleanings and the total duration of the *CIP*, t_C, can be determined, and, hence, the number of *CIP*s, as some global constraints concerning the total amount of beer to be processed must be satisfied. In particular, with reference to Fig. 4, these constraints are:

$$\begin{aligned} N_C \int_0^{t_C} Q_p dt &= V_T \\ N_C t_C &\leq t_T \end{aligned} \tag{1}$$

Where N_C is the number of *CIP*s, t_C is the elapsed time of a chemical cycle, V_T is total amount of filtered beer to be processed, which is known, Q_p the permeate flow and t_T the assigned time for filtering. Notice that N_C from (1) can be a real number, as the last *CIP* cycle can finish incomplete. So, the global problem can be reduced to the optimization of a chemical cleaning cycle, assuming of all them uniform, provided that the constraints (1) are explicitly considered in the optimization. In this way, the large time scale covered by the processing of the beer assignment can be shortened and the corresponding time scale integrated in a shorter time scale problem.

This problem is displayed in Fig. 5. The cost function to maximize is the global one, covering the total number of *CIP* cycles, but the decision variables and the dynamic model are reduced to the ones of a chemical cycle. In particular, the cost function J_T is formulated as the economic target:

$$J_T = N_C \begin{bmatrix} \alpha_0 \int_0^{t_C} Q_p dt - \alpha_1 \int_0^{t_C} p Q_p dt \\ -\alpha_2 \int_0^{t_C} p_F Q_F dt - \alpha_3 \int_0^{t_C} Q_F dt - \alpha_4 N_B - \alpha_5 \end{bmatrix} \tag{2}$$

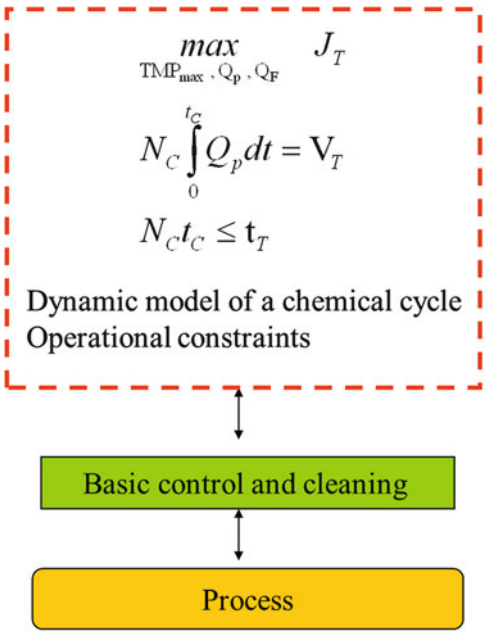

Fig. 5 The optimizing control problem and its implementation

Which includes several terms with α_i representing prices: α_0 price of beer, α_1 price of energy in permeate pump, α_2 price of energy in cross flow pump, α_3 price of cross flow cooling, α_4 cost of a backflush, α_5 cost of a chemical cleaning and N_C the number of *CIPs*. Notice that the first term tries to maximize the filtered beer per *CIP*, contributing indirectly to maximizing the membrane life which expands for a certain number of *CIPs*.

The constraints are the ones that apply to a *CIP*, with the addition of the global ones (1), and include ranges for the process variables and other operational limitations. Concerning the decision variables, the two flows Q_F and Q_P, cross flow and permeate, as well as the maximum trans-membrane pressure to fire a backflush cleaning, TMP_{max}, were chosen. They can be given different parameterizations, compromising between increasing the degrees of freedom of the problem and increasing the computation time.

The third element deals with the way the optimization is performed considering the discontinuities over a *CIP*. The embedded logic approach [8] has been used, which follows a sequential method for solving the dynamic optimization problem, including the discontinuous operation as part of the model simulation according to Fig. 6.

The dynamic model follows [9] and comprises equations for hydraulic flows, growing cake, membrane fouling, etc. It has been implemented in the simulation environment EcosimPro [1], an object oriented language implementing a good treatment of discontinuities and facilities for optimization according to a sequential approach. The integration of the model starts from the current point and implements

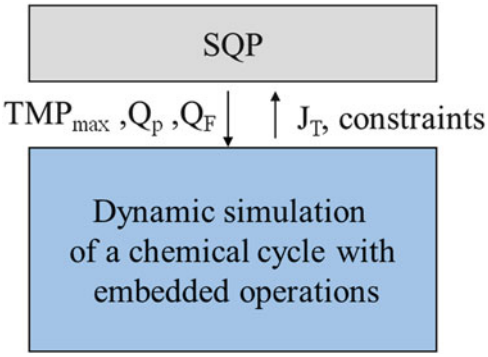

Fig. 6 Logic embedded approach to discontinuous dynamic optimization

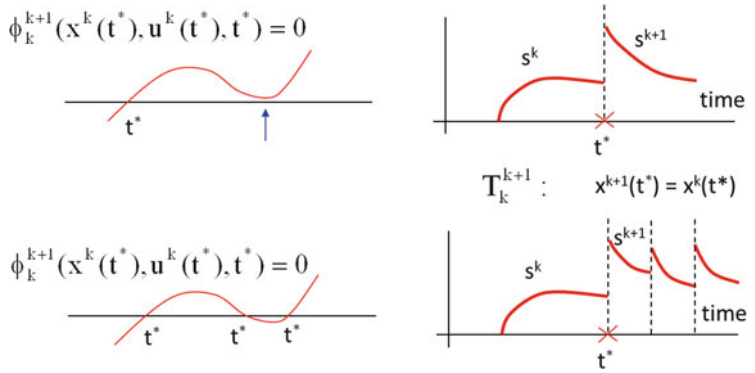

Fig. 7 Transition function and sensibilities with state discontinuities

a backflush cleaning each time the *TMP* reaches the value TMP_{max}, starting again the filtration from the new conditions of the membrane pores until the elapsed time between two backflushes is less or equal to T_C, in which case a chemical cleaning takes place and the simulation ends. So, N_C and J_T are evaluated, as well as the constraints and their values are sent back to the optimization algorithm. As no explicit integer variables are used, in principle, an NLP method such as SQP (Sequential Quadratic Programming), could be used.

Nevertheless, notice that the discontinuities involved are not input discontinuities but state ones and this may create problems in the computation of the cost function gradient in the optimization. This type of discontinuities are fired when a certain transition function $\phi(x, u)$ in the model changes sign, according to the time evolution of the model states x, as in the upper part of Fig. 7, where a change from model k to model $k + 1$ at time instant t^* is represented besides the corresponding sensibilities s. Of course, the sensibilities, defined as the derivatives of the states with respect to the decision variables, may jump at t^*, but, in principle, this jump does not means that the gradient of the cost function is discontinuous.

The problem may appear if the situation represented in the bottom of Fig. 7, takes place, because here, a small change in the decision variable, changes the number of discontinuities, impeding the right computation of the gradients. This is exactly the situation with our problem, in which a small change in the value of TMP_{max} may change by one the number of backflushes, that is of discontinuities of a *CIP*.

To avoid this problem, an alternative was used in the paper, based in recognizing that the integer number of possible backflushes typically doesn't change much in practice. It consist in solving in parallel a number of similar problems, but, each one with a fixed and different number of backflushes NB, and then select the one with the best cost function J_T. As N_B is fixed, the above mentioned problem disappears and an NLP method can be used, which speeds up the computations. Of course, the price to pay is repeating the NLP problem a small number of times (3–6 being adequate) but it compensates as there is no need to formulate the problem as a mixed-integer optimization one. In order to impose a fix number of backflushes in the operation, notice that two new constraints must be added in order to obtain realistic solutions:

$$\begin{aligned} \Delta t_{N_B} &\leq T_c \\ \Delta t_{N_B-1} &> T_c \end{aligned} \tag{3}$$

That corresponds with forcing the last backflush to fulfill the condition of a chemical cleaning and the previous one not to fulfilling it.

3 Results and Implementation

The proposed approach has been tested with a realistic simulation of the beer filtration process, coupled to an optimizing controller as the one described in section two. The numbers correspond to a small pilot plant and are not representative of an industrial installation, but describe quite well the possibilities of the method. The assignment was to process a volume of $V_T = 0.051\,\mathrm{m}^3$ in less than 5 days. The process constraints were given by:

$$\begin{aligned} 0.008 \leq Q_F \leq 0.021 \quad & \mathrm{m}^3/h \\ 0.6 \leq TMP_{max} \leq 1.9 \quad & \text{bars} \\ 0.0002 \leq Q_P \leq 0.0007 \quad & \mathrm{m}^3/\mathrm{h} \end{aligned} \tag{4}$$

In Fig. 8, results of three parallel dynamic optimizations, each one identical, but with different values of N_B, taken as 5, 6 and 7, are given. In this case, the associated cost function J_T to the case N_B equal to 6 is the best one, so that this value was selected. The optimal solution corresponds to a benefit of 34.9317 Euros, and the assignment was performed in less than 5 days, with 9.66 *CIPS*, which means 9 full cycles and a partial one at the end. The optimal decision variables were:

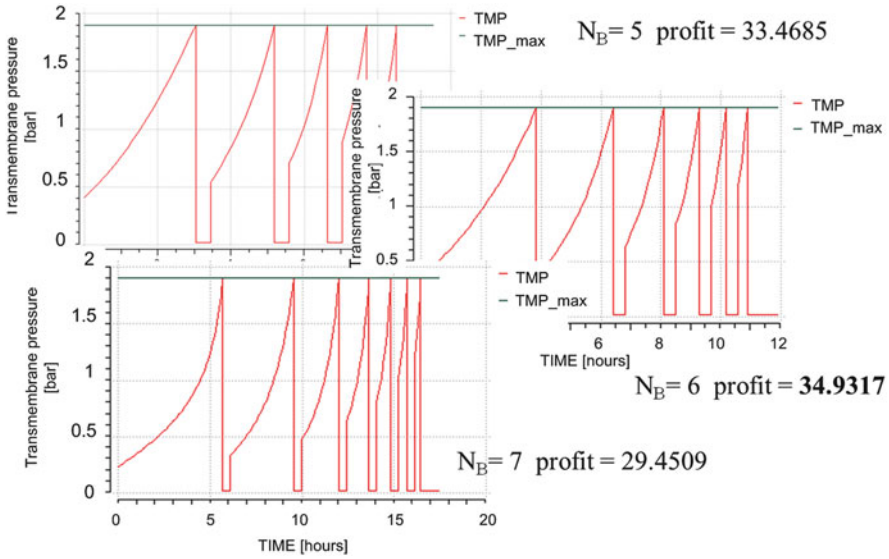

Fig. 8 Optimal solution of three parallel cases with NB selected as 5, 6 and 7 backflushes

Regarding implementation, it is worth mentioning that the differential algebraic system of equations (DAE), resulting after discretization along the fiber axial direction of the original partial differential equations (PDE) first-principle model, reaches a substantial size: with 16 discretizations, it presents a number of 647 equations in total, with 68 dynamic states.

The required numerical optimization to obtain the best possible working regime for the installation is of a dynamical nature. In order to be able to apply the standard NLP techniques, the so called direct sequential approach [2] is adopted in this case. The input to the plant, or control vector, is parameterized, and then, the resulting initial value problem (IVP) is solved, meaning that the dynamic equations of the model are numerically integrated for a duration corresponding to a complete *CIP* cycle. The modelling and numerical integration has been implemented by means of the EcosimPro tool. The specific integration software used was DASSL [7], while the numerical programming has been performed by an SQP-type algorithm as implemented in SNOPT [5].

The obtained computation time of around 10 min, using a 1.83 GHz standard PC with 1 GB of RAM, it is not negligible. It can be attributed to the combination of already mentioned factors, in particular the size of the model and the control vector parameterization strategy adopted. In relation with the latter, it should be emphasized that the model has to be integrated for a whole *CIP* cycle horizon at every step of the optimization algorithm. Furthermore, this integration is performed, not once, but several times for each SQP iteration, in order to automatically obtain

Table 1 Decision variables values

Variable	Value	Units
N_B	6	
Q_F	0.008	m^3
TMP_{max}	1.9	bars
Q_p	0.00048444	m^3/h
Chemical cleaning cycle	11.8916	h
Computation time	623	s

information concerning the gradients using discrete perturbations for each input. In any case, although there is room for improvement, the control processing time already obtained, opens the door to real-time implementation given the time scales of the process.

Further work is needed to include several elements that are required in an industrial plant. Among them, it is necessary to update the model with on-line measurements from the plant in order to tune it according to the different beer types and membrane quality that one can find over the process operation. At the same time, closed loop operation requires the incorporation in the problem formulation of the shrinking time horizon that appears in real-time operation as time passes. In relation with this, it should be said that the values for the three input degrees of freedom appearing in Table 1, are the optimal constant values which were obtained for the entire exercise in an open loop fashion. The implementation of a closed loop receding horizon scheme, would, on the other hand, immediately provide, at each sampling instant, new values for the same control inputs, that would be optimal at the new conditions, taking into consideration the presence of disturbances.

4 Conclusions

A novel approach has been presented to reformulate an optimization problem that combines different time scales and decision levels as well as continuous and discrete decisions. Furthermore, the model presents a variable structure in the sense that it has several discontinuities whose existence cannot be known beforehand since they depend on the dynamical evolution of the state. The approach is based on the idea of optimizing control, mixing economic optimization and dynamic control, as well as in the use of embedded logic optimization with a sequential approach to dynamic optimization. Finally, the discontinuity analysis has led to incorporate parallel computation to solve the hybrid problem in an efficient way. The method has been tested in a simulated process giving promising results.

References

1. Agrupados, E.: EcosimPro, User Manual. www.ecosimpro.com (2010)
2. Biegler, L.T., Grossmann, I.E.: Retrospective on optimization. Comput. Chem. Eng. **28**, 1169–1192 (2004)
3. Blankert, B.: Short to medium term optimization of dead-end ultra-filtration. PhD thesis, University of Groningen, The Netherlands (2007)
4. Engell, S.: International scientific & technical encyclopedia (iste). J. Process Control **17**, 203–219 (2007)
5. Gill, P.E., Murray, W., Saunders, M.A.: Software for large-scale nonlinear programming. Department of Management Science and Engineering, Stanford University, Stanford (2008)
6. González, A.I., Zamarreño, J.M., Prada, C.: Controlador predictivo con significación económica: aplicaciones simuladas a procesos batch. Ingeniería Electrónica, Automática y Comunicaciones **XXIII**, 26–32 (2002)
7. Petzold, L.: A Description of DASSL: A Differential/Algebraic System Solver. Technical report 82-8637. Sandia National Laboratories, Livermore, CA, USA (1982)
8. Rodríguez, M., Sarabia, D., Prada, C.: Hybrid predictive control of a simulated chemical plant. In: Taming Heterogeneity and Complexity of Embedded Control, pp. 617–634. ISTE, London/Newport Beach (2007)
9. Van der Sman, R.G.M.: Simulation of confined suspension flow at multiple scale lengths. Soft Matter **22**, 4376–4387 (2009)

Energy-Aware Lease Scheduling in Virtualized Data Centers

Nguyen Quang-Hung, Nam Thoai, Nguyen Thanh Son, and Duy-Khanh Le

Abstract Energy efficiency has become an important measurement of scheduling algorithms in virtualized data centers. One of the challenges of energy-efficient scheduling algorithms, however, is the trade-off between minimizing energy consumption and satisfying quality of service (e.g. performance, resource availability on time for reservation requests). We consider resource needs in the context of virtualized data centers of a private cloud system, which provides resource leases in terms of virtual machines (VMs) for user applications. In this paper, we propose heuristics for scheduling VMs that address the above challenge. On performance evaluation, simulated results have shown a significant reduction on total energy consumption of our proposed algorithms compared with an existing First-Come-First-Serve (FCFS) scheduling algorithm with the same fulfillment of performance requirements. We also discuss the improvement of energy saving when additionally using migration policies to the above mentioned algorithms.

1 Introduction

Cloud computing [4] has been developed as a utility computing model and is driven by economies of scale. Reduction in energy consumption (kWh) for cloud systems, which are built up from virtualized data centers [3, 11], is of high concern for any cloud provider. Energy-aware scheduling of VMs in virtualized data centers is still challenging [1, 3, 7, 10]. There are several works that have been proposed to address the problem of energy-efficient scheduling of VMs in cloud data centers. Some works [1, 10] proposed scheduling algorithms to change adaptatively processor speed when executing user applications such that the changing processor speed

N. Quang-Hung (✉) • N. Thoai • N. Thanh Son
Faculty of Computer Science and Engineering, HCMC University of Technology, VNUHCM, 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
e-mail: hungnq2@cse.hcmut.edu.vn; hungnq2@gmail.com; nam@cse.hcmut.edu.vn; sonsys@cse.hcmut.edu.vn

D.-K. Le
Department of Computer Science, National University of Singapore, Singapore, Singapore
e-mail: leduykha@comp.nus.edu.sg

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_16

method meets user requirements and reduces power consumption of processors when executing user applications. Some other works proposed algorithms that consolidate VMs onto a small set of physical servers in a virtualized datacenter [3, 7] such that power consumption of physical servers is minimized. However, the challenge of reducing energy consumption while preserving quality of service (e.g. performance or resource availability on time for reservation request) remains.

Sotomayor et al. [11, 12] have proposed a lease-based model for resource provisioning problems and presented FCFS-based scheduling algorithms to meet user performance. The presented scheduling algorithms in that works, however, have never involved energy efficiency. In this paper, we introduce an energy-aware lease scheduling problem with trade-off between minimizing of energy consumption and satisfying quality of service. We concern on the provision of hardware resources. The software requirements on provisioning resource are out of scope of this paper. Using VMs incurs some overheads (e.g. transferring VM images); therefore, these overheads of VMs should be considered in the problem of scheduling VM-based leases. The resource allocation problem of VMs with multiple resources is NP-hard. Each VM requires multiple resources such as CPU, memory, I/O to execute its applications. The resource allocation problem can be seen as a d-dimensional Vector Bin Packing problem (VBP_d) [8], in which each physical server with multiple resources is considered as a d-dimensional bin, and each virtual machine is a d-dimensional item with various sizes of requested resources (e.g. CPU, memory). The VBP_d is claimed as NP-hard problem for $\forall d \geq 1$ [8].

In recent research, Fan et al. [5] claimed a linear relationship between power consumption (in Watts) on a physical server and its load (i.e., CPU utilization). The authors estimate that the power consumption of an idle (0 % CPU utilization) server is equal or greater than 50 % of the power consumption of the server at full load (100 % CPU utilization). Barroso and Hölzle [2] have proposed a case of energy-proportional computing where all components in a computer could be turned on/off on demand. In this paper, we propose an energy-aware scheduling algorithm to map user lease requests onto physical servers. The objective of our scheduling algorithm is to find an optimal schedule that has a minimum number of active physical servers and finishes all user lease requests while satisfying user lease requirements. Our scheduling algorithm includes two phases: power-aware VM allocation and re-scheduling. Our proposed allocation algorithm uses the minimum number of physical servers on mapping of the ready leases (in scheduler's queue). We also solve a re-scheduling problem by suspending, migrating, and resuming leases from physical servers that have CPU utilization lower than a pre-defined low-threshold. These low load physical servers could be put into energy saving modes (e.g. stand-by, suspend to disk, or turn idle nodes off) to avoid unwanted power consumption (e.g. 50 %) in idle nodes [3].

The remainder of the paper is organized as follows. In Sect. 2, we discuss the works that are related to our approach and energy-aware scheduling of virtual machines in virtualized data centers. We present the lease scheduling problem and the proposed energy-aware scheduling and migration algorithms in Sect. 3. The

results of our simulation study are reported and discussed in Sect. 4. The last section gives conclusions and future work.

2 Related Works

Sotomayor et al. [11, 12] proposed a lease-based model and implemented First-Come-First-Serve (FCFS) [6] and back-filling [6] algorithms to schedule best effort, immediate and advanced reservation leases. The FCFS and back-filling algorithms consider only one performance metric such as waiting time and slowdown, without mentioning energy efficiency. To maximize performance, these scheduling algorithms tend to choose free load servers (i.e. those with the highest-ranking scores) when allocating a new lease. Therefore, a lease with just a single VM can be allocated on a big, multi-core physical server. This could waste a lot of energy. The authors also proposed a migration algorithm for preempting a best-effort lease in case the scheduler needs more resources for an advanced reservation lease. However, the authors did not use the migration algorithm on dynamic consolidation of VMs to turn low utilization servers off for energy saving. Instead, our allocations will choose working physical servers and turn off other free load servers. We also improve the migration algorithm to allow migration of leases that are running on low utilization servers, and turn these servers off.

Albers et al. [1] reviewed some energy-efficient algorithms which are used to minimize flow time by changing processor speed according to job size. Laszewski et al. [10] proposed scheduling heuristics and presented application experience for reducing power consumption of parallel tasks in a cluster with the Dynamic Voltage Frequency Scaling (DVFS) technique. We did not use the DVFS technique to reduce energy consumption on data centers.

Previous research [3, 7] presented scheduling algorithms that place virtual machines (VMs) in virtualized data centers to minimize energy consumption. Beloglazov et al. [3] presented a modified best-fit decreasing (denoted as MBFD) heuristic for placement of VMs and VM migration policies under adaptive thresholds in virtualized data centers. The MBFD sorts all VMs in a decreasing order of CPU demands and tends to allocate a VM to an active physical server that would take the minimum increase of power consumption. The MBFD can reduce energy consumption in a heterogeneous environment. On the other hand, choosing a host with least increasing power consumption can lead to performance inefficiency. The MBFD will prefer a lower-performance host rather than a higher-performance host if each processor in the lower-performance host consumes less power than each processor in the higher-performance host does. The MBFD is also not concerned about the duration time of VMs. In contrast, our proposed allocation algorithms account for the duration time of VMs and will greedily allocate VMs belonging to a lease to the same physical machine. The previous migration policies [3] did not concern on overheads of migration (e.g. suspend, resume, and migration time) of VMs. We study effects of the overheads of migration of VMs on a schedule plan.

An optimum allocation of each independent VM is studied in [7]. In the paper, the authors developed a score-based allocation method to calculate the scores matrix of allocations of m VMs to n physical servers. A score is the sum of many factors such as power consumption, hardware and software fulfillment, resource requirement. These studies are unsuitable for the following lease scheduling in this paper. We consider the case where each user lease has a limited duration time and contains a group of concurrent VMs (e.g. each MPI job requires tens to thousands of VMs concurrently).

3 Problem Description

Given a set of leases L_i (i ∈ [1;n]) to be scheduled on a set of physical servers M_j (j ∈ [1;m]). We extend the resource model that is defined in [11]. A user requests some leases. A user i^{th} lease requests (1) a set of rn_i identical virtual machines (VMs), (2) start time (st_i), and (3) duration of the lease (dur_i). In the user i^{th} lease, each k^{th} VM requires u_{ik} percent of CPU utilization (e.g. each 100 % is one core), r_{ik} MB of memory, d_{ik} MB of disk image, and b_{ik} MB/s of network bandwidth. A lease can be a best-effort or an advanced reservation lease that is without or with user specified start time. Each physical server has total U percent of CPU utilization, R megabytes (MB) of memory, D MB of available file system, Bw MB/s of network bandwidth.

In this paper, we use the following energy consumption model proposed in [3,5]:

$$P_j = P_{idle} + (P_{max} - P_{idle}) \times CPU_j \tag{1}$$

where P_{idle}, P_{max}, and P_j are idle power, maximum power, and total system power of a single physical server (M_j), and CPU_j is the server's CPU utilization where $0 \leq CPU_j \leq 1$.

The objective is to find an optimal schedule that maps all user lease requests into the smallest number of physical servers in order to minimize total energy consumption of all activated physical machines and to satisfy QoS (e.g. performance, or resource is available on time for advanced reservation leases [11]). Formally, we formulate the static VM allocation problem as following:

$$\textbf{Minimize} \sum_{j=1}^{m} (P_{idle} + (P_{max} - P_{idle}) \times CPU_j) \times y_j$$

subject to

$$\sum_{i=1}^{n} \sum_{k=1}^{rn_i} u_{ik} x_{ikj} \leq U_j \times y_j, \qquad j = 1, \ldots, m \tag{2}$$

$$\sum_{i=1}^{n} \sum_{k=1}^{rn_i} r_{ik} x_{ikj} \leq R_j \times y_j, \qquad j = 1, \ldots, m \tag{3}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{rn_i} b_{ik}x_{ikj} \le Bw_j \times y_j, \qquad j = 1,\ldots,m \tag{4}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{rn_i} d_{ik}x_{ikj} \le D_j \times y_j, \qquad j = 1,\ldots,m \tag{5}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ikj} = 1, \qquad k = 1,\ldots,rn_i \tag{6}$$

$$CPU_j = \frac{\sum_{i=1}^{n}\sum_{k=1}^{rn_i} u_{ik}x_{ikj}}{U} \qquad j = 1,\ldots,m \tag{7}$$

where the binary variables $x_{ikj} \in \{0,1\}$ and $y_j \in \{0,1\}$. $x_{ikj} = 1$ if and only if the k^{th} VM of the lease L_i is allocated on the server M_j, and $y_j = 1$ if and only if the server M_j is allocating resources for at least one VM and $y_j = 0$ if and only if the server M_j is in a sleep state. (That is we assume that a server in sleep state does not consume energy). Equations (2)–(5) are constraints on resources of each physical server, Eq. (6) describes the fact that each VM will be allocated on only one physical machine. The CPU utilization of a physical machine is calculated by Eq. (7). We assume that the CPU utilization is unchanged during an interval of two continuous events of the scheduler. The energy consumption (E_j) of a physical machine formulates as:

$$E_j = \int_0^T P_j(t)dt \tag{8}$$

The makespan of a schedule (C_{max}), is defined as the maximum of the completion time of all leases and formulated as: $C_{max} = max\{C(L_i)|i = 1,\ldots,n\}$, where the $C(L_i)$ is completion time of a lease L_i. The $C(L_i)$ formulated as $C(L_i) = (st_i + dur_i + t_i^{mig} + t_i^{sus} + t_i^{trans})$, where st_i, dur_i, t_i^{mig}, t_i^{sus}, t_i^{trans} are start time, duration time, migration time, suspend time, and transferring time of image-disks of some VMs of the lease respectively.

3.1 A Special Case

Given a set of leases L_i (i ∈ [1;n]) to be scheduled on a set of identical physical servers M_j (j ∈ [1;m]). Let us assume that all user leases request only one VM. We formulate the special lease scheduling with a single-VM problem as following:

$$\textbf{Minimize} \sum_{j=1}^{m} E_0 \times T_j + \sum_{i=1}^{n} e_i$$

where E_0 is the base energy consumption of the physical server in a unit of time, T_j is the working time of the physical server M_j (j ∈ [1;m]), e_i is the energy consumption for executing the user lease L_i (i ∈ [1;n]).

3.2 Scheduling Algorithm

Our lease scheduling problem is on-line scheduling. The scheduling algorithm is triggered by an event of a new lease or at a regular interval. Firstly, the algorithm sorts the list of leases (e.g. best-effort leases, immediate leases, etc.) in a queue that are ready to run in decreasing order by lease duration. A lease that has longest duration time will be mapped first. Secondly, the algorithm uses a heuristic (FF-MAP-H2L or FF-MAP-L2H) for mapping leases onto physical servers in order to minimize the number of active physical servers. The two allocation algorithms, FF-MAP-H2L and FF-MAP-L2H, which are discussed in our previous works [9], both use two ways in sorting the list of physical servers (i.e. in the order of highest to lowest ranking scores of physical servers and reverse). They allocate a new lease to some active physical servers such that every VM in the new lease is allocated successfully. They always sort free load physical servers at the tail of the sorted list of physical servers. Our energy-aware lease scheduling algorithm is presented in Algorithm 1.

Algorithm 1 Energy-aware lease scheduling

Input: leases in queue, set of physical hosts
Output: None or a mapping of scheduled leases
1: Q = Sort ready leases in queue in decreasing order of their durations.
2: **For** each lease l in the sorted lease queue Q
3: Use **FF-MAP-H2L** or **FF-MAP-L2H** to map the lease l to the first active physical server.
4: **End For**
5: **If** all leases in the queue are mapped successfully, return the mapping of scheduled leases.
6: **Else** return None.

In this paper, we extend the FF-MAP-H2L with migration, called (i) PMIG-LxHy-FF-MAP-H2L and (ii) MIG-LxHy-FF-MAP-H2L. Both of the two algorithms (i) and (ii) do re-scheduling by migrating all of the running leases on physical servers M_k ($k \in [1;m]$) that have resource utilization less than a defined low threshold (x) (e.g. 0.4) and medium threshold (y) (e.g. 0.8). Then the scheduler sets the servers M_k passive and puts them in energy-saving mode (e.g. sleep, shut down). A system administrator sets our defined low and medium thresholds. The algorithm (i) differs from the algorithm (ii) by adding one more step to check whether there are enough available resources in set S_{med}, where $S_{med} = \{M_j | \forall j \in [1;m] \wedge x < cpuload(h) \leq y\}$, or not before it re-schedules all of the running leases on low utilization servers.

We also consider the overheads for migrating leases in both PMIG-LxHy-FF-MAP-H2L and MIG-LxHy-FF-MAP-H2L. Given a lease L_i with set of L_{iv} VMs, the overhead for migrating the lease L_i includes migration time t_i^{mig}, t_i^{sus} suspend time and t_i^{res} resume time of the set of the lease's VMs. The migration time includes t_i^{trans} transferring time of image-disks of these VMs. The scheduler can estimate the migration time, suspend and resume time before re-schedule the migrated leases in future. A. Beloglazov's work [3] did not consider the migration overheads.

For example, consider a lease with two (2) VMs where each VM requires 1,024 MB of physical memory, 4,096 MB of hard disk, a 100 MB/s network, and a physical memory bandwidth of 32 MB/s. Then, we have: $t_i^{sus} = t_i^{res} = 2 \times (1,024/32) = 64.00$ s, $t_i^{mig} = 2 \times (4,096/100) = 81.92$ s. The total migration time that is the sum of migration, suspend and resume times is 145.92 s. Consequently, the migration time causes the lease's waiting time increase.

4 Experimental Study

The system architecture of an energy-efficient resource manager for private clouds was proposed in our previous work [9]. Our proposed system has been deployed on a system with a cloud management software (e.g. OpenNebula) and a resource management (e.g. Haizea) in order to set up a private cloud. Figure 1 shows the proposed system architecture (a) and lease scheduler (b) for provision resources.

We use a script, which is provided by Haizea [11], to run and convert 30 days of a log trace in Parallel Archive Workload (SDSC-BLUE-2000-3.1-cln.swf [15]). We did not change information on the number of jobs, the job arrival time, time to finish the jobs during the conversion. Each simulation will create a total of 5,108 leases. Each lease has a various number of identical VMs with the same size (e.g. single core, 1,024 MB of RAM). We assume that the deployment of VMs on physical servers does not incur overheads. We assume that the simulated cloud data center has 1,000 homogeneous physical servers. Each physical server has a 16/32-core CPU. Overheads of re-scheduling include the suspend/resume rate of 32 MB/s and the network bandwidth of 100 Mbps.

We experimented with the following lease allocation algorithms:

(1) Non Power-Aware Greedy (**NPA Greedy**): The original greedy algorithm in Haizea [11].

(2–3) Our scheduling algorithm with **FF-MAP-L2H**, **FF-MAP-H2L**.

(4–6) The PMIG-LxHy-FF-MAP-H2L with three settings at 0.5, 0.4 and 0.3 low-threshold values and 0.8 high-threshold value that are denoted as **PMIG-L50H80-FF-MAP-H2L, PMIG-L40H80-FF-MAP-H2L and PMIG-L30H80-FF-MAP-H2L**.

(7–9) **MIG-L50H80-FF-MAP-H2L, MIG-L40H80-FF-MAP-H2L and MIG-L30H80-FF-MAP-H2L**: Running the MIG-LxHy-FF-MAP-H2L with three settings at 0.5, 0.4 and 0.3 low-threshold values and 0.8 high-threshold value

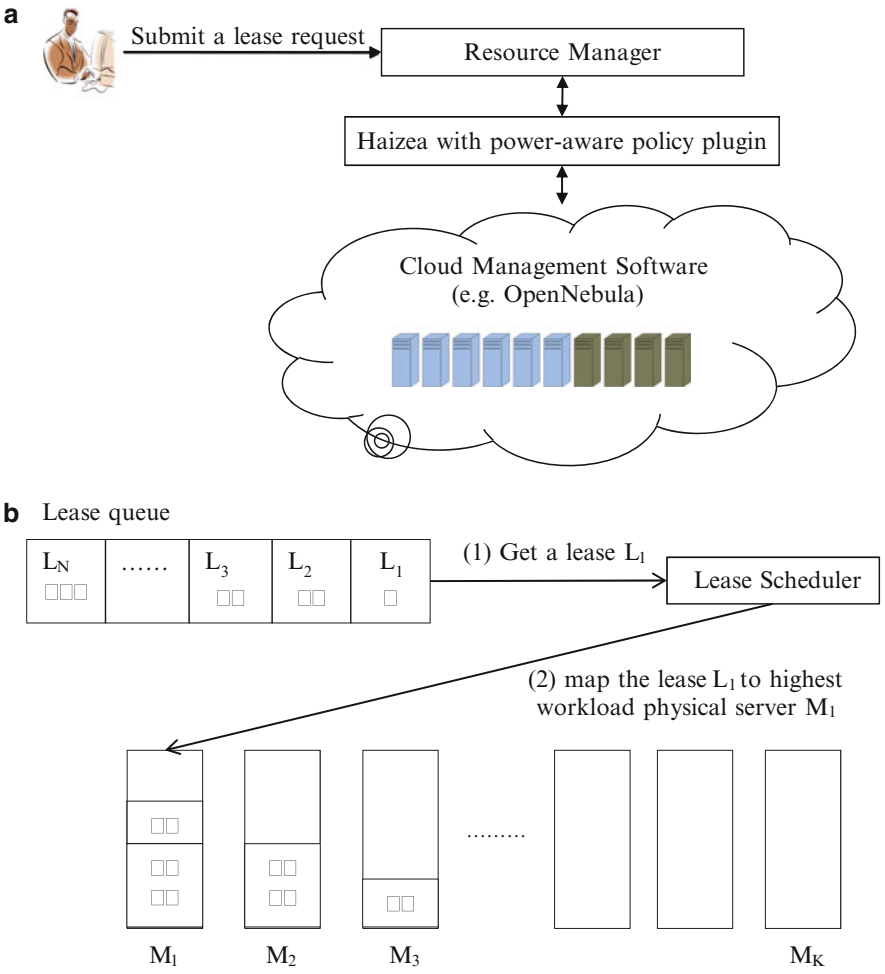

Fig. 1 The system architecture: (**a**) System architecture and (**b**) Lease scheduler

Table 1 Power consumption (Watt) of two HP Proliant servers (Source from [13, 14])

Platform	$P_{idle}(W)$	$P_{max}(W)$
HP Proliant DL585 G5 (2.7 GHz, AMD Opteron 8384)	299	521
HP Proliant DL785 G5 (2.30 GHz, AMD Opteron 8376 HE)	444	799

We collect experimental data on two physical server models: (i) HP Proliant DL585 G5 (2.7 GHz, AMD Opteron 8384, 16 GB of physical memory) [13]; and (ii) HP Proliant DL785 G5 (2.30 GHz, AMD Opteron 8376 HE, 32 GB of physical memory) [14]. Table 1 shows the average active power of both server models. Tables 2 and 3 show simulation results of the above lease allocation algorithms

Table 2 Total energy consumption (kWh), total waiting time, and makespan (C_{max}) of lease allocation algorithms. Each server has 16 cores and 16 GB of physical memory and the power model of HP Proliant DL585 G5 ($P_{min} = 299$ W, $P_{max} = 521$ W), $T_{suspend} = T_{resume} = 32$ MB/s, network bandwidth is 100 Mbps

Algorithm	Energy (kWh)	Total waiting time (h)	C_{max} (h)	Total migrated leases
(1) NPA Greedy	3,287.59	0.000	735.757	0
(2) FF-MAP-H2L	2,736.07	0.000	735.757	0
(3) FF-MAP-L2H	2,741.61	0.000	735.757	0
(4) PMIG-L50H80-FF-MAP-H2L	2,644.36	355.869	737.246	483
(5) PMIG-L40H80-FF-MAP-H2L	2,625.84	222.711	735.828	300
(6) PMIG-L30H80-FF-MAP-H2L	2,654.22	175.804	736.943	223
(7) MIG-L50H80-FF-MAP-H2L	2,682.05	158.893	735.757	134
(8) MIG-L40H80-FF-MAP-H2L	2,660.86	71.347	735.757	165
(9) MIG-L30H80-FF-MAP-H2L	2,674.44	25.438	735.757	112

Table 3 Total energy consumption (kWh), total waiting time, C_{max} of lease allocation policies. Each server has 32 cores, 32 GB of physical memory and the power model of HP Proliant DL785 G5 ($P_{min} = 444$ W, $P_{max} = 799$ W), $T_{suspend} = T_{resume} = 32$ MB/s, network bandwidth is 100 Mbps

Algorithm	Energy (kWh)	Total waiting time (h)	C_{max} (h)	Total migrated leases
(1) NPA Greedy	3,676.35	0.000	735.757	0
(2) FF-MAP-H2L	2,260.60	0.000	735.757	0
(3) FF-MAP-L2H	2,282.37	0.000	735.757	0
(4) PMIG-L50H80-FF-MAP-H2L	2,165.67	757.395	736.943	464
(5) PMIG-L40H80-FF-MAP-H2L	2,167.33	195.388	736.989	297
(6) PMIG-L30H80-FF-MAP-H2L	2,171.52	137.541	735.828	225
(7) MIG-L50H80-FF-MAP-H2L	2,215.98	56.566	735.757	109
(8) MIG-L40H80-FF-MAP-H2L	2,207.44	520.333	735.757	113
(9) MIG-L30H80-FF-MAP-H2L	2,197.66	55.699	735.757	118

on a simulated cluster with 16 and 32 core architectures and compare their total energy consumption (kWh) to the NPA Greedy algorithm [11]. Figure 2 shows the total energy consumption (kWh) of each allocation algorithm.

The results show that the energy-aware lease scheduling has the total waiting time and C_{max} equal to that of the NPA in the experiments. Compared to the NPA, the energy-aware lease scheduling with both FF-MAP-H2L and FF-MAP-L2H reduces the total energy consumption in both 16-core and 32-core cases. Our proposed algorithms reduced total energy consumption that is linear increasing in the number of cores in each host. Moreover, using the FF-MAP-H2L with migration algorithms at three (0.5, 0.4, 0.3) threshold values, called PMIG-L50H80-FF-MAP-H2L, PMIG-L40H80-FF-MAP-H2L, PMIG-L30H80-FF-MAP-H2L, MIG-L50H80-FF-MAP-H2L, MIG-L40H80-FF-MAP-H2L and MIG-L30H80-FF-MAP-H2L, also

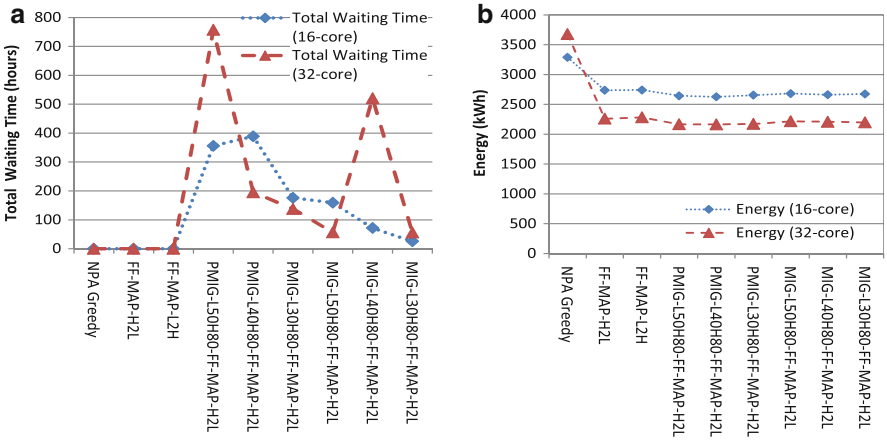

Fig. 2 The total energy consumption (kWh) for the investigated algorithms. (**a**) Total waiting time. (**b**) Total energy consumption

reduced the total energy consumption more than the FF-MAP-H2L, FF-MAP-L2H and NPA without migration. A disadvantage of these migration algorithms, however, is the decreasing performance, i.e. these migration algorithms increase the total waiting time of migrated leases when we consider overheads in migration and rescheduling these migrated leases. Consequently, C_{max} can be increased.

5 Conclusions and Future Work

This work presents an energy-aware lease scheduling problem and proposes a scheduling algorithm for lease scheduling problems to minimize the total energy consumption. The simulation results show that our algorithms reduce the total energy consumption significantly compared with an existing FCFS-based algorithm in the Haizea. Our algorithms are also beneficial on multi-core architectures, i.e. the more cores the machines have, the more the energy consumption is reduced.

In future, we are interested in cloud systems with heterogeneous resources. The cloud systems will provide resources to many types of leases such as best-effort, advanced reservation, and immediate leases at the same time. We will investigate the VM placement problem with multiple resources (e.g. CPU, RAM, network bandwidth, etc.) and scheduling algorithms to solve the special case of energy-aware lease scheduling.

References

1. Albers, S.: Energy-efficient algorithms. Commun. ACM **53**(5), 86–96 (2010)
2. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. Computer **40**(12), 33–37 (2007)
3. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener. Comput. Syst. **28**(5), 755–768 (2012)
4. Buyya, R., Yeo, C., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. **25**(6), 599–616 (2009)
5. Fan, X., Weber, W.-D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. ACM SIGARCH Comput. Archit. News. **35**, 13 (2007)
6. Feitelson, D.G., Rudolph, L., Schwiegelshohn, U.: Parallel job scheduling – a status report. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2004. LNCS, vol. 3277, pp. 1–16. Springer, Heidelberg (2005)
7. Goiri, Í., Nou, R., Berral, J., Guitart, J., Torres, J.: Energy-aware scheduling in virtualized datacenters. In: IEEE International Conference on Cluster Computing, CLUSTER 2010, Heraklion, pp. 58–67 (2010)
8. Panigrahy, R., Talwar, K., Uyeda, L., Wieder, U.: Heuristics for vector bin packing. Technical report, Microsoft Research (2011)
9. Quang-Hung, N., Thoai, N., Son, N.T.: Performance constraint and power-aware allocation for user requests in virtual computing lab. J. Sci. Technol. (Vietnam), **49**(4A), 383–392 (2011)
10. von Laszewski, G., Wang, L., Younge, A.J., He, X.: Power-aware scheduling of virtual machines in DVFS-enabled clusters. In: IEEE International Conference on Cluster Computing and Workshops, 2009, New Orleans, pp. 1–10 (2009). doi:10.1109/CLUSTR.2009.5289182
11. Sotomayor, B.: Provisioning Computational resources using virtual machines and leases. PhD Thesis submited to The University of Chicago, US (2010)
12. Sotomayor, B., Keahey, K., Foster, I.: Combining batch execution and leasing using virtual machines. In: Proceedings of the Eighteenth International Symposium on High Performance Distributed Computing (HPDC'08), Boston, 23–27 June 2008, pp. 87–96 (2008)
13. SPECpower ssj2008 results for HP ProLiant DL585 G5 (2.70 GHz, AMD Opteron 8384). http://bit.ly/JrkskF
14. SPECpower ssj2008 results for HP ProLiant DL785 G5 (2.30 GHz, AMD Opteron 8376 HE). http://bit.ly/K99RfD
15. The San Diego Supercomputer Center (SDSC) Blue Horizon log. http://bit.ly/JUQsiP

Mathematical Models of Perception and Generation of Art Works by Dynamic Motions

Alexander Schubert, Katja Mombaur, and Joachim Funke

Abstract This paper presents a study on the role of dynamic motions in the creation and perception processes of action-art paintings. Although there is a lot of interest and some qualitative knowledge around, there are no quantitative models in the scientific computing sense about this process yet. To create such models and implement them on a robotic platform is the objective of our work. Therefore, we performed motion capture experiments with an artist and reconstructed the recorded motions by fitting the data to a rigid-body model of the artist's arm. A second model of a 6-DOF robotic platform is used to generate new motions by means of optimization and optimal control algorithms. Additionally, we present an image analysis framework that computes certain image characteristics related to aesthetic perception and a web tool that we developed to perform online sorting and cluster studies with participants. We present first results concerning motion reconstruction and perception studies and give an outlook to what will be the next steps towards an autonomous painting robotic platform.

1 Introduction

The cognitive processes of generating and perceiving abstract art are – in contrast to figurative art – mostly unknown. Within the process of perceiving representational art works, the effect of meaning is highly dominant. In abstract art, with the lack of this factor, the processes of perception are much more ambiguous, relying on a variety of more subtle qualities. In this work, we focus on the role of dynamic motions performed during the creation of an art work as one specific aspect that influences our perception and aesthetic experience.

A. Schubert (✉) • K. Mombaur
Interdisciplinary Center for Scientific Computing, University of Heidelberg, Heidelberg, Germany
e-mail: alexander.schubert@iwr.uni-heidelberg.de; katja.mombaur@iwr.uni-heidelberg.de

J. Funke
Institute of Psychology, University of Heidelberg, Heidelberg, Germany
e-mail: joachim.funke@psychologie.uni-heidelberg.de

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_17

1.1 Action Paintings: Modern Art Works Created by Dynamic Motions

The term "action painting" was first used in the essay "The American Action Painters" by Harold Rosenberg in 1952 [1]. While the term "action painting" is commonly used in public, art historians sometimes also use the term "Gestural Abstraction". Both terms emphasize the process of creating art, rather than the resulting art work, which reflects the key innovation that arose with this new form of painting in the 1940s to the 1960s. The artists often consider the physical act of painting itself as the essential aspect of the finished work. The most important representative of this movement is Jackson Pollock (1912–1958), who introduced this new style around 1946. Clearly, artists like Pollock do not think actively about dynamic motions performed by their bodies the way, mathematicians from the area of modeling and optimal control do. But from a mathematical and biomechanical point of view it is very exciting that one of the main changes they applied to their painting style in order to achieve their aim of addressing the subconscious mind has been a shift in the manner they carry out their motions during the creational process

1.2 Understanding the Perception and Generation of Art Works

Since humans possess many more degrees of freedom than needed to move a hand (or any end-effector that they might be using for painting, like brushes or pencils), the motions executed by an artist can be seen as a superposition of goal directed motions and implicit, unconscious motions. The former are carried out to direct his hand to the desired position, the latter are the result of some unconscious process defining a particular style of the motion. From a mathematical perspective, this can be seen as an implicitly solved optimal control problem with a certain cost function

Fig. 1 An action painting in the style of Jackson Pollock, painted by "JacksonBot"

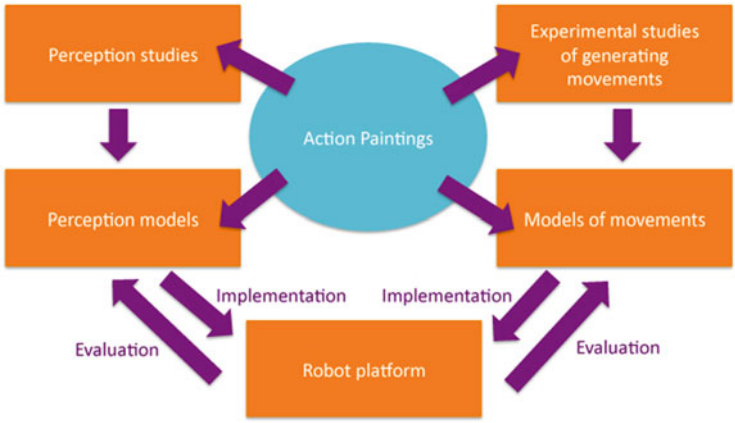

Fig. 2 Schematic overview of experimental and computational parts of study

relating to smoothness, jerk, stability or energy costs. The assumption that human motion can be described in this manner has been widely applied and verified, for example in human locomotion. For details, see [2] or [19].

When looking at action paintings, we note that this form of art generation is a very extreme form of this superposition model with a negligible goal-directed part. Therefore, it is a perfect basis to study the role of (unconscious) motion dynamics on a resulting art work.

The goal of our project is to use state-of-the-art tools from scientific computing to analyze the impact of motion dynamics both on the creational and perceptual side of action-painting art works. Figure 2 shows a schematic overview of the experimental and theoretical parts of our project. On the one hand, we perform perception studies, in which participants are shown different action paintings and then have to describe how they perceive these paintings. On the basis of these experiments, models for the perception of action paintings are established. On the other hand, we have conducted motion capture studies in which an artist generated action paintings. The painting process was recorded using several inertia sensors on the artist's arm and hand which provide both kinematic and dynamic data. On the basis of these recordings, we reconstructed and analyzed the artist's motion. Results from both approaches – on perception and on the generation of action art – will later be implemented on a robot for validation purposes. In this paper, we present some preliminary results on modeling, motion reconstruction as well as on perception studies and our image analysis framework.

1.3 Paper Outline

This paper is organized as follows: In Sect. 2, we will give an introduction to the current theory of art perception and an overview of the tools we developed for

image analysis and online perception experiments. In Sect. 3, we first briefly discuss the mathematical background of our work by introducing optimal control problems and the direct multiple shooting method. Then, we describe the reconstruction of recorded motions from an artist using multibody dynamics and optimal control theory. Thereafter, we present our plan to create new motions for our robotic platform by solving an optimal control problem to compute the joint torques. Finally, in Sect. 4, we conclude our current findings and present the next steps in our project plan.

2 Modeling the Perception of Art Works

When we talk about models for art perception in this paper, we have to state that we do not want to create a new qualitative model for art perception but we want to find quantitative data that link the motion dynamics of the creation process to viewers' aesthetic experience when looking at the painting. Once we find this data, we aim to integrate it into existing perception models, possibly modifying or improving them. Our main goal is, however, to develop a simple mathematical model that allows our robotic platform to continuously monitor its painting process and to adapt its motion dynamics considering previously given goals.

2.1 Previous Work/State of the Art

The perception of art, especially abstract art, is still an area of ongoing investigations. Therefore, no generally accepted theory including all facets of art perception exists. There are, however, different theories that can explain different aspects of art perception. One example of a theory of art perception is the one presented by Leder et al. in [3] (see Fig. 3). In the past, resulting from an increasing interest in embodied cognition and embodied perception, there has been a stronger focus on the nature of human motion and its dynamics regarding neuroscience or rather neuroaesthetics as well as psychology and history of art. There are several results, showing that we perceive motion and actions with a strong involvement of those brain regions that are responsible for motion and action generation. These findings support the theory that the neural representations for action perception and action production are identical (see, e.g. [4]). The relation between perception and embodied action simulation also exists for static scenes (see, e.g. [5]) and ranges even to the degree, where the motion is implied only by a static result of this very motion. For example, Knoblich et al. showed in [6] that the observation of a static graph sign evokes in the brain a motor simulation of the gesture, which is required to produce this graph sign. Finally, in [7], D. Freedberg and V. Gallese proposed that this effect of reconstructing motions by embodied simulation mechanisms will also be found when looking at "art works that are characterized by the particular

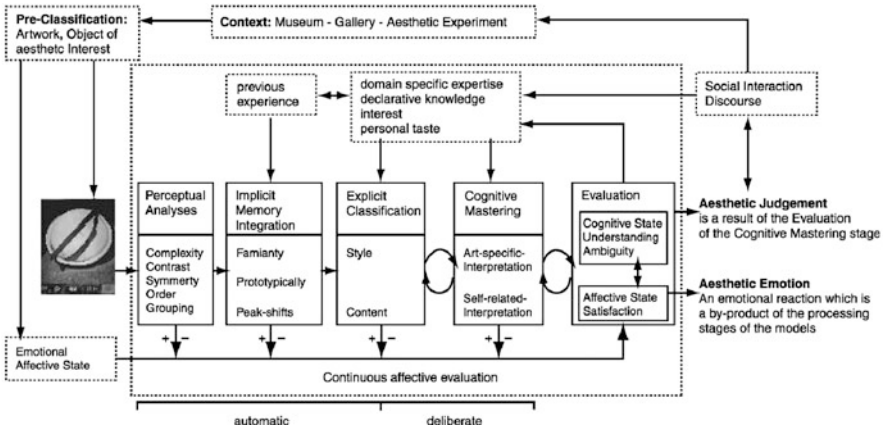

Fig. 3 Figure taken from Leder et al. [3]

gestural traces of the artist, as in Fontana and Pollock" – a conjecture that has first been observed empirically by Taylor et al. in [8].

2.2 Perception Experiments

This section describes our perception experiments which are performed using a web interface that we created for this purpose.

We performed two pre-studies to find out, whether human contemplators can distinguish robot paintings from human-made paintings and how they evaluate robot paintings created by the robot JacksonBot [17] using motions that are the result of an optimal control problem with different mathematical objective functions. In the first study, we showed nine paintings to 29 participants, most of whom were laymen in arts and only vaguely familiar with Jackson Pollock. Seven paintings were original art works by Jackson Pollock and two paintings were generated by the robot platform JacksonBot. We asked the participants to judge which of the paintings were original paintings by Pollock and which were not, but we intentionally did not inform them about the robotic background of the "fake" paintings. As might be expected, the original works by Pollock had a higher acceptance rate, but, very surprisingly, the difference between Pollock's and JacksonBot's paintings was not very high (2.74 ± 0.09 vs. 2.85 ± 0.76, on a scale of 1–5).

In the second study, the participants were shown ten paintings created solely by the robot platform, but with two different objective functions (maximizing and minimizing overall angular velocity in the robot arm) in the optimal control problem. The participants easily distinguished the two different painting styles.

After the pre-studies, we developed a more sophisticated web-based platform for further, more detailed investigations on this subject.

The first goal of our detailed perception experiments is to find out about the way, viewers judge action-art paintings regarding similarity. Therefore, we present a set of stimuli consisting of original action-art paintings by Pollock and other artists and added images, that were painted by our robot platform. Participants are then asked to perform different tasks with these stimuli.

The web-interface provides three different study types for perception analysis. In the first task, the viewers are presented three randomly chosen paintings and asked to arrange them on the screen according to their similarity. As a result, for every set of three paintings A, B, C, we obtain a measure $d_{ABC} = \frac{dist_{AB}}{dist_{BC}}$ for the similarity of two paintings in comparison with another pair of two paintings.

In the second task, people are basically asked to perform a standard sorting study, i.e. they are asked to combine similar paintings in groups and to give some information about their reasons for the chosen groups (Fig. 4). The results of this task are used to validate the information obtained by the previous one and, additionally, they are used to gain more information about the attributes and traits, people seem to use while grouping.

Finally, participants are shown images individually and are asked to judge them on different absolute scales. The results from this task are used to obtain an overall scaling for the first two tasks.

Once, we have obtained this information for a sufficient amount of robot paintings, we can use standard procedures from statistics like fuzzy cluster analysis or multidimenstional scaling to determine whether viewers differentiate between paintings created by different objective functions or rather whether they rate paintings created by the same objective function as similar. Additionally, we can link the given cluster descriptions to certain objective functions (e.g. paintings created by maximum jerk motions might be clustered together and be described as “aggressive” or “dynamic”).

Fig. 4 Interface for web-based perception studies

2.3 Perception Models

As stated in Sect. 2, we want to develop a model that allows our robotic platform to monitor its painting process using a camera system and – based on an evaluation of its current status – to change its movement according to predefined goals. Therefore, we developed an image analysis software tool based on OpenCV for details, see [9] that uses a variety of different filters and image processing tools that are related to aesthetic experience. For an overview on the software, see [10]. To give only one example, Taylor et al. showed in [11] that fractal-like properties of art works might be of interest, particularly when looking at action-art paintings. We address the question of fractal-like properties by computing two values: the fractal dimension D using the "box counting" method and the Fourier power spectrum using FFT. The fractal dimension is calculated by overlapping the given image with a continuously refining two-dimensional grid of width ϵ. If $N(\varepsilon)$ is the number of "boxes" that cover a part of the object of interest, the fractal dimension is given by:

$$D = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}} \tag{1}$$

By linking these low-level image features to the viewer's judgements described in the previous paragraph, the robot will be able to predict the most likely judgement of a viewer and to adapt its movement accordingly.

3 Modeling the Generation of Art Works by Dynamic Motions

As mentioned in Sects. 1.1 and 1.2, the generation of action paintings uses motions that arise from the subconscious of the artists. Therefore, we cannot try to generate similar motions by traditional path planning. Instead, we apply our approach of generating motions as the result of an optimal control problem, which is much more suited to address this type of motions.

3.1 Mathematical Background

To perform mathematical computations on motion dynamics, we first need to create models of a human and the robot arm. In this case, by "model", we mean a physical multi-body model consisting of rigid bodies which are connected by different types of joints (prismatic or revolute). Depending on the number of bodies and joints, we end up with an certain number of degrees of freedom and a set of generalized

variables q (coordinates), $\dot{q}$ (velocities), $\ddot{q}$ (accelerations), and τ (joint torques). Given such a model, we can fully describe its dynamics by means of

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \tag{2}$$

where $M(q)$ is the joint space inertia matrix and $N(q, \dot{q})$ contains the generalized non-linear effects. Once we have such a model, we can formulate an optimal control problem using $x = [q, \dot{q}]^T$ as states and $u = \tau$ as controls. The OCP can be written in a general form as:

$$\min_{x,u,T_1} \int_{T_0}^{T_1} L(t, x(t), u(t), p)dt + \Phi_M(T_1, x(T_1)) \tag{3}$$

subject to:

$$\dot{x} = f(t, x(t), u(t), p) \tag{4}$$

$$g(x(t), u(t), p) \geq 0 \tag{5}$$

$$r_{T_0}(x(T_0), p) + r_{T_1}(x(T_1), p) = 0 \tag{6}$$

where p contains several model parameters which in our case are fixed and g contains constraints like joint and torque limitations. Note, that all the dynamic computation from our model is included in the RHS of diff.eq. (4), The objective function is given by the sum of the Lagrange term $\int_{T_0}^{T_1} L(t, x(t), u(t), p)dt$ and the Mayer term $\Phi_M(T_1, x(T_1))$. The former is used to address objectives that have to be evaluated over the whole time horizon (such as minimizing jerk), the latter is used to address objectives that only need to be evaluated at the end of the time horizon (such as overall time). In our case, we will often only use the Lagrange term. For details about the specific problems we used, see Sects. 3.3 and 3.4.

To solve such a problem numerically, we apply a direct multiple shooting method which was developed by Bock and Plitt [12] and is implemented in the software package MUSCOD-II, which is maintained and developed further at IWR. It discretizes the continuous formulation of our optimal control problem by dividing the time horizon in several so-called multiple shooting intervals I_j. This discretization is used both for controls and states, the latter are parameterized as starting values s_j for an initial value problem on each multiple shooting interval I_j. The controls are given by simple base functions $\bar{u}|_{I_j}$ (e.g. piece-wise constant, piece-wise linear or spline functions) for each interval. Additional continuity conditions

$$x(t_{j+1}, s_j, \bar{u}|_{I_j}) - s_{j+1} = 0$$

are added for each multiple-shooting-node to ensure a continuous solution. Further discretization of the constraints and objective function leads to a nonlinear optimization problem:

$$\min_{y} F(y) \tag{7}$$

subject to:

$$g(y) \geq 0 \tag{8}$$

$$h(y) = 0 \tag{9}$$

where y contains the variables s_j, T_1 and the parameters describing the control base functions $\bar{u}|_{I_j}$. This problem is then solved by using a specially tailored sequential quadratic programming (SQP) method. For a more detailed description of the algorithm, see [12, 13]. Regarding dynamics computation, we use the Rigid Body Dynamics Library (RBDL) [14] which is an highly efficient C++ library for forward and inverse rigid body dynamics and includes all major algorithms like the articulated body algorithm and a recursive Newton-Euler algorithm.

3.2 Previous Work/State of the Art

Optimization and optimal control techniques are very powerful tools that can be applied concerning many aspects of our research. In this specific case, we use optimization methods to compute the full trajectory of our robotic platform. Our basic approach is that humans are unwittingly applying optimization in different areas like motion control or complex problem solving. As mentioned in Sect. 1.2, this approach of characterizing human motions as solution of an optimal control problem has been successfully applied in several areas, particularly in walking and running motions (see [2, 15]), but also (very recently) regarding emotional body language during human walking (see [16]). Concerning the application of our approach on painting motions, a first proof of concept has been given by our previous robotic platform "JacksonBot". Even though with "JacksonBot", the optimization was purely kinematic with no respect to motion dynamics, paintings created using different optimality conditions were clearly distinguished by viewers (see [17]).

3.3 Experiments with Artists

In order to study the way, real human artists move during action-painting, we performed motion-capture studies. We started with several experiments where we recorded the motion of a collaborating artist and plan to redo the same experiments with other artists for validation purposes. We used three inertia sensors to record dynamic data $D_{capture}$ for each of the three segments of the artist's arm (hand, lower arm, upper arm). To fit this data to our 9 DOF model of a human arm that is based on

data from deLeva [18], we formulated an optimal control problem which generates the motion $x(t) = [q(t), \dot{q}(t)]^T$ and the controls $u(t) = \tau(t)$ that best fit the captured data with respect to the model dynamics f.

$$\min_{\alpha} \sum_{i} ||D^*(x(t_i;\alpha)) - D_{mocap}(t_i)||_2^2, \tag{10}$$

$D^*(x(t;\alpha))$ resulting from a solution of

$$\min_{x,u} \int_{T_0}^{T_1} \left[\sum_{i=1}^{n} \alpha_i L_i(t, x(t), u(t), p) \right] dt \tag{11}$$

subject to:

$$\dot{x}(t) = f(t, x(t), u(t), p) \tag{12}$$

$$r_{T_0}(x(T_0), p) + r_{T_1}(x(T_1), p) = 0 \tag{13}$$

$$g(x(t), u(t), p) \geq 0 \tag{14}$$

The constraints in this case are given by the limited angles of the human arm joints and torque limitations of the arm muscles. Figure 5 shows the computed states and the fit quality of the acceleration data for a very dynamic, jerky motion. Note that for this type of motion, the fact that the angle values are approaching the joint limitations is plausible.

3.4 Motion Generation for Robot Platform by Means of Optimal Control

To generate new motions for our robotic platform (a 6-DOF-KUKA arm) we created a 6-DOF rigid-body-model of the arm. We now can compute end-effector trajectories as results of optimal control problems with different objective functions. The mathematical problem is described and solved using the optimal control code MUSCOD-II as it has been described in Sect. 3.1. In this case, we include all limitations of our KUKA arm using the inequality constraints $g(x(t), u(t), p) \geq 0$ and choose from a set of different objective functions L derived either from our motion capture experiments or motivated from physical extremes (e.g. maximizing the torque or minimizing the variance of the angular velocities in all joints).

The paintings created by the robot based on (a superposition of) these objective functions will be added to the paintings already present in the framework of our perception studies. This has two major advantages compared to human-created paintings: First, we know the exact details about the underlying motion dynamics

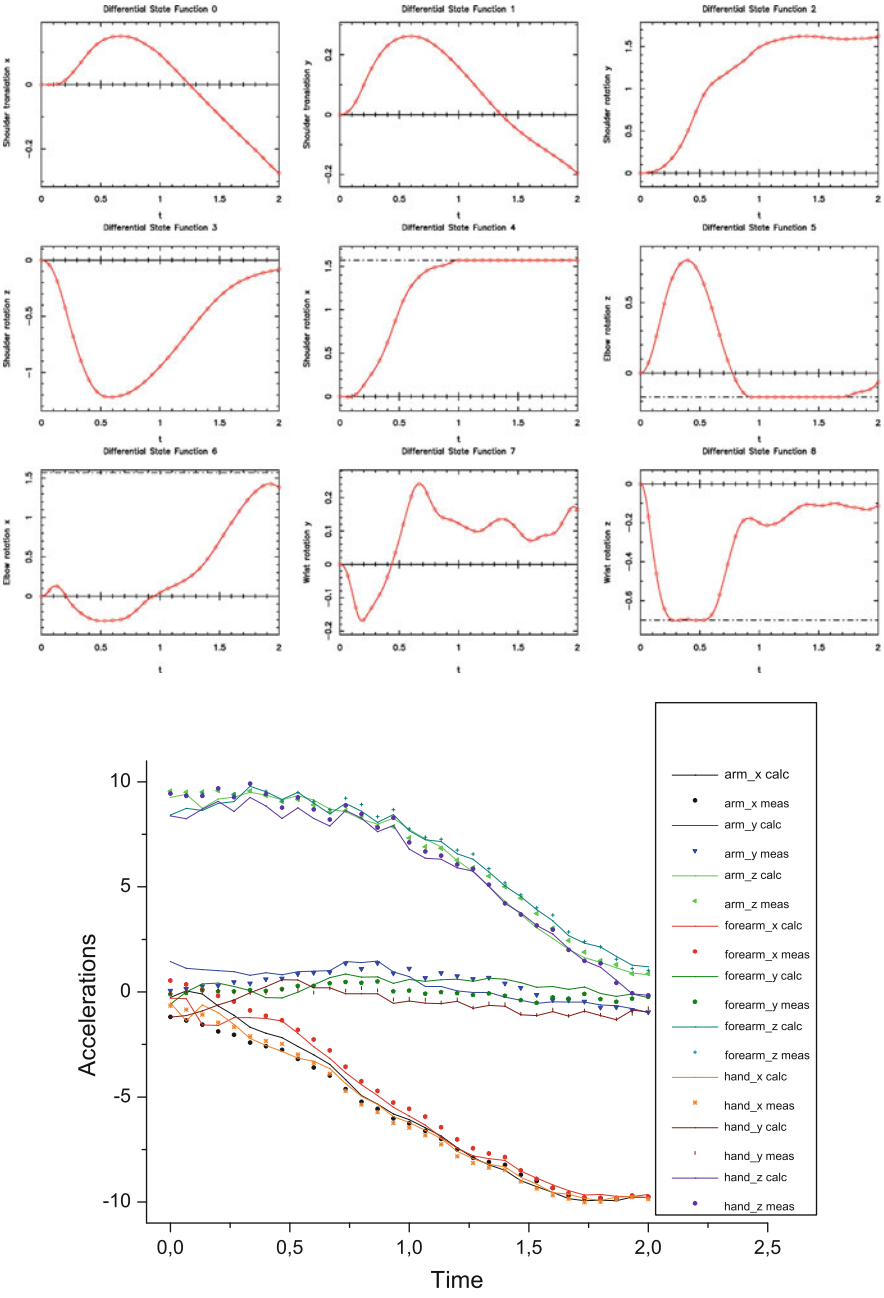

Fig. 5 Computed trajectories for joint angles (*above*) and comparison of computed (*dots*) and measured (*lines*) accelerations (*below*)

and can therefore derive correlations more easy. Second, we can easily create images specifically suited to an area of interest in our perception study.

4 Summary

An overview of our approach to investigate the influence of dynamic motions on modern art works was presented. We successfully reconstructed artist's motions from dynamic motion-capture data using a rigid-body model of the artist's arm. We described the advantages of our optimal control approach to this specific type of human motions and portrayed the combination of several tools for perception studies and image analysis with a robotic platform in order to uncover the subconscious nature of action-painting motions. In the next step, we will use the motion capture data obtained from experiments with our collaborating artist not only to reconstruct the motion, but to use an inverse optimal control approach (like successfully used in a similar case by Mombaur et al. in [19]) to retrieve the underlying objective functions of these motions. To do so, we will use an efficient direct all-at-once approach as presented by Hatz et al. in [20]. We will link these objectives both to low-level image features detected by our image analysis framework and viewers' judgements derived from our online-tool. That way, we aim to build a database containing all this information as a foundation to create a feedback for the robot painting process.

Acknowledgements The authors want to thank the Simulation and Optimization research group of IWR at the University of Heidelberg for the permission to work with the software package MUSCOD-II. We greatfully acknowledge financial and scientific support that was given by the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences, funded by DFG (Deutsche Forschungsgemeinschaft). We also greatfully acknowledge financial travel support granted by DAAD (Deutscher Akademischer Austauschdienst). Finally, we want to thank Nicole Suska for the possibility to perform motion capture experiments with her.

References

1. Rosenberg, H.: The American action painters. Art News 51/8, 22 1952
2. Felis, M., Mombaur, K.: Modeling and optimization of human walking. Cognitive Systems Monographs, Springer LNEE. **18**, 31–42 (2013)
3. Leder, H., Belke, B., Oeberst, A., Augustin, D.: A model of aesthetic appreciation and aesthetic judgments. Br. J. Psychol. **95**(4), 489–508 (2004)
4. Buxbaum, L.J., Kyle, K.M., Menon, R.: On beyond mirror neurons: internal representations subserving imitation and recognition of skilled object-related actions in humans. Cogn. Brain Res. **25**, 226–239 (2005)
5. Urgesi, C., Moro, V., Candidi, M., Aglioti, S.M.: Mapping implied body actions in the human motor system. J. Neurosci. **26**, 7942–7949 (2006)

6. Knoblich, G., Seigerschmidt, W., Flach, R., Prinz, W.: Authorship effects in the prediction of handwriting strokes: evidence for action simulation during action perception. Q. J. Exp. Psychol. **55**(3), 1027–1046 (2002)
7. Freedberg, D., Gallese, V.: Motion, emotion and empathy in esthetic experience. Trends Cogn. Sci. **11**, 197–203 (2007)
8. Taylor, J.E.T., Witt, J.K., Grimaldi, P.J.: Uncovering the connection between artist and audience: viewing painted brushstrokes evokes corresponding action representations in the observer. Cognition **125**(1), 26–36 (2012)
9. Bradski, G.: The OpenCV library. Dr. Dobb's J. Softw. Tools **25**, 120–126 (2000)
10. Ducati, D.: Computergestützte Bildanalyse in der Robotik. Thesis, IWR, University of Heidelberg (2012)
11. Taylor, R.P., Micolich, A.P., Jonas, D.: Fractal analysis of Pollock's drip paintings. Nature **399**, 422 (1999)
12. Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings 9th IFAC World Congress, Budapest. Pergamon Press, pp. 242–247 (1984)
13. Leineweber, D.B., Bauer, I., Bock, H.G., Schlöder, J.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization – part I: theoretical aspects. Comput. Chem. Eng. **27**, 157–166 (2003)
14. Felis, M.: RBDL – the Rigid Body Dynamics Library (2011). http://rbdl.bitbucket.org. Cited15May2012
15. Schultz, G., Mombaur, K.: Modeling and optimal control of human-like running. IEEE/ASME Trans. Mechatron. **15**(5), 783–792, (2010, published online 2009)
16. Felis, M., Mombaur, K., Berthoz, A.: Mathematical modeling of emotional body language during human walking. In: Modeling, Simulation and Optimization of Complex Processes - HPSC 2012, Proceedings of the 5th International Conference on HPSC, March 5-9, 2012, Hanoi, Vietnam
17. Raschke, M., Mombaur, K., Schubert, A.: An optimisation-based robot platform for the generation of action paintings. Int. J. Arts Technol. **4**(2), 181–195 (2011)
18. de Leva, P.: Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. J. Biomech. **29**(9), 1223–1230 (1996)
19. Mombaur, K., Truong, A., Laumond, J-P.: From human to humanoid locomotion – an inverse optimal control approach. Auton. Robots **28**(3), 369–383 (2010)
20. Hatz, K., Schlöder, J.P., Bock, H.G.: Estimating parameters in optimal control problems. SIAM J. Sci. Comput. **34**(3), A1707–A1728 (2012)

An Eulerian Interface-Sharpening Algorithm for Compressible Gas Dynamics

Keh-Ming Shyue

Abstract We describe a novel Eulerian interface-sharpening approach for the efficient numerical resolution of contact discontinuities arising from inviscid compressible flow in more than one space dimension. The algorithm uses the single-phase compressible Euler equations as the model system, and introduces auxiliary differential terms to the model so as to neutralize numerical diffusion that is inevitable when the original Euler system is solved by a diffused interface method. A standard fractional-step method is employed to solve the proposed model equations in two steps, yielding an easy implementation of the algorithm. Preliminary results obtained using an anti-diffusion based model system are shown to demonstrate the feasibility of the algorithm for practical problems.

1 Introduction

Computing a non-oscillatory, positivity-preserving, sharply resolved volume fraction function, denoted by $\alpha \in [0, 1]$, for the initial-value problem of the volume-fraction transport equation

$$\partial_t \alpha + \mathbf{u} \cdot \nabla \alpha = 0 \tag{1}$$

with discontinuous initial data is of fundamental importance in many practical problems of interest. One simple example is concerned with an unsteady, incompressible, viscous, two-phase flow that is governed by the incompressible Navier-Stokes equations,

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p I_N) &= \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} + \mathbf{f}_\sigma, \end{aligned} \tag{2}$$

K.-M. Shyue (✉)
Department of Mathematics, National Taiwan University, Taipei 10617, Taiwan
e-mail: shyue@ntu.edu.tw

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_18

where $\mathbf{u}$ denotes the velocity vector, ρ the density, p the pressure, I_N the $N \times N$ identity matrix (N the number of spatial dimensions), $\boldsymbol{\tau} = \epsilon \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)$ the stress tensor, $\mathbf{g}$ the gravitational field, and $\mathbf{f}_\sigma = -\sigma\kappa\nabla\alpha$ the capillary force. We assume that the fluids of interest consist of two different phases, gas and liquid, for instance, separated by immiscible interfaces, where in regions $\alpha = 0$ and $\alpha = 1$, the fluid is single phase (gas or liquid), while in regions $0 < \alpha < 1$, we have a (gas-liquid) two-phase coexistent phase. In the latter case, it is a common practice to set the density as well as the material quantities such as the dynamic viscosity ϵ and the surface-tension coefficient σ by the solution of (1) via a simple α-weighted average,

$$z = \alpha z_1 + (1 - \alpha) z_2$$

for $z = \rho, \epsilon, \sigma$; z_k the kth phasic variable of z. In addition to that, from the given set of volume fractions, the normal direction $\nabla\alpha$ and the curvature $\kappa = \nabla \cdot (\nabla\alpha/|\nabla\alpha|)$ at the interface that contributes to the capillary force $\mathbf{f}_\sigma$ on the right-hand side of the momentum equations may be calculated via numerical means.

For incompressible two-phase flow governed by (1) and (2), interface sharpening of some kind (cf. [1, 8, 15–17] and references therein) is a popular technique that is applied together with an underlying advection scheme to compute a sharp solution profile of (1); this yields an accurate definition of the aforementioned physical and geometrical quantities present in (2) near the interfaces, and is viable to the remaining parts of the flow solver. Among various interface-sharpening approaches, in this work, we are interested in a class of methods that is based on the inclusion of a differential source term to (1) in a form,

$$\partial_t \alpha + \mathbf{u} \cdot \nabla\alpha = \frac{1}{\mu} \mathcal{D}_\alpha, \tag{3a}$$

as a numerical model for interface-sharpening, where $\mu \in \mathbb{R}$ is a free parameter. In the work proposed by Olsson, Kreiss, and coworker (cf. [9, 10]), the term $\mathcal{D}_\alpha$ is of an interface-compression type as

$$\mathcal{D}_\alpha := \nabla \cdot \left[(\varepsilon \nabla\alpha \cdot \mathbf{n} - \alpha\,(1 - \alpha))\, \mathbf{n} \right], \tag{3b}$$

where both a nonlinear convection and a linear diffusion term are introduced in the model. Here $\mathbf{n} = \nabla\alpha/|\nabla\alpha|$ is the unit normal, and $\varepsilon > 0$ is the diffusion coefficient which is assumed to be in the order of the spatial mesh size. On the other hand, in the work advocated by So, Hu, and Adams [13], it takes simply the linear diffusion term, but is of an anti-diffusion one, as

$$\mathcal{D}_\alpha := -\nabla \cdot (\varepsilon \nabla\alpha), \tag{3c}$$

where the diffusion coefficient ε is assumed to be in the order of the velocity vector in absolute value which mimics the diffusion rate from the modified equation of the numerical method.

Our goal here is to describe a novel approach that generalizes (3) for interface-sharpening of discontinuous volume fractions in incompressible flow to more general interfaces (i.e., contact discontinuities) that are governed by the Euler equations in the compressible single-phase flow; an extension of this approach to the multi-phase case will be reported elsewhere. The proposed model that we are going to describe in Sect. 2 will be formulated in such a way that a standard fractional-step method can be applied, yielding a simple and yet accurate algorithm for numerical approximation.

2 Mathematical Models

The basic physical conservation laws for the inviscid, non-heat conducting, single-phase, compressible flow in Cartesian coordinates take the form

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p I_N) &= 0, \\ \partial_t E + \nabla \cdot (E \mathbf{u} + p \mathbf{u}) &= 0. \end{aligned} \tag{4}$$

We assume that the constitutive law for the fluid phase of interest satisfies a Mie-Grüneisen equation of state of the form

$$p(\rho, e) = p_{\text{ref}}(\rho) + \Gamma(\rho)\rho \left[e - e_{\text{ref}}(\rho) \right]. \tag{5}$$

Here e denotes the specific internal energy, $\Gamma = (1/\rho)(\partial_e p)|_\rho$ is the Grüneisen coefficient, and p_{ref}, e_{ref} are the properly chosen states of the pressure and the internal energy along some reference curve (e.g., along an isentrope, a single shock Hugoniot, or the other empirically fitting curves) in order to match the experimental data of the material being examined. For simplicity, each of the expressions Γ, p_{ref}, and e_{ref} is taken as a function of the density only, see Sect. 4 for an example. We have $E = \rho e + \rho \mathbf{u} \cdot \mathbf{u}/2$ denoting the total energy as usual.

To derive our compressible model for interface-sharpening that may be used in a diffused interface method for numerical approximation, as in our previous work for compressible multiphase flow solver (cf. [12]), we begin by considering an interface only problem (i.e., a contact discontinuity in gas dynamics) where both the pressure and the velocity are assumed to be constants in the whole domain, while the density is having jumps across some interfaces. Then from (4), we find easily the basic transport equations for the interfaces as

$$\partial_t \rho + \mathbf{u} \cdot \nabla \rho = 0, \tag{6a}$$

$$\mathbf{u} \left(\partial_t \rho + \mathbf{u} \cdot \nabla \rho \right) = 0, \tag{6b}$$

$$\frac{\mathbf{u} \cdot \mathbf{u}}{2} \left(\partial_t \rho + \mathbf{u} \cdot \nabla \rho \right) + \left[\partial_t (\rho e) + \mathbf{u} \cdot \nabla (\rho e) \right] = 0. \tag{6c}$$

With the interface-sharpening model (3) for the volume fraction in mind, it should be sensible to assume a variant model for the density as

$$\partial_t \rho + \mathbf{u} \cdot \nabla \rho = \frac{1}{\mu} \mathcal{D}_\rho, \tag{7a}$$

where the term $\mathcal{D}_\rho$ can be defined analogously based on either the interface-compression or the anti-diffusion formulation. Having that, to ensure the velocity remains at a constant state across the interfaces Eq. (6b) should be modified by

$$\mathbf{u} \left(\partial_t \rho + \mathbf{u} \cdot \nabla \rho \right) = \frac{1}{\mu} \mathbf{u} \, \mathcal{D}_\rho. \tag{7b}$$

Furthermore, to ensure the pressure retains in equilibrium also, using the equation of state (5) and Eq. (6a), together with a proper smoothness assumption of the density, it is not difficult to show that Eq. (6c) should be modified by

$$\frac{\mathbf{u} \cdot \mathbf{u}}{2} \left(\partial_t \rho + \mathbf{u} \cdot \nabla \rho \right) + \left[\partial_t \left(\rho e \right) + \mathbf{u} \cdot \nabla \left(\rho e \right) \right] = \frac{1}{\mu} \left[\frac{\mathbf{u} \cdot \mathbf{u}}{2} + \partial_\rho (\rho e) \right] \mathcal{D}_\rho, \tag{7c}$$

where we have $\partial(\rho e)/\partial\rho = e_{\text{ref}} + \rho e'_{\text{ref}} - (\Gamma p'_{\text{ref}} + (p - p_{\text{ref}})\Gamma')/\Gamma^2$; $z' = dz/d\rho$ for $z = \Gamma$, p_{ref}, and e_{ref}.

Since in general we are interested in shock wave problems as well, we should apply the interface-sharpening terms described above only locally near the interfaces. For this reason, it is common to introduce an interface indicator, denoted by H_I, to the model so that it takes effect near the interfaces only, and has no effect on the other genuinely nonlinear shock and rarefaction waves (cf. [11]).

With that, in summary, the interface-sharpening model we propose to solve inviscid compressible single-phase flows with the Mie-Grüneisen equation of state (5) in Cartesian coordinates takes the form

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= \frac{1}{\mu} H_I \mathcal{D}_\rho, \\ \partial_t \left(\rho \mathbf{u} \right) + \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u} + p I_N \right) &= \frac{1}{\mu} H_I \mathcal{D}_{\rho \mathbf{u}}, \\ \partial_t E + \nabla \cdot \left(E \mathbf{u} + p \mathbf{u} \right) &= \frac{1}{\mu} H_I \mathcal{D}_E. \end{aligned} \tag{8}$$

Here, without causing any confusion, in Eq. (8) we have used the notations $\mathcal{D}_{\rho\mathbf{u}} := \mathbf{u}\mathcal{D}_\rho$, and $\mathcal{D}_E := (\mathbf{u} \cdot \mathbf{u}/2 - \partial(\rho e)/\partial\rho)\mathcal{D}_\rho$.

To end this section, for the ease of the latter discussion, it is useful to write (8) into a dimension-wise expression by

$$\partial_t q + \sum_{j=1}^{N} \partial_{x_j} f_j(q) = \frac{1}{\mu}\psi(q), \tag{9a}$$

with q, f_j, and ψ defined respectively by

$$q = (\rho, \rho u_1, \ldots, \rho u_N, E)^T, \tag{9b}$$

$$f_j = \left(\rho u_j, \rho u_1 u_j + p\delta_{j1}, \ldots, \rho u_N u_j + p\delta_{jN}, E u_j + p u_j\right)^T, \tag{9c}$$

$$\psi = H_I\left(\mathcal{D}_\rho, \mathcal{D}_{\rho u_1}, \ldots, \mathcal{D}_{\rho u_N}, \mathcal{D}_E\right)^T, \tag{9d}$$

where δ_{ij} is the Kronecker delta.

3 Numerical Methods

To approximate (9) numerically, a fractional step method that consists of the following steps in each time iteration is employed:

(1) Solve the model equation without interface-sharpening terms

$$\partial_t q + \sum_{j=1}^{N} \partial_{x_j} f_j(q) = 0 \tag{10a}$$

using a state-of-the-art shock-capturing method over a time step Δt.

(2) Iterate the equation with the interface-sharpening terms

$$\partial_\tau q = \psi(q) \tag{10b}$$

using a simple explicit method, over a time step $\Delta\tau$ towards a "sharp layer"; $\tau = t/\mu$ is a scaled time variable.

Note that in step 1 we have used a standard high-resolution finite-volume method based on a wave-propagation viewpoint for the numerical approximation of Eq. (10a) (cf. [5]), and in step 2 we have employed an explicit first-order in time and second-order in space finite difference method for discretizing Eq. (10b) (cf. [7, 11, 14]). In this work, the local interface indicator H_I is defined as a Heaviside function of form

$$H_I(z) = \begin{cases} 1 & \text{if } z \geq z_0, \\ 0 & \text{otherwise,} \end{cases}$$

where the variable z can be taken as some measure of the physical quantities such as density, entropy, pressure, and velocity. Alternatively, it can be taken on an

augmented variable such as the volume fraction that is introduced to Eq. (8) for that matter. Here z_0 is some prescribed tolerance on z, see [11] for the other approach.

In the numerical results shown in Sect. 4, an anti-diffusion based model equation with $\mathcal{D}_\rho = -\nabla \cdot (\varepsilon \nabla \rho)$ is used in the model for approximation. Here the diffusion coefficient ε is chosen to be a function of the local velocity that varies both in space and time. To stabilize the computation of $\nabla \rho$ and so the flux $\varepsilon \nabla \rho$, the MINMOD limiter is imposed in step 2 of the method (cf. [2, 3]). As to the stopping criterion towards a "sharp layer", in practice, only 1 or 2 iterations are sufficient for the interface-sharpening purpose.

4 Numerical Examples

We now present sample results obtained using our interface-sharpening method with anti-diffusion based model equations in both one and two dimensions. Additional results that further validate the proposed method will be reported elsewhere.

Example 4.1. Our first test problem is the classical Sod shock tube problem in one dimension, where initially the state variables are $(\rho,\ u_1,\ p)_L = (1,\ 0,\ 1)$ and $(\rho,\ u_1,\ p)_R = (0.125,\ 0,\ 1)$, respectively. Here L is the state used for $x_1 \in [0, 0.5)$, and R is the state used for $x_1 \in [0.5, 1]$. The fluid inside the domain is gas modeled by the ideal gas equation of state $p(\rho, e) = (\gamma - 1)\rho e$ with $\gamma = 1.4$. There are non-reflecting outflow boundaries on the left and right sides.

In Fig. 1, we show interface-sharpening results for the density, velocity, and pressure at time $t = 0.15$ using a 100 grid. It is easy to see that our interface-sharpening algorithm works in a satisfactory matter on the interface without introducing any spurious oscillations in the pressure, while retaining the same solution structure in the region of shock and rarefaction waves.

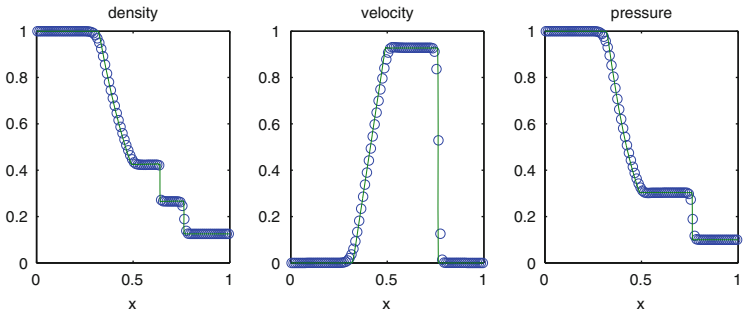

Fig. 1 Interface-sharpening result for the Sod shock tube problem at time $t = 0.15$ using a 100 grid. The *solid line* shown in the graph is the exact solution

Example 4.2. Our second example in one dimension is an impact problem in which a pre-compressed semi-infinite aluminum slab at rest with $(\rho,\ p) = (4{,}000\ \mathrm{kg/m^3},\ 7.93 \times 10^9\ \mathrm{Pa})$ is being hit by an ambient aluminum slab traveling at the speed 2 km/s from the right to the left with the reference state $(\rho,\ p) = (\rho_0,\ p_0)$. We assume that the constitutive law of an aluminum satisfies the Mie-Grüneisen equation of state (5) with Γ, p_{ref}, and e_{ref} defined by

$$\Gamma(\rho) = \Gamma_0(1-\eta)^{\alpha}, \quad p_{\mathrm{ref}}(\rho) = \frac{\rho_0 c_0^2 \eta}{(1-s\eta)^2}, \quad e_{\mathrm{ref}}(\rho) = \frac{\eta}{2\rho_0}\left(p_0 + p_{\mathrm{ref}}(\rho)\right),$$

where the numerical values of the material constants are taken to be $\rho_0 = 2{,}785\,\mathrm{kg/m^3}$, $p_0 = 0$, $c_0 = 5{,}328\,\mathrm{m/s}$, $s = 1.338$, $\Gamma_0 = 2$, and $\alpha = 1$; $\eta = 1 - \rho_0/\rho$.

In this setup, it is not difficult to show that the exact solution of this problem would consist of a leftward going shock wave to the stationary aluminum, a material interface, and a rightward going shock wave to the moving aluminum. Figure 2 shows results with and without interface-sharpening at time $t = 50\,\mu\mathrm{s}$ using a 200 grid. From the figure, we observe a slight improvement on the interface structure, see [12] also for a similar calculation.

Example 4.3. We are next concerned with a passive evolution of a two-dimensional square column of size $(x_1, x_2) \in [0.3, 0.7] \times [0.3, 0.7]\,\mathrm{m^2}$ in a unit square domain with uniform equilibrium pressure $p = 10^5\,\mathrm{Pa}$ and constant particle velocity $(u_1,\ u_2) = (10^2\ \mathrm{m/s},\ 10^2\ \mathrm{m/s})$. In this test, the density in the region inside a square column is $1{,}500\,\mathrm{kg/m^3}$, and it is $1{,}000\,\mathrm{kg/m^3}$ otherwise. We use the linearized Mie-Grüneisen equation of state $p(\rho, e) = (\gamma - 1)\,\rho e + c_0^2\,(\rho - \rho_0)$ to model the material in the whole domain with the material-dependent quantities taken to be $\gamma = 4.4$, $\rho_0 = 1{,}000\,\mathrm{kg/m^3}$, $c_0 = 1{,}624\,\mathrm{m/s}$. Figure 3 shows contour plots of the density obtained using the method with and without interface-sharpening at time $t = 0.02\,\mathrm{s}$ using a 100×100 grid. An excellent interface-sharpening result

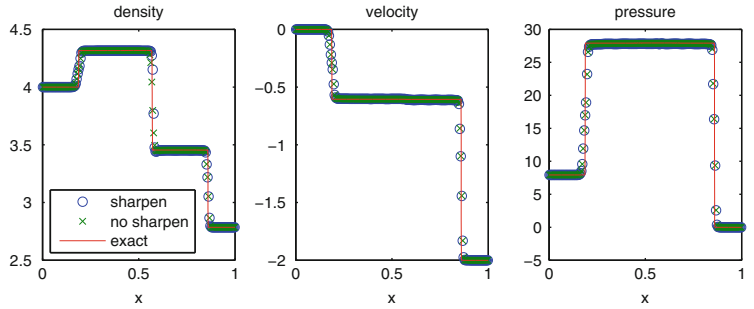

Fig. 2 Interface-sharpening result for the aluminum impact problem at time $t = 50\,\mu\mathrm{s}$ using 200 grids. The *solid line* shown in the graph is the exact solution and the symbol "×" is the result obtained using the method without interface-sharpening

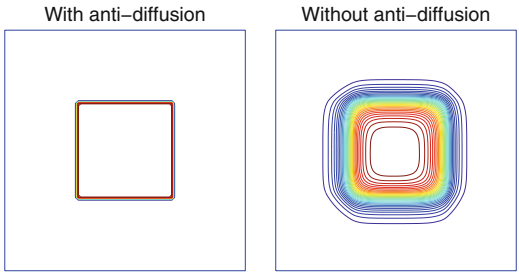

Fig. 3 Numerical results for a passive evolution of a square column. Contours of the density are shown at time $t = 0.02$ s obtained using the method with and without interface-sharpening on a 100×100 grid

is observed, whereas a severely diffused interface is seen using the standard high-resolution method. Here periodic boundary conditions are used on all sides.

Example 4.4. Our second example in two dimensions is a Mach 3 shock wave in air interacting with a heavier circular gas column. In this test, we take a shock tube of size $(x_1, x_2) \in [-1, 1] \times [0, 0.5]\,\mathrm{m}^2$, and consider a planarly leftward-moving shock wave with initial position $x_1 = 0.7$ m and states in the pre- and post-shock as

$$(\rho,\ u_1,\ u_2,\ p)_{\text{pre-shock}} = \left(1\ \mathrm{kg/m^3},\ 0,\ 0,\ 10^5\ \mathrm{Pa}\right),$$

$$(\rho,\ u_1,\ u_2,\ p)_{\text{post-shock}} = \left(3.857\ \mathrm{kg/m^3},\ -831.479\ \mathrm{m/s},\ 0,\ 1.033 \times 10^6\ \mathrm{Pa}\right),$$

respectively. In addition to that, we assume a stationary heavier circular gas column with radius 0.2 m and center $(0.4, 0)$ m lying in front of the shock. Inside the gas column the flow is in standard atmospheric condition with density $\rho = 10\,\mathrm{kg/m^3}$; this gives us one example that the interface is accelerated by a shock wave coming from the light-fluid to the heavy-fluid region, yielding a transmitted shock wave, an interface, and a reflected shock after the interaction. As in Example 4.1, the fluid under consideration is an ideal gas with $\gamma = 1.4$.

In Fig. 4, we show schlieren images of density obtained using the method with and without interface-sharpening at six different times $t = 2^i \times 10^{-1}$ ms for $i = 1, 2, \ldots, 6$, with a 800×200 grid. It is interesting to see that as far as the global wave structure (i.e., the shape and location of the incident, transmitted and reflected waves) is concerned, we observe similar behavior of the solutions between those two. However, a sharper resolution of the contact line is observed when our anti-diffusion method is in use.

Example 4.5. Finally, we are interested in a model blast wave problem with complex wave interactions and a general equation of state. As initial condition, we have a stationary circular gaseous explosive charge of radius 0.1 m and center $(0, 0.25)$ m located in a rectangular domain $(x_1, x_2) \in [-1, 1] \times [0, 1]\,\mathrm{m}^2$. Inside the circular region, the density and pressure are $\rho = 1{,}700\ \mathrm{kg/m^3}$ and $p = 10^{12}$ Pa, while outside the circular region, we have $\rho = 1{,}000\,\mathrm{kg/m^3}$ and $p = 5 \times 10^{10}$ Pa. The material in the entire domain is modeled by the Jones-Wilkins-Lee equation of state for gaseous explosives (cf. [4]) in that it takes the form (5) with

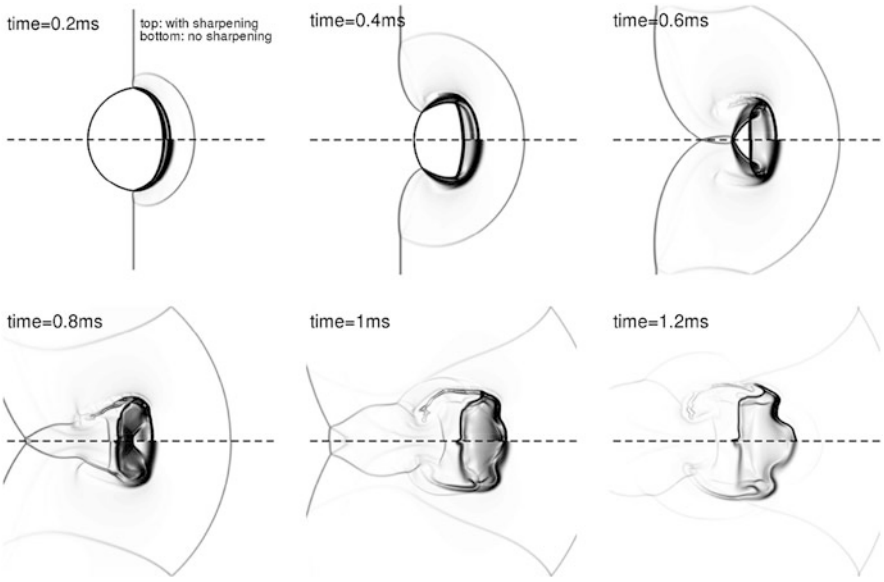

Fig. 4 Numerical results for a Mach 3 shock wave in air interacting with a heavier circular gas column. Schlieren images of density are shown at six different times $t = 2^i \times 10^{-1}$ ms for $i = 1, 2, \ldots, 6$, obtained using the method with and without interface-sharpening (drawn on the *top* and *bottom* parts of each graph, respectively) with a 800×200 grid

$$\Gamma(V) = \Gamma_0, \qquad p_{\text{ref}}(V) = \mathcal{A}\, \exp\left(\frac{-\mathcal{R}_1 V}{V_0}\right) + \mathcal{B}\, \exp\left(\frac{-\mathcal{R}_2 V}{V_0}\right),$$
$$e_{\text{ref}}(V) = \frac{\mathcal{A}\, V_0}{\mathcal{R}_1} \exp\left(\frac{-\mathcal{R}_1 V}{V_0}\right) + \frac{\mathcal{B}\, V_0}{\mathcal{R}_2} \exp\left(\frac{-\mathcal{R}_2 V}{V_0}\right) - e_0,$$

where $V = 1/\rho$ is the specific volume. The material-dependent quantities we use in the simulations are $\Gamma_0 = 0.28$, $\rho_0 = 1{,}640\,\text{kg/m}^3$, $e_0 = 0$, $\mathcal{A} = 494\,\text{GPa}$, $\mathcal{B} = 1.21\,\text{GPa}$, $\mathcal{R}_1 = 4.94$, and $\mathcal{R}_2 = 1.21$. The boundary conditions are solid walls on the top- and bottom-side, and non-reflecting on the left- and right-side.

In this problem, due to the pressure difference, breaking of the circular membrane occurs instantaneously, yielding an outward-going shock wave, an inward-going rarefaction wave, and a contact discontinuity lying in between. At a later time, this outward-going shock wave is reflected from the bottom wall, and so the inward-going rarefaction is bounced back from the explosive center; this generates complex wave interactions afterwards.

Figure 5 shows schlieren images of density at three different times $t = 18$, 36, and $72\,\mu\text{s}$. Here we have performed the computations using both the anti-diffusion based interface-sharpening method with a 400×200 grid, and also the local adaptive mesh refinement (AMR) version of the method without anti-diffusion (cf. [6]). In

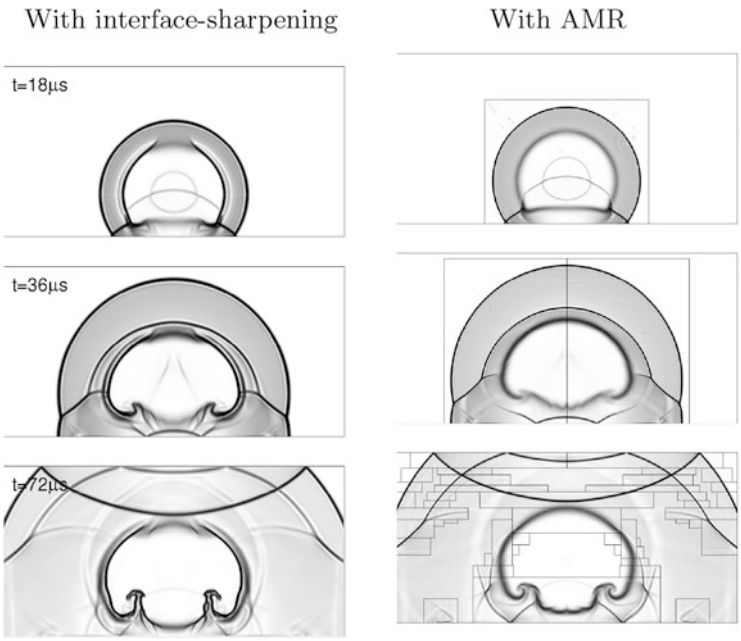

Fig. 5 Numerical results for a blast wave problem in two dimensions. Schlieren images of density are shown at three different times $t = 18, 36, 72\,\mu$s obtained using the methods with anti-diffusion (*the left column*) and with local adaptive mesh refinement (*the right column*)

the later AMR runs, the base grid used here is 200×100 and with a two-level of grid refinement; the refinement ratio is 4 for each level of grid, i.e., in the refined region the mesh size is twice smaller than in the anti-diffusion runs. From the figure, we observe the same qualitative structure of solution between them, especially on the structure of shock waves; this is as expected because for the Euler equations without any source terms the shock wave is stable under mesh refinement. This is not the case, however, for interfaces (contact discontinuities); a difference in the solution then occurs due to the perturbation of complex wave interactions upon them.

5 Conclusion

We have described a class of interface-sharpening methods for single-phase compressible flow with interfaces. Numerical validation of the proposed methods using an anti-diffusion based model system has been performed. It shows the feasibility of the algorithm for sharpening compressible interfaces numerically in one and two dimensions. Ongoing work is to validate the method using the interface-compression based model, and extend the method to compressible multiphase flow and to mapped grids with complex geometries.

Acknowledgements This work was supported in part by National Science Council of Taiwan Grants #96-2115-M-002-008-MY3 and 99-2115-M-002-005-MY2.

References

1. Boris, J.P., Book, D.L.: Flux-corrected transport I. SHASTA, a fluid transport algorithm that works. J. Comput. Phys. **11**, 38–69 (1973)
2. Breuß, M., Welk, M.: Staircasing in semidiscrete stabilized inverse linear diffusion algorithms. J. Comput. Appl. Math. **206**, 520–533 (2007)
3. Breuß, M., Brox, T., Sonar, T., Weickert, J.: Stabilized nonlinear inverse diffusion for approximating hyperbolic PDEs. In: Kimmel, R., Sochen, N., Weickert, J. (Eds.) Proceedings Scale Space 2005, Springer LNCS 3459, Hofgeismar, pp. 536–547. Springer (2005)
4. Dobratz, B.M., Crawford, P.C.: LLNL Explosive handbook: properties of chemical explosives and explosive simulants. UCRL-52997, LLNL (1985)
5. LeVeque, R.J.: Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge/New York (2002)
6. LeVeque, R.L.: Conservation law package (clawpack), 2003. Available at the http://depts.washington.edu/clawpack
7. LeVeque, R.J.: Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. SIAM, Philadelphia (2007)
8. Noh, W.F., Woodward, P.: SLIC (simple line interface calculation). In: van de Vooren, A.I., Zandbergen, P.J. (Eds.) Proceedings of 5th International Conference on Numerical Methods in Fluid Dynamics, Enschede. Springer, Berlin/Heidelberg (1976)
9. Olsson, E., Kreiss, G.: A conservative level set method for two phase flow. J. Comput. Phys. **210**, 225–246 (2005)
10. Olsson, E., Kreiss, G., Zahedi, S.: A conservative level set method for two phase flow II. J. Comput. Phys. **225**, 785–807 (2007)
11. Shukla, R.K., Pantano, C., Freund, J.B.: An interface capturing method for the simulation of multi-phase compressible flows. J. Comput. Phys. **229**, 7411–7439 (2010)
12. Shyue, K.-M.: A fluid-mixture type algorithm for compressible multicomponent flow with Mie-Grüneisen equation of state. J. Comput. Phys. **171**, 678–707 (2001)
13. So, K.K., Hu, X.Y., Adams, N.A.: Anti-diffusion method for interface steepening in two-phase incompressible flow. J. Comput. Phys. **230**, 5155–5177 (2011)
14. So, K.K., Hu, X.Y., Adams, N.A.: Anti-diffusion interface sharpening technique for two-phase compressible flow simulations. J. Comput. Phys. **231**, 4304–4323 (2012)
15. Štrubelj, L., Tiselj, I.: Two-fluid model with interface sharpening. Int. J. Numer. Methods Eng. **85**, 575–590 (2011)
16. Ubbink, O., Issa, R.I.: A method for capturing sharp fluid interfaces on arbitrary meshes. J. Comput. Phys. **153**, 26–50 (1999)
17. Xiao, F., Honma, Y., Kono, T.: A simple algebraic interface capturing scheme using hyperbolic tangent function. Int. J. Numer. Mech. Fluids **48**, 1023–1040 (2005)

Numerical Simulation of the Damping Behavior of Particle-Filled Hollow Spheres

Tobias Steinle, Jadran Vrabec, and Andrea Walther

Abstract In light of an increasing awareness of environmental challenges, extensive research is underway to develop new light-weight materials. A problem arising with these materials is their increased response to vibration. This can be solved using a new composite material that contains embedded hollow spheres that are partially filled with particles. Progress on the adaptation of molecular dynamics towards a particle-based numerical simulation of this material is reported. This includes the treatment of specific boundary conditions and the adaption of the force computation. First results are presented that showcase the damping properties of such particle-filled spheres in a bouncing experiment.

1 Motivation

The quality of industrial machines suffers because of vibration due to their operation. To improve the quality of such products, it is important to find possibilities to suppress this unwanted side-effect. Adding massive material to the assembly is the classical way to achieve this. However, to reduce manufacturing costs and the impact on the environment (for instance, by saving energy), it is also desirable to construct machines that are as lightweight as possible. Therefore, an extensive effort in research of lightweight materials is underway. The Fraunhofer Institute of Advanced Manufacturing Technology and Materials (IFAM) in Dresden studies and manufactures structures constructed from hollow spheres. These structures can be used, e.g., in a sandwich configuration within the casings of machines, cf. Fig. 1.

To combine lightweight materials with vibration damping, research is now being focused on particle-filled hollow sphere structures, cf. Fig. 2. The inclusion of a

T. Steinle • A. Walther (✉)
Institute of Mathematics, University of Paderborn, Paderborn, Germany
e-mail: tobias.steinle@upb.de; andrea.walther@upb.de

J. Vrabec
Thermodynamics and Energy Technology, University of Paderborn, Paderborn, Germany
e-mail: jadran.vrabec@upb.de

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_19

ceramic powder inside the hollow spheres leads to a high damping [6]. Such a material possesses several properties that make its application very attractive. For instance, because of the metallic nature of these spheres, they are resistant to solvents and because of their small size, as an ensemble they can easily be adapted to a broad variety of shapes. To couple the spheres, different techniques can be employed, for instance, using glue, solder or embedding them into a matrix.

As a measure for the damping, the time between two bounces of a single sphere is used. The IFAM has an experimental test setup for this purpose that records the sound of the collisions with a fundament. Obtaining a numerical simulation to study the damping influence of the particles in this bouncing experiment is the goal of this work. This would allow for parameter studies, for example with respect to the optimal size or number of particles within the sphere. First simulations were conducted by Blase [1] for the two-dimensional case. She used a collision detection approach such that the system temporally evolved from collision to collision between the hull of the hollow sphere and a particle or between two particles. As

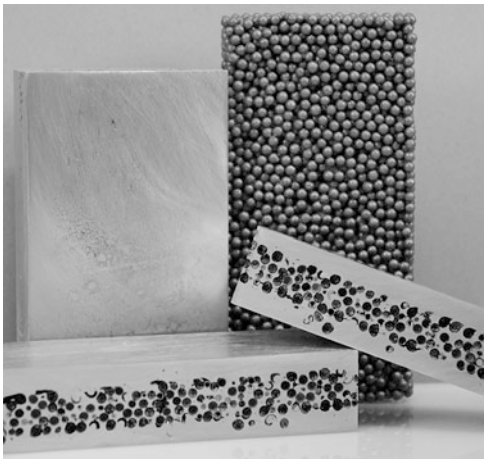

Fig. 1 Hollow sphere structures (©Fraunhofer IFAM-Dresden)

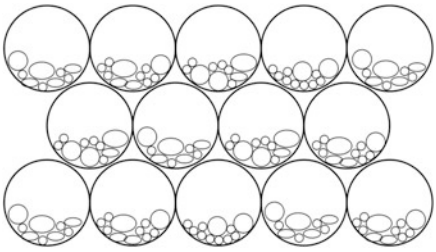

Fig. 2 Schematic cross-section

the number of particles increases, this detection scheme breaks down due to the numerical effort. It has to be noted that within a single sphere, up to $2 \cdot 10^5$ particles have to be considered. Therefore, the high complexity of this approach requires a different method when expanding the simulation to the three-dimensional case. This alternative approach should thus be based on a time integration method.

2 Molecular Dynamics

For the simulation of the dynamic behavior of the filled spheres, the tracking of the translational and rotational movement of all enclosed particles is required. To achieve this, a variety of methods is available. One of them is molecular dynamics (MD), which typically considers the trajectories of molecules on the nanoscopic scale. MD simulations are able to track trillions of molecules [2]. Based on a program that is available at the University of Paderborn and was originally developed by Mader [7], an adaption to the present case is underway.

To compute the translational propagation of a particle i, the Newtonian equation of motion has to be considered

$$\ddot{\mathbf{x}}_i = m_i^{-1}\mathbf{F}_i \ ,$$

where $\mathbf{x}_i$ is the position of the particle in space, and $\mathbf{F}_i$ and m_i denote the force acting on the particle and its mass, respectively. Rewriting this second order ordinary differential equation (ODE) as a system of first order ODE leads to

$$\dot{\mathbf{v}}_i = \mathbf{a}_i = m_i^{-1}\mathbf{F}_i \ , \quad \dot{\mathbf{x}}_i = \mathbf{v}_i \ .$$

Here, $\mathbf{v}_i$ is the velocity and $\mathbf{a}_i$ the acceleration of the particle i. Using a Taylor expansion, these equations can be numerically solved over time using an explicit Leapfrog scheme [4]

$$\mathbf{v}_i^{n+\frac{1}{2}} = \mathbf{v}_i^{n-\frac{1}{2}} + dt\, m_i^{-1}\mathbf{F}_i^n, \tag{1}$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + dt\, \mathbf{v}_i^{n+\frac{1}{2}} \ . \tag{2}$$

Note that the solutions for the velocities and positions of the particles are obtained alternatingly at each half time-step, giving the Leapfrog scheme its name.

Analogously to the translational movement, for the rotational movement of particle i, the angular velocity and the orientation $\boldsymbol{\theta}_i$ in space are needed. The

angular velocity $\boldsymbol{\omega}_i$ is defined in terms of the angular momentum $\mathbf{j}_i$ and the moment of inertia tensor $\mathbf{I}_i$

$$\boldsymbol{\omega}_i = \dot{\boldsymbol{\theta}}_i = \mathbf{I}_i^{-1}\mathbf{j}_i \ . \tag{3}$$

The torque $\boldsymbol{\tau}_i$ is the rate of change of the angular momentum

$$\boldsymbol{\tau}_i = \frac{d\,\mathbf{j}_i}{dt} \ . \tag{4}$$

Subsequently, Taylor expansions of Eqs. (3) and (4) can be integrated to obtain the desired physical properties. For the orientation $\mathbf{q}_i$ in space, Fincham's explicit rotational quaternion algorithm [3] was used that reduces the complexity of this calculation by saving one vector product evaluation. For the main part of the algorithm, Fincham proposed a quaternion representation matrix $Q(\mathbf{q}_i)$ and a modified angular velocity $\mathbf{w}_i$ to obtain the spatial orientation. The change of the orientation over time is

$$\frac{d\,\mathbf{q}_i}{dt} = Q(\mathbf{q}_i)\mathbf{w}_i \ .$$

On that basis, $\mathbf{q}_i$ can be obtained with a Taylor expansion. The main equations of the rotational leapfrog scheme are

$$\begin{aligned}
\mathbf{j}_i^{n+\frac{1}{2}} &= \mathbf{j}_i^{n-\frac{1}{2}} + \boldsymbol{\tau}_i^n \\
\mathbf{q}_i^{n+1} &= \mathbf{q}_i^n + \frac{dt}{2} Q(\mathbf{q}_i^{n+\frac{1}{2}})\mathbf{w}_i^{n+\frac{1}{2}} \ .
\end{aligned}$$

Combined with Eqs. (1) and (2), this then gives a full representation of the translational and rotational movement of a particle over time.

The forces and torques acting on the particles in are based on a pair-potential U which describes the force that one particle exerts on another particle as a function of their mutual distance. The force is given by the derivative of the potential with respect to their distance r_{ij}

$$\mathbf{F}_{ij} = -\frac{dU}{d\,\mathbf{r}_{ij}} \ .$$

A popular interaction potential used in MD is the Lennard-Jones 12-6 potential

$$U(r_{ij}) = 4\varepsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right) \ , \tag{5}$$

where σ denotes the size parameter of the considered particle and ε is the depth of the potential well (cf. Fig. 3).

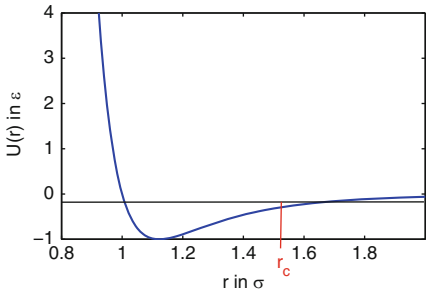

Fig. 3 Lennard-Jones potential

During simulation, the inter-particle potential (5) is only evaluated up to a certain cut-off distance r_c around each particle, because particles that are far apart from each other interact only weakly. Additionally, cutting off the potential saves computational effort. Due to the steep slope of the potential, particles that are very close to each other will exert a large repulsive force, while particles further apart will attract each other. This behavior models real-world atomistic forces (repulsion by electronic overlap and attraction by dispersion) reasonably well.

3 Adapting Molecular Dynamics to Particle Dynamics

While MD methods are well suited for simulating the behavior of molecules on the nanoscopic scale, several adjustments have to be made for the present application where mesoscopic particles are considered. For instance, the periodic boundary conditions, that are usually assumed in MD, have to be changed to reflective boundary conditions for the interactions of the particles with the hollow sphere. In this work, two approaches were followed.

The first approach uses the conservation law of momentum to derive the velocities of the particles $\hat{\mathbf{v}}_p$ and the hollow sphere $\hat{\mathbf{v}}_s$ after their eventual collision

$$m_p|\mathbf{v}_p| + m_s|\mathbf{v}_s| = \hat{m}_p|\hat{\mathbf{v}}_p| + \hat{m}_s|\hat{\mathbf{v}}_s| \ . \tag{6}$$

For each time step, the masses and velocities of particles that are reflected by the hull are accumulated in a substitute particle with corresponding total mass m_m and velocity $\mathbf{v}_m$. Combining the definition of the coefficient of restitution (COR)

$$\varepsilon = -\frac{|\hat{\mathbf{v}}_p| - |\hat{\mathbf{v}}_s|}{|\mathbf{v}_p| - |\mathbf{v}_s|} \ ,$$

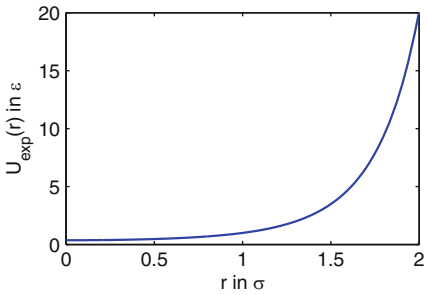

Fig. 4 Exponential barrier potential

with Eq. (6), the solution for the velocities of the reflected particles and the hull after central inelastic collisions are obtained by

$$\begin{aligned} \hat{\mathbf{v}}_s &= \mathbf{v}_s + \tfrac{m_m}{m_s+m_m}(1+\varepsilon)(\mathbf{v}_m - \mathbf{v}_s), \\ \hat{\mathbf{v}}_p &= \mathbf{v}_p + \tfrac{m_s}{m_s+m_p}(1+\varepsilon)(\mathbf{v}_s - \mathbf{v}_p)\ . \end{aligned} \tag{7}$$

To account for the shape of the sphere, the resulting velocities of the particles are subjected to a reflection with the tangential plane at the point of impact. If $\mathbf{n}$ is the normal vector of that tangential plane and $\hat{\mathbf{v}}_p$ the velocity of the particle after the reflection, the new velocity vector $\hat{\mathbf{v}}_{pr}$ is obtained by rotating $\hat{\mathbf{v}}_p$ around $\mathbf{n}$ by 180°. Therefore, the resulting velocity of the skewed collision is

$$\hat{\mathbf{v}}_{pr} = \hat{\mathbf{v}}_p - 2\frac{\hat{\mathbf{v}}_p \cdot \mathbf{n}}{\|\mathbf{n}\|^2}\mathbf{n}\ ,$$

with $\hat{\mathbf{v}}_p$ according to Eq. (7). If the COR is known, this gives an exact computation of the reflection. However, this discrete mechanical approach disrupts the otherwise continuous, potential-based nature of MD.

The second approach thus uses an interaction potential between the particles and for the spheric hull. As particles approach the hull of the sphere, the forces arising from the potential are evaluated. To prevent particles from exiting the sphere, an exponential barrier potential was used

$$U_{exp}(r) = \delta \cdot (\exp(r) - r), \tag{8}$$

where δ controls the force of the interaction and r is the distance between the particle and the hull of the sphere (cf. Fig. 4). For the collisions between particles, the LJ potential was used. To avoid attracting forces, the cut-off radius was selected as

$$r_c = 2^{\frac{1}{6}}\sigma.$$

Fig. 5 Particle consisting of eight spheres

In this way, the potential is 0 when particles have a distance greater than r_c where it would normally be attracting. Also, the force reaches 0 at r_c while staying continuously differentiable. This homogenizes the overall approach to use only continuous potentials for all interactions.

The particles that are filled into the hollow sphere are manufactured using a complex process during which they assume different shapes and sizes. Common shapes include cylinders and spheres as well as particles with rugged edges and hooks. In MD, the same variation has to be considered, when simulating different molecule species. There, molecules are assembled from several spheres in a molecule-fixed coordinate system. This approach can be used for the particles considered here as well. Different shapes can be built from elementary spherical shaped parts and the number of each particle type can be specified as input for the algorithm (cf. Fig. 5).

For the initial configuration, instead of the regular lattice often used in MD, the particles were aligned in a spherical shape using spherical coordinates such that they do not overlap. The particle types were then randomly distributed throughout the initial configuration. In the present simulation, the particles were dropped first in the spherical hull and were given time to settle at the bottom. After this preparation, the particle-filled sphere was dropped towards the fundament. Thus, for the damping behavior, the initial setup was irrelevant. Figure 6 shows the initial configuration of $2 \cdot 10^4$ particles using the visualization tool MegaMol of the VISUS group at the University of Stuttgart [5].

4 Results

To study the damping behavior of the particle-filled spheres, the experimental setup of the IFAM was modeled in the computer. A sphere was dropped from a certain height h_0. To measure the damping, the time interval Δt between the first two bounces was sampled. The comparison with the bouncing time interval of an empty sphere gives a good indication of the damping due to the particles. According to

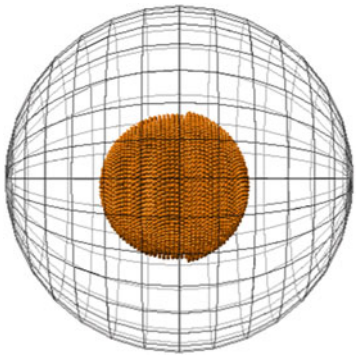

Fig. 6 Initial configuration with $2 \cdot 10^4$ particles

Jehring et al. [6], the resulting COR of the system can then be obtained using the equations of the vertical throw. It evaluates as

$$\varepsilon_r = \sqrt{\frac{g}{8h_0}\Delta t^2}\,,$$

where g is the gravitational acceleration. It was aimed at a comparison of this experiment with the present numerical simulations. A single hollow sphere with a diameter of 3 mm and a mass of 12.518 mg was considered that was dropped onto a fundament from a height of 1.5 mm. The spherical particles in the hollow sphere had a uniform diameter of 31 μm and a mass of 41 ng. For the reflection of the sphere on the surface, a barrier potential similar to Eq. (8) was used. The sphere was treated as a particle approaching the fundament with the imposed barrier potential for the interaction and rebound. The cut-off radius was set to 35 μm. The numerical results were obtained on the OCuLUS computer at the Paderborn Center for Parallel Computing (PC^2). Table 1 lists the preliminary results that were obtained with a time step of 1 μs. Figure 7 shows the results for a variety of different time steps. For too large time steps, such as 5 μs, the simulation leads to false results due to inadequate numerical integration of the equations of motion. However, the choice of the time step size has no significant influence on the damping behavior as soon as it is small enough to discretize the particle motions and a reasonably large number of particles is used in the experiment.

As can be seen, the number of particles has a large effect on the damping. It is expected from the physical experiments that as the number of particles increases, reducing Δt further at first, the damping properties of the filled hollow sphere will eventually start to deteriorate and approach the behavior of a solid sphere. Quantifying the number of particles when this happens is of great interest.

Table 1 Time interval Δt between the first and second bounce on the fundament

Particles	Δt (ms)	ε_r	Particles	Δt (ms)	ε_r
50	34.66	0.3134	2,500	34.33	0.3104
100	34.66	0.3134	5,000	34.02	0.3076
250	34.54	0.3123	7,500	32.92	0.2977
500	34.56	0.3125	10,000	32.41	0.2930
750	34.52	0.3121	12,500	31.97	0.2891
1,000	34.62	0.3130	15,000	31.62	0.2859
1,500	34.46	0.3115	17,500	31.35	0.2835
2,000	34.35	0.3106	20,000	30.76	0.2781

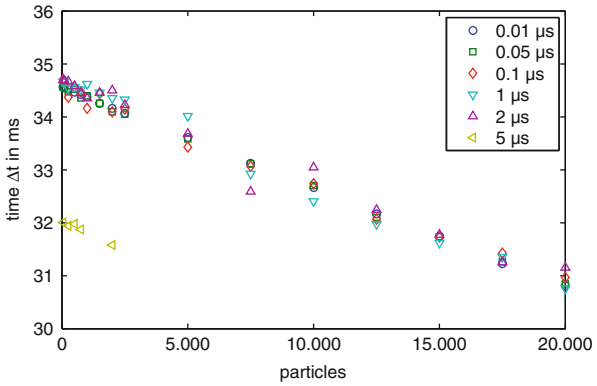

Fig. 7 Simulation results for different time step lengths

5 Future Work

As can be seen from the computation times in Fig. 8, the complexity of the standard MD algorithm is $O(N^2)$, where N is the number of particles. Therefore, when larger particle numbers have to be considered, adaptations must be made to achieve reasonable computing times. One approach is the linked cell algorithm [4]. This algorithm retains the basic structure of the MD algorithm, but makes a crucial adaptation in the force computation algorithm. By dividing the simulation volume into equally sized cells when computing the forces that act on a certain particle, only particles in the same cell and in neighboring cells have to be considered. This reduces the complexity to $O(N)$ [4]. However, the linked cell algorithm works best when the particles are evenly distributed throughout the simulation volume. In the present case, the particles tend to accumulate over time at the bottom of the hollow sphere, which results in a few cells that are crowded and many cells that do not contain any particles. To cope with this aggregation, the cells need to adapt to that

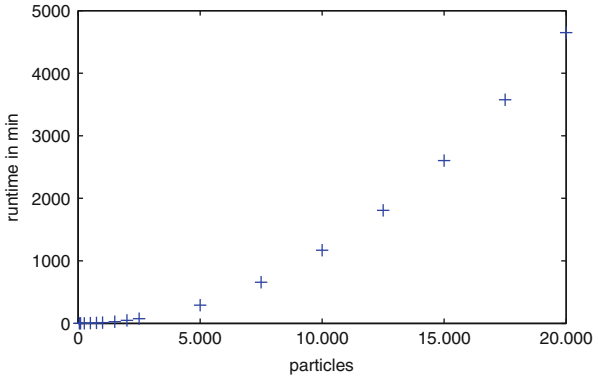

Fig. 8 Computation times for 400,000 steps and $dt = 1\,\mu s$

situation to retain the efficiency of the algorithm. This will be the subject of future work.

Increasing the particle number will allow to directly compare results from numerical simulation to physical experiments. To simulate the material on a larger scale, a significant number of filled spheres must be considered. The simulation therefore has to be extended in this respect. The nature of coupling between the different parts needs to be evaluated carefully as different types of assembly can be manufactured, e.g. by gluing or soldering. Deformations of the spheres occurring in the material may have to be considered as well. Natural parallelization by assigning a single sphere to each process should yield reasonable computing times for the numerical simulation of this complex material when combined with the linked cell algorithm.

6 Conclusion

Progress on the simulation of particle-filled spheres by adaption of molecular dynamics was presented. Preliminary results show the effect of an increasing number of particles on the damping of a filled hollow sphere. The dramatic effect of the particles seen in the physical experiment can also be seen in the numerical simulation. The complexity of the algorithm has to be improved, e.g., by an adapted linked cell algorithm, to allow for larger particle numbers. The modeling of friction between particles also plays an important role and there are several approaches that have to be evaluated.

References

1. Blase, D.: Simulation partikelgefüllter Hohlkugeln in zwei Raumdimensionen. Diploma thesis, TU Dresden (2008)
2. Eckhardt, W., Heinecke, A., Bader, R., Brehm, M., Hammer, N., Huber, H., Kleinhenz, H.-G., Vrabec, J., Hasse, H., Horsch, M., Bernreuther, M., Glass, C.W., Niethammer, C., Bode, A., Bungartz, H.-J.: 591 TFLOPS multi-trillion particles simulation on SuperMUC. In: Kunkel, J.M., Ludwig, T., Meuer, H.W. (eds.) Supercomputing: 28th International Supercomputing Conference, ISC 2013, Leipzig, Germany, June 16-20, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7905, pp. 1–12. Springer, Berlin/Heidelberg (2013)
3. Fincham, D.: Leapfrog rotational algorithms. Molec. Sim. **8**, 165–178 (1992)
4. Griebel, M., Knapek, S., Zumbusch, G., Caglar, A.: Numerische Simulation in der Moleküldynamik. Springer, Berlin/Heidelberg (2004)
5. Grottel, S., Reina, G., Dachsbacher, C., Ertl, T.: Coherent culling and shading for large molecular dynamics visualization. Comput. Graph. Forum (Proc. EUROVIS 2010) **29**(3), 953–962 (2010)
6. Jehring, U., Kieback, B., Stephani, G., Quadbeck, P., Courtois, J., Hahn, K., Blase, D., Walther, A.: Lightweight-materials made from particle filled metal hollow spheres. In: Proceedings Metfoam, Bratislava (2009)
7. Mader, D.: Molekulardynamische Simulationen nanoskaliger Strömungsvorgänge. Diploma thesis, University Stuttgart (2004)

FSSP Algorithms for Square and Rectangular Arrays

Hiroshi Umeo

Abstract The synchronization in cellular automata has been known as firing squad synchronization problem (FSSP) since its development. The firing squad synchronization problem on cellular automata has been studied extensively for more than fifty years, and a rich variety of synchronization algorithms has been proposed not only for one-dimensional arrays but also for two-dimensional arrays. In the present paper, we focus our attention to the two-dimensional array synchronizers that can synchronize any square/rectangle arrays and construct a survey on recent developments in their designs and implementations of optimum-time and non-optimum-time synchronization algorithms for two-dimensional arrays.

1 Introduction

Synchronization of large scale networks is an important and fundamental computing primitive in parallel and distributed systems. We study a synchronization problem that gives a finite-state protocol for synchronizing cellular automata. The synchronization in cellular automata has been known as firing squad synchronization problem (FSSP) since its development, in which it was originally proposed by J. Myhill in Moore [8] to synchronize all parts of self-reproducing cellular automata. The problem has been studied extensively for more than 50 years [1–28].

In the present paper, we focus our attention to two-dimensional (2D) array synchronizers that can synchronize square/rectangle arrays and construct a survey on recent developments in designs and implementations of optimum-time and non-optimum-time synchronization algorithms for the two-dimensional arrays. Specifically, we attempt to consider the following questions:

- Is there any new 2D FSSP algorithm other than classical ones?
- What is the smallest 2D synchronizer?

H. Umeo (✉)
University of Osaka Electro-Communication,
Neyagawa-shi, Hastu-cho, 18-8, Osaka, 572-8530, Japan
e-mail: umeo@cyt.osakac.ac.jp

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_20

- How can we synchronize 2D arrays with the general at any position?
- How do the algorithms compare with each other?
- Can we extend the 2D synchronizers proposed so far to three-dimensional arrays?

Generally speaking, in the design of 2D synchronizers, configurations of a one-dimensional synchronization algorithm are mapped onto a 2D array through a mapping scheme so that all of the cells on the 2D array would fall into a final synchronization state simultaneously. The mapping schemes we consider include a rotated L-shaped mapping, a zebra mapping, a diagonal mapping, and a one-sided recursive-halving marking based mapping. All mappings will be employed efficiently in the design of 2D FSSP algorithms for square and rectangle arrays. Due to the space available we omit the details of those algorithms and their implementations.

2 Firing Squad Synchronization Problem

2.1 FSSP on 1D Arrays

The firing squad synchronization problem (FSSP, for short) is formalized in terms of the model of cellular automata. Consider a one-dimensional (1D) array of finite state automata. All cells (except the end cells) are identical finite state automata. The array operates in lock-step mode such that the next state of each cell (except the end cells) is determined by both its own present state and the present states of its right and left neighbors. Thus, we assume the nearest left and right neighbors. All cells (*soldiers*), except one *general* cell, are initially in the *quiescent* state at time $t = 0$ and have the property whereby the next state of a quiescent cell having quiescent neighbors is the quiescent state. At time $t = 0$ the *general* cell is in the *fire-when-ready* state, which is an initiation signal to the array.

The FSSP is stated as follows: Given a 1D array of n identical cellular automata, including a *general* at one end that is activated at time $t = 0$, we want to design the automata $M = (\mathcal{Q}, \delta)$ such that, *at some future time*, all the cells will *simultaneously* and, *for the first time*, enter a special *firing* state, where $\mathcal{Q}$ is a finite state set and $\delta : \mathcal{Q}^3 \rightarrow \mathcal{Q}$ is a next-state function. The tricky part of the problem is that the same kind of soldier having a fixed number of states must be synchronized, regardless of length n of the array. The set of states and next state function must be independent of n. Figure 1 is a space-time diagram for the optimum-step firing squad synchronization algorithm. The general at left end emits at time $t = 0$ an infinite number of signals which propagate at $1/(2^{k+1}-1)$ speed, where k is positive integer. These signals meet with a reflected signal at half point, quarter points, ..., etc., denoted by $\odot$ in Fig. 1. It is noted that these cells indicated by $\odot$ are synchronized. By increasing the number of synchronized cells exponentially, eventually all of the cells are synchronized.

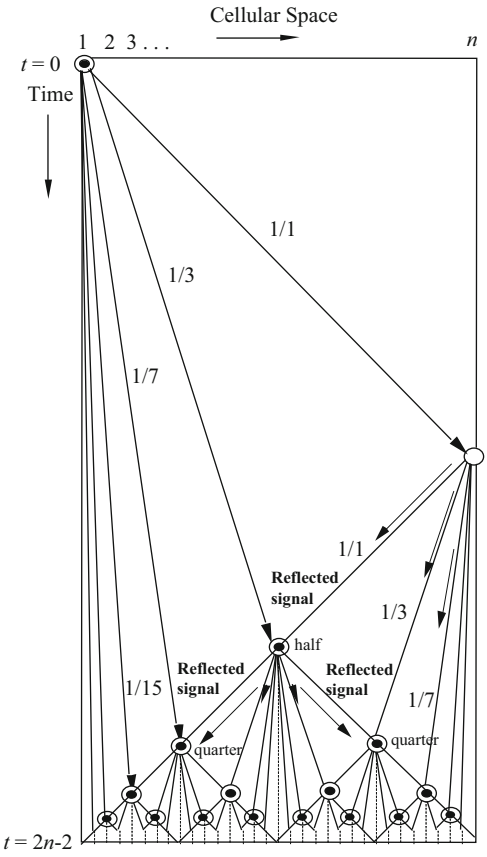

Fig. 1 Space-time diagram for optimum-time firing squad synchronization algorithm

The problem was first solved by J. McCarthy and M. Minsky who presented a $3n$-step algorithm for 1D cellular array of length n. In 1962, the first optimum-time, i.e. $(2n-2)$-step, synchronization algorithm was presented by Goto [4], with each cell having several thousands of states. Waksman [28] presented a 16-state optimum-time synchronization algorithm. Afterward, Balzer [1] and Gerken [3] developed an eight-state algorithm and a seven-state synchronization algorithm, respectively, thus decreasing the number of states required for the synchronization. Mazoyer [7] developed a six-state synchronization algorithm which, at present, is the algorithm having the fewest states for 1D arrays. In the sequel we use the following theorem as a base algorithm in the design of 2D array algorithms.

Theorem 1 (Goto [4], Waksman [28]). *There exists a cellular automaton that can synchronize any 1D array of length n in optimum* $2n-2$ *steps, where an initial general is located at a left or right end.*

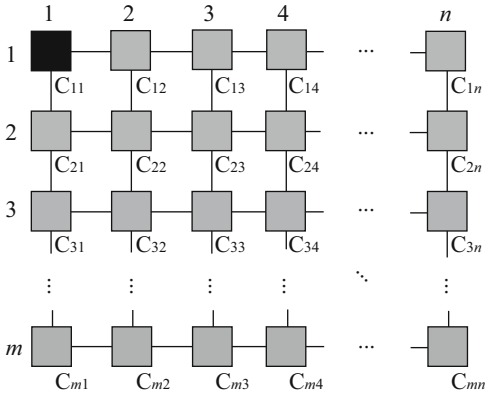

Fig. 2 A two-dimensional (2D) cellular automaton

2.2 FSSP on 2D Arrays

Figure 2 shows a finite 2D array consisting of $m \times n$ cells. Each cell is an identical (except the border cells) finite-state automaton. The array operates in lock-step mode in such a way that the next state of each cell (except border cells) is determined by both its own present state and the present states of its north, south, east and west neighbors. Thus, we assume the von Neumann-type four nearest neighbors. All cells (*soldiers*), except the north-west corner cell (*general*), are initially in the quiescent state at time $t = 0$ with the property that the next state of a quiescent cell with quiescent neighbors is the quiescent state again. At time $t = 0$, the north-west corner cell $C_{1,1}$ is in the *fire-when-ready* state, which is the initiation signal for synchronizing the array. The firing squad synchronization problem is to determine a description (state set and next-state function) for cells that ensures all cells enter the *fire* state at exactly the same time and for the first time.

A rich variety of synchronization algorithms for 2D arrays has been proposed. Concerning the rectangle synchronizers, see Beyer [2], Shinahr [10], Schmid [9], Szwerinski [11], Umeo [12], Umeo [13], and Umeo, Hisaoka, and Akiguchi [14]. As for square synchronization which is a special class of rectangles, several square synchronization algorithms have been proposed by Beyer [2], Shinahr [10], and Umeo, Maeda, and Fujiwara [20]. In recent years, Umeo and Kubo [18] developed a seven-state square synchronizer, which is a smallest implementation of the optimum-time square FSSP algorithm, known at present. One can easily see that it takes $2n - 2$ steps for any signal to travel from $C_{1,1}$ to $C_{n,n}$ due to the von Neumann neighborhood. Concerning the time optimality of the two-dimensional square synchronization algorithms, the following theorems have been established.

Theorem 2 (Beyer [2], Shinahr [10]). *There exists no 2D cellular automaton that can synchronize any square array of size $n \times n$ in less than $2n - 2$ steps, where the general is located at one corner of the array.*

Theorem 3 (Shinahr [10]). *There exists a 17-state cellular automaton that can synchronize any square array of size $n \times n$ at exactly $2n-2$ optimum steps.*

The lower bound of the time complexity for synchronizing rectangle arrays is as follows:

Theorem 4 (Beyer [2], Shinahr [10]). *There exists no cellular automaton that can synchronize any rectangle array of size $m \times n$ in less than $m+n+max(m,n)-3$ steps, where the general is located at one corner of the array.*

Theorem 5 (Beyer [2], Shinahr [10]). *There exists a cellular automaton that can synchronize any rectangle array of size $m \times n$ in exactly $m+n+max(m,n)-3$ steps, where the general is located at one corner of the array.*

3 Rotated L-Shaped Mapping Based Algorithm $\mathcal{A}_1$

The first 2D synchronization algorithm was developed independently by Beyer [2] and Shinahr [10]. It is based on a simple mapping which embeds a 1D optimum-time FSSP algorithm onto L-shaped sub-arrays composing a 2D array. We refer the embedding as *rotated L-shaped mapping*.

The algorithm for 2D square arrays operates as follows: By dividing an entire square array of size $n \times n$ into n rotated L-shaped 1D arrays, shown in Fig. 3 (left), in such a way that the length of the ith (from outside) L-shaped array is $2n-2i+1$ ($1 \leq i \leq n$). One treats the square synchronization as n independent 1D synchronizations with the general located at the bending point of the L-shaped array. We denote the ith L-shaped array by L_i and its horizontal and vertical segment is denoted by $\mathrm{L}_i^{\mathrm{h}}$ and $\mathrm{L}_i^{\mathrm{v}}$, respectively. Note that a cell at each bending point of the L-shaped array is shared for each synchronization by the two segments. See Fig. 3 (left). Concerning the synchronization of L_i, it can be easily seen that a general is generated by the cell $\mathrm{C}_{i,i}$ at time $t = 2i-2$ with the four nearest von-Neumann neighborhood communication, and the general initiates the horizontal (row) and vertical (column) synchronizations on $\mathrm{L}_i^{\mathrm{h}}$ and $\mathrm{L}_i^{\mathrm{v}}$, each of length $n-i+1$ using an optimum-time synchronization algorithm which can synchronize arrays of length ℓ in $2\ell-2$ steps (Theorem 1). For each i, $1 \leq i \leq n$, the ith L-shaped array L_i can be synchronized at time $t = 2i-2+2(n-i+1)-2 = 2n-2$. Thus the square array of size $n \times n$ can be synchronized at time $t = 2n-2$ in optimum-steps. In Fig. 3 (left), each general is represented by a black circle • in a shaded square and a wake-up signal for the synchronization generated by the general is indicated by a horizontal and vertical arrow. Shinahr [10] gave a 17-state implementation based on Balzer's eight-state synchronization algorithm (Balzer [1]). Later, it has been shown in Umeo, Maeda and Fujiwara [20] that nine states are sufficient for the optimum-time square synchronization:

Theorem 6 (Umeo, Maeda, and Fujiwara [20]). *There exists a nine-state 2D CA that can synchronize any $n \times n$ square array in $2n-2$ steps.*

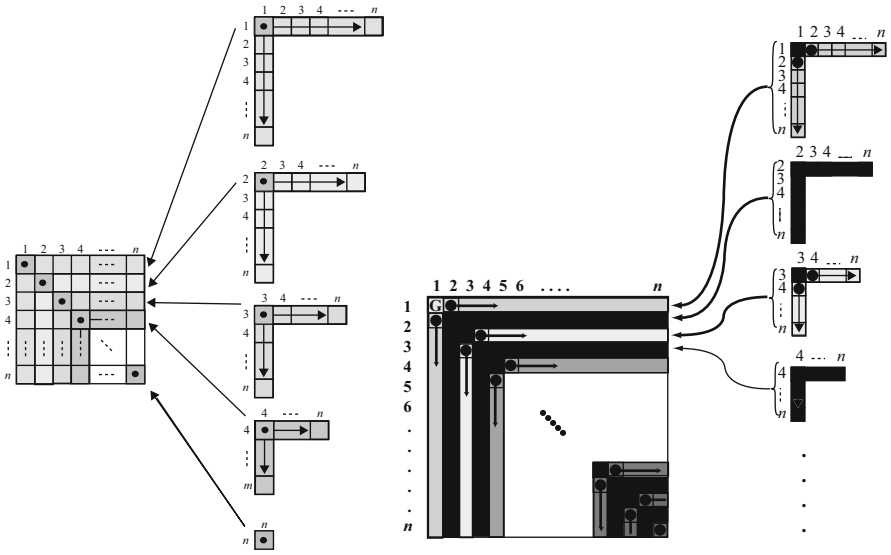

Fig. 3 A synchronization scheme based on *rotated L-shaped mapping* for $n \times n$ square cellular automaton (*left*) and *zebra mapping* for $n \times n$ square cellular automaton (*right*)

The first optimum-time rectangle synchronization algorithm was developed by Beyer [2] and Shinar [10] based on the rotated L-shaped mapping. The rectangular array of size $m \times n$ is regarded as $\min(m,n)$ rotated L-shaped 1D arrays, where they are synchronized independently using the *generalized* firing squad synchronization algorithm. The configurations of the generalized synchronization on 1D array are mapped onto the 2D array. Thus, an $m \times n$ array synchronization problem is reduced to independent $\min(m,n)$ 1D generalized synchronization problems such that $\mathcal{P}(m, m+n-1), \mathcal{P}(m-1, m+n-3), \ldots, \mathcal{P}(1, n-m+1)$ in the case $m \leq n$ and $\mathcal{P}(m, m+n-1), \mathcal{P}(m-1, m+n-3), \ldots, \mathcal{P}(m-n+1, m-n+1)$ in the case $m > n$, where $\mathcal{P}(k, \ell)$ means the 1D generalized synchronization problem for ℓ cells with a general on the kth cell from left end. Beyer [2] and Shinahr [10] presented an optimum-time synchronization scheme in order to synchronize any $m \times n$ arrays in $m + n + max(m,n) - 3$ steps. Shinahr [10] has given a 28-state implementation. Umeo, Ishida, Tachibana, and Kamikawa [16] gave a precise construction of the 28-state automaton having 12,849 rules.

Theorem 7 (Shinahr [10]). *There exists a 28-state cellular automaton that can synchronize any $m \times n$ rectangular arrays in optimum-time $m + n + max(m,n) - 3$ steps.*

4 Zebra Mapping Based Algorithm $\mathcal{A}_2$

In this section we first consider a state-efficient optimum-time square synchronization algorithm $\mathcal{A}_2$ proposed in Umeo and Kubo [18]. The algorithm is a variant of the L-shaped mapping. We show that seven states are sufficient for the optimum-time square synchronization. The proposed algorithm is basically based on the rotated L-shaped mapping scheme presented in the previous section. However, it is quite different from it in the following points. The mapping onto square arrays consists of two types of configurations: one is a one-cell smaller synchronized configuration and the other is a filled-in configuration with a stationary state. The stationary state remains unchanged once filled-in by the time before the final synchronization. Each configuration is mapped alternatively onto an L-shaped array *in a zebra-like fashion*. The mapping is referred to as *zebra mapping*. Figure 3 (right) illustrates the zebra mapping which consists of an embedded synchronization layer and a filled-in layer. In our construction we take the Mazoyer's 6-state synchronization rule as an embedded synchronization algorithm. See Mazoyer [7] for the six-state transition rule set. Figure 4 shows some snapshots of the synchronization process operating in optimum-steps on a 13×13 square array. The readers can see how those two types of configurations are mapped in the zebra-like fashion. The constructed seven-state cellular automaton has 787 transition rules, which can be found in Umeo and Kubo [18].

Theorem 8 (Umeo and Kubo [18]). *The seven-state square synchronization algorithm $\mathcal{A}_2$ can synchronize any $n \times n$ square array in optimum $2n-2$ steps.*

As for the rectangular arrays, Umeo and Nomura [23] constructed a ten-state 1629-rule 2D cellular automaton that can synchronize any $m \times n$ rectangle arrays in $m+n+\max(m,n)-2$ steps. See Fig. 5 for its snapshots. Note that the time complexity is one step larger than optimum.

Theorem 9 (Umeo and Nomura [23]). *There exists a ten-state 2D CA that can synchronize any $m \times n$ rectangle arrays in $m+n+\max(m,n)-2$ steps.*

5 Diagonal Mapping Based Algorithm $\mathcal{A}_3$

In this section we study a synchronization algorithm based on diagonal mapping. With the diagonal mapping, configurations of 1D cellular array can be embedded onto a square/rectangle array divided along the principal diagonal. Here we give a decomposition of a square array. We divide n^2 cells on a square array of size $n \times n$ into $2n-1$ groups g_k, $-(n-1) \leq k \leq n-1$ along the principal diagonal such that

$$g_k = \{C_{i,j} \mid j - i = k\},\ -(n-1) \leq k \leq n-1.$$

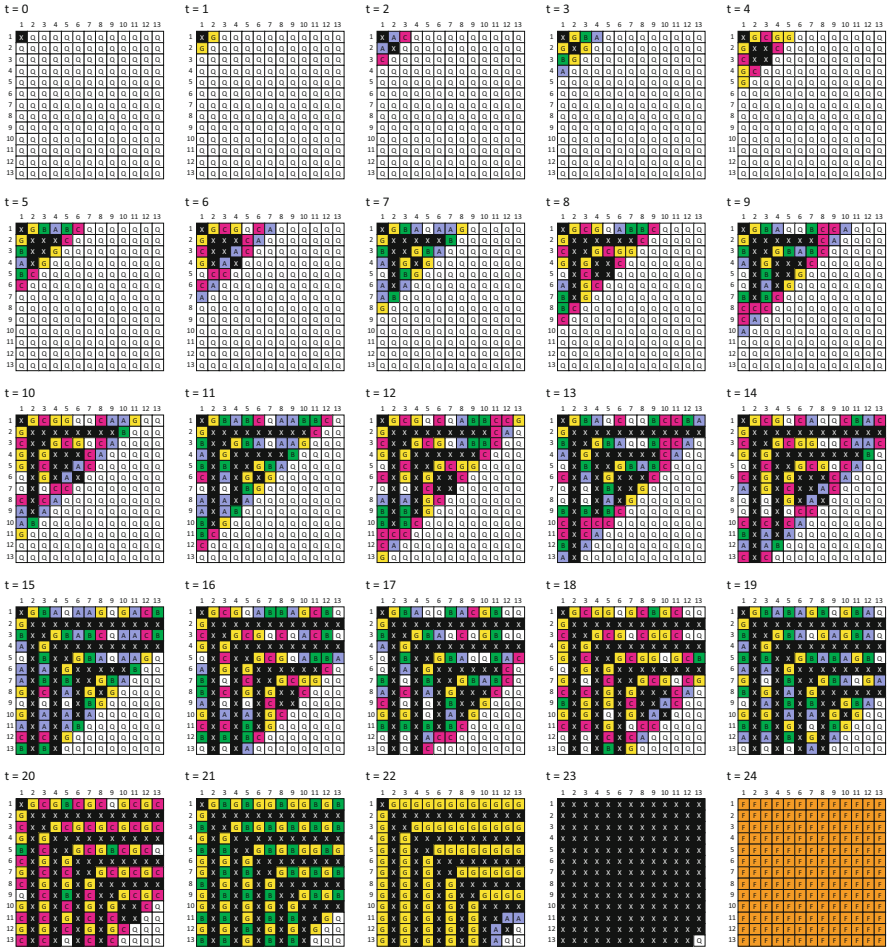

Fig. 4 Snapshots of the seven-state zebra-type square synchronizer on a 13×13 array

Each cell in g_k on the 2D square simulates the state of its corresponding cell C_k in the 1D array of length $2n-1$, $-(n-1) \leq k \leq n-1$. It has been shown in Umeo, Hisaoka, and Akiguchi [14] that any 1D generalized FSSP algorithm with an Inner-Independent Property $\mathcal{Z}$ (below) can be easily embedded onto 2D rectangle arrays *without introducing additional states*. The statement can also be applied to square arrays.

Inner-Independent Property $\mathcal{Z}$: Let S_i^t denote the state of C_i at step t. We say that an FSSP algorithm has *Inner-Independent Property $\mathcal{Z}$*, where any state S_i^t appearing *in the area $\mathcal{Z}$* can be computed from its left and right neighbor states S_{i-1}^{t-1} and S_{i+1}^{t-1} but it never depends on its own previous state S_i^{t-1}.

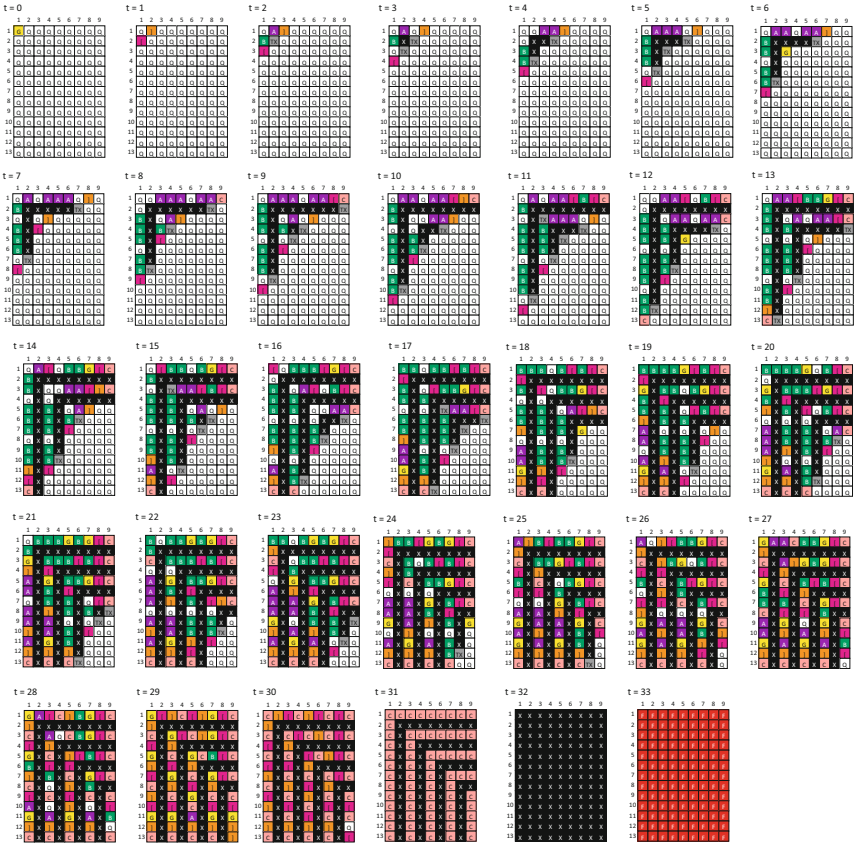

Fig. 5 Snapshots of the ten-state zebra-type rectangle synchronizer on a 13 × 9 array

A special 15-state generalized FSSP algorithm with the *Inner-Independent Property* $\mathcal{Z}$ can be realized in Ishii, Yanase, Maeda, and Umeo [6]. The 15-state algorithm with the property $\mathcal{Z}$ can be embedded on any square arrays without introducing additional states, yielding a 15-state optimum-time square synchronization algorithm.

Theorem 10 (Ishii, Yanase, Maeda, and Umeo [6]). *There exists a 15-state cellular automaton that can synchronize any* $n \times n$ *square array in optimum* $2n - 2$ *steps.*

As for the rectangle case, Umeo, Hisaoka, and Akiguchi [14] constructed a 12-state optimum-time rectangle synchronizer. See Umeo, Hisaoka, and Akiguchi [14] for details.

Theorem 11 (Umeo, Hisaoka, and Akiguchi [14]). *There exists a 12-state cellular automaton that can synchronize any $m \times n$ square array in optimum $m + n + \max(m, n) - 3$ steps.*

6 One-Sided Recursive-Halving Marking Based Algorithm $\mathcal{A}_4$

In this section we present an optimum-time synchronization algorithm $\mathcal{A}_4$ which is based on a marking called *one-sided recursive-halving marking*. The marking scheme prints a special mark on cells in a cellular space defined by one-sided recursive-halving. The marking itself is based on a well-known optimum-time one-dimensional synchronization algorithm. Let S be a one-dimensional cellular space consisting of cells C_i, C_{i+1}, ..., C_j, denoted by $[i \ldots j]$, where $j > i$. Let $|S|$ denote the number of cells in S, that is $|S| = j - i + 1$ for $S = [i \ldots j]$. A cell $C_{(i+j)/2}$ in S is a center cell of S, if $|S|$ is odd. Otherwise, two cells $C_{(i+j-1)/2}$ and $C_{(i+j+1)/2}$ are center cells of S.

The one-sided recursive-halving marking for a given 1D cellular space $[1 \ldots n]$ is defined as follows:

One-Sided Recursive-Halving Marking

```
begin
  S := [1...n];
  while |S| > 1 do
    if |S| is odd then
      mark a center cell C_x in S
      S := [x...n];
    else
      mark center cells C_x and C_{x+1} in S
      S := [x + 1...n];
end
```

We have developed a simple implementation of the one-sided recursive halving marking on a 13-state cellular automaton. It can be easily seen that any 1D cellular space of length n with the one-sided recursive-halving marking initially can be synchronized in optimum $n - 1$ steps.

Now we consider a square array of size $n \times n$ with an initial general G on $C_{1,1}$. The square is regarded as consisting of two triangles: upper and lower halves separated by a diagonal, shown in Fig. 6 (left). Each upper and lower half triangle consists of n columns and n rows, each denoted by c_k and r_k, $1 \leq k \leq n$, such that:

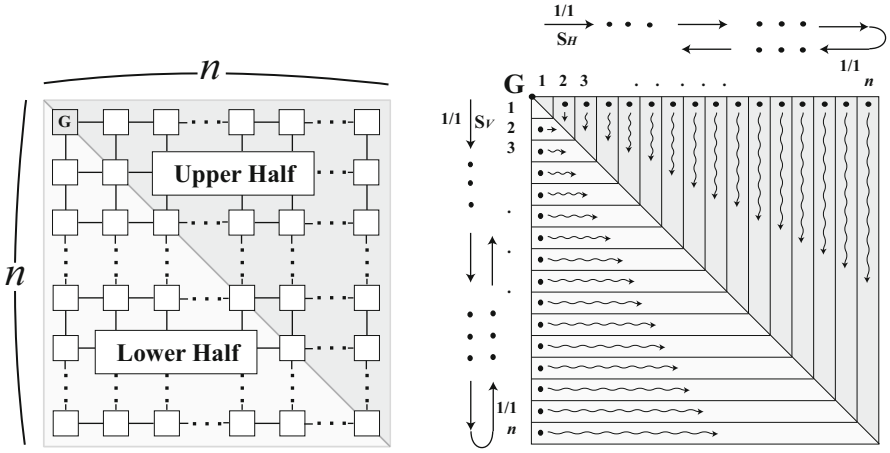

Fig. 6 A square array is decomposed into an upper and lower triangle (*left*) and an illustration of the synchronization scheme in each triangle (*right*)

$$c_k = \{C_{i,k} \mid 1 \leq i \leq k\}, \qquad r_k = \{C_{k,j} \mid 1 \leq j \leq k\}.$$

Note that the length of c_k and r_k is k for $1 \leq k \leq n$. An overview of the algorithm $\mathcal{A}_4$ is:

- Each upper and lower half triangle is synchronized independently.
- At time $t = 0$ the array begins to prepare printing the one-sided recursive halving mark on each column and row, each starting from top of each column and a left end of each row, respectively, in the triangles. The marking operation will be finished before the arrival of the first wake-up signal for the synchronization.
- Simultaneously, the general generates two signals $\mathbf{s}_H$ and $\mathbf{s}_V$ at time $t = 0$. Their operations are as follows:
  - **Signal $\mathbf{s}_H$**: The s_H-signal travels along the first row at 1/1-speed and reaches $C_{1,n}$ at time $t = n - 1$. Then it reflects there and returns the same route at 1/1-speed, and reaches $C_{1,1}$ again at time $t = 2n - 2$. On the return way, it generates a general on $C_{1,i}$ at time $t = n - 1 + n - (i - 1) = 2n - i$, at every visit of $C_{1,i}$, where $1 \leq i \leq n$. See Fig. 6 (right). The general is denoted by a black circle •. A ripple-like line, staring from the symbol •, shown in Fig. 6 (right), illustrates the initiation of the synchronization process initiated by the general. The general initiates a synchronization for the ith column, and yields a successful synchronization at time $t = 2n - 2$. Note that the length of the ith column is i and the synchronization is started at time $t = 2n - i$, for any $1 \leq i \leq n$. In this way, the upper half triangle can be synchronized in $2n - 2$ steps.
  - **Signal $\mathbf{s}_V$**: The s_V-signal travels along the 1st column at 1/1-speed and reaches $C_{n,1}$ at time $t = n - 1$. Then it reflects there and returns the same

Table 1 A list of FSSP algorithms for square arrays

Algorithms & implementations	# of states	# of rules	Time complexity	Communication model	Mapping
Beyer [2] Algorithm A_1	—	—	$2n-2$	O(1)-bit	L-shaped
Shinahr [10] Algorithm A_1	17	—	$2n-2$	O(1)-bit	L-shaped
Umeo, Maeda and Fujiwara [20] Algorithm A_1	9	1718	$2n-2$	O(1)-bit	L-shaped
Umeo and Kubo [18] Algorithm A_2	7	787	$2n-2$	O(1)-bit	Zebra
Umeo, Maeda, Hisaoka, and Teraoka [21] Algorithm A_3	6	942	$4n-4$	O(1)-bit	Diagonal
Ishii et al. [6] Algorithm A_3	15	1614	$2n-2$	O(1)-bit	Diagonal
Umeo, Uchino and Nomura [25] Algorithm A_4	37	3271	$2n-2$	O(1)-bit	Recursive-Halving
Gruska, Torre and Parente [5] Algorithm A_1	—	—	$2n-2$	1-bit	L-shaped
Umeo and Yanagihara [26] Algorithm A_1	49	237	$2n-2$	1-bit	L-shaped

route at 1/1-speed, and reaches $C_{1,1}$ again at time $t = 2n-2$. On the return way, it generates a general on $C_{i,1}$ at time $t = n-1+n-(i-1) = 2n-i$, at every visit of $C_{i,1}$, where $1 \leq i \leq n$. The general initiates a synchronization for the ith row. Note that the length of the ith row is i. The ith row can be synchronized at time $t = 2n-2$, for any i, $1 \leq i \leq n$. Thus, the lower half triangle can be synchronized in $2n-2$ steps.

We implemented the algorithm $\mathcal{A}_4$ on a 2D cellular automaton. The constructed cellular automaton has 37 internal states and 3,271 transition rules. Thus we have:

Theorem 12 (Umeo, Uchino, and Nomura [25]). *The synchronization algorithm $\mathcal{A}_4$ can synchronize any $n \times n$ square array in optimum $2n-2$ steps.*

As for the rectangle case, Umeo, Nishide, and Yamawaki [22] constructed a 2D cellular automaton with 384-states and 112,690-rules. See Umeo, Nishide, and Yamawaki [22] for details.

Table 2 A list of FSSP algorithms for rectangle arrays

Algorithms & implementations	# of states	# of rules	Time complexity	Communication model	Mapping
Beyer [2] Algorithm A_1	—	—	$m+n+max(m,n)-3$	O(1)-bit	L-shaped
Shinahr [10]	28	—	$m+n+max(m,n)-3$	O(1)-bit	L-shaped
Umeo et al. [16] Algorithm A_1	28	12849*			
Umeo and Nomura [23] Algorithm A_2	10	1629	$m+n+max(m,n)-2$	O(1)-bit	Zebra
Umeo, Hisaoka and Akiguchi [14] Algorithm A_3	12	1532	$m+n+max(m,n)-3$	O(1)-bit	Diagonal
Umeo, Maeda, Hisaoka, and Teraoka [21] Algorithm A_3	6	942	$2m+2n-4$	O(1)-bit	Diagonal
Umeo, Nishide and Yamawaki [22] Algorithm A_4	384	112690	$m+n+max(m,n)-3$	O(1)-bit	Recursive-Halving

Theorem 13 (Umeo, Nishide, and Yamawaki [22]). *The algorithm $\mathcal{A}_4$ can synchronize any $m \times n$ rectangular array in $m+n+max(m,n)-3$ optimum steps.*

7 Conclusions

In this paper, we have presented a survey on recent developments of optimum-time and non-optimum-time FSSP algorithms for 2D arrays. In Tables 1 and 2 we present a list of implementations of square/rectangle FSSP algorithms for cellular automata with O(1)-bit and 1-bit communications.

The O(1)-bit communication model, discussed in this paper, is a usual cellular automaton in which the amount of communication bits exchanged in one step between neighboring cells is assumed to be O(1) bits. The 1-bit communication model is a subclass of the O(1)-bit model, in which inter-cell communication is restricted to 1-bit communication.

For a long time only one FSSP algorithm based on the rotated L-shaped mapping proposed by Beyer [2] and Shinar [10] has been known. The readers can see how a rich variety of 2D FSSP algorithms exists. Some algorithms can be easily extended to 3D arrays. The embedding schemes developed in this paper would be useful for further implementations of multi-dimensional synchronization algorithms.

Acknowledgements A part of this work is supported by Grant-in-Aid for Scientific Research (C) 21500023.

References

1. Balzer, R.: An 8-state minimal time solution to the firing squad synchronization problem. Inf. Control **10**, 22–42 (1967)
2. Beyer, W.T.: Recognition of topological invariants by iterative arrays. Ph.D. Thesis, MIT, pp. 144 (1969)
3. Gerken, H.D.: Über Synchronisations problem bei Zellularautomaten, Diplomarbeiten, Institut für Theoretische Informatik, Technische Universität Braunschweig, pp. 50, (1987)
4. Goto, E.: A minimal time solution of the firing squad problem. Dittoed Course Notes for Applied Mathematics 298, Harvard University, pp. 52–59 (1962)
5. Gruska, J., Torre, S.L., Parente, M.: The firing squad synchronization problem on squares, toruses and rings. Int. J. Found. Comput. Sci. **18**(3), 637–654 (2007)
6. Ishii, S., Yanase, H., Maeda, M., Umeo, H.: State-efficient implementations of time-optimum synchronization algorithms for square arrays. Technical Report of IEICE, Circuit and Systems, pp. 13–18 (2006)
7. Mazoyer, J.: A six-state minimal time solution to the firing squad synchronization problem. Theor. Comput. Sci. **50**, 183–238 (1987)
8. Moore, E.F.: The firing squad synchronization problem. In: Moore, E.F., (Ed.) Sequential Machines, Selected Papers, pp. 213–214. Addison-Wesley, Reading (1964)
9. Schmid, H.: Synchronisationsprobleme für zelluläre Automaten mit mehreren Generälen. Diplomarbeit, Universität Karsruhe, (2003)
10. Shinahr, I.: Two- and three-dimensional firing squad synchronization problems. Inf. Control **24**, 163–180 (1974)
11. Szwerinski, H.: Time-optimum solution of the firing-squad-synchronization-problem for n-dimensional rectangles with the general at an arbitrary position. Theor. Comput. Sci. **19**, 305–320 (1982)
12. Umeo, H.: Firing squad synchronization algorithms for two-dimensional cellular automata. J. Cell. Autom. **4**, 1–20, (2008)
13. Umeo, H.: Firing squad synchronization problem in cellular automata. In: Meyers, R.A. (Ed.) Encyclopedia of Complexity and System Science, vol. 4, pp. 3537–3574. Springer, Berlin, Heidelberg (2009)
14. Umeo, H., Hisaoka, M., Akiguchi, S.: Twelve-state optimum-time synchronization algorithm for two-dimensional rectangular cellular arrays. In: Proceedings of 4th International Conference on Unconventional Computing: UC 2005, Sevilla. LNCS 3699, pp. 214–223 (2005)
15. Umeo, H., Hisaoka, M., Teraoka, M., Maeda, M.: Several new generalized linear- and optimum-time synchronization algorithms for two-dimensional rectangular arrays. In: Margenstern, M. (Ed.) Proceedings of 4th International Conference on Machines, Computations and Universality: MCU 2004, Saint Petersburg. LNCS 3354, pp. 223–232 (2005)
16. Umeo, H., Ishida, K., Tachibana, K., Kamikawa, N.: A transition rule set for the first 2-D optimum-time synchronization algorithm. In: Proceedings of the 4th International Workshop on Natural Computing, PICT 2, Himeji, pp. 333–341. Springer (2009)
17. Umeo, H., Kamikawa, N., Nishioka, K., Akiguchi, S.: Generalized firing squad synchronization protocols for one-dimensional cellular automata – a survey. Acta Phys. Pol. B, Proc. Suppl. **3**, 267–289 (2010)
18. Umeo, H., Kubo, K.: A seven-state time-optimum square synchronizer. In: Proceedings of the 9th International Conference on Cellular Automata for Research and Industry, Ascoli Piceno. LNCS 6350, pp. 219–230. Springer (2010)

19. Umeo, H., Kubo, K.: Recent developments in constructing square synchronizers. In: Proceedings of the 10th International Conference on Cellular Automata for Research and Industry, Santorini. LNCS 7495, pp. 171–183. Springer (2012)
20. Umeo, H., Maeda, M., Fujiwara, N.: An efficient mapping scheme for embedding any one-dimensional firing squad synchronization algorithm onto two-dimensional arrays. In: Proceedings of the 5th International Conference on Cellular Automata for Research and Industry, Geneva. LNCS 2493, pp. 69–81. Springer (2002)
21. Umeo, H., Maeda, M., Hisaoka, M., Teraoka, M.: A state-efficient mapping scheme for designing two-dimensional firing squad synchronization algorithms. Fundam. Inform. **74**, 603–623 (2006)
22. Umeo, H., Nishide, K., Yamawaki, T.: A new optimum-time firing squad synchronization algorithm for two-dimensional rectangle arrays – one-sided recursive halving based. In: Löwe, B. et al. (Eds.) Proceedings of the International Conference on Models of Computation in Context, Computability in Europe 2011, CiE 2011, Sofia. LNCS 6735, pp. 290–299 (2011)
23. Umeo, H., Nomura, A.: Zebra-like mapping for state-efficient implementation of two-dimensional synchronization algorithms (2014, manuscript in preparation)
24. Umeo, H., Uchino, H.: A new time-optimum synchronization algorithm for rectangle arrays. Fundam. Inform. **87**(2), 155–164 (2008)
25. Umeo, H., Uchino, H., Nomura, A.: How to synchronize square arrays in optimum-time. In: Proceedings of the 2011 International Conference on High Performance Computing and Simulation (HPCS 2011), Istanbul, pp. 801–807. IEEE (2011)
26. Umeo, H., Yanagihara, T.: Smallest implementations of optimum-time firing squad synchronization algorithms for one-bit-communication cellular automata. In: Proceedings of the 2011 International Conference on Parallel Computing and Technology, PaCT 2011, Kazan. LNCS 6873, pp. 210–223 (2011)
27. Umeo, H., Yamawaki, T., Shimizu, N., Uchino, H.: Modeling and simulation of global synchronization processes for large-scale-of two-dimensional cellular arrays. In: Proceedings of International Conference on Modeling and Simulation, AMS 2007, Phuket, pp. 139–144 (2007)
28. Waksman, A.. An optimum solution to the firing squad synchronization problem. Inf. Control **9**, 66–78 (1966)

Optimization Issues in Distributed Computing Systems Design

Krzysztof Walkowiak and Jacek Rak

Abstract In recent years, we observe a growing interest focused on distributed computing systems. Both industry and academia require increasing computational power to process and analyze large amount of data, including significant areas like analysis of medical data, earthquake, or weather forecast. Since distributed computing systems – similar to computer networks – are vulnerable to failures, survivability mechanisms are indispensable to provide the uninterrupted service. Therefore, in this paper we propose a novel $1 + 1$ protection mechanism. We formulate an ILP model related to optimization of survivable distributed computing systems. The objective is to allocate computational tasks to computing nodes and dimension network capacity in order to minimize the operational cost of the computing system and satisfy survivability constraints. To facilitate high computational complexity caused by NP-completeness in solving the ILP problem, we propose additional cut inequalities that can be applied for the branch-and-cut algorithm. We consider the cut-and-branch variant of the B&C algorithm. To construct additional cut inequalities we use the idea of cover inequalities and mixed integer rounding (MIR) inequalities. Results of experiments conducted using CPLEX solver are provided and discussed.

1 Introduction

Recently, distributed computing systems are gaining much attention due to a growing demand to process large computational tasks in many different fields, e.g., collaborative visualization of large scientific databases, financial modeling, medical data analysis, bioinformatics, experimental data acquisition, climate/weather modeling, earthquake simulation, astrophysics and many others [13, 15, 18]. Development of distributed computing systems triggers the need to make research on a wide range

K. Walkowiak (✉)
Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, PL-50-370 Wroclaw, Poland
e-mail: krzysztof.walkowiak@pwr.wroc.pl; krzysztof.walkowiak@pwr.edu.pl

J. Rak
Gdansk University of Technology, Narutowicza 11/12, PL-80-233 Gdansk, Poland
e-mail: jrak@pg.gda.pl

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_21

of various topics. In this work, we focus on one of these aspects and address the problem of scheduling and capacity design in the overlay computing systems.

The distributed computing systems like Grids can be developed using special dedicated high-speed networks [18], as well as the Internet can be used as the backbone network for overlay-based systems [16, 17]. We focus on the latter case, as overlays offer considerable network functionalities (e.g., diversity, flexibility, manageability) in a relatively simple and cost-effective way, as well as regardless of physical and logical structure of underlying networks. Efficiency of distributed computing systems may be significantly impacted by failures of network elements (i.e., nodes, or links) most frequently caused by human errors (e.g., cable cuts), or forces of nature (e.g., earthquakes, hurricanes). Therefore, network survivability, i.e., ability to provide the continuous service after a failure [5], becomes a crucial issue. Most commonly, it is assured by means of additional (backup) resources (e.g., transmission links/paths, computing units) used after the failure affecting the components of the main communication path (called working path) as the main network resources of task processing. To provide survivability for distributed systems, we propose a proactive approach based on the $1 + 1$ method applied in connection-oriented computer networks [5, 14, 17]. In a non-failure operating state of the system, results (output data) are concurrently sent from two computing nodes to all receivers requesting information. After a failure, one of the considered computing nodes remains operational, implying that results of computation are provided with no violation at all.

Survivability of distributed computing systems (in particular including protection of computing units), is a relatively new topic, and only a few proposals exist in the literature. However, most of them assume that the computing system uses a dedicated optical network, e.g., [3, 4, 9, 16]. The main novelty of this work is that we consider a survivable distributed system working as an overlay network, i.e., a network that is built on top of an existing underlying network providing basic networking functionalities, including routing and forwarding. In this overlay structure, the underlying network can be based on either wired or even wireless communications (the latter one realized e.g., according to 802.11s standard of wireless mesh networks with highly directional antennas providing mutually non-interfering links).

It is worth noting that issues of survivable distributed systems design are addressed in many currently ongoing projects aimed to determine the architecture of Future Internet. In particular, they are one of the main aims of the Polish IIP project [7], and in particular with respect to the assumed IIP activities, including Internet 3D (Task 3.3), or medical digital libraries (Task 3.2.4).

The main contributions of the paper are threefold. (i) A new ILP model for scheduling and capacity design in overlay computing system with additional survivability constraints. (ii) Cut inequalities proposed to facilitate solving of the ILP model. (iii) Numerical results presenting the performance of the overlay computing system in various conditions and effectiveness of additional cuts.

The rest of the paper is organized as follows. In Sect. 2, we present the system architecture, and formulate the ILP model. The respective cut inequalities are

introduced in Sect. 3. Section 4 includes results of numerical experiments. Section 5 concludes the work.

2 ILP Model of Scheduling and Capacity Design in Survivable Distributed Computing Systems

The ILP model of a survivable distributed computing system is formulated according to real overlay systems as well as according to assumptions presented in previous works on optimization of distributed computing systems [1, 3, 4, 7, 9, 10, 16, 17, 19]. Note that the presented model is generic, thus the model can be applied to optimize various kinds of computing systems including Grids and public resource computing systems [13, 15, 18].

2.1 Notation

The computing system considered here consists of nodes indexed by $v = 1, 2, \ldots, V$ representing computing elements (individual computers or clusters), as well as sources of input data and destinations of output data. The system works on top of an overlay network (e.g., Internet), and each node is connected by an access link to the network. Connectivity between nodes is provided by virtual links of the overlay realized by paths consisting of links deployed in the underlying network. According to [1], nodes' capacity constraints are typically adequate in overlay networks. Additionally, in overlays the underlaying physical network is typically assumed to be overprovisioned, and the only bottlenecks are access links [19]. Therefore, the only network capacity constraints in the model refer to access links. Since the access link capacity is to be dimensioned, integer variable z_v denotes the number of capacity modules allocated to the access link of node v. We assume that each node v is assigned to a particular ISP (Internet Service Provider), which offers high speed access link with a capacity module m_v given in Mbps (e.g., Fast Ethernet). Each node is already equipped with some computers, and p_v denotes the processing power of node v given by a number of uniform tasks that node v can perform in 1 s.

The computing system is to process a set of computational tasks of the same required processing power, e.g., a number of FLOPS and indexed by $r = 1, 2, \ldots, R$. Each task r belongs to a particular computational project k. Each task r can be processed independently, i.e., we assume that there is no dependency between individual tasks. For each task r, there is a source node that produces the input data, and one or more destination nodes that receive the output data including results of computations (processing). Constants s_{rv} and t_{rv} are used to denote the source and destination nodes of each task, i.e., s_{rv} is 1, if node v is the source node of task

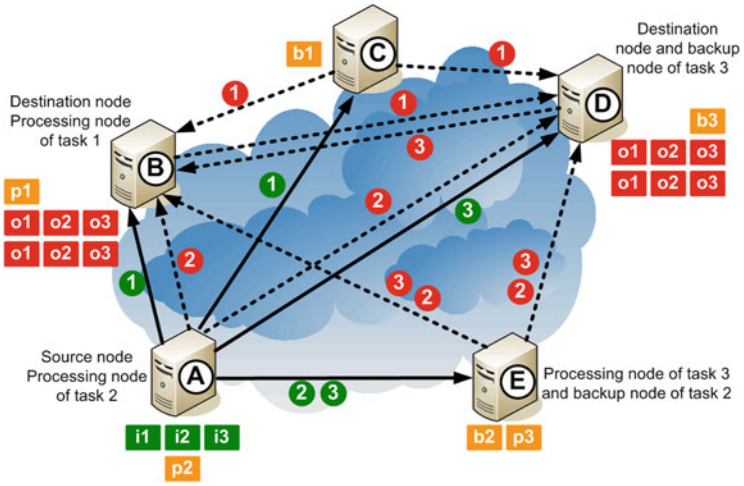

Fig. 1 Example of a survivable distributed computing system

r; 0 otherwise. In the same way, t_{rv} is 1, if node v is the destination node of task r; 0 otherwise. Constants a_r and b_r denote the transmission rate of input data and output data, respectively, per task r given in bps (bits per second).

The workflow of the system is as follows. The project input data is transferred from the source node providing input data to one or more computing nodes that process the data. Next, the output data (results of computations) is sent from each computing node to one or more destination nodes. Similar to [4, 10, 16], we assume that computational projects are long-lived, i.e., they are established for a relatively long time (e.g., days, weeks). The input and output data associated with the project is continuously generated and transmitted. Thus, computational and network resources can be allocated in the system using offline optimization methods.

A simple example to illustrate the system architecture of the survivable distributed computing system is shown in Fig. 1. We assume that the system contains five computing nodes denoted as `A`, `B`, `C`, `D`, and `E`, which are connected to the overlay network. There are three tasks to be processed in the system. The workflow is as follows. Node A is the source node of all tasks. Therefore, node A provides input data related to the tasks (rectangles labeled `i1`, `i2` and `i3`). Nodes `B` and `D` are destinations of all tasks. Rectangles labeled `p1`, `p2`, and `p3` denote the places where a particular task is processed. Rectangles labeled `o1`, `o2`, and `o3` denote results of computations related to tasks `1`, `2` and `3`, respectively. Due to the survivability requirements, each of the three tasks is processed at two separate nodes – primary tasks are marked with rectangles `p1`, `p2`, and `p3`, while backup tasks are labelled `b1`, `b2` and `b3`, respectively. For example, task `1` is processed at nodes `B` and `C`. Moreover, in the illustration we report network flows generated to deliver the input and output data. Solid line denotes the flow of input data. Circle with a number inside denotes the indices of tasks the input data is related to. Dotted line denotes

the flow of output data. Again, the numbers in circles indicate task indices the data belongs to. Assignment of tasks to processing nodes generates network traffic. For instance, node A uploads two copies of task `1` and `3` input data, and one copy of task `2` input data.

Many computational tasks processed in distributed systems are of great importance and need execution guarantees, e.g., medical applications, business analysis, weather forecasts, etc. However, distributed systems – similar to communication networks – are subject to various unintentional failures caused by natural disasters (hurricanes, earthquakes, floods, etc.), overload, software bugs, human errors, and intentional failures caused by maintenance action or sabotage [17]. Such failures influence network infrastructure connecting computing nodes, e.g., access link failure, underlying physical network link failure, etc. Moreover, elements of distributed computing systems are also subject to various breakdowns (e.g., hardware failure, power outage, software bug, etc.). Therefore, in distributed computing systems, to provide guarantees on computational tasks completion, execution and delivery of results need to be enhanced with some survivability mechanisms.

We consider a failure that leads to a situation when the results to be obtained at one of the computing nodes are not delivered to the requesting destination nodes (e.g., due to an access link failure, backbone link failure, backbone node failure, etc., and processing issues including node hardware failure, power outage, etc.). To protect the distributed computing system against these kinds of failures, we introduce a similar approach as in connection-oriented networks, i.e., $1 + 1$ protection developed in the context of Automatic Protection Switching (APS) networks [5]. The key idea is to assign to each computational task two computing nodes: primary and backup. Both nodes simultaneously process the same input data and next send results to all destination nodes. To make the system more flexible, not all tasks are to be protected – parameter α_r is 1, if task r requires protection; 0, otherwise.

Moreover, we assume that the maximum number of computing nodes involved in one project cannot be larger than S. For instance, if $S = 1$, then all uniform tasks of a particular project can be computed only in one node. When $S = V$, the number of computing nodes is not limited. We refer to S as a split ratio. The motivation behind this parameter follows from management issues, i.e., less computing nodes (lower value of the split ratio) facilitates the management of the computing system.

The objective of our model is to minimize the operational cost (OPEX) of the computing system including expenses related to two elements: transmission and processing. Constant ξ_v given in euro/month denotes the whole OPEX cost related to one capacity module allocated for node v and includes leasing cost of the capacity module paid to the ISP as well as all other OPEX costs like energy, maintenance, administration, etc. Constant Ψ_v denotes the OPEX cost related to processing of one uniform task in node v. The Ψ_v cost is defined in euro/month and contains all expenses necessary to process the uniform computational tasks including both processing and storage issues (e.g., energy, maintenance, hardware amortization, etc.).

2.2 ILP Model

Indices

$v,\ w = 1, 2, \ldots, V$ computing nodes
$k = 1, 2, \ldots, K$ computational projects
$r = 1, 2, \ldots, R$ computational tasks

Constants

p_v maximum processing rate of node v
a_r transmit rate of input data per task r (Mbps)
b_r transmit rate of output data per task r (Mbps)
s_{rv} $= 1$, if v is the source node of task r; 0 otherwise
t_{rv} $= 1$, if v is the destination node of task r; 0 otherwise
t_r number of destination nodes for task r, i.e., $t_r = \Sigma_v t_{rv}$
S split ratio
y_v OPEX cost related to processing of one task in node v (euro/month)
ψ_v OPEX cost related to one capacity module of node v (euro/month)
m_v size of the capacity module for node v (Mbps)
α_r $= 1$, if task r requires protection; 0, otherwise
δ_{rk} $= 1$, if task r belongs to project k; 0, otherwise

Variables

x_{rv} $= 1$, if task r is allocated to primary computing node v; 0, otherwise (binary)
y_{rv} $= 1$, if task r is allocated to backup computing node v; 0, otherwise (binary)
z_v capacity of node v access link expressed in the number of capacity modules (non-negative integer)
u_{kv} $= 1$, if project k uses computing node v; 0, otherwise (binary)

Objective It is to determine scheduling of tasks to primary and backup nodes as well as dimension network access links to minimize the operational cost of the system, i.e.:

$$\textit{minimize } C = \Sigma_v z_v \xi_v + \Sigma_r \Sigma_v (x_{rv}\psi_v + y_{rv}\psi_v) \tag{1}$$

Constraints

(a) Each computing node v has a limited processing power p_v. Therefore, each node cannot be assigned with more tasks to calculate than it can process. Both primary and backup nodes must be taken into account:

$$\Sigma_r (x_{rv} + y_{rv}) \leq p_v; \qquad v = 1, 2, \ldots, V \tag{2}$$

(b) Download capacity constraint – incoming flow of each node cannot exceed the capacity of the access link

$$\Sigma_r(1-s_{rv})a_r(x_{rv}+y_{rv})+\Sigma_r t_{rv}b_r(1-x_{rv}+\alpha_r-y_{rv}) \le z_v m_v; \quad v=1,2,\ldots,V \tag{3}$$

(c) Upload capacity constraint – outgoing flow of each node cannot exceed the capacity of the access link:

$$\Sigma s_{rv}a_r(1-x_{rv}+\alpha_r+y_{rv})+\Sigma_r(t_r-t_{rv})b_r(x_{rv}+y_{rv}) \le z_v m_v; \quad v=1,2,\ldots,V \tag{4}$$

(d) Each task r must be assigned to exactly one primary node:

$$\Sigma_v x_{rv}=1; \quad r=1,2,\ldots,R \tag{5}$$

(e) If task r is to be protected (i.e., $\alpha_r = 1$), it must be assigned to a backup node:

$$\Sigma_v y_{rv}=\alpha_r; \quad r=1,2,\ldots,R \tag{6}$$

(f) In order to provide survivability, primary and backup nodes must be disjoint:

$$(x_{rv}+y_{rv}) \le 1; \quad r=1,2,\ldots,R \quad v=1,2,\ldots,V \tag{7}$$

(g) Definitions of variables to determine participation of a node in a computational project – both primary and backup nodes are considered:

$$x_{rv} \le u_{kv}; \quad r=1,2,\ldots,R \quad v=1,2,\ldots,V \quad k=1,2,\ldots,K \quad \delta_{rk}=1 \tag{8}$$

$$y_{rv} \le u_{kv}; \quad r=1,2,\ldots,R \quad v=1,2,\ldots,V \quad k=1,2,\ldots,K \quad \delta_{rk}=1 \tag{9}$$

(h) The split of each computational project cannot exceed the given limit S:

$$\Sigma_v u_{kv} \le S; \quad k=1,2,\ldots,K \tag{10}$$

3 Cut Inequalities

The problem (1)–(10) is NP-hard, since it is equivalent to the network design problem with modular link capacities [14]. Therefore, we propose here to use additional cut inequalities that can be applied in construction of the branch-and-cut (B&C) algorithm. We consider the cut-and-branch variant of the B&C algorithm, in which cut inequalities are added to the root node of the solution tree. It means that all generated cuts are valid throughout the whole B&C tree [12].

The first cut is a lower bound on y_v values. Notice that if node v is a destination node of task r ($t_{rv} = 1$), it must receive the output data (results of computations) related to task r. Node v can receive the data in two ways: either as the input data

(that is later processed in this node to obtain the output data), or as the output data. However, notice that if node v is the source node of task r ($s_{rv} = 1$), the considered node is not obliged to download the data related to task r. First, we consider the case when task r is not to be protected ($\alpha_r = 0$). The download capacity of node v related to processing of task r must then exceed the following value $t_{rv}\,(1 - s_{rv})min(a_r, b_r)$, i.e., the minimum of input and output data rates of task r is selected. If task r is to be protected ($\alpha_r = 1$), node v receives results of computations twice. However, according to constraint (7), the same node cannot serve as both a primary and a backup computing node of task r, i.e., data related to task r cannot be downloaded in both cases (primary and backup processing) as the input data. Consequently, two cases are possible: (i) node v downloads input and output data of task r; (ii) node v downloads twice output data of task r. Summarizing, the download capacity of v related to processing of protected task r must exceed the value of $t_{rv}(1-s_{rv})min(a_r + b_r, 2b_r)$. Let d_{rv} denote the lower bound of a download flow related to node v and task r:

$$d_{rv} = \begin{cases} t_{rv}(1 - s_{rv})min(a_r, b_r); & \alpha_r = 0 \\ t_{rv}(1 - s_{rv})min(a_r + b_r, 2b_r); & \alpha_r = 1 \end{cases} \tag{11}$$

In analogous way, we analyze the upload capacity of node v related to processing of task r. If node v is the source node of task r ($s_{rv} = 1$), the data related to task r is to be delivered to all destination nodes of this task. Again, the data can be sent either as input data (rate a_r) to another processing node, or node v performs task r and sends the output data (rate b_r) to ($t_r - t_{rv}$) nodes (all destination nodes except for itself). If task r is not protected ($\alpha_r = 0$), the upload capacity of node v related to task r must exceed the s_{rv} value equal to $\min(a_r, (t_r - t_{rv})b_r)$. When task r is protected ($\alpha_r = 1$), due to constraint (7), it is again not possible that node v serves as both a primary and a backup node, performs task r twice, and consequently sends the output data of both a primary and a backup task. Therefore, two cases are possible: (i) node v uploads input data (rate a_r) to primary and backup nodes; (ii) node v calculates task r (for instance as a primary node) and sends the output data (rate b_r) to destination nodes, the data related to backup task is sent as input data (rate a_r). The upload capacity of node v related to task r must thus exceed the value of s_{rv} equal to $min(2a_r, a_r + (t_r - t_{rv})b_r)$. Let e_{rv} denote the lower bound of upload flow related to node v and task r:

$$e_{rv} = \begin{cases} s_{rv}min(a_r, (t_r - t_{rv})b_r); & \alpha_r = 0 \\ s_{rv}min(2a_r, a_r + (t_r - t_{rv})b_r); & \alpha_r = 1 \end{cases} \tag{12}$$

Combining definitions (11)–(12), and with MIR (Mixed Integer Rounding) approach [6–11], we can formulate the following cuts:

$$\lceil \Sigma_r d_{rv}/m_v \rceil \leq z_v; \quad v = 1, 2, \ldots, V \tag{13}$$

$$\lceil \Sigma_r e_{rv}/m_v \rceil \leq z_v; \quad v = 1, 2, \ldots, V \tag{14}$$

Moreover, we propose to use cuts based on the Cover Inequality (CI) approach [2]. Two constraints are taken into account, i.e., the processing limit (3) and the split limit (10). To limit the number of possible cover inequalities, we first solve linear relaxation of the model (1)–(10). Next, in the obtained solution, we identify variables x_{rv}, y_{rv} and u_{kv} that are not integer. For these variables, the CI approach is next applied in the context of constraints (3) and (10).

4 Results

In this section, we present results of computational experiments. The ILP model introduced in Sect. 2 was used to obtain the optimal results using the branch-and-cut algorithm offered by CPLEX 11.0 solver [8]. The goal of experiments was twofold. First, we examined how the split ratio and protection scope parameters influence the OPEX cost of the distributed computing system. Second, we evaluated efficiency of the additional cut inequalities. Experiments were run on a PC computer with IntelCore i7 processor and 4 GB RAM. Since there are no existing benchmark systems, we generated at random two sets of example systems according to parameter values presented in Table 1. For each size of the system, we created 12 different sets of systems. Moreover, 11 configurations of various protection requirements were investigated with the following values of the protected tasks percentage (PTP): 0, 10, 20, ..., 100 %. Thus, the overall number of individual cases tests was 132 (i.e., 12×11). Two values of the split ratio were used in the experiments: 4 and the maximum possible value (20 in the case of 20-node systems, and 30 in the case of 30-node systems, accordingly). Execution time of CPLEX was set to 20 min. Since, when using the default settings of the optimality gap (i.e., 0.0001), CPLEX was not able to stop calculations within this time limit, we set the optimality gap to 0.01 instead of the default value of 0.0001.

The first part of experiments was to examine the influence of the split ratio and protection scope on the overall OPEX cost. To show the aggregate results, for each unique system, we calculated the relative OPEX cost normalized using the

Table 1 Parameters of analyzed systems

	20-node systems	30-node systems
Number of computing nodes	20	30
Number of projects	10	20
Number of tasks per project	30–70	40–80
Cost of capacity module	120–400	120–400
Processing cost of one unit	50–150	50–150
Processing limit of one node	10–40	10–40
Number of destination nodes	1–4	1–6
Input and output data rates	5–15	5–15

Table 2 Average relative cost as a function of protected tasks percentage and split ratio

	20-node systems		30-node systems	
PTP (%)	$S = 4$ (%)	$S = 20$ (%)	$S = 4$ (%)	$S = 30$(%)
0	100	100	100	100
10	112	113	113	113
20	125	124	124	124
30	139	139	137	137
40	151	151	150	149
50	163	163	163	162
60	176	176	174	173
70	189	189	188	187
80	200	200	200	199
90	211	211	207	206
100	224	224	223	222

Table 3 Cut inequalities performance as a function of PTP

	No cuts		MIR		CI		MIR + CI	
PTP (%)	Time (s)	BB nodes	Time (s)	BB nodes	Time (s)	BB nodes	Time (s)	BB nodes
0	63	523	66	536	158	1,478	65	527
10	57	479	56	481	50	513	305	4,527
20	94	521	90	521	1,947	32,938	440	4,763
30	893	9,035	866	9,097	805	9,020	503	4,522
40	129	539	125	542	340	2,108	130	497
50	983	5,726	952	5,764	2,317	24,695	3,179	30,287
60	928	4,333	898	4,337	955	7,559	328	1,253
70	5,250	27,844	5,082	27,883	4,714	41,262	2,002	9,378
80	1,256	4,299	1,213	4,299	3,807	17,587	1,589	8,586
90	11,530	74,287	11,324	74,307	4,500	20,705	2,056	9,516

cost obtained for PTP = 0 %. The average value for each value of PTP was next computed. In Table 2, we report results for two sets of systems and two split ratio values.

The trend in Table 2 is similar for both types of systems, i.e., the relative cost grows linearly with the increase of the PTP parameter. Moreover, there is a very small difference between both presented values of the split ratio. Detailed comparison of two cases of the split ratio shows that the average gap between both values of the split ratio (i.e., 4 versus 20 (30)) is less than 1 %. Thus, the influence of the split ratio on the OPEX cost is very limited. Another important conclusion is that the average cost of systems with full protection (PTP = 100 %) is about 122–124 %, compared to the case of unprotected tasks (PTP = 0 %) for all reported cases.

The second goal of experiments was to verify the performance of additional cut inequalities. In Table 3, we report results in terms of execution time and the number

of B&B nodes obtained for one of 30-node systems. We can see that in general, application of both additional cuts (MIR and CI) reduces both the execution time and the number of B&B nodes. However, results of some individual cases show non-stability, i.e., additional cuts sometimes do not improve performance.

5 Concluding Remarks

In this paper, we addressed the problem of providing protection of information flows against failures of network elements forming the communication layer of distributed computing systems. In particular, we focused on multipath routing design with disjoint working and transmission paths, called $1 + 1$ protection scheme, that has never been addressed in the literature before. An ILP model was formulated to provide optimal routing results. It is worth noting that our method, enhanced with additional cut inequalities, turned out to be able to find solutions in short time.

Acknowledgements The work was supported in part by Ministry of Science and Higher Education, Poland, under the European Regional Development Fund, Grant No. POIG.01.01.02-00-045/09-00 Future Internet Engineering.

References

1. Akbari, B., Rabiee, H., Ghanbari, M.: An optimal discrete rate allocation for overlay video multicasting. Comput. Commun. **31**(3), 551–562 (2008)
2. Barnhart, C., Hane, C.A., Vance, P.H.: Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. Oper. Res. **48**(2), 318–326 (2000)
3. Buysse, J., De Leenheer, M., Dhoedt, B., Develder, C.: Providing resiliency for optical grids by exploiting relocation: a dimensioning study based on ILP. Comput. Commun. **34**(12), 1389–1398 (2011)
4. Develder, C., Buysse, J., Shaikh, A., Jaumard, B., De Leenheer, M., Dhoedt, B.: Survivable optical grid dimensioning, anycast routing with server and network failure protection. In: Proceedings of IEEE ICC 2011, Kyoto, pp. 1–5 (2011)
5. Grover, W.D.: Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking. Prentice Hall PTR, Upper Saddle River (2003)
6. Gunluk, O.: Branch-and-cut algorithm for capacitated network design problems. Math. Program. **86**, 17–39 (1999)
7. http://www.iip.net.pl
8. ILOG CPLEX, 12.0 User's Manual, France (2007)
9. Jaumard, B., Shaikh, A.: Maximizing access to IT services on resilient optical grids. In: Proceedings of 3rd International Workshop on Reliable Networks Design and Modeling (RNDM), Budapest, pp. 151–156 (2011)
10. Kacprzak, T., Walkowiak, K., Wozniak, M.: Optimization of overlay distributed computing systems for multiple classifier system – heuristic approach. Log. J. IGPL. doi:10.1093/jigpal/jzr020 (2011)
11. Marchand, H., Wolsey, L.: Aggregation and mixed integer rounding to solve MIPs. Oper. Res. **49**, 363–371 (2001)

12. Mitchell, J.: Branch-and-cut methods for combinatorial optimization problems. In: Handbook of Applied Optimization, Oxford University Press, Oxford/New York (2002)
13. Nabrzyski, J., Schopf, J., Weglarz, J. (eds.): Grid Resource Management: State of the Art and Future Trends. Kluwer Academic, Boston (2004)
14. Pioro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufmann, Amsterdam/Boston (2004)
15. Shen, X., Yu, H., Buford, J., Akon, M. (eds.): Handbook of Peer-to-Peer Networking. Springer, New York/London (2009)
16. Thysebaert, P., De Leenheer, M., Volckaert, B., De Turck, F., Dhoedt, B., Demeester, P.: Scalable dimensioning of resilient lambda grids. Future Gener. Comput. Syst. **24**(6), 549–560 (2008)
17. Vasseur, J.P., Pickavet, M., Demeester, P.: Network Recovery. Elsevier, Burlington (2004)
18. Wilkinson, B.: Grid Computing: Techniques and Applications. Chapman & Hall/CRC Computational Science. Chapman & Hall, London (2009)
19. Zhu, Y., Li, B.: Overlay networks with linear capacity constraints. IEEE Trans. Parallel Distrib. Syst. **19**(2), 159–173 (2008)