

Interactive Optimization of Path Planning for a Robot Enabled by Virtual Commissioning



Ruth Fleisch, Doris Entner, Thorsten Prante and Reinhard Pfefferkorn

Abstract Optimized path planning contributes to reducing the non-productive time of material handling in fully automated manufacturing. This paper presents a case study from the machine-tool industry sector about optimization of a path planning algorithm with the goal to minimize the time a material handling gantry robot requires to follow a feedback path, i.e. feeding a just cut part again to the saw which had just cut it, in order to realize more and more complex cutting patterns. Particularities of the case study configuration led to the application of an interactive optimization approach based on the definition and manipulation of rules for smoothing of initially planned paths and the exploration of the impacts of the rules on the time the material handling robot requires for traversing these paths by means of visual examination as well as by virtual commissioning. The achieved results were deployed in plants for cutting wooden or metal panels.

R. Fleisch (✉) · D. Entner · T. Prante
Design Automation, V-Research GmbH – Industrial Research and Development,
Stadtstraße 33, 6850 Dornbirn, Austria
e-mail: ruth.fleisch@v-research.at

D. Entner
e-mail: doris.entner@v-research.at

T. Prante
e-mail: thorsten.prante@v-research.at

R. Pfefferkorn
Schelling Anlagenbau GmbH, Gebhard-Schwärzler-Straße 34,
6858 Schwarzach, Austria
e-mail: Reinhard.Pfefferkorn@schelling.at

Introduction

Today's markets demand for customized products to fulfill increasingly client specific requirements. For cut-to-size plants, which automatically divide panels into requested sizes, the thereto necessary dealing with small lot sizes or lot-size one products results in a higher complexity of cutting patterns to enable efficient usage of the panels and thus producing as little waste as possible. As a consequence, the complexity of cut-to-size plant layouts increases as well, e.g. caused by feedbacks in their material flow (Fleisch et al. 2013). At the same time, besides high throughput and overall performance, a main requirement of operators of cut-to-size plants is a compact and space-efficient plant layout. To successfully tackle all of these requirements and challenges a new type of plant has been designed which implements material-flow feedback with reduced space usage. In order to realize the feedback loop, a gantry robot was chosen for panel transportation.

This paper reports the method and some results of minimizing the time required by this robot to move from start to end positions while at the same time obeying space limitations. Thereto, the robot must be able to move along any path and simultaneously rotate around the vertical axis while staying within spatial boundaries. While the finding of a valid path and aspects concerning the linking of the algorithm to the control system are described in detail in Fleisch et al. (2016), this paper focuses on the optimization of the path planning algorithm.

The following section describes the optimization problem and the rationales for choosing an interactive optimization approach. Section "[Gantry Robot and Integration into Plant Control System](#)" presents the material handling robot covered in the here presented case study and the integration of its control system into the overall plant control system. Section "[Related Work](#)" outlines related work, while the path planning algorithm and aspects concerning its optimization are introduced in Section "[Path Planning and Its Optimization Hook](#)". This is followed by a section about the optimization environment and procedure as well as, relating thereto, the virtual commissioning of the robot. Section "[Conclusion](#)" concludes the paper.

Optimization Problem and Approach

In order to determine which version of the path planning algorithm performs best, the following optimization loop was performed: The path planning algorithm generates a solution (a smoothed geometric path obeying certain spatial limitations), which, for evaluation, is fed into a simulation of the motion controller of the gantry robot. This simulation performs exactly the same task as the real motion controller does and therefore enables virtual production runs of a virtual plant with the real cutting patterns (i.e. virtual commissioning) towards reliable and exact temporal evaluation of the results of the respective version of the path planning algorithm. Depending on the results of this evaluation the adaption of the path planning algorithm is performed

or not (optimization loop). Adaption of the path planning algorithm is achieved in terms of definition and reworking, respectively a set of rules, which controls the smoothing of the initial result of the path planning step, i.e. a geometric path.

In other words, the objective function of the optimization problem assigns a time value calculated by means of virtual commissioning to a set of rules for smoothing a path. The behavior of the motion controller used for temporal evaluation of the results of path planning towards their optimization is unknown, i.e. a black box, in the path planning step. Hence, the optimization of a path with respect to time doesn't take place when planning a path during the operation of the plant, but instead is optimized upfront during the design and development of the robot control system in the context of overall plant (type) design and development.

The rationales for choosing an interactive approach towards fulfilling this task are the following. First, the smoothing rules to be applied in the path planning algorithm as well as their impact on the temporal behavior of the material handling robot moving from specified start to end positions were not fully understood. Both the determination of the right set of smoothing rules for geometrical manipulation of generated paths and the analysis of their impact on the temporal behavior of the robot are rather complex and partly creative tasks. Thus, in a first step, an environment needed to be created which allows knowledgeable domain experts to experiment with and understand the relationship between smoothing rules and the to-be optimized time variable. This forms the basis for designing in a next step a set of parameterized smoothing rules which will be more easily approachable by fully automated optimization procedures.

Second, it is as important for plant manufacturers to be able to exchange components or fabricators which contribute to their plants as it is not uncommon for technical systems (i.e. these components) that a new release manifests in changed behavior. Thus, plant manufacturers ensure internal process reliability by taking precautionary measures such as decoupling optimization environments from specific technical vendors and solutions, which in turn also works towards reusability of the optimization environment.

The third rationale for choosing an interactive optimization approach over a fully automated one for the described optimization task is due to the particularities of the development of control systems based on virtual commissioning. In daily practice, it is of utmost importance for gaining valid optimization results as to component behavior to keep up with the updates of each part of the overall plant system (including even overall plant process control), their interfaces and their interaction (see Fig. 8) during the rapidly changing design and development process of a plant component such as a material handling gantry robot tailored for optimized performance in cut-to-size plants employed in customer-specific manufacturing. Complementing what was stated as to the first rationale for choosing an interactive optimization approach, this makes it even more difficult to design a rich, realistic optimization model that describes all relevant aspects of the optimization problem in advance to actually performing the optimization.

These rationales led to the development of an interactive optimization environment with, on the one hand, visual support for domain expert users to analyze the impact of

smoothing rule set manipulation by comparing input and output of the path planning step and, on the other hand, reliable and exact temporal metrics provided by virtual production runs enabled by virtual commissioning.

Gantry Robot and Integration into Plant Control System

In order to realize that parts produced by sawing a panel into pieces can be fed again to the saw for further partitioning (i.e. feedback loops), in traditional cut-to-size plants, angular transfer conveyors allowing movements of panels in straight lines and turning in right angles only as well as turning devices for rotating a panel have typically been used. In new more compact and space-efficient cut-to-size plants with feedback in the material flow panels need be moved in the form of a curve while they are rotated at the same time. Therefore, a new material handling robot has been developed for transporting panels, e.g., from a saw to a feedback conveyor and vice versa (see Fig. 1).

The gantry robot picks up a panel with the aid of a suction unit and then pulls it horizontally over a brush table from a starting to an end point within a few seconds. During this movement, the robot can simultaneously rotate around the vertical axis. To prevent collisions with other panels or plant components, spatial limitations induced by the plant itself as well as by panels moving in it have to be met which can vary for each piece of material to be transported. To this end, adaptive and intelligent path planning is required to provide a fully automated and optimized transportation system with feedbacks in the material flow.

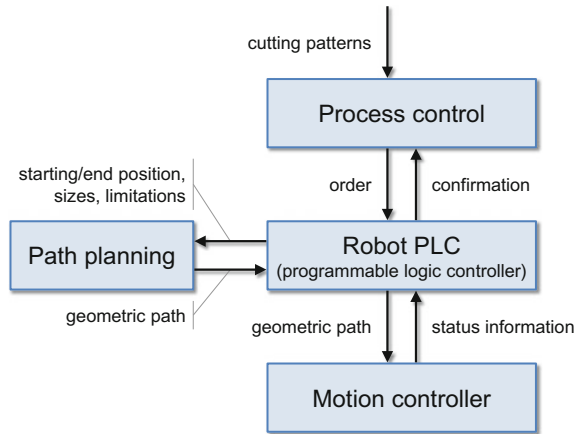
The path planning for the robot has to be incorporated into the central hierarchical control system of the plant. In the presented case study, it is linked to the robot PLC, since at this level the information necessary for the input to the path planning algorithm is available (for an overview, see Fig. 2).

The inputs to the highest layer of the control system, i.e. the process control, are cutting patterns, according to which the panels or their parts, respectively are cut by the saw and run through the plant. The process control logic controls the material flow in the plant by sending orders to PLCs (programmable logic controllers) of the different

Fig. 1 Gantry robot transporting a panel outputted by a saw towards other panels on a feedback conveyor



Fig. 2 Integration of the path planning algorithm into a hierarchical control system at PLC level



plant components like the saw itself, roller tracks, or the materials handling robot. Examples of such orders for the robot PLC are “travel to the part which is just cut and pick it up” or “transport the part from the saw to feedback conveyor and put it down”.

According to the orders, the robot PLC controls for example the suction unit of the robot or defines and transmits the input for the path planning algorithm, consisting of the starting and end position as well as panel and (robot) mechanical system size and spatial limitations. For each movement of the robot a solution path, consisting of a list of coordinates and rotation angles, has to be calculated by the path planning. This (smoothed) path is transferred to the robot PLC which passes it on to the motion controller together with other information.

The motion controller subject to this case study does not calculate a valid, collision-free geometric path itself, but requires one as input, on the basis of which it performs the trajectory planning, i.e. calculating a time-parameterized curve in a black box manner. Moreover, the motion controller controls the actuators for movements in the x-y-plane and rotations as well as sends status information back to the robot PLC.

After completing the movement of the robot from the starting to the end position, the robot PLC confirms completion of the order to the process control.

Related Work

The related work for the presented optimization of the path planning algorithm comprises three areas: motion planning, virtual commissioning as well as empirical and interactive optimization.

The problem of robot motion planning can be divided into three sub-problems: path planning, i.e. the specification of the geometric path avoiding obstacles, trajec-

tory planning, i.e. the specification of the time evolution along this geometric path, and path tracking as performed by the low-level control loops (Haschke et al. 2008). In the following, only the related work as to path planning and trajectory planning is discussed. Concerning path planning, there are several approaches like potential-field-based techniques, combinatorial methods producing roadmaps or sampling-based planning. The latter has been successfully applied in practice for many challenging problems (Siciliano and Khatib 2008) and a variant of it is realized within the project presented in this paper. Sampling-based planners rely on a collision checking module instead of explicitly representing the environment (Karaman and Frazzoli 2011) and sample the space of all possible placements of the robot for the collision-free ones (LaValle 2006).

Finding not only a feasible path for the robot, but also one that optimizes one or more criteria for a given high-level task is an important issue (Luna et al. 2013). If the optimization shall be with respect to time, trajectory planning is commonly involved. In Wu et al. (2000), for example, first, a shortest path composed of circular arcs and straight lines is obtained for a wheeled mobile robot and then, a time optimal velocity profile is generated. In Haschke et al. (2008), time-optimal trajectories are also determined based on a given geometric path. In contrast, a given geometric path is smoothed additionally in the course of planning time-optimal trajectories in Hauser and Ng-Thow-Hing (2010). In Ratliff et al. (2009), the requirement that the input path for trajectory planning has to be collision-free is even dropped. Instead, avoiding obstacles is included in planning trajectories which optimize over a variety of dynamic and task-based criteria. Finally, Shareef and Trächtler (2016) present a method for simultaneous path planning and trajectory optimization.

If it is sufficient to regard the path length as proxy for travel time (Richardson and Olson 2011), only the path planning problem itself can be considered for optimization. Besides shortening the path (Luna et al. 2013; Campana et al. 2015), different metrics like smoothness or obstacle clearance (Luna et al. 2013; Richardson and Olson 2011) or more general cost functions (Karaman and Frazzoli 2011; Campana et al. 2015) can be applied for optimizing a geometric path. Kretschmann (2007) presents an example of time-optimization, which uses geometrical features of the path, in order to avoid time-consuming calculation of the travelling time. The relationship between the geometrical parameters and the time-optimal trajectories is elaborated with the help of simulation. Theoretical considerations presented by Kretschmann prove that the cost function based on the geometrical parameters is a sufficiently reliable indicator for the travelling time.

The solution approach presented in this paper follows the idea of deciding which rules are best for smoothing the geometric paths within path planning by evaluating the rules as to temporal behavior of the robot. This optimization procedure is enabled by virtual commissioning of the robot, which involves the trajectory planning.

The purpose of virtual commissioning is to test manufacturing systems together with their control programs through simulation to produce reliable and precise behavioral forecasts before on-site installation and ramp-up (Hoffmann et al. 2010). Beyond that, virtual commissioning enables optimization in relation to control systems. In Svensson et al. (2012), for example, a simulation-based optimization method is pre-

sented where a combination of real industrial control systems and of a virtual manufacturing system is used for the simulation part. In their case study based on this test bed, optimization was applied for tuning process parameters of an automotive sheet-metal press line.

Empirical optimization is used in the field of code compiling, especially for library generation, to choose parameter values or even an algorithm from a suite of algorithms by generating different versions of the program, running all of them on the actual hardware and selecting the version which results in the best performance (Yotov 2003; Epshteyn 2006). This idea of evaluating multiple code versions is also applied in the project presented in this paper, however, a part of the test system is not the real one but virtual, i.e. the motion controller simulation.

Interactive optimization means that the user participates actively in the optimization process. More precisely, “an interactive approach recognizes some limits to modeling and parameter setting in a real situation, and values the user’s expertise in the application domain that can be exploited by the optimization system” (Meignan et al. 2015). This means, that “with an adequate interaction between an optimization system and its users, the optimization model can be enriched to fit the real problem, the search process can be guided for improving its efficiency, and the user can better understand the system” (Meignan et al. 2015).

In Meignan et al. (2015), a classification of interactive optimization methods is proposed according to the purpose of the interaction and the role of the user: In the case of problem-oriented interaction, the user aims at modifying the optimization problem by adjusting the existing constraints or objectives or by enriching the problem, i.e. defining new constraints or objectives. The target of search-oriented interaction is improving the performance of the optimization procedure. The user can achieve this by tuning strategic parameters of the optimization procedure, by supporting the search with information related to decision variables (guiding), or by acting as a search procedure (assisting).

The here presented optimization approach belongs to the assisting-category, as the user has to define, implement and rework a set of smoothing rules, that is the next element of the solution space for which the time has to be calculated. On the other hand, the approach belongs to the guiding-category as well, because the user analyzes the link between the implemented set of rules and the time for executing the path and uses this information to change the set of rules.

Path Planning and Its Optimization Hook

Before discussing the set of smoothing rules as the hook for optimizing travel time of the material handling robot for a path computed by the path planning algorithm, this section, first, summarizes details of the path planning algorithm and its specific interplay with the motion controller deployed in this case study. Further information on the latter topic can be found in Fleisch et al. (2016).

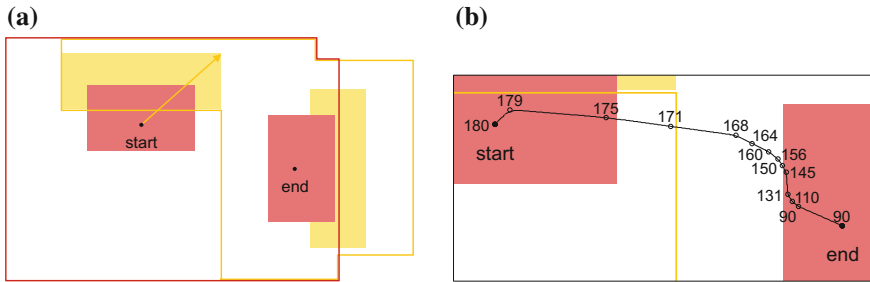


Fig. 3 **a** Panel (yellow, size: 2160 mm \times 750 mm, arrow marks the offset) and part of the mechanical system (red), their limitations (solid lines), and rotation point in starting and end position. **b** A solution path for the rotation point of the robot. The rotation angles for each point are given in degrees

Path Planning and Its Interplay with the Motion Controller

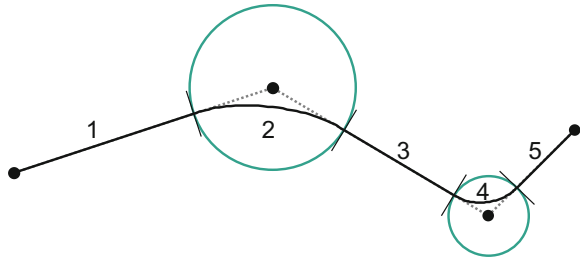
The developed path planning algorithm computes a smoothed geometric path for every movement of the material handling robot from a starting to an end point in the horizontal plane. Such a path avoids obstacles and contains information for simultaneous rotation around the vertical axis of the robot. The movement can take place with or without material, the latter, e.g., in order to collect a panel.

The path planning algorithm comprises the following three steps: (step 1) computation of a valid geometric path, (step 2) smoothing this path (see Fig. 6), and (step 3) determining a feasible blending parameter for each position of the path (see Fig. 4). Subsequently, the final curve to be travelled along by the robot (i.e. trajectory planning) is determined by its motion controller, which calculates in a black box manner a time-parameterized curve corresponding to the input it received from the path planning algorithm.

Here, we first describe input and output of the path planning algorithm, followed by the challenges related to the black box calculation of the time-parameterized curve by the motion controller and how they influence the path planning algorithm.

Figure 3a visualizes the main input data to the path planning algorithm of a real-world example: A panel to be transported is always rectangular and must not exit a given area. This ensures that collisions are prevented, e.g. of the panel with other panels in the plant or with plant components. A part of the mechanical system of the robot is also represented as a rectangle and must stay inside its own limitations. The admissible areas for the rectangles are simple polygons, featuring internal angles of 90° or 270° . Thus, in the case of transporting a panel, the main input parameters for the path planning algorithm are the sizes of both rectangles, the offsets of the rectangles relative to the rotation point of the robot, the spatial limitations, and the starting and end position. A position is composed of the x- and y-coordinates of the rotation point of the robot in the horizontal plane and further includes the associated rotation angle for this point.

Fig. 4 The respective blending parameters are the radii of the depicted circles. 1, 3 and 5 are linear segments, 2 and 4 are polynomial segments



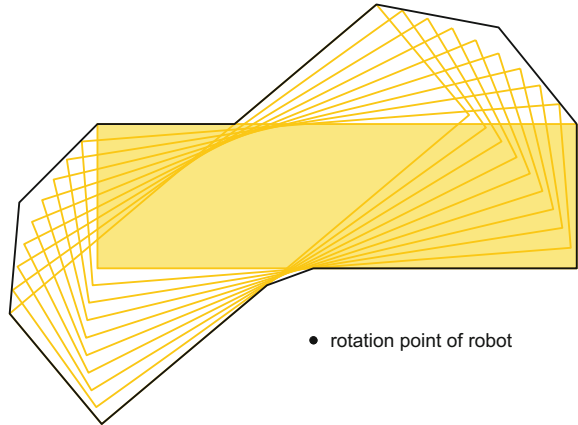
The output of the path planning algorithm is a smoothed geometric path for the rotation point of the robot from the starting to the end position such that the relevant spatial limitations are not violated (see Fig. 3b). The geometric path is a list of positions where the number of positions must not exceed 25—this requirement is introduced by the motion controller. Each position is enriched by a blending parameter (step 3) which indicates the extent of blending of the movement along the polygonal chain at the respective point (see Fig. 4).

The list of positions together with the blending parameter for every point is passed onto the motion controller of the robot, where it serves as a basis for generating a trajectory and for controlling the movement. The points of the path in the x-y-plane are connected by linear interpolation with a polynomial blend (see Fig. 3b). Thus, the plane curve consists of linear and polynomial segments.

If the polygonal chain is blended at a point, the rotation angles corresponding to the transition points between the blending segment (e.g. segment 2 in Fig. 4) and the preceding (segment 1) as well as the subsequent (segment 3) linear segment are calculated by comparing the proportions of the path length. However, the rules of how the rotation angle is interpolated are not available for path planning, since they are implemented in conjunction with the trajectory planning in the motion controller. Thus, the matching of rotation angles to the plane curve can be computed only at the transition points of the segments. Therefore, it has to be assumed that, in theory, the rotation to be executed along a segment can take place at any arbitrary point of that segment. This has an effect as to collision avoidance and can result in, e.g., a higher number of positions required for specifying a path in case of tight spatial limitations and only little space left to route. Dealing with this effect is explained below.

For computing a valid geometric path a variant of a sampling-based planner was implemented which samples the space of all possible placements of the robot for the collision-free ones (Fleisch et al. 2016). The sampling is realized as an incremental search, where the global search direction is along the medial axis of the polygonal limitation. In the case that a panel is moved, the admissible area of the panel is used for determining the medial axis as the material is usually more critical than the mechanical system regarding the available space. In a step of the incremental search, the next feasible position is found by varying the position or the step size. For the purpose of checking the feasibility of the next position, i.e. collision detection, the area which is covered by a rectangle rotating from the starting to the end angle of

Fig. 5 Polygon (black) encompassing a rotating rectangle



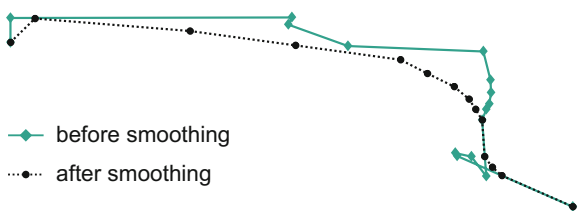
the step is approximated by a surrounding polygon (see Fig. 5). This polygon must not be outside the limitations when it is moved along the segment from the current to the next point.

Optimization Hook: Set of Smoothing Rules

As the behavior of the motion controller concerning trajectory planning and interpolation of the rotation angle is unknown, i.e. a black box, in path planning (steps 1-3), the optimization of a path with respect to time is not possible within the path planning step. Instead, improvements can only be based on geometrical features of the path. Since a smoother path is assumed to be typically more time-efficient, the task to be tackled by interactive optimization is to determine which rules for smoothing the paths shall be implemented in order to minimize the time which the robot needs for following the paths.

As elaborated in Section “[Path Planning and Its Interplay with the Motion Controller](#)”, the path planning algorithm integrates the smoothing by first computing a valid geometric path (step 1), then smoothing the solution (step 2) and, finally, determining a feasible blending parameter for each position (step 3). Figure 6 shows an example of an initial valid path and, based on it, a smoothed path.

Fig. 6 Valid path before and after smoothing (rotation angles are not depicted)



The input for the smoothing part of the algorithm is a valid geometric path, which means a polygonal chain together with a rotation angle for each of its points. Following this path ensures that both the panel and the modeled part of the mechanical system do not exit their admissible areas. The polygonal chain is iteratively smoothed by eliminating or modifying positions based on smoothing rules while maintaining the spatial limitations. An example for a rule which chooses a position for eliminating or changing, is the selection of the most acute angle, formed by the line segments of the polygonal chain. Further options for changing a position are for instance translating a point to the centroid of the triangle with the chosen point and the preceding as well as the subsequent point of the polygonal chain as vertices or replacing a position by the center of the preceding and subsequent position (both the center in the x - y -plane and the center of the rotation angle). Further rules are based among other aspects on the following heuristics for smoothing a path:

- the elimination of self-intersections of the polygonal chain,
- shortening the total length of the polygonal chain,
- avoidance of short line segments of the polygonal chain,
- the reduction of the number of positions, or
- a more uniform distribution of the rotation angles along the polygonal chain.

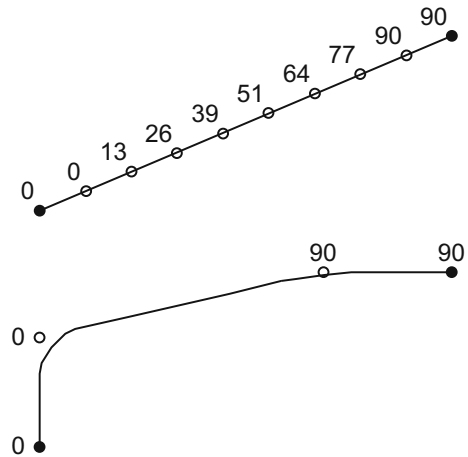
For the determination of the best rules to be implemented, an interactive optimizing procedure is applied which includes virtual production runs.

Optimization Environment and Procedure

As illustrated just before, the optimization as to the time the material handling gantry robot needs to follow a path is carried out by interactive definition of the smoothing rules. One of the challenges in doing so comes from the fact that it is not necessarily the shortest path yielding the shortest travel time since, for example, the robot has to slow down for turning sharply in the x - y -plane and hence a longer path with obtuse angles can be preferable to a shorter one with acute angles. Thus, the optimization of the path planning provides a possibility to find a trade-off between the path length and the sizes of the path angles for example.

Another illustrative example which seems counter intuitive at first sight is depicted in Fig. 7. The two shown paths are the outcome for the same input data but determined by two different sets of smoothing rules. This example again reveals that a mere visual examination does not suffice but the exact calculation of times by virtual commissioning of the robot is relevant, since the latter delivers an unexpected result: Due to the behavior of the motion controller, a straight line in the x - y -plane defined by 10 positions is inferior regarding time to a non-linear, longer path specified by only 4 positions, where both paths represent a rotation by 90° and the distance between the starting and end point is 1.69 m. It takes the robot 8.04 s to follow the straight line and 7.21 s to follow the non-linear path, yielding a difference of about 10%. This illustrates that the number of positions for defining a path has a great impact on the

Fig. 7 It takes the robot more time to follow the first path than to follow the second one. Rotation angles in degrees are depicted



material-handling time for the robot (motion controller) at hand. The high number of positions of the linear path is required for distributing the rotation angle along the straight line due to the close proximity of the boundaries.

To get a first glance of the impact of the smoothing rules on the geometry of the paths, the paths are visually examined. In addition, for the optimization, the exact execution times for moving the panels are determined. Since this calculation and data concerning temporal aspects such as velocity, acceleration, or jerk are not available at the path planning level, the motion controller of the material handling gantry robot with its trajectory planning is necessary, as the times depend on its behavior. Consequently, the evaluation with respect to time of the implemented rules is performed by virtual commissioning of the robot.

Virtual Commissioning of the Material Handling Robot

Apart from selecting the best rules for smoothing within the frame of path planning and thereby optimizing the material handling robot with regard to its temporal behavior, virtual commissioning of the robot facilitates also the general development of the path planning algorithm, testing its integration into the control system as well as the coordination of the interaction between the robot PLC and the motion controller. This leads to improved quality of the algorithm and reduced commissioning time and costs of the plant.

In order to establish a system for optimizing the path planning, an industrial computer has been set up where the process control, the PLC and the path planning algorithm are installed. The same computer can be used later in the real plant as well. In contrast, a simulator of the motion controller is employed in place of the real one. Figure 8 depicts the components of the system for virtual commissioning of the robot.

Fig. 8 For optimization, a simulator of the motion controller is used

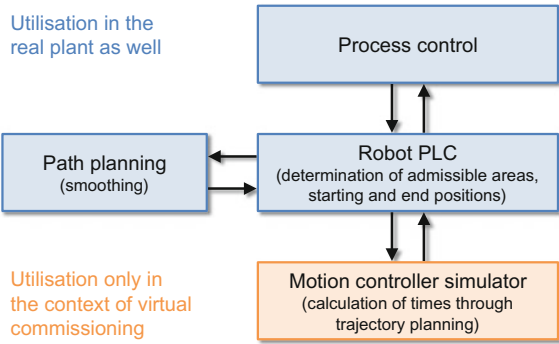
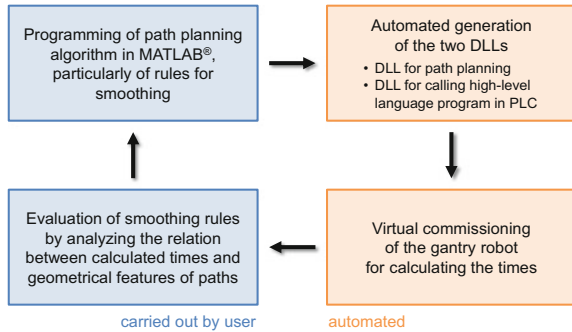


Fig. 9 Optimization loop



The challenge related to virtual commissioning of the robot is to keep up with the updates of each component of the system, their interfaces and their interaction during the rapidly changing design and development phase of the robot.

Details of the Implementation and of the Optimization Loop

As evaluating of the path planning algorithm in the frame of interactive optimization should be as efficient as possible, the path planning algorithm is implemented by using a high-level programming language instead of directly programming in PLC code. This is more suitable for developing complex algorithms. In the present case, the algorithm is developed in MATLAB®. MATLAB Coder™ generates C and C++ code from MATLAB® code and, in a second step, optionally machine-readable code in the form of a DLL (dynamic link library). In order to call high-level language programs from the control program, another DLL is required. The generation of the two DLLs based on the high-level programming language code of the algorithm is automated so as to enable more efficient working with the optimization system and that modifications of the rules in the algorithm are ready for evaluation quickly. Figure 9 gives an overview of the interactive optimization procedure.

The loop starts with defining the smoothing rules and programming them in MATLAB[®]. This has to be accomplished by the user, whereas the two DLLs are automatically generated on the basis of the MATLAB[®] code and are made available for deployment on the system for virtual commissioning. In order to obtain the traversal times of the robot, a production run is simulated by means of the virtual commissioning system. For this, the production list in form of cutting patterns is loaded in the graphical user interface of the process control and the virtual production is started. A log file is created for every path which contains the input and output data of the path planning algorithm and data recorded by the simulator of the motion controller. The recorded data is a list of time stamps and for each time stamp the corresponding position of the robot (x- and y-coordinates of the rotation point and rotation angle). With this data, the movement of the robot (with or without a panel) following the path can be visualized. It is the same motion as in the real world or as depicted in the graphical user interface of the process control, but with the advantage that the spatial limitations and the solution path are illustrated. The visualization supports the analysis of the relation between the calculated times and the geometrical features of the path and thus, facilitates definition and evaluation of the smoothing rules. Based on the findings, the set of smoothing rules can be reworked.

The MATLAB[®] code of the algorithm is structured in such a way that only local alterations are necessary for implementing new smoothing rules which keeps the effort to a minimum and contributes to the robustness of the code. Together with the automated generation of the DLLs, this achieves reduced amount of user input for optimization and thus increases the efficiency of the optimization loop.

Conclusion

The trend towards customer-specific production, for example in the furniture industry, has led to the development of cut-to-size plants with feedbacks in their material flow. In order to realize feedbacks in a compact and space-efficient way as well as to achieve the target of time-optimized panel handling, a new material handling gantry robot has been developed. This paper presented an interactive approach to optimizing the path planning algorithm of the robot as to its temporal behavior when traversing planned paths. Thereto, the smoothing rules applied to initially generated paths were used as optimization hook. As traversal times of the robot are not available in the path planning step, a system for virtual commissioning of the robot was established. It enables virtual production runs and thereby the temporal evaluation of the smoothing rules in order to find the optimal ones as to their impacts on the behavior of the robot. During the optimization loop user input is required, on the one hand, for analyzing the relation between geometrical features of a path and the calculated time which the robot needs to follow the path and, on the other hand, for adapting the smoothing rules of the path planning algorithm. Virtual commissioning has the advantage that it can be conducted already during the design and development process of the materials handling gantry robot and not as late as during commissioning on a customer's site.

Potential extensions and future work concerning path planning optimization include the automation of the evaluation of the smoothing rules (lower box on the left-hand side of Fig. 9) to improve and accelerate the utilization and operation of the optimization system.

Acknowledgements This work was carried out within the COMET K-Project #843551 “Advanced Engineering Design Automation (AEDA)” funded by the Austrian Research Promotion Agency FFG.

References

- Campana M, Lamiroux F, Laumond J-P (2015) A simple path optimization method for motion planning. Rapport LAAS n 15108
- Epshteyn A et al (2006) Analytic models and empirical search: a hybrid approach to code optimization. In: Proceedings of the 18th international conference on languages and compilers for parallel computing. Springer, Berlin, pp 259–273
- Fleisch R, Schöch R, Prante T, Pfelegerl R (2013) Consistent use of emulation across different stages of plant development—The case of deadlock avoidance for cyclic cut-to-size processes. In: Winter simulation conference (WSC 2013), pp 2565–2576
- Fleisch R, Schöch R, Prante T, Pfefferkorn R (2016) A path planning algorithm for a materials handling gantry robot and its validation by virtual commissioning. In: Advances in manufacturing technology XXX, pp 169–174
- Haschke R, Weitnauer E, Ritter H (2008) On-line planning of time-optimal, jerk-limited trajectories. In: 2008 IEEE/RSJ international conference on intelligent robots and systems, pp 3248–3253
- Hauser K, Ng-Thow-Hing V (2010) Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts. In: 2010 IEEE international conference on robotics and automation, pp 2493–2498
- Hoffmann P, Schumann R, Maksoud TM, Premier GC (2010) Virtual commissioning of manufacturing systems a review and new approaches for simplification. In: ECMS, pp 175–181
- Karaman S, Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *Int J Rob Res* 30:846–894
- Kretschmann R (2007) Zeitoptimale Bahnplanung für Industrieroboter. Available at: [http://www.mftech.de/diplom_zeitoptimale_bahnplanung_industrieroboter_uebersicht\(2007\).pdf](http://www.mftech.de/diplom_zeitoptimale_bahnplanung_industrieroboter_uebersicht(2007).pdf). Accessed 27 Apr 2017
- LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge
- Luna R, Şucan IA, Moll M, Kavraki LE (2013) Anytime solution optimization for sampling-based motion planning. In: 2013 IEEE international conference on robotics and automation, pp 5068–5074
- Meignan D, Knust S, Frayret J-M, Pesant G, Gaud N (2015) A review and taxonomy of interactive optimization methods in operations research. *ACM Trans Interact Intell Syst* 5:17:1–17:43
- Ratliff N, Zucker M, Bagnell JA, Srinivasa S (2009) CHOMP: gradient optimization techniques for efficient motion planning. In: IEEE international conference on robotics and automation
- Richardson A, Olson E (2011) Iterative path optimization for practical robot planning. In: 2011 IEEE/RSJ international conference on intelligent robots and systems, pp 3881–3886
- Shareef Z, Trächtler A (2016) Simultaneous path planning and trajectory optimization for robotic manipulators using discrete mechanics and optimal control. *Robotica* 34:1322–1334
- Siciliano B, Khatib O (2008) Springer handbook of robotics. Springer Science & Business Media, Berlin
- Svensson B, Danielsson F, Lennartson B (2012) Time-synchronised hardware-in-the-loop simulation—Applied to sheet-metal press optimisation. *Control Eng Pract* 20:792–804

- Wu W, Chen H, Woo P-Y (2000) Time optimal path planning for a wheeled mobile robot. *J Rob Syst* 17:585–591
- Yotov K et al (2003) A comparison of empirical and model-driven optimization. In: *Proceedings of the ACM SIGPLAN 2003 conference on programming language design and implementation*. ACM, New York, pp 63–76