2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Increasing Efficiency of Optimization-based Path Planning for Robotic Manipulators

Hao Ding[†*], Gunther Reißig[⋆], and Olaf Stursberg[†]

Abstract— **Path planning for robotic manipulators interacting with obstacles is considered, where an end-effector is to be driven to a goal region in minimum time, collisions are to be avoided, and kinematic and dynamic constraints are to be obeyed. The obstacles can be time-varying in their positions, but the positions should be known or estimated over the prediction horizon for planning the path. This non-convex optimization problem can be approximated by Mixed Integer Programs (MIPs), which usually leads to a large number of binary variables, and hence, to inacceptable computational time for the planning. In this paper, we present a geometric result whose application drastically reduces the number of binary decision variables in the aforementioned MIPs for 3D motion planning problems. This leads to a reduction in computational time, which is demonstrated for different scenarios.**

I. Introduction

Path planning of robotic manipulators has been investigated taking into account safety, efficiency, and optimality. In general, the task can be summarized as follows: *Given an initial configuration, the best way (with respect to a given performance criterion) of driving the end effector of a robotic manipulator into a specified goal region has to be determined. Kinematic and dynamic constraints, including those that guarantee the avoidance of collisions with obstacles, must be obeyed. The obstacles can be time-varying in their positions, but the positions should be known or estimated over the prediction horizon for planning the path.*

Since this non-convex optimization problem is hard to be solved efficiently, several motion planning algorithms have been proposed which are based on solving approximations to the original problem. A real-time motion planning method using potential fields by combining repulsive potentials for obstacles with an attractive potential of the goal was proposed in [1]. Such a method can be applied online, but optimality of motion is not considered and the inclusion of robot kinematics and dynamics is difficult. Methods operating in the configuration space, such as cell decomposition [2] and mixed-integer programming [3], take advantage of the fact that the configuration of the robot reduces to a single point. However, to find proper and efficient obstacle representation is a big challenge for robots with many degrees of freedom [4]. To account for the drawbacks of these two classes of methods, sampling-based approaches like rapidly-exploring random trees (RRTs, [5]) and probabilistic roadmaps (PRMs, [6]) have been proposed. The idea is to plan the path in the configuration space, but checking collisions in the workspace using forward kinematics. As variants, kinodynamic planning [7], RRT* [8], elastic roadmaps [9], and model-based optimization embedded into RRTs [10] have been investigated. However, all sampling-based approaches rely on a quantization of the configuration space, of which only a finite number of samples is considered during motion planning.

In addition to the previously mentioned techniques, it has been proposed to solve mixed-integer linear program (MILP) approximations of path planning problems directly in the workspace, where selected points on the links (*particles*) are used to represent the robot geometry as well as obstacle avoidance constraints [11]. The technique has been extended in [12] to incorporate state-dependent and time-varying constraints. Depending on the number of particles used, this approach often leads to a large number of binary variables, and hence, to inacceptable computational time for the planning. For 2D scenarios, we have recently presented a geometric result whose application to the aforementioned MILPs drastically reduces the number of binary variables as well as the computational effort for solving path planning problems. In this paper, we extend that result to 3D scenarios. For the sake of simplicity, only time-optimal control will be considered, i.e., the time for steering the end effector from a prescribed initial position to a goal is to be minimized, but the approach we propose also applies to more general cost functions. We represent the dynamics of the robot by bounds on the joint velocities. This is justified because the joint positions of industrial robots are usually controlled by built-in lower-level controllers. Depending on the capability of the latter, bounds on velocities can be chosen to guarantee that the robot can successfully track the optimized trajectories. This simple dynamic model is also useful, e.g. for human robot interaction and collaboration, where additional constraints on the joint velocities are imposed depending on the obstacle movement.

The remainder of this paper is organized as follows: In Section II, the problem of optimal motion planning for robotic manipulators in the workspace is defined and approximated by an MILP. In Section III, we present our main result, a theorem related to the three dimensional geometric aspects of collision avoidance, whose application yields equivalent MILPs with a drastically reduced number

[†]Institute of Control and System Theory, Dept. of Electrical Engineering and Computer Science, University of Kassel, Germany, {hao.ding, stursberg}@uni-kassel.de.

[⋆]Chair of Control Eng. (LRT-15), Dept. Aerospace Eng., University of the Federal Armed Forces Munich, D-85577 Neubiberg (Munich), Germany, http://www.reiszig.de/gunther/

[*]Group of Robotics and Manufacturing, ABB Corporate Research, Ladenburg, Germany

This work is partially supported by the project EsIMiP funded by the Bavarian Research Foundation (AZ-852-08).

of binary decision variables. This leads to savings in the computational effort for solving motion planning problems, which is demonstrated in Section IV for a 2-link robotic manipulator interacting with obstacles in three dimensions.

II. Motion Planning Problem in the Workspace

The following notation is used throughout the paper:

- *Workspace*: the Euclidean space $Z = \mathbb{R}^3$ with points $(z^1, z^2, z^3)^T \in Z$.
- *Configuration space*: $C = \mathbb{R}^n$; points $(q_1, q_2, \ldots, q_n)^T \in C$ represent joint angles of the robotic manipulator with n degrees of freedom.
- $z_j \in Z$ denotes the position of joint j including the position of the end effector in the workspace with $j \in \{1, \ldots, n+1\}$.

The idea of motion planning considered in the paper is to determine optimized trajectories of joint positions in the workspace in order to drive the end effector to the goal set. The paths of all robot links has to be free of collisions with the obstacles. Accordingly, the motion planning problem in the workspace can be formulated as:

$$\min \quad \sum_{\tau=0}^{g} J(z(\tau), \tau) \tag{1a}$$

$$s.t. \quad (z_j(t+1) - z_j(t)) \in$$

$$[V_j^-(z(t), F(t)),\ V_j^+(z(t), F(t))] \cdot \Delta t, \tag{1b}$$

$$\forall t \in \{0, 1, \ldots, g-1\}, \quad \forall j \in \{1, \ldots, n+1\} \tag{1c}$$

$$z_j(0) = (z_j^{1,0}(0), z_j^{2,0}(0), z_j^{3,0}(0))^T,\ z_{n+1}(g) \in G, \tag{1d}$$

$$H(z(t)) \cap F(t) = \emptyset, \quad \forall t \in \{0, 1, \ldots, g\}, \tag{1e}$$

where $z = (z_1, \ldots, z_{n+1})$ and:

- the joint velocities in the workspace are bounded to $[V_j^-(z(t), F(t)),\ V_j^+(z(t), F(t))]$;
- Δt denotes the sampling time;
- the initial joint positions in the workspace $z_j(0)$ and the goal set G for the end effector $z_{n+1}(g)$ are specified;
- $H(z(t))$ represents the geometry of the robotic manipulator (including the kinematics), which must not intersect with the forbidden region $F(t)$.

The above optimization problem is non-convex. We here follow the approach initially proposed in [11], where selected points of the robot geometry (*particles*) are used to formulate obstacle avoidance constraints, and the overall non-convex optimization problem is approximated by a mixed-integer linear program (MILP).

A. Collision Avoidance

1) For a Single Point: The condition that a point lies outside a polyhedral obstacle can be formulated as a mixed-integer linear constraint, as is explained below.

Let the obstacle $\mathcal{P} \subseteq Z = \mathbb{R}^3$ be a polyhedron, i.e., a finite intersection of half-spaces,

$$\mathcal{P} = \{\xi \mid A \cdot \xi < b\}, \tag{2}$$

where $A \in \mathbb{R}^{N \times 3}$ is a matrix with rows A_k, $k \in \{1, \ldots, N\}$, and $b = (b_1, \ldots, b_N)^T$. The condition that the point ξ lies outside the obstacle $\mathcal{P}$, $\xi \notin \mathcal{P}$, is equivalent to the requirement that at least one among the N scalar inequalities in $A \cdot \xi \geq b$ is satisfied. This, in turn, is equivalent to the set of constraints

$$A \cdot \xi \geq b + (v - \mathbf{1}) \cdot M, \tag{3}$$

$$\sum_{k=1}^{N} v_k = 1, \tag{4}$$

where $v = (v_1, \ldots, v_N)^{\mathrm{T}}$ is a vector of binary variables $v_k \in \{0, 1\}$ and $\mathbf{1} = 1^{N \times 1}$ is a vector of ones. M is a sufficiently large constant[1], hence the name 'big-M method' [13]. The constraint (3) says that the kth scalar inequality in $A \cdot \xi \geq b$ is enforced if and only if $v_k = 1$, and the constraint (4) ensures that the latter equality holds for exactly one k.

If (3) and (4) are constraints in an optimization problem, points ξ (e.g. joint positions) in the obstacle $\mathcal{P}$ are infeasible, and hence, collision is avoided. In the presence of multiple obstacles, an analogous set of constraints, with separate matrices A and b and separate binary variables v for each obstacle, can be used.

2) For Robotic Manipulators: The constraints in (1e) have been introduced to avoid collisions with obstacles. Assuming that the links form straight lines between adjacent joints, a finite number of particles are chosen along the link in order to represent the geometry of the robot, namely $H(z(t))$ in (1e). The particles are chosen by:

$$p_{sj} = \lambda_s \cdot z_j + (1 - \lambda_s) \cdot z_{j+1}, \tag{5}$$

where p_{sj} denotes the sth particle on link j. By selecting $\lambda_s \in [0, 1]$ with $s \in \{1, \ldots, S\}$, where S is the number of particles per link, the position p_{sj} of the particle is defined. In order to guarantee that none of the introduced particles is contained in $\mathcal{P}$, we substitute p_{sj} for ξ in (3) to arrive at the constraint

$$A_k \cdot p_{sj}(t) \geq b_k + (v_{j,s,k}(t) - 1) \cdot M \tag{6}$$

for each face k of the obstacle $\mathcal{P}$, each link j, each particle s associated with link j, and each time $t \in \{0, \ldots, g\}$. In addition, the constraint

$$\sum_{k=1}^{N} v_{j,s,k}(t) = 1 \tag{7}$$

is required for each j, s and t. Here, $v_{j,s,k}(t)$ is a binary variable to be introduced in the MILP as a decision variable, whereas $p_{sj}(t)$ is just an abbreviation for the term (5) containing the decision variables $z_j(t)$.

The number of particles has to be chosen such that the geometry of any link is represented with sufficient accuracy. For compensating the error made by approximating collision avoidance constraints with the help of particles, the obstacle $\mathcal{P}$ is to be enlarged by a suitable safety margin, which will be discussed in Sec. IV.

[1]The way of choosing the value of M can be found in [13, p. 205].

B. Kinematic and Dynamic Constraints

Since the joints of the robot are connected by straight links, the kinematic constraints imply that the distance between the joints q_j and q_{j+1} is equal to the length $r_{j,j+1}$ of the corresponding link. This requirement can be represented by the constraints

$$(z_{j+1}(t) - z_j(t))^T \cdot (z_{j+1}(t) - z_j(t)) = r_{j,j+1}^2 \quad (8)$$

with $j \in \{1, \ldots, n\}$ and $t \in \{0, 1, \ldots, g\}$. See [11].

The quadratic constraints in (8) may be approximated by a circumscribing polyhedron and an inscribing polyhedron [14], [12]. If $z_{j+1}(t) - z_j(t)$ lies in the region between the polyhedra, the kinematic constraints are satisfied approximately. The fact that $z_{j+1}(t) - z_j(t)$ lies inside the circumscribing polyhedron is expressed by:

$$A_{cs} \cdot (z_{j+1}(t) - z_j(t)) \leq B_{cs}, \quad (9)$$

where A_{cs} and B_{cs} specify the circumscribing polyhedron. Likewise, $z_{j+1}(t) - z_j(t)$ must lie outside the inscribing polyhedron which is formulated by:

$$A_{is} \cdot (z_{j+1}(t) - z_j(t)) \geq B_{is} + (u(t) - \mathbf{1}) \cdot M, \quad (10)$$

where $u(t) = (u_1(t), \ldots, u_{N_{is}}(t))^{\mathrm{T}}$ is a vector of binary variables $u_k \in \{0, 1\}$, N_{is} the number of faces of the inscribing polyhedron, $\mathbf{1} = 1^{N \times 1}$ a vector of ones, and M a large constant. In complete analogy to Section II-A.1, the kth scalar inequality in $A_{is} \cdot (z_{j+1}(t) - z_j(t)) \geq B_{is}$ is enforced if and only if $u_k(t) = 1$. To guarantee that $z_{j+1}(t) - z_j(t)$ lies outside the polyhedron specified by (10), at least one among the N_{is} inequalities must be satisfied, which can be ensured by the following condition:

$$\sum_{k=1}^{N_{is}} u_k(t) = 1. \quad (11)$$

The conjunction of (9), (10), and (11) approximates the quadratic equality constraint (8) using linear inequalities and binary decision variables.

As mentioned before, the dynamic constraints considered here are the velocity limits of the joints in the workspace. Varying velocity limits in different regions to account for the safety of operation or for time and energy efficiency can also be formulated using binary variables [12]. In the paper, for simplification, velocity limits (compare to (1b)) are considered as constants for each joint in the workspace:

$$V_j^- \cdot \Delta t \leq (z_j(t+1) - z_j(t)) \leq V_j^+ \cdot \Delta t. \quad (12)$$

The velocity limits may be different for different joints in Z.

C. Objective Function

Different criteria like average kinetic energy or transition time may be used [11], [12]. Here, only the minimization of the transition time is considered, which can be realized by the following cost function [15], [12]:

$$J = -\sum_{t=0}^{g} a(t) \quad (13)$$

with

$$a(0) \leq a(1) \leq \cdots \leq a(g), \quad (14)$$

where the binary variable $a(t)$ satisfies the constraint

$$z_{n+1}(t) \notin G \Rightarrow a(t) = 0. \quad (15)$$

III. Main Result

Using constraints of the form (3) to avoid collisions, as explained in Sections II-A.1 and II-B, may introduce a huge number of binary variables. This, in turn, often leads to intolerable computational effort for solving the resulting MILPs. In this section we extend the two-dimensional geometric result in [15] to the three-dimensional case, which will allow us to drastically reduce the number of binary variables. To this end, we first introduce some basic concepts from the theory of linear programming [16]:

Let the polyhedron $\mathcal{P}$ be given by $\mathcal{P} = \{\xi \in \mathbb{R}^m | A \cdot \xi \leq b\}$, where $A \in \mathbb{R}^{N \times m}$ and $b \in \mathbb{R}^N$. The system $A \cdot \xi \leq b$ is *redundant* if one of its scalar inequalities is implied by the others, and *non-redundant*, otherwise. The polyhedron $\mathcal{P}$ itself as well as any intersection of $\mathcal{P}$ with one of its supporting hyperplanes is called a *face* of $\mathcal{P}$. A face containing a single point is called a *vertex*, and each compact one-dimensional face is called an *edge*. A face of maximal dimension among the faces of $\mathcal{P}$ that are different from $\mathcal{P}$ is called a *facet*, and a face that does not contain any other face of $\mathcal{P}$ is said to be *minimal*. Finally, $\mathcal{P}$ is *full-dimensional* if its interior is non-empty, and $\mathcal{P}$ is *simple* if each of its vertices is contained in exactly m facets of $\mathcal{P}$.

III.1 Theorem. *Let $\mathcal{P} = \{\xi \in \mathbb{R}^3 | A \cdot \xi \leq b\}$ be a full-dimensional, simple polyhedron, $A \in \mathbb{R}^{N \times 3}$, $b \in \mathbb{R}^N$, $N > 2$, and assume that the system $A \cdot \xi \leq b$ of linear inequalities is non-redundant. Let $x, y \in \mathbb{R}^3$, $x \neq y$. Then exactly one of the following two statements holds.*

(i) The line segment $[\![x, y]\!]$ joining x and y intersects the interior of $\mathcal{P}$.

(ii) There exists a one-dimensional face E of $\mathcal{P}$ with the following property: If $f_1, f_2 \in \{1, \ldots, N\}$ are (the uniquely determined) indices that satisfy

$$E = \{\xi \in \mathcal{P} \,|\, A_{f_1} \cdot \xi = b_{f_1},\ A_{f_2} \cdot \xi = b_{f_2}\}, \quad (16)$$

then for each point $p \in [\![x, y]\!]$ it holds that

$$A_{f_1} p \geq b_{f_1} \text{ or } A_{f_2} p \geq b_{f_2}. \quad (17)$$

Obviously, the two statements are mutually exclusive. What we need to show is that (ii) holds whenever (i) does not hold. We sketch a proof which works by reduction to the two-dimensional case.

Assume that the line segment $[\![x, y]\!] = [\![(x_1, x_2, x_3), (y_1, y_2, y_3)]\!] \subseteq \mathbb{R}^3$ does not intersect the interior of $\mathcal{P}$. By means of a linear change of coordinates, one may obtain from $\mathcal{P} \subseteq \mathbb{R}^3$ a full-dimensional polyhedron $\widetilde{\mathcal{P}} \subseteq \mathbb{R}^2$ with non-redundant representation

$$\widetilde{\mathcal{P}} = \left\{ \xi \in \mathbb{R}^2 \,\middle|\, \widetilde{A} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} \leq \tilde{b} \right\}, \quad (18)$$

such that the line segment $[\![(x_1, x_2), (y_1, y_2)]\!] \subseteq \mathbb{R}^2$ does not intersect the interior of $\widetilde{\mathcal{P}}$. It can be shown that our claim follows from the two-dimensional result given below provided that (18) involves more than two inequalities, which is the most difficult case.

III.2 Theorem ([15]). *Let (18) be a full-dimensional polyhedron, $\widetilde{A} \in \mathbb{R}^{N\times 2}$, $\widetilde{b} \in \mathbb{R}^N$, $N > 2$, and assume that the system $\widetilde{A} \cdot \xi \leq \widetilde{b}$ of linear inequalities is non-redundant and that the rows of $\widetilde{A}$ are ordered counterclockwise. Let $\tilde{x}, \tilde{y} \in \mathbb{R}^2$, $\tilde{x} \neq \tilde{y}$. Then exactly one of the following two statements holds.*

(i) The line segment $[\![\tilde{x}, \tilde{y}]\!]$ joining $\tilde{x}$ and $\tilde{y}$ intersects the interior of $\widetilde{\mathcal{P}}$.

(ii) There is some $i \in \{1, \ldots, N\}$ such that for each point $p \in [\![\tilde{x}, \tilde{y}]\!]$ it holds that $\widetilde{A}_i p \geq \tilde{b}_i$ or $\widetilde{A}_{i+1} p \geq \tilde{b}_{i+1}$. Here, $\widetilde{A}_i$ denotes the ith row of $\widetilde{A}$, and $\widetilde{A}_{i+1}$ stands for $\widetilde{A}_1$ if $i = N$.

A direct application of Theorem III.1 shows that in order to avoid collisions it suffices to introduce two constraints

$$A_{f_1(E)} p_{sj}(t) \geq b_{f_1(E)} + (v_{j,E}(t) + w_{j,s}(t) - 2)M, \quad (19a)$$

$$A_{f_2(E)} p_{sj}(t) \geq b_{f_2(E)} + (v_{j,E}(t) - w_{j,s}(t) - 1)M \quad (19b)$$

for each one-dimensional face E of the obstacle $\mathcal{P}$, each link j, each particle s associated with link j, and each time $t \in \{0, \ldots, g\}$. In addition, a constraint

$$\sum_E v_{j,E}(t) = 1 \quad (20)$$

is required for each j and t, where the sum is over all one-dimensional faces E of $\mathcal{P}$. Here, $v_{j,E}(t), w_{j,s}(t) \in \{0, 1\}$ are binary variables to be introduced in the MILP as decision variables, and for each one-dimensional face E of $\mathcal{P}$, $f_1(E)$ and $f_2(E)$ denote the uniquely determined indices that satisfy (16).

In the above formulation (19)-(20), the binary variable $v_{j,E}(t)$ can be seen as selecting a pair of adjacent facets of the obstacle $\mathcal{P}$ that share a common edge, rather than a facet of $\mathcal{P}$ as with the formulation in Sections II-A.1 and II-A.2, and the binary variable $w_{j,s}(t)$ selects one of the facets of the already selected pair. Indeed, (19a) is only enforced if $v_{j,E}(t) = w_{j,s}(t) = 1$, and (19b) is only enforced if $v_{j,E}(t) = 1$ and $w_{j,s}(t) = 0$.

The crucial point here is that v does not anymore depend on the particle index s, which greatly reduces the number of binary decision variables. In fact, the number of binary variables introduced for the purpose of formulating collision avoidance constraints is

$$n \cdot (g + 1) \cdot (S + N_e)$$

in contrast to

$$n \cdot (g + 1) \cdot S \cdot N$$

for the method of Sections II-A.1 and II-A.2, where N_e denotes the number of one-dimensional faces of $\mathcal{P}$.

IV. Computational Results

The proposed method is applied to a 2-link robotic manipulator in $\mathbb{R}^3$ interacting with two and three obstacles, respectively. Its performance is compared to that of the method from [12].

For the 2-link scenario of Fig. 1, the robotic manipulator interacts with two obstacles. One of them is the 8-face polyhedron $\mathcal{P}$ shown in Fig. 1, and the other is a table-like obstacle represented by the constraint $z_j \geq -0.2$ for $j \in \{1, 2\}$.

Our computational results are based on the following parameters. The sampling time is $\Delta t = 0.2$, the link length is 0.3, and the bounds on the velocities in the workspace of the middle joint and the end-effector are 0.2 and 0.3, respectively. The end effector is to be driven to the goal region $G = [0.19, 0.21] \times [0.49, 0.51] \times [-0.01, 0.01]$ from the initial position determined by $z_1 = (0, 0, 0)^T$ (base joint), $z_2 = (0.3, 0, 0)^T$ (middle joint), and $z_3 = (0.6, 0, 0)^T$ (end-effector). Particles are equally distributed on each link, which include the end-effector of the corresponding link.

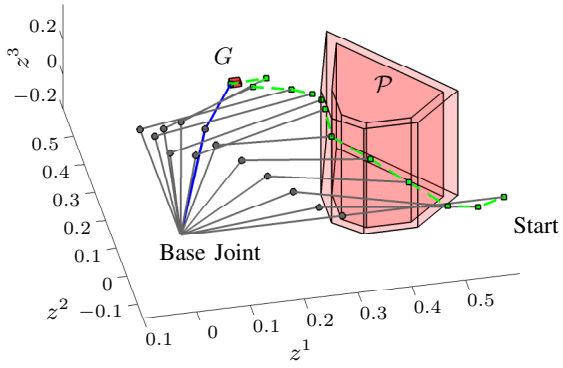


Fig. 1. A 2-link robotic manipulator interacting with a 8-face obstacle $\mathcal{P}$. The final configurations are shown in blue, and the computed motion of the end effector which reaches the goal within 12 steps, is illustrated by a sequence of green dashed line segment.

The method proposed in this paper is capable of planning the motion of sampled systems only. However, the resulting motion can be forced to satisfy the constraints between the sampling instants if the method is applied to a robust version of the original specification, e.g. [17]. For the scenario considered here, such a robust version is obtained by enlarging the obstacles by a suitable safety margin, which is shown as the larger polyhedron in Fig. 1. That margin is chosen to additionally account for the error made by approximating the link length constraint (8), the error made by approximating collision avoidance constraints with the help of particles, as well as the error made by neglecting the links' width.

The method proposed in the present paper and the previous method from [12] have been implemented in C++ using Concert Technology libraries in CPLEX Studio Academic Research 12.2 and run on a single thread of an $i5$-CPU with

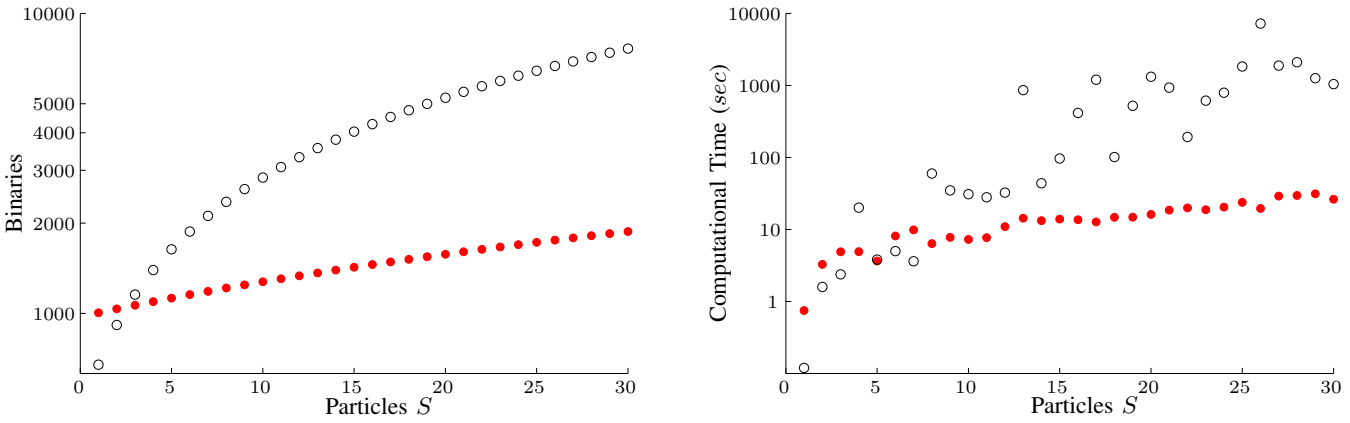
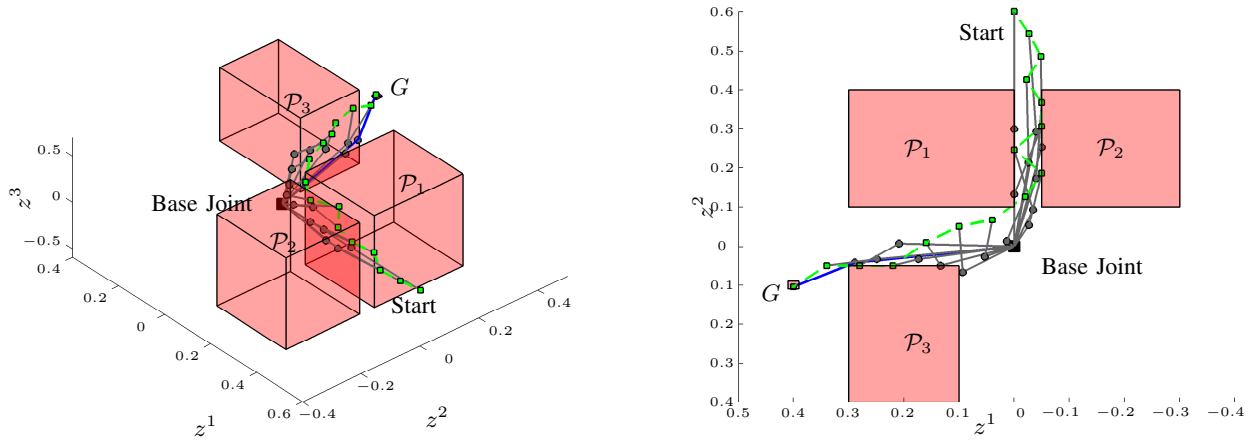


Fig. 2. Number of binary variables in the MILP (left) and the computational time (right), as a function of the number S of particles per link for the scenario of the 2-link robot interacting with the 8-face obstacle. (● corresponds to the methods proposed in the present paper, and ○, to that from [12]).



(a) Side view.

(b) Top view.

Fig. 3. A 2-link robotic manipulator interacting with 3 obstacles.

2.67 GHz clock rate. In order to separate the effects of different MILP formulations from unknown heuristics that are built in to CPLEX, the *MIP Strategy Probe*[2] is turned off. The quadratic constraints of each link length are approximated by two inscribing and circumscribing polyhedra of 14 faces each, see Section II-B. The 'big-M method' of Section II-A.1 is applied with $M = 100$. We use the horizon $g = 14$, and the goal can be reached within 12 steps.

The results presented in Fig. 2 show that the MILP formulation proposed in the present paper drastically reduces the number of binary decision variables. In addition, the computational time has been reduced in most of the cases, especially when the number of particles is large.

A more challenging scenario of a 2-link robot with 3 obstacles of 6 faces each is illustrated in Fig. 3. We choose a prediction horizon of $g = 16$, and the goal is reached at the 15-th step. Specifically, for $S = 5$ and $g = 9$ in this scenario, the solution was determined within 0.497 seconds using 4 threads.

Further tests which we do not report here in detail have shown that the computational effort may vary widely even for scenarios that look quite similar. In addition, if all the powerful built-in heuristics are used, CPLEX often shows an excellent performance regardless of any specific MILP formulation, especially if the number of particles is small. In this paper, we have combined the idea of reducing the number of binary variables with some but not all of the aforementioned built-in heuristics. Our results on 2D planning problems [15] show that a combination with a larger set of built-in heuristics leads to even more impressive reductions in computational effort.

V. Conclusions and Future Work

Motion planning for robotic manipulators with polyhedral obstacles and velocity constraints for the joint positions is considered in the paper. It has been approximated by Mixed-Integer Linear Programs (MILPs). We have presented

[2] *Probe* determines the extent by which variables should be probed prior to the branching phase. Depending on the particular optimization problem, that option may have a dramatic influence on the solution time [18].

and proved a geometric theorem in $\mathbb{R}^3$ whose application drastically reduces the number of binary decision variables in the MILPs, compared to previous results from [12]. In addition, computational time during the MILP solution is reduced in most cases, especially with larger number of particles, over previous methods.

The combination of the proposed theorem in $\mathbb{R}^3$ with suitable heuristics in CPLEX is a matter of current research. An ambitious goal is to generate the path of the robotic manipulator online with identification of dynamic obstacles (like human operators [19]) from measured data, which has been realized in a human-robot-interaction scenario in [20]. The computation will be accelerated by using the moving horizon scheme known from model predictive control rather than optimizing at once over the complete time span required to reach the goal. Our results indicate that the method proposed in this paper can be successfully applied in this setting in order to plan a robot's motion online.

We also work on extending the method to cover more general dynamic constraints than just velocity bounds, by combining our results with a recently proposed method for computing reachable sets of nonlinear systems [21].

REFERENCES

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[2] J. Latombe, *Robot Motion Planning*. Kluwer, 1991.

[3] H. Ding, M. Zhou, and O. Stursberg, "Optimal Motion Planning for Manipulators with Dynamic Obstacles using Mixed Integer Linear Programming," in *IEEE Mediterranean Conf. on Control and Automa.*, 2009, pp. 934–939.

[4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Model and Control*. John Wiley & Sons, Inc., 2006.

[5] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. on Robot. and Autom.*, 2000, pp. 995–1001.

[6] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration space," *IEEE Transactions on Robot. and Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[7] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. of Robot. Research*, vol. 21, no. 3, pp. 233–255, 2002.

[8] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. of Robot. Research*, vol. 30, no. 7, pp. 846–894, 2011.

[9] Y. Yang and O. Brock, "Elastic roadmaps - motion generation for autonomous mobile manipulation," *Autonomous Robots*, vol. 28, pp. 113–130, 2010.

[10] H. Ding, G. Schnattinger, B. Passenberg, and O. Stursberg, "Improving motion of robotic manipulators by an embedded optimizer." in *IEEE Conf. on Autom. Sci. and Eng.*, 2010, pp. 204–209.

[11] L. Blackmore and B. Williams, "Optimal manipulator path planning with obstacles using disjunctive programming," in *Proc. of the American Control Conference*, 2006, pp. 3200–3202.

[12] H. Ding, M. Zhou, and O. Stursberg, "Optimal path planning in the workspace for articulated robots using mixed integer programming," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2009, pp. 5770–5775.

[13] H. P. Williams, *Model building in mathematical programming*, 2nd ed. WILEY, 1985.

[14] A. Richards and J. How, "Mixed-integer programming for control," in *Proc. of the American Control Conference*, 2005, pp. 2676–2683.

[15] H. Ding, G. Reißig, D. Gross, and O. Stursberg, "Mixed-integer programming for optimal path planning of robotic manipulators," in *IEEE Conf. on Autom. Sci. and Eng.*, 2011, accepted.

[16] A. Schrijver, *Theory of linear and integer programming*. Chichester: John Wiley & Sons Ltd., 1986, a Wiley-Interscience Publication.

[17] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica J. IFAC*, vol. 45, no. 2, pp. 343–352, 2009.

[18] IBM, *User's Manual for CPLEX*, 2010.

[19] H. Ding, G. Reißig, K. Wijaya, D. Bortot, K. Bengler, and O. Stursberg, "Human arm motion modeling and long-term prediction for safe and efficient human-robot-interaction," in *IEEE Int. Conf. Robot. and Autom.*, 2011, pp. 5875–5880.

[20] H. Ding, K. Wijaya, G. Reißig, and O. Stursberg, "Optimizing motion of robotic manipulators in interaction with human operators," in *Int. Conf. on Intell. Robot. and Appl.*, ser. Lecture Notes in Computer Science (LNCS), S. Jeschke, Ed. Springer, 2011.

[21] G. Reißig, "Computing abstractions of nonlinear systems," *IEEE Trans. Automat. Control*, vol. 56, 2011, accepted, avail. via http://www.reiszig.de/gunther/pubs/i11abs.html.