

Online Adaption of Motion Paths to Time-Varying Constraints using Homotopies

Damian Kontny* Olaf Stursberg*

* *Control and System Theory, Dept. of Electrical Engineering and Computer Science, University of Kassel (Germany). e-mail: {dkontny, stursberg}@uni-kassel.de*

Abstract: This paper presents a method for optimal point-to-point control of linear discrete-time systems with time-varying non-convex state constraints, as arising for problems of avoiding moving obstacles. While common approaches like MPC with mixed-integer programming (MIP) can quickly become time-demanding, the presented approach efficiently computes circumventing near-optimal trajectories by using homotopy properties. In a first step, a range of offline selected trajectories is used to span a region of homotopic trajectories, for which the transitioning behavior between these is determined by semi-definite programming (SDP). The online part then determines with low computational effort a collision-free and near-optimal homotopic trajectory. The procedure maps the moving obstacles for relevant collision-critical time steps into the homotopy space, and determines a suitable trajectory by a tree-search of moderate size. The circumventing trajectories and the resulting computation times are illustrated by simulation.

© 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Real-time control, obstacle avoidance, constrained control, model predictive control, optimal trajectory, non-convex optimization.

1. INTRODUCTION

This paper addresses the problem of controlling a linear discrete time system from an initial to a final state, while minimizing a cost function and considering input constraints as well as non-convex time-varying state constraints. A practical instance of such a problem is human-robot-cooperation, where the robot has to avoid collisions with a human worker while maintaining the nominal operation close to optimal. Another example from the field of autonomous driving is to aim at preventing accidents with pedestrians or other cars. A difficulty of solving such problems is the large online computational effort, rendering standard formulations often not applicable. The goal of this paper is to derive a method which provides optimized and efficient solutions for such scenarios.

A possible approach for optimized point-to-point transition with constraints is *model predictive control* (MPC) (see e.g. Mayne et al. [2000]). In this field different methods were proposed to reduce the online effort, like *move-blocking strategies* (Gondhalekar and Imura [2010]), soft constrained formulations by penalty terms or barrier functions (Wang and Boyd [2010]), (Richards [2015]), or efficient direct multiple shooting (Kirches et al. [2012]), (Diehl et al. [2002]).

While for convex problems, the computation times typically can be reduced sufficiently, the situation for non-convex problems (as described above) is different. Nonlinear programming often leads not to suitable computation times and considerably suboptimal solutions. As alternative, MPC with mixed-integer-programming (MIP) as discussed in (Blackmore and Williams [2006]), (Ding et al. [2011]) for a human-robot scenario, suffers from large amounts of binary variables for encoding convex partitioned regions, over which a suitable sequence can be determined by tree search. Even for small problems, the combina-

torial complexity limits the real-time applicability. Therefore, methods like *explicit MPC* (Tøndel et al. [2003]), (Zeilinger et al. [2011]) were developed, which partition the solution space and offline determine controllers by the use of multi-parametric programming. While the computation times are lower, this method is paired with high memory requirements, and thus limited to low dimensions and/or short prediction horizons. In addition, the adaption to time-varying state constraints is an open problem.

Trajectory planning in non-convex spaces is also the focus in a variety of publications in the field of human-robot-interaction. Representative methods are *potential field methods* (Khatib [1986]), cell-decomposition based on graph search (Newman and Branicky [1991]), and sampling-based methods like *rapidly exploring random trees* (RRT) (Kuffner and LaValle [2000]) and RRT* (Karaman and Frazzoli [2011]), (Ferguson and Stentz [2007]). Most of them do not consider the system dynamics, are not applicable for higher dimensions, or do not aim at trajectory optimization.

A predecessor of this work (Kontny and Stursberg [2016b]) deals with the problem by offline synthesizing controllers for homotopic trajectories and online avoiding collisions for appearing static obstacles by a procedure of selecting collision free homotopic trajectories and passing the obstacle along one of its vertices. This concept was extended to the consideration of input constraints in the synthesis procedure, see (Kontny and Stursberg [2016a]). Compared to the predecessor work, this paper introduces: a solution for providing trajectories with reduced costs, the extension of the method to systems which are not controllable in one step, and the additional consideration of predicted moving obstacle positions.

The paper is structured such that Sec. 2 introduces the considered type of homotopies, Sec. 3 provides the problem statement, and Sec. 4 describes the offline part of the procedure to determine the transition behavior between homotopic trajectories. In the online part (Sec. 5), a collision avoidance procedure is introduced which determines an optimized homotopic trajectory that passes the obstacle. Finally, a numerical example is shown in Sec. 6, before the paper ends with conclusion.

2. HOMOTOPIC FUNCTIONS

Let a linear discrete-time system be given by:

$$x_{k+1} = Ax_k + Bu_k, \quad (1)$$

with time $k \in \mathbb{N}_0$, state vector $x_k \in \mathbb{R}^{n_x}$, and inputs from the feasible set $u_k \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$.

Definition 1. Given the system dynamics (1), a system is said to be *r-step controllable*, if there exists a control input $u_k \in \mathbb{R}^{n_u}$ such that for any initial state x_0 , the state can be driven to a final state x_f in at least r time-steps. With $r = n_x/n_u$ time-steps. Δ

This is fulfilled if the controllability matrix $R = (B, AB, A^2B, \dots, A^{n_x-1}B) \in \mathbb{R}^{n_x \times n_u n_x}$ has full row rank i.e. $\text{rank}(R) = n_x$. Constraints on inputs or states are irrelevant for this property.

Let so-called *base trajectories* of (1) be given by $\hat{x}^i = (x_0^i, \dots, x_N^i)$ and $\hat{u}^i = (u_0^i, \dots, u_{N-1}^i)$, where $i \in \mathcal{M} := \{0, 1, \dots, n_c\}$ is indexing the i -th base trajectory of $n_c + 1$ many. They are chosen to span a region, in which the online circumvention of an obstacle is possible. All base trajectories are defined in the same time domain, i.e. the same number of time steps with final time $N \in \mathbb{N}$, and the same initial and final states $x_0^i = x_s$, $x_N^i = x_f$, $i \in \mathcal{M}$. They are designed in such a way, that sufficient reserves in the input space are available, to later transition from one to another. Let the set $\mathcal{X} := \{\hat{x}^0, \dots, \hat{x}^{n_c}\}$ contain all base trajectories, which are computed offline and are thus known in the online procedure. When choosing the base trajectories, knowledge about possible position occurrence of the obstacles should be considered.

A trajectory $\hat{x}^i$ can be interpreted as the image of a function: $\hat{x}^i = F^i(\hat{u}^i)$. The following definition introduces *homotopic* trajectories which are in between the trajectories $\hat{x}^i$:

Definition 2. For a set of $n_c + 1$ continuous functions $F^i : \mathbb{R}^{n_u \times T} \rightarrow \mathbb{R}^{n_x \times T}$, $i \in \mathcal{M}$, a *vectorized homotopy* is defined by: $H : (\mathbb{R}^{n_u \times T})^{n_c} \times [0, 1]^{n_c} \rightarrow \mathbb{R}^{n_x \times T}$. The second argument is a vector of homotopy parameters $\lambda = (\lambda^1, \dots, \lambda^{n_c})^T$ with:

$$\lambda^i \in [0, 1], \quad \sum_{i=1}^{n_c} \lambda^i \leq 1. \quad (2)$$

With $F = (F^1(\hat{u}^1) - F^0(\hat{u}^0), \dots, F^{n_c}(\hat{u}^{n_c}) - F^0(\hat{u}^0))^T$ and $\lambda^0 := 1 - \sum_{i=1}^{n_c} \lambda^i$, the *linear vectorized homotopy function* is given by:

$$\begin{aligned} H(\hat{u}^0, \dots, \hat{u}^{n_c}, \lambda) &= \sum_{i=0}^{n_c} \lambda^i \cdot F^i(\hat{u}^i) \\ &= F^0(\hat{u}^0) + \sum_{i=1}^{n_c} (F^i(\hat{u}^i) - F^0(\hat{u}^0)) \cdot \lambda^i = F^0(\hat{u}^0) + F \cdot \lambda \end{aligned} \quad (3)$$

Δ

This definition states that the homotopic functions are linear interpolations of the base trajectories. Since the dynamics of (1) is also linear, the homotopic trajectories share the same characteristics and satisfy the same convex input or state constraints as the base trajectories. Since (3) relates to complete

trajectories, a homotopic state $x_k(\lambda_k)$ for a single point of time k lies in between the states x_k^i , $i \in \mathcal{M}$, and is identified by the homotopy value λ_k . This leads to the following formulation of a homotopic state and input at time k :

$$x_k(\lambda_k) := x_k^0 + D_{x_k} \lambda_k, \quad u_k(\lambda_k) := u_k^0 + D_{u_k} \lambda_k, \quad (4)$$

with the homotopy vector $\lambda_k := (\lambda_k^1, \dots, \lambda_k^{n_c})^T \in \mathbb{R}^{n_c}$ at time k , matrices $D_{x_k} = (x_k^1 - x_k^0, \dots, x_k^{n_c} - x_k^0) \in \mathbb{R}^{n_x \times n_c}$, and $D_{u_k} = (u_k^1 - u_k^0, \dots, u_k^{n_c} - u_k^0) \in \mathbb{R}^{n_u \times n_c}$ for $k \in \mathcal{K} := \{0, \dots, N\}$. A trajectory with constant homotopy vector $\tilde{\lambda}$ is denoted by $\hat{x}(\tilde{\lambda})$, and $\hat{u}(\tilde{\lambda})$ respectively.

Since for an *r-step controllable* system (see Def. 1), only every r -th time $k = t \cdot r$, with $t \in \mathcal{T} = \{0, \dots, \lfloor N/r \rfloor\} \subseteq \mathbb{N}_0$, can satisfy the homotopic state equation (4), new state notations are introduced. The notation is required since in Sec. 4, a auxiliary system is introduced which describes the transitioning behavior between homotopic trajectories depending on this timeline. Each state x_{tr} for $t \in \mathcal{T}$ corresponds to the new notation $z_t \triangleq x_{tr}$, $z_{t+1} \triangleq x_{(t+1)r}$ etc. Consequently, the homotopy state equation (4) can be rewritten to:

$$z_t(\lambda_{tr}) = z_t^0 + D_{z_t} \lambda_{tr}, \quad (5)$$

which satisfies (4) for every r -th time step. Simultaneously, a new homotopy state notation $\mu \in \mathbb{R}^{n_c}$ describing every r -th homotopy state λ_{tr} is introduced. The new homotopy state notation μ is connected as follows with the homotopy state λ : $\mu_t \triangleq \lambda_{tr}$, $\mu_{t+1} \triangleq \lambda_{(t+1)r}$. Thus (5) can be described by:

$$z_t(\mu_t) = z_t^0 + D_{z_t} \mu_t. \quad (6)$$

Definition 3. Let the set $\mathcal{L}$ contain all base trajectories $\hat{x}^i \in \mathcal{X}$ described by their homotopy values $\tilde{\mu}^i \in \mathcal{L} := \{\tilde{\mu}^0, \dots, \tilde{\mu}^{n_c}\}$, with $\tilde{\mu}^i \in \{0, 1\}^{n_c}$ and let the sum over all components of μ^i be $\sum_{i=1}^{n_c} \mu^i \leq 1$. Δ

3. PROBLEM DEFINITION

The task is to bring the system (1) from an initial state $x_{k_0} = x_s$ at time $k = k_0$ to a final state x_f , while avoiding collisions with a moving obstacle $\mathcal{P}_{x_k}$ in the state space and satisfying input constraints $u_k \in \mathcal{U} \subset \mathbb{R}^{n_u}$. The obstacle is considered to be known for a prediction horizon of H steps, and is defined as a polytopic region $\mathcal{P}_{x_{k+j}} := \{x_{k+j} \mid C_{k+j} x_{k+j} \leq d_{k+j}\} \subseteq \mathbb{R}^{n_x}$, with $j \in \mathcal{J} := \{0, \dots, H\}$, $C \in \mathbb{R}^{c \times n_x}$, and $d \in \mathbb{R}^c$. The position prediction may be obtained from estimation using an appropriate model or for communicated information on the planned motion of $\mathcal{P}_{x_{k+j}}$. The goal is to find a feasible and optimized trajectory $\hat{x}^*, \hat{u}^*$ by solving the optimization problem:

$$\begin{aligned} \min_{x_{k_0+j}, u_{k_0+j}} \quad & J(x_{k_0+j}, u_{k_0+j}) = \sum_{j=0}^H (x_{k_0+j} - x_f)^T Q (x_{k_0+j} - x_f) \\ & + (u_{k_0+j} - u_f)^T R (u_{k_0+j} - u_f) \\ \text{s.t.} \quad & (1), u_{k+j} \in \mathcal{U}, x_{k_0+j} \notin \mathcal{P}_{x_{k_0+j}}, \forall j \in \mathcal{J}, x_{k_0} = x_s, \end{aligned} \quad (7)$$

for the prediction horizon H . The performance function is quadratic with positive-definite weighting matrices $Q \in \mathbb{R}^{n_x \times n_x}$, and $R \in \mathbb{R}^{n_u \times n_u}$. The control problem is implemented in a receding horizon fashion. Within this implementation, only the first step of the control strategy is applied, and the calculation is repeated for an incremented initial time $k_0 := k_0 + 1$. The initial state for the next iteration is then given by the actual measured state. Due to the non-convex obstacle avoidance constraint, a possible approach to solve the considered problem is by MIP,

or specifically mixed-integer quadratic programming (MIQP). This leads typically to a large number of binary variables used to formulate, whether x_{k_0+j} lies outside of any bounding plane of $\mathcal{P}_{\|_{r+1}}$ for each point of time $k_0 + j$, $j \in \mathcal{J}$. Even if the obstacle and x_{k_0+j} are far away from each other in the considered time horizon, the problem must nevertheless be formulated as a MIQP, leading to a large share of unnecessary computations.

The method presented here is based on computing offline trajectories to build a homotopy function, which spans a region for circumventing the obstacle. This enables a fast online procedure to bring the system to a homotopic trajectory which passes the moving obstacle. In order to enable that a solution of the problem can be found by this method, the following assumption is introduced:

Assumption 4. Let the set of trajectories $\mathcal{X}$ contain at least one trajectory $\hat{x}^i$, $i \in \mathcal{M}$, such that for any $\hat{x}_k^i \in \hat{x}^i$ a collision with the obstacle can be avoided: $\hat{x}_k^i \notin \mathcal{P}_{x_k}$, $\forall k \in \{0, \dots, N\}$. Δ

While the violation of this assumption implies an ill-posed problem, the assumption alone is not sufficient to guarantee a trajectory free of collision, since the transition from the current trajectory to a homotopic feasible one must be also realized without collisions.

4. TRANSITIONS BETWEEN HOMOTOPIC TRAJECTORIES

Consider the task of steering (1) from a state $z_{t_0} \triangleq x_{k_0}$ satisfying (6) to a future state on a homotopic trajectory. Let z_{t_0} be identified by μ_{t_0} , and the new trajectory (i.e. the goal of the transition) be referenced by $\bar{\mu}$. While the inputs for the currently executed trajectory and the targeted one are known from the homotopy function (4), the transition from the actually executed trajectory to the final one requires additional inputs δu_k , leading to the overall input:

$$\tilde{u}_k(\lambda_k) := u_k(\lambda_k) + \delta u_k. \quad (8)$$

To restrict the speed of changing between homotopic trajectories, constraints on δu_k are formulated:

$$\delta u_{\min} \leq \delta u_k \leq \delta u_{\max}, \quad (9)$$

for all $k \in \mathcal{K}$, with $\delta u_{\min}, \delta u_{\max} \in \mathbb{R}^{n_u}$.

Now since every r -th state $x_{tr} \triangleq z_t$ is reachable according to Def. 1 (with $\text{rank}(R) = n_x$), an auxiliary linear time-varying (LTV) system is defined, which describes the change of the homotopy state μ_t , $t \in \mathcal{T}$ to a final value denoted by $\bar{\mu}$ as follows:

$$\mu_{t+1} = \tilde{A}_t(\mu_t - \bar{\mu}) + \bar{\mu}. \quad (10)$$

Since the transitioning behavior is freely parameterizable, the matrix $\tilde{A}_t$ is chosen to be time-varying and diagonal $\tilde{A}_t = \text{diag}(a_{i,t}) \in \mathbb{R}^{n_c \times n_c}$ with elements $a_{i,t}$, where index i denotes the i -th element, and t specifies the point of time. Additionally the variables $a_{i,t}$ have to satisfy $|a_{i,t}| \leq 1$ for all $i \in \{1, \dots, n_c\}$ and $t \in \mathcal{T}$, for reasons of stability. Thus, (10) describes how the homotopy values μ_t (or respectively every r -th value λ_{tr}) evolves from its current value to $\bar{\mu}$ in terms of the matrices $\tilde{A}_t$.

Lemma 5. The input constraints (9) hold for every time step $k \in \mathcal{K}$, if the following inequality holds for all $t \in \mathcal{T}$:

$$\begin{aligned} \Delta U_{\min} &\leq (B, AB, \dots, A^{r-1}B)^{-1} D_{z_{t+1}} (\tilde{A}_t(\mu_t - \bar{\mu}) + \bar{\mu} - \mu_t) \\ &\leq \Delta U_{\max}, \end{aligned} \quad (11)$$

with $\Delta U_{\min} = (\delta u_{\min}^T, \dots, \delta u_{\min}^T)^T \in \mathbb{R}^{n_{ur}}$, and $\Delta U_{\max} = (\delta u_{\max}^T, \dots, \delta u_{\max}^T)^T \in \mathbb{R}^{n_{ur}}$.

Proof. Starting from an initial state $z_{t_0}(\mu_{t_0})$, the state vector $z_t(\mu_t)$ can be described by the following equation:

$$z_t(\mu_t) = A^{tr-t_0r} z_{t_0}(\mu_{t_0}) + \sum_{l=t_0r}^{tr-1} A^{tr-l-1} B \tilde{u}_l(\lambda_l). \quad (12)$$

Starting from time $t-1$, (12) determines $z_t(\mu_t)$ by:

$$z_t(\mu_t) = A^r z_{t-1}(\mu_{t-1}) + \sum_{l=tr-r}^{tr-1} A^{tr-l-1} B \tilde{u}_l(\lambda_l). \quad (13)$$

The value λ_{tr-r} in (13) corresponds to $\lambda_{tr-r} \triangleq \mu_{t-1}$. The other intermediate values of λ_l for $l \in \{tr-r+1, \dots, tr-1\}$ are located in between μ_{t-1} and $\mu_t \triangleq \lambda_{tr}$. Since this intermediate values are not existing for an r -step controllable system, these values are set to: μ_{t-1} , corresponding to a zero-order hold of μ_{t-1} . Furthermore, the states $z_{t-1}(\mu_{t-1})$ on the right side and $z_t(\mu_t)$ on the left are replaced according to (6), one gets:

$$\begin{aligned} z_t^0 + D_{z_t} \mu_t &= A^r (z_{t-1}^0 + D_{z_{t-1}} \mu_{t-1}) + \\ &\quad (B, AB, \dots, A^{r-1}B) \begin{pmatrix} u_{tr-1}^0 + D_{u_{tr-1}} \mu_{t-1} + \delta u_{tr-1} \\ u_{tr-2}^0 + D_{u_{tr-2}} \mu_{t-1} + \delta u_{tr-2} \\ \vdots \\ u_{tr-r}^0 + D_{u_{tr-r}} \mu_{t-1} + \delta u_{tr-r} \end{pmatrix}. \end{aligned} \quad (14)$$

Replacing μ_t on the left side according to (10) and rewriting the terms leads to:

$$\begin{aligned} z_t^0 + D_{z_t} (\tilde{A}_{t-1}(\mu_{t-1} - \bar{\mu}) + \bar{\mu}) &= A^r z_{t-1}^0 + \underbrace{(B, AB, \dots, A^{r-1}B)}_{D_{z_t}} \begin{pmatrix} u_{tr-1}^0 \\ u_{tr-2}^0 \\ \vdots \\ u_{tr-r}^0 \end{pmatrix} + \\ &\quad \underbrace{\left(A^r D_{z_{t-1}} + (B, AB, \dots, A^{r-1}B) \begin{pmatrix} D_{u_{tr-1}} \\ D_{u_{tr-2}} \\ \vdots \\ D_{u_{tr-r}} \end{pmatrix} \right)}_{D_{z_t}} \mu_{t-1} + (B, AB, \dots, A^{r-1}B) \begin{pmatrix} \delta u_{tr-1} \\ \delta u_{tr-2} \\ \vdots \\ \delta u_{tr-r} \end{pmatrix}. \end{aligned} \quad (15)$$

Incrementing $t := t+1$, solving (15) for the vector of the additional inputs, and constraining it from the left and right side yields:

$$\Delta U_{\min} \leq \begin{pmatrix} \delta u_{tr+r-1} \\ \delta u_{tr+r-2} \\ \vdots \\ \delta u_{tr} \end{pmatrix} \leq \Delta U_{\max}, \quad (16)$$

which equals (11) and therefore satisfies (9). $\blacksquare$

For the scenario as described in Sec. 3, it is assumed that the system should transition as fast as possible to a collision-free and optimized trajectory (described by the transition from μ_t to $\bar{\mu}$) after detection of the obstacle $\mathcal{P}_x$. Selecting a suitable candidate $\bar{\mu}$ is discussed in the next section. In order to make the best possible use of the constrained input signal (11), the matrices $\tilde{A}_t$, are chosen to be time-varying, since $D_{z_{t+1}}$ in (11) is also changing over time.

For fast transitioning between homotopic trajectories, the eigenvalues of the matrices $\tilde{A}_t$ must be minimized subject to the input constraints. As given in (Boyd et al. [1994]), the eigenvalue problem (EVP) is to minimize the maximum eigenvalue of $\tilde{A}_t$. The matrix $\tilde{A}_t$ depends affinely on the variables $a_{i,t}$, such

that $\tilde{A}_t = W_{0,t} + \sum_{i=1}^{n_c} a_{i,t} W_{i,t}$ is a linear matrix. This allows to formulate the convex EVP with the input constraints as a semi-definite program (SDP):

$$\min_{a_{i,t}, \gamma} \gamma \quad (17)$$

$$\text{s.t. } \tilde{A}_t - \gamma I \leq 0, \forall t \in \mathcal{T} \quad (18)$$

$$(11), \forall \{\mu_t, \bar{\mu}\} \in \{0, 1\}^{n_c}, \forall t \in \mathcal{T}. \quad (19)$$

The constraint (11) has to be satisfied for all possible combinations resulting from $\mu_t \in \{0, 1\}^{n_c}$ and $\bar{\mu} \in \{0, 1\}^{n_c}$, and for all times $t \in \mathcal{T}$. Because $\tilde{A}_t$ is chosen to be diagonal, (17)–(19) reduces to a general linear programming problem. The dynamics (10) specifies the transitioning behavior between homotopic trajectories such that the transition is as fast as possible with respect to the input constraints.

5. ONLINE COLLISION AVOIDANCE

The online procedure starts by determining the cost optimal homotopic goal trajectory encoded by $\bar{\mu}^*$ for a given prediction horizon H , while first neglecting the obstacle. If the resulting trajectory fails to be free of collision, a fast online procedure is initiated, which determines an optimized value for $\bar{\mu}$ with respect to the performance function in (7), such that the system passes the obstacle. The advantage of restricting the problem of collision avoidance to homotopic trajectories is that time steps with collisions in the homotopy space can be identified quickly, and that only these time steps are relevant for the circumvention procedure.

5.1 Determining the Best Homotopic Trajectory

If no obstacles are present, the task is to find an optimal value of $\bar{\mu}$. It can be shown that the cost function (7) can be reformulated from $J(x_{k+j}, u_{k+j})$ to $J(\bar{\mu})$, $j \in \mathcal{J}$. This essentially reduces the search space of the problem from $(H+1) \cdot n_u \cdot n_x$ to n_c variables. The initial homotopy vector is $\mu_{t_0} = \mu_{init}$ at time $t = t_0$, and the corresponding state x_{k_0} is known with $k_0 = t_0 \cdot r$ according to (6). The prediction of up to H time steps leads to a considered time span $k \in \mathcal{K}_H = \{k_0, \dots, k_0 + H\}$, beginning from k_0 , or respectively $t \in \mathcal{T}_H = \{t_0, \dots, t_0 + \lfloor H/r \rfloor\}$. The optimization problem can be reformulated to:

$$\min_{\bar{\mu}} \sum_{j=0}^H (x_{t_0r+j} - x_f)^T Q (x_{t_0r+j} - x_f) + (u_{t_0r+j} - u_f)^T R (u_{t_0r+j} - u_f) \quad (20)$$

s.t. :

$$x_{k+1} = Ax_k + B\tilde{u}_k, \forall k \in \{tr, \dots, tr + r - 1\} \quad (21)$$

$$\tilde{u}_k = u_k^0 + D_{u_k} \mu_t + \delta u_k, \forall k \in \{tr, \dots, tr + r - 1\} \quad (22)$$

$$(B, AB, \dots, A^{r-1}B)^{-1} D_{z_{t+1}} (\tilde{A}_t (\mu_t - \bar{\mu}) + \bar{\mu} - \mu_t) = \begin{pmatrix} \delta u_{tr+r-1} \\ \delta u_{tr+r-2} \\ \vdots \\ \delta u_{tr} \end{pmatrix} \quad (23)$$

$$\mu_{t+1} = \tilde{A}_t (\mu_t - \bar{\mu}) + \bar{\mu} \quad (24)$$

$$z_t(\mu_t) = z_t^0 + D_{z_t} \mu_t \quad (25)$$

$$\forall t \in \mathcal{T}_H. \quad (26)$$

The values of $\tilde{A}_t$ are determined according to Sec. 4. As noticed, the optimization problem contains no constraints on the input signal, while these have already been integrated in the

matrices $\tilde{A}_t$ by the solution of (17)–(19). The optimal solution of the homotopy value, based on the initial time t_0 , is given by $\bar{\mu}_{|t_0}^*$. The index denotes that the optimal homotopy value is computed based on time t_0 and is constant for the future times of the prediction horizon. The resulting optimal state trajectory is denoted by $x_{t_0r+j|t_0r}^*$, $j \in \mathcal{J}$. The computation of (20)–(26) can be performed very fast because the quadratic optimization problem is low-dimensional and only subject to equality constraints.

5.2 Transforming the Moving Obstacle Into the Homotopy Space

Now, the resulting state trajectory is checked against collision, i.e. $x_{t_0r+j|t_0r}^* \notin \mathcal{P}_{x_{t_0r+j}}$ is evaluated $\forall j \in \mathcal{J}$. If collisions are found, the state space obstacle $\mathcal{P}_{x_{t_0r+j}}$ is transformed for each j into the homotopy space, denoted by $\mathcal{P}_{\lambda_{t_0r+j}}$. Since it was shown that an r -step controllable system conforms only every r -th time step to the homotopy function (4), a collision check may only be allowed to any r -th step. The related polytope is $\mathcal{P}_{x_{t_0r+sr}} := \{x_{t_0r+sr} | C_{t_0r+sr} x_{t_0r+sr} \leq d_{t_0r+sr}\} \subseteq \mathbb{R}^{n_x}$, or respectively with $s \in \{0, \dots, \lfloor H/r \rfloor\}$:

$$\mathcal{P}_{z_t} := \{z_t | C_{tr} z_t \leq d_{tr}\}, t \in \mathcal{T}_H. \quad (27)$$

The polytope $\mathcal{P}_{z_t}$ mapped into the homotopy space is denoted by $\mathcal{P}_{\mu_t}$ and determined by inserting (6) into (27):

$$C_{tr}(z_t^0 + D_{z_t} \mu_t) \leq d_{tr} \quad (28)$$

$$\underbrace{C_{tr} D_{z_t}}_{C_{\mu_t}} \mu_t \leq \underbrace{d_{tr} - C_{tr} z_t^0}_{d_{\mu_t}} \quad (29)$$

$$\Rightarrow \mathcal{P}_{\mu_t} := \{\mu_t | C_{\mu_t} \mu_t \leq d_{\mu_t}\}. \quad (30)$$

The polytope $\mathcal{P}_{\mu_t}$ exists for time t if the intersection:

$$\mathcal{P}_{\mu_t} \cap z_t(\mu_t) \neq \emptyset \quad (31)$$

is non-empty. This can be easily checked if the following linear optimization problem has a feasible solution:

$$\min_{\mu_t} b^T \mu_t \quad (32)$$

$$\text{s.t. } C_{\mu_t} \mu_t \leq d_{\mu_t} \quad (33)$$

$$\mu_t \in [0, 1]^{n_c}, \sum_{i=1}^{n_c} \mu_t^i \leq 1, \quad (34)$$

where $b \neq 0 \in \mathbb{R}^{n_c}$ can be chosen arbitrarily. The set of all times t for which (32)–(34) is satisfied, is denoted by:

$$\mathcal{T}_I := \{t | (32) - (34) \text{ satisfied}\} \subseteq \mathcal{T}_H. \quad (35)$$

If $\mathcal{T}_I = \emptyset$, no collision of the moving obstacle and the system occurs for $t \in \mathcal{T}_H$.

The transformation of $\mathcal{P}_{z_t}$ into $\mathcal{P}_{\mu_t}$ may lead to redundant half-planes in $\mathcal{P}_{\mu_t}$, which can be removed by linear programming, as described in Caron et al. [1989]. Let the w -th row of C_{μ_t} be denoted by $C_{\mu_t}^w$, and the w -th component of d_{μ_t} by $d_{\mu_t}^w$. Further, let $\mathcal{W}_t := \{1, \dots, c_t\}$ be the set of all inequalities at time t . The w -th row is redundant, iff the linear program:

$$\max_{\mu_t} C_{\mu_t}^w \mu_t \quad (36)$$

$$\text{s.t. } C_{\mu_t}^i \mu_t \leq d_{\mu_t}^i, \forall i \in \mathcal{W}_t \setminus \{w\} \quad (37)$$

has an optimal value less or equal to $d_{\mu_t}^w$. Solving the problem c_t -times yields for $t \in \mathcal{T}_I$, the simplified polytope $\tilde{\mathcal{P}}_{\mu_t} := \{\mu_t | \tilde{C}_{\mu_t} \mu_t \leq \tilde{d}_{\mu_t}\}$ with $\tilde{C}_{\mu_t} \in \mathbb{R}^{\tilde{c}_t \times n_c}$, $\tilde{d}_{\mu_t} \in \mathbb{R}^{\tilde{c}_t}$, and w -th rows $\tilde{C}_{\mu_t}^w$ and $\tilde{d}_{\mu_t}^w$, $w \in \tilde{\mathcal{W}}_t := \{1, \dots, \tilde{c}_t\}$.

5.3 Homotopy Control Algorithm (HCA)

If a collision is possible, the goal is to select an optimized homotopy value $\bar{\mu}_{\cdot|t_0}^*$, with initial state $z_{t_0} \triangleq x_{k_0}$ and corresponding μ_{t_0} for which the resulting trajectory satisfies $x_{t_0r+j|t_0r}^* \notin \mathcal{P}_{x_{t_0r+j}}$, $j \in \mathcal{J}$. The for computing $\bar{\mu}_{\cdot|t_0}^*$, see Alg. 1 can be separated into four parts.

1. Optimal trajectory, without obstacle (line 0) The algorithm 1 computes the optimal value $\bar{\mu}_{\cdot|t_0}^*$ for the prediction horizon H by solving the optimization problem (20)-(26). The optimal homotopy value $\bar{\mu}_{\cdot|t_0}^*$ defines the optimal state sequence $x_{t_0r+j|t_0r}^*$, $j \in \mathcal{J}$. The resulting trajectory is checked for collisions with $\mathcal{P}_{x_{t_0r+j}}$ (line 1). If no collisions are found, the algorithm terminates successfully. If the trajectory collides with $\mathcal{P}_{x_{t_0r+j}}$, the following collision avoidance procedure starts.

2. Best Free Base Trajectory (line 2) First, the costs for all $\bar{\mu}^i \in \mathcal{L}$ are computed according to (20)-(26), and the transitions from the actually executed trajectory towards the base trajectories (encoded by $\bar{\mu}^i$) are checked for collisions. The least costly solution $\bar{\mu}^{i^*}$ within the set $\mathcal{L}$ is selected, leading to the best free resulting trajectory when transitioning to a base trajectory. This step prevents the following step from running into an infeasible branch, since the selected base trajectory is always feasible.

3. Tree-Search (line 3-15) A best-first strategy is executed to decide how the moving obstacle should be passed, and to guarantee that the resulting trajectory of $z_t(\mu_t)$ determined from (10), for all $t \in \mathcal{T}_I$ is free of collisions. A node of the tree models an element $g_p \in \mathcal{T}_I$, with p indexing the tree level starting with $p := 1$. The node g_p is said to be explored, if all successors $w_{g_p} \in \mathcal{W}_{g_p}$ are visited. If a node g_p is explored, the node is removed from the set $\mathcal{T}_I$:

$$\mathcal{T}_I := \mathcal{T}_I \setminus g_p, \quad (38)$$

The best node based on g_p is denoted by $w_{g_p}^*$ and is stored in the set:

$$\mathcal{O} := \mathcal{O} \cup w_{g_p}^*. \quad (39)$$

The best-first search selects for node p one polytope $\tilde{\mathcal{P}}_{\mu_{g_p}}$ with $g_p \in \mathcal{T}_I$. Then, each successor $w_{g_p} \in \mathcal{W}_{g_p}$ is determined by solving the optimization problem (20)-(26) with the additional constraints:

$$\tilde{C}_{\mu_{g_p}}^{w_{g_p}} \bar{\mu} \geq \tilde{d}_{\mu_{g_p}}^{w_{g_p}} \quad (40)$$

$$\tilde{C}_{\mu_{g_p}}^{w_{g_p}} \mu_{g_p} \geq \tilde{d}_{\mu_{g_p}}^{w_{g_p}} \quad (41)$$

$$\tilde{C}_{\mu_{g_p}}^{w_{g_p}} \bar{\mu}^{i^*} \geq \tilde{d}_{\mu_{g_p}}^{w_{g_p}}. \quad (42)$$

Constraint (40) requires to select a goal trajectory $\bar{\mu}$, which generally passes the obstacle $\tilde{\mathcal{P}}_{\mu_{g_p}}$ along the w_{g_p} -th half-space. Since this does not ensure that the transient behavior during the change of the trajectories is also free of collision, constraint (41) is introduced, which prevents the system to collide at time g_p with $\tilde{\mathcal{P}}_{\mu_{g_p}}$. The homotopy state μ_{g_p} can be computed recursively from (10):

$$\mu_{g_p} = \prod_{i=g_p-1}^{t_0} \tilde{A}_i(\mu_{t_0} - \bar{\mu}) + \bar{\mu}. \quad (43)$$

The last constraint (42) guarantees, that the w_{g_p} -th half-space also contains the previously selected homotopy value of the

best base trajectory $\bar{\mu}^{i^*}$ from part (2.). This guarantees that a feasible solution of the tree-search can always be found. After the exploration of all successors $w_{g_p} \in \mathcal{W}_{g_p}$ at time g_p , the one producing the lowest costs is denoted by $w_{g_p}^*$ and stored in the set (39). The set $\mathcal{T}_I$ is updated according to (38).

The procedure is repeated by selecting a new $g_p \in \mathcal{T}_I$ and computing the successors $w_{g_p} \in \mathcal{W}_{g_p}$ according to (20)-(26) with the constraints (40)-(42), and additionally (40)-(42) for all $w_{g_p} \in \mathcal{O}$.

The tree-search terminates immediately if no more collisions between $z_t(\mu_t)$ and $\mathcal{P}_{z_t}$, for all $t \in \mathcal{T}_I$ can be found (line 9-13), or if the set $\mathcal{T}_I$ is empty. The best-first search terminates with optimized homotopy value $\bar{\mu}_{\cdot|t_0}^*$.

4. Pushing (line 16-18) The pushing procedure is executed if a collision is detected for the intermediate states x_{t_0r+j} , $j \in \mathcal{J}$. Hence, for time steps which are not considered during the collision avoidance in Step 3.. This part then pushes the trajectory out of the obstacle. Therefore, the solution of the optimized homotopy value $\bar{\mu}_{\cdot|t_0}^*$ is changed piecewise towards the known collision free trajectory encoded by $\bar{\mu}^{i^*}$ of step 2., according to:

Algorithm 1. Homotopy Control Algorithm (HCA)

Given: (1), $\hat{x}^i \in \mathcal{X}$, $\tilde{A}_t$, t_0 , $k_0 = t_0r$, H , $\mathcal{P}_{x_{t_0r+j}}$, $\mathcal{T}_H$

Define: $\mathcal{O} := \emptyset$

0. compute $\bar{\mu}_{\cdot|t_0}^*$ according to (20)-(26)

1. **if** $\exists j \in \mathcal{J} : x_{t_0r+j} \in \mathcal{P}_{x_{t_0r+j}}$ **do**

2. compute the best free base trajectory $\bar{\mu}^{i^*}$

3. transform state obstacles $\mathcal{P}_{x_{t_0r+j}}$ into the homotopy space $\tilde{\mathcal{P}}_{\mu_t}$ as described in Sec. 5.2 and determine intersection times $t \in \mathcal{T}_I$.

4. **while** $\mathcal{T}_I \neq \emptyset$ **do**

5. select $q_p \in \mathcal{T}_I$

6. compute for each each successor $w_{q_p} \in \mathcal{W}_{q_p}$ the corresponding optimal $\bar{\mu}_{g_p}^*$ acc. (20)-(26) and (40)-(42), and additionally (40)-(42) for all elements of $\mathcal{O}$

7. store best successor $w_{q_p}^*$ into $\mathcal{O} := \mathcal{O} \cup w_{q_p}^*$

8. compute $z_t(\mu_t)$ with the obtained $\bar{\mu}_{g_p}^*$ and (10)

9. **if** $\exists t \in \mathcal{T}_H : z_t(\mu_t) \in \mathcal{P}_{z_t}$ **do**

10. update $\mathcal{T}_I$ acc. to (38)

11. **else**

12. **break while**

13. **end if**

14. **end while**

15. optimal homotopy value $\bar{\mu}_{\cdot|t_0}^*$, and corresponding optimal trajectory $x_{t_0r+j|t_0r}^*$, with $j \in \mathcal{J}$

16. **if** $\exists j \in \mathcal{J} : x_{t_0r+j} \in \mathcal{P}_{x_{t_0r+j}}$ **do**

17. push $\bar{\mu}_{\cdot|t_0}^*$ towards base trajectory $\bar{\mu}^{i^*}$ acc. to (44)

18. **end if**

19. $t_0 := t_0 + 1$

20. **end if**

Return $(\bar{\mu}_{\cdot|t_0}^*, x_{t_0r+j|t_0r}^*)$ obtained from executing $u_{t_0r+j|t_0r}^*$

Fig. 1. HCA determining circumventing homotopic trajectories for linear systems with moving obstacle.

$$\bar{\mu}_{\cdot|t_0}^* := \bar{\mu}_{\cdot|t_0}^* + \alpha(\bar{\mu}^{i^*} - \bar{\mu}_{\cdot|t_0}^*), \quad (44)$$

by iteratively increasing $\alpha \in [0, 1]$.

The complete procedure is shown in Alg. 1. After computation according to Alg. 1, the first step of the control strategy is applied to the system, the prediction horizon is shifted one step forward, and the computation is repeated.

6. EXAMPLE

The proposed homotopy control method is applied to a point mass moving in a 3-D space, according to the dynamics:

$$\ddot{x}(t) = -\frac{c}{m}\dot{x}(t) + \frac{1}{m}F_x(t) \quad (45)$$

$$\ddot{y}(t) = -\frac{c}{m}\dot{y}(t) + \frac{1}{m}F_y(t) \quad (46)$$

$$\ddot{z}(t) = -\frac{c}{m}\dot{z}(t) + \frac{1}{m}F_z(t), \quad (47)$$

with input vector $\mathbf{u}(t) = (F_x(t), F_y(t), F_z(t))$, the mass m , and a friction constant c . The variables x, y, z describe the cartesian coordinates of the 3-D space in this example. The state vector of system (45)-(47) is given by $\mathbf{x}(t) = (x(t), \dot{x}(t), y(t), \dot{y}(t), z(t), \dot{z}(t))^T$. The system is discretized (ZOH) with discretization time $T = 0.1$ seconds (s). Using parameters $m = 2$ and $c = 1$, the discrete-time system $\hat{\mathbf{x}}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$ has the following matrices:

$$\mathbf{A} = \begin{pmatrix} 1 & 0.0975 & 0 & 0 & 0 & 0 \\ 0 & 0.9512 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.0975 & 0 & 0 \\ 0 & 0 & 0 & 0.9512 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.0975 \\ 0 & 0 & 0 & 0 & 0 & 0.9512 \end{pmatrix}, \quad \mathbf{B} = 1e^{-2} \begin{pmatrix} 0.25 & 0 & 0 \\ 4.88 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 4.88 & 0 \\ 0 & 0 & 0.25 \\ 0 & 0 & 4.88 \end{pmatrix}. \quad (48)$$

The system is 2-step controllable according to Def. 1. Totally three pairs of base trajectories $(\hat{x}^i, \hat{u}^i)$ are chosen to span a region for circumvention, i.e. $n_c = 2$. The final time for all $i \in \mathcal{M}$, is $N = 60$. All state trajectories start at the common state $\mathbf{x}_0^i = \mathbf{x}_s = [0, 0, 0, 0, 0, 0]^T$, and end in the final state $\mathbf{x}_N^i = \mathbf{x}_s = [5, 0, 5, 0, 5, 0]^T$. The base trajectories are shown in Fig. 2 by the three black dotted lines. The upper and lower bound on the additional input $\delta\mathbf{u}_k$, which limit the speed of motion between the homotopic trajectories, is given by $\delta\mathbf{u}_{\max} = [100, 100, 100]^T$ and $\delta\mathbf{u}_{\min} = [-100, -100, -100]^T$. The obstacle $\mathcal{P}_x$ starts at time $k = 0$ at the upper right area in the state space (x, y, z) , shown by the transparent polytope with dashed edges in Fig. 2. It moves towards the lower left region until time $k = 10$, where it remains for the rest of the time. The final position is shown by the green polytope. For better illustration, the cost function (20) is parametrized by $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ chosen as the identity matrix and $\mathbf{R} \in \mathbb{R}^{2 \times 2}$ as diagonal matrix with values $1e^{-3}$, implying that trajectories with lower costs reach the final state along a straighter trajectory.

The computation is performed in Matlab using a PC with an Intel Core i7 (3.4GHz). The blue trajectory (see Fig. 2) is determined by the homotopy approach starting at initial time $k_0 = 0$ with a prediction horizon of $H = 10$. It can be seen that the trajectory first follows an optimal homotopic trajectory, and from $k = 5$ reacts to the moving polytope by transitioning to another homotopic trajectory. This can also be seen in Fig. 3 by the yellow colored bars, which show the computation times for the homotopy control method with receding horizon. It

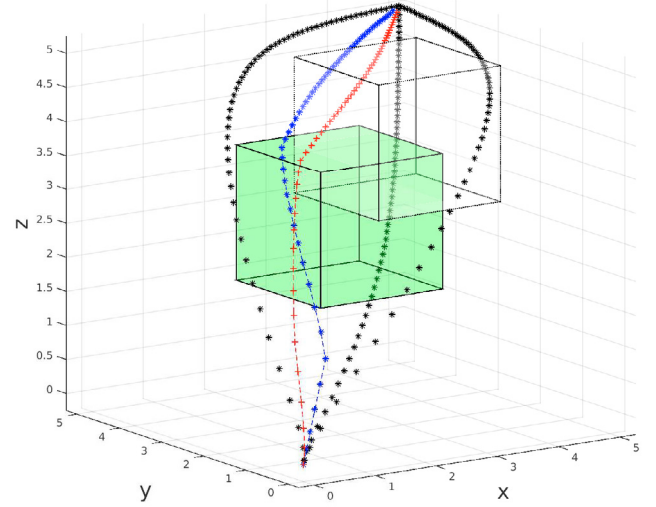


Fig. 2. Simulation of the base trajectories $\hat{x}^1, \hat{x}^2, \hat{x}^3$ (black stars), the obtained trajectory (blue stars) by the homotopy control method with $H = 10$, and the trajectory (red crosses) for $H = 60$. The obstacle $\mathcal{P}_{x_k}$ moves from initial position (transparent dashed polytope) to the final position (green polytope).

can be identified that during the first steps the computation time is very low, with approx. 3 milliseconds (ms). When the obstacle becomes relevant during the prediction horizon, the computation time raises up to 18 ms, and finally falls again when the obstacle is passed. Since the system is 2-step controllable in $r = 2$ steps, the computation has to be performed only in every second time step, which results in a total amount of 30 time steps of computation, as shown in Fig. 3.

For the same scenario, but with a prediction horizon of $H = 60$, the solution of the homotopy control method is given by the red trajectory in Fig. 2. Here, it can be seen that the system reacts at the very beginning to the moving obstacle, and finds a solution which is 2% better in costs, compared to the first case with smaller prediction horizon. The computation time (see Fig. 3 blue bars) is approx. 35 ms at the beginning. Compared to the smaller prediction horizon, the computational load for the entire time is larger. However, for comparison, the computation times of a standard MPC based on MIP solved with CPLEX yields solver times of approx. 800 ms at each iteration (while the costs are 37% better for the second scenario).

Thus, the computation times are drastically reduced by approx. 96% overall, when using the proposed scheme.

7. CONCLUSION

Based on homotopic trajectories, an approach to determination of optimized point-to-point trajectories with input constraints and time-varying moving obstacles is considered. The proposed method identifies time steps at which collisions with a moving obstacle may occur. These are used in a collision avoidance scheme based on tree-search. While the determination of the homotopic trajectories and the determination of the transitioning behavior are accomplished offline, a time-consuming online computation of complete trajectories is avoided, making the online computation efficient. The main advantage is that the homotopy control algorithm performs its computations over an

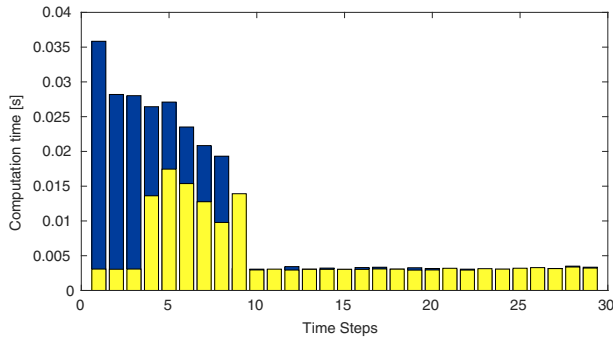


Fig. 3. Comparison between computation times of Alg. 1 with prediction horizon $H = 10$ (yellow bars), and $H = 60$ (blue bars). The computation times are shown for the entire simulation of the receding horizon scheme.

entire trajectory (which is coded by the homotopy parameter) and not as usually done in trajectory optimization for each time step. This reduces the optimization problem to a low-dimensional problem. Furthermore, due to the homotopic space spanned by the base trajectories, a subset of time steps was identified which contains time steps which are relevant for collision avoidance. By considering only these in the optimization, the efficiency of the proposed method increases significantly. The simulations show that the proposed method computes close to optimal trajectories with significantly lower computational effort than standard MPC.

Future work will extend this method to time varying final states and to nonlinear systems.

ACKNOWLEDGEMENTS

Partial financial support by the European Commission through the project UNCOVERCPS (grant No. 643921) is gratefully acknowledged.

REFERENCES

- Blackmore, L. and Williams, B. (2006). Optimal manipulator path planning with obstacles using disjunctive programming. In *American Control Conference*, 3200–3202.
- Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*. SIAM.
- Caron, R.J., McDonald, J.F., and Ponik, C.M. (1989). A degenerate extreme point strategy for the classification of linear constraints as redundant or necessary. *Journal of Optimization Theory and Applications*, 62(2), 225–237.
- Diehl, M., Bock, H., Schlöder, J., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. of Process Control*, 12(4), 577 – 585.
- Ding, H., Reissig, G., and Stursberg, O. (2011). Increasing Efficiency of Optimization-based Path Planning for Robotic Manipulators. In *50th IEEE Conf. on Decision and Control*, 1399–1404.
- Ferguson, D. and Stentz, A. (2007). Anytime, dynamic planning in high-dimensional search space. In *IEEE Conf. on Robotics and Automation*, 1310–1315.
- Gondhalekar, R. and Imura, J. (2010). Least-restrictive move-blocking model predictive control. *Automatica*, 46(7), 1234 – 1240.
- Karaman, S. and Frazzoli, E. (2011). Incremental sampling-based algorithms for optimal motion planning. *J. of Robotics Research*, 30(7), 846–894.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *J. of Robotics Research*, 5, 90–98.
- Kirches, C., Wirsching, L., Bock, H., and Schlöder, J. (2012). Efficient direct multiple shooting for nonlinear model predictive control on long horizons. *J. of Process Control*, 22(3), 540 – 550.
- Kontny, D. and Stursberg, O. (2016a). Fast control using homotopy properties for obstacle-avoidance of systems with input constraints. In *IEEE Conf. on Computer Aided Control System Design*, 654–660.
- Kontny, D. and Stursberg, O. (2016b). Fast optimizing control for non-convex state constraints using homotopy properties. In *55th IEEE Conf. on Decision and Control*, 4894–4900.
- Kuffner, J. and LaValle, S. (2000). RRT-connect: An efficient approach to single-query path planning. In *IEEE Conf. on Robotics and Automation*, 995–1001.
- Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789 – 814.
- Newman, W. and Branicky, M. (1991). Real-time configuration space transforms for obstacle avoidance. *J. of Robotics Research*, 6, 650–667.
- Richards, A. (2015). Fast model predictive control with soft constraints. *European Journal of Control*, 25, 51 – 59.
- Tøndel, P., Johansen, T., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3), 489 – 497.
- Wang, Y. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Tr. on Control Systems Technology*, 18(2), 267–278.
- Zeilinger, M., Jones, C., and Morari, M. (2011). Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. *IEEE Tr. on Automatic Control*, 56(7), 1524–1534.