# A Task Space Approach for Planar Optimal Robot Tube Following

Matthias Oberherber, Hubert Gattringer, Andreas Müller and Michael Schachinger
*Institute of Robotics, Johannes Kepler University Linz, Altenberger Str. 69, Linz, Austria*

Keywords: Robotics, Optimal Path Planning, Optimal Tube Following, Dynamic Modeling, Polygonal Paths.

Abstract: The classical optimal path following problem considers the problem of moving optimally along a predefined geometric path under technological restrictions. In contrast to optimal path following, optimal tube following allows deviations from the initial path within a predefined tube to reduce cost even more. The present paper proposes a modern approach that treats this non-convex problem in task space. This novel method also provides a simple way to derive optimal trajectories within a tube described in terms of polygonal lines. Numerical examples are presented that allow to compare the proposed method to existing joint space approaches.

## 1 INTRODUCTION

Intelligent path planning and motions exploiting the existing hardware in an optimal way are essential for modern production lines. Nevertheless, it is still common to use lines as geometric paths and motion profiles like trapezoidal acceleration or minimum jerk to follow these lines (Siciliano et al., 2009). An improvement for tasks with a defined robot motion can be achieved by the optimal path following problem. The latter consist in following a given path in shortest time possible under kinematic and dynamic constraints. It is an established approach to divide the problem into the geometric path planning followed by a subsequent optimization of the path following trajectory (Bobrow et al., 1985) (Pfeiffer and Johanni, 1987). Nowadays very efficient algorithms are available to solve this kind of problem like (Pham, 2013; Reynoso-Mora et al., 2013; Verscheure et al., 2009b).

If only the start and end points are defined, optimal point-to-point motion planning can be used. In this course the path is no longer defined fix, it is optimized under various restrictions. Beside the kinematic and dynamic restrictions of the robot, also obstacles have to be considered (Shiller and Dubowsky, 1988). The optimization can be performed in task space (Rajan, 1985) or joint space (Antonelli et al., 2005). Usually the calculation times for this kind of problem are rather high, compared to the execution times.

Often a predefined path do not have to be followed exactly, but within a certain surrounding. (Debrouwere et al., 2014) proposed a method called optimal tube following, where the path can vary within a predefined tube around an initial path to reduce the costs compared to the initial path. They solved the problem by introducing a joint angle parametrization, tube approximation and optimization of the path within the tube to reduce execution times.

Robotic tasks are typically given in task space, why the paths are usually also planed there. Due to this the approach proposed in the present paper is based on an optimization of the geometric end-effector (EE) path. This path is defined via splines, which provide many useful properties concerning optimization. We will give a comparison to the joint space approach in (Debrouwere et al., 2014). Further we show how this idea can be extended to achieve optimal trajectories starting with polygonal lines as geometric path, as they are used e.g. in mobile robotics (Zou et al., 2006; Zaverucha, 2005).

The focus of the present paper lies on time optimal trajectories. Nevertheless, the proposed approaches can also be used for other optimality criteria like energy consumption.

In the course of this paper derivatives with respect to the path parameter are marked with a prime $x' = \frac{dx}{ds}$, time derivatives with a dot $\dot{x} = \frac{dx}{dt}$. Small bold letters are used for vectors, capital bold letters for matrices.

The paper is organized as follows: Section 2 starts with the geometric path planning followed by the kinematic and dynamic modeling of the robot. The time-optimal path following problem is introduced in section 3. Afterwards, the tube restriction is defined in section 4. The joint space time-optimal tube following is briefly introduced in section 5. Our task space approach is proposed in section 6. In section 7 we



Oberherber, M., Gattringer, H., Müller, A. and Schachinger, M.
A Task Space Approach for Planar Optimal Robot Tube Following.
DOI: 10.5220/0005980303270334
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 2*, pages 327-334
ISBN: 978-989-758-198-4

give a comparison between the joint space and task space approach. Further we present a numerical example for an industrial robot to show the approach's functionality concerning polygonal initial paths. Finally, we conclude our paper and give an insight into future work in section 8.

## 2 PATH PLANNING AND MODELING

The following section briefly covers the geometric path planning, the kinematic and dynamic modeling of the manipulator. Further, the parametrization of the Equations of Motion (EoM) along the scalar path parameter is introduced.

### 2.1 Geometric Path Planning

Spline functions provide many useful properties like the convex hull property, local support property, continuity or the invariance against translation, rotation and scaling (Piegl and Tiller, 1997). These properties, and the existence of very fast algorithms for numerically calculations, like de Boor's algorithm (DeBoor, 1978), make splines to an interesting tool in the fields of robotics. A three-dimensional spline curve, describing e.g. the EE-position $\boldsymbol{r}_\mathrm{E}$ of a robot can be written as

$$\boldsymbol{r}_\mathrm{E}(s) = \sum_{l=1}^{n_\mathrm{D}} \boldsymbol{d}_l N_l^d(s), \tag{1}$$

with the scalar path parameter $s = [s_\mathrm{B}, s_\mathrm{E}]$ ($s_\mathrm{B}$ - begin of the path, $s_\mathrm{E}$ - end of the path). $N_l^d(s)$ denote the B-spline basis functions of degree $d$, while $\boldsymbol{d}_l$ are the control points (CPs). There are different ways to define the basis functions as local or global support functions. Once the path is defined, the derivatives of the path with respect to the path parameter $\boldsymbol{r}'_E(s), \boldsymbol{r}''_E(s)$ can be calculated easily by differentiation of the basis functions. For details we refer to (DeBoor, 1978) and (Piegl and Tiller, 1997). An exemplary spatial EE-path, defined via splines, is shown Figure 1.

### 2.2 Kinematic Model

The forward kinematics of a manipulator provides the EE- coordinates $\boldsymbol{z}_\mathrm{E}^T(s) = [\boldsymbol{r}_\mathrm{E}^T(s), \boldsymbol{\varphi}_\mathrm{E}^T(s)]$ (position $\boldsymbol{r}_\mathrm{E}$ , orientation $\boldsymbol{\varphi}_\mathrm{E}$, e.g. in Cardan angles) as a function of the joint positions $\boldsymbol{q}$ of the robot. Conversely, the inverse kinematics $\boldsymbol{q}(s) = \boldsymbol{f}^{-1}(\boldsymbol{z}_\mathrm{E}(s))$ provides the joint angles as a function of desired EE - coordinates. There are different ways to solve the locally, but not

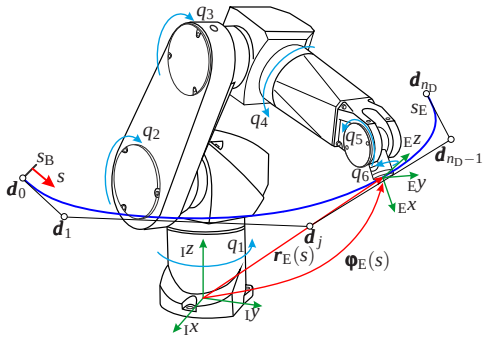


Figure 1: Six-axis industrial robot with joint positions $\boldsymbol{q}$.

globally unique problem of the inverse kinematics. We use a numerical approach (Siciliano et al., 2009) based on the relation

$$\begin{pmatrix} \boldsymbol{v}_\mathrm{E} \\ \boldsymbol{\omega}_\mathrm{E} \end{pmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2}$$

with the EE - velocities $\boldsymbol{v}_\mathrm{E}$, $\boldsymbol{\omega}_\mathrm{E}$ represented in the inertial frame and the geometric Jacobian

$$\boldsymbol{J} = \begin{pmatrix} \frac{\partial \boldsymbol{v}_\mathrm{E}}{\partial \boldsymbol{q}} \\ \frac{\partial \boldsymbol{\omega}_\mathrm{E}}{\partial \boldsymbol{q}} \end{pmatrix}. \tag{3}$$

With the Jacobian and $\boldsymbol{r}'_\mathrm{E}(s)$, $\boldsymbol{r}''_\mathrm{E}(s)$, the prime quantities of the joint angles can be calculated as

$$\boldsymbol{q}' = \boldsymbol{J}^{-1}\boldsymbol{r}'_\mathrm{E} \tag{4}$$

$$\boldsymbol{q}'' = \boldsymbol{J}^{-1}\left[\boldsymbol{r}''_\mathrm{E} - \boldsymbol{J}'\boldsymbol{q}'\right]. \tag{5}$$

Near to singularities the Jacobian gets singular. Due to this a robust inversion $\boldsymbol{J}^+ = (\boldsymbol{J}^T\boldsymbol{J} + \delta^2\mathbf{I})^{-1}\boldsymbol{J}^T$ is used, with the identity matrix $\mathbf{I}$ and a damping factor $\delta$ that is only active near to singularities.

Using the chain rule for differentiation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\mathrm{d}x}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t} = x'\dot{s} \tag{6}$$

and the identity $\ddot{s} = (\mathrm{d}\dot{s}/\mathrm{d}s)(\mathrm{d}s/\mathrm{d}t) = (\dot{s}^2)'/2$, the joint velocities and accelerations are

$$\dot{\boldsymbol{q}}(s) = \boldsymbol{q}'(s)\dot{s} \tag{7}$$

$$\ddot{\boldsymbol{q}}(s) = \boldsymbol{q}''(s)\dot{s}^2 + \frac{1}{2}\boldsymbol{q}'(s)(\dot{s}^2)'. \tag{8}$$

### 2.3 Dynamic Model

A dynamic model of the robot is necessary to be able to include torque constraints in the optimization. It is also used for simulation purposes. The dynamic model can be written in form of the EoM as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{9}$$

with the position dependent positive definite mass matrix $\boldsymbol{M}(\boldsymbol{q})$. Joint angles $\boldsymbol{q}$ are the minimal coordinates of the system. The vector $\boldsymbol{g}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ contains all

nonlinear terms, like Coriolis-, centrifugal-, friction-, or gravitational forces. The motor torques are denoted with $\boldsymbol{\tau}$. We derive the EoM with the help of the Projection Equation (Bremer, 2008). For the path tracking problem a representation of the EoM in the parameter range is necessary, and can be written as

$$\boldsymbol{\tau} = \boldsymbol{a}(s)\,(\dot{s}^2)' + \boldsymbol{b}(s)\,\dot{s}^2 + \boldsymbol{c}(s) + \boldsymbol{d}_{\mathrm{v}}(s)\dot{s}. \tag{10}$$

A method, based on the Projection Equation, to derive the parameters $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}_{\mathrm{v}}$ analytically is proposed in (Gattringer et al., 2014). Numerical methods to derive them are presented in (Johanni, 1988) and (Geu Flores and Kecskemthy, 2012).

## 3 PATH FOLLOWING PROBLEM

The time optimal robot path following (PF) problem regards the problem of generating trajectories that follow predefined EE paths in shortest possible time, taking into account kinematic and dynamic constraints. Since the path is defined in the parameter range, the aim is to calculate an optimal relation $t(s)$ between time and path parameter. We are looking for optimal solutions for the cycle time $t_{\mathrm{E}}$ of a process. Since this time is the solution of the optimization, it is a priori unknown. A change of the integration variable from $t$ to $s$ leads to

$$t_{\mathrm{E}} = \int_0^{t_{\mathrm{E}}} 1\,\mathrm{d}t = \int_{s_{\mathrm{B}}}^{s_{\mathrm{E}}} \frac{1}{\dot{s}(s)}\,\mathrm{d}s. \tag{11}$$

By introducing the abbreviation $z = \dot{s}^2$ the following optimization problem in parameter space

$$\begin{aligned} \min_{z(\cdot)} \; & \int_{s_{\mathrm{B}}}^{s_{\mathrm{E}}} \frac{1}{\sqrt{z(s)}}\mathrm{d}s \qquad (12)\\ \text{s.t. } & \underline{\dot{\boldsymbol{q}}} \le \boldsymbol{q}'(s)\sqrt{z(s)} \le \overline{\dot{\boldsymbol{q}}} \\ & \underline{\ddot{\boldsymbol{q}}} \le \boldsymbol{q}''(s)z(s) + \frac{1}{2}\boldsymbol{q}'(s)z'(s) \le \overline{\ddot{\boldsymbol{q}}} \\ & \underline{\boldsymbol{\tau}} \le \boldsymbol{a}(s)z'(s) + \boldsymbol{b}(s)z(s) + \boldsymbol{c}(s) + \boldsymbol{d}_{\mathrm{v}}(s)\sqrt{z(s)} \le \overline{\boldsymbol{\tau}}. \end{aligned}$$

has to be solved. It is assumed that the joint velocity restrictions $\underline{\dot{\boldsymbol{q}}}, \overline{\dot{\boldsymbol{q}}}$, joint acceleration restrictions $\underline{\ddot{\boldsymbol{q}}}, \overline{\ddot{\boldsymbol{q}}}$ and torque restrictions $\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}$ are constant along the path.

There exist several different approaches to solve the classical path following problem. In the 1980's (Bobrow et al., 1985) proposed a method based on numerical integration and the search of switching points in the phase plane, while (Shin and McKay, 1986) used a dynamic programming approach to solve this kind of problem. Current works in this research field are for example an online log-barrier optimization (Verscheure et al., 2009b), or a Second Order Cone Program (SOCP) reformulation of the problem (Verscheure et al., 2009a). These approaches make use of a convex formulation of the optimization problems. Nevertheless, (12) is non-convex, due to the inclusion of viscous friction (parameter $\boldsymbol{d}_{\mathrm{v}}$). (Reynoso-Mora et al., 2013) proposed a method called convex-relaxation, that empowers them to reformulate the optimization as a SOCP anyway.

## 4 TUBE-RESTRICTION

The tube should constrain the EE-position within a defined distance $r_{\mathrm{T}}(s_i)$ to the initial path. Debreuwere et al. proposed different ways to approximate the tube. They show, that a linear approximation is computationally more expensive than a quadratic approximation, see Figure 2. In this paper we will use the quadratic approximation, defined as

$$\begin{aligned} C_{\mathrm{r_E}} = \{ & \boldsymbol{r}_{\mathrm{E}}(s_i) | \qquad (13)\\ & (\boldsymbol{r}_{\mathrm{E}}(s_i) - \boldsymbol{r}_{\mathrm{E},0}(s_i))^T (\boldsymbol{r}_{\mathrm{E}}(s_i) - \boldsymbol{r}_{\mathrm{E},0}(s_i)) \le r_{\mathrm{T}}(s_i)^2 \big\}, \end{aligned}$$

where $C_{\mathrm{r_E}}$ is the defined space of the tube. The better performance of the quadratic approximation goes in contrast with a higher deviation of the approximated to the original tube. An approximation of the tube is necessary to allow a tangential relocation of a discrete path point $\boldsymbol{r}_{\mathrm{E}}(s_i)$ against the initial path $\boldsymbol{r}_{\mathrm{E},0}(s_i)$, see Figure 2. For a detailed explanation of the tube definitions we refer to (Debrouwere et al., 2014).

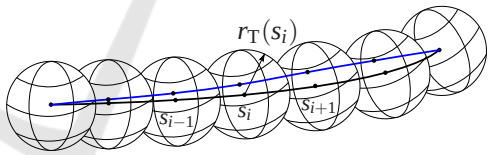


Figure 2: Quadratic approximation of the tube with the initial path $\boldsymbol{r}_{\mathrm{E},0}(s)$ in black and a modified path $\boldsymbol{r}_{\mathrm{E}}(s)$ in blue.

## 5 JOINT SPACE APPROACH

(Debrouwere et al., 2014) solves the time optimal PF problem in joint space, with the joint angles $\boldsymbol{q} \in \mathbb{R}^{n_{\mathrm{DOF}}}$ as minimal coordinates. These are parametrized along the path within an upper $\overline{\boldsymbol{q}}$ and a lower $\underline{\boldsymbol{q}}$ boundary using a parameter $0 \le \boldsymbol{\lambda}(s) \le 1$. Thus, they can be written as

$$\boldsymbol{q}(s) = \boldsymbol{\lambda}(s)\underline{\boldsymbol{q}}(s) + (1 - \boldsymbol{\lambda}(s))\,\overline{\boldsymbol{q}}(s). \tag{14}$$

Parameter $\boldsymbol{\lambda}(s) = [\lambda_1(s), \ldots, \lambda_{n_{\mathrm{DOF}}}(s)]^T$ can be defined as a spline curve with $k_{\mathrm{c}}$ CPs $\boldsymbol{\gamma}$. These CPs represent the optimization variables and the optimization

problem can be written as

$$\min_{z(\cdot),\boldsymbol{\gamma}} \int_{s_B}^{s_E} \frac{1}{\sqrt{z(s)}} ds \tag{15}$$

$$\text{s.t. } \underline{\dot{\boldsymbol{q}}} \le \boldsymbol{q}'(s)\sqrt{z(s)} \le \overline{\dot{\boldsymbol{q}}}$$

$$\underline{\ddot{\boldsymbol{q}}} \le \boldsymbol{q}''(s)z(s) + \frac{1}{2}\boldsymbol{q}'(s)z'(s) \le \overline{\ddot{\boldsymbol{q}}}$$

$$\underline{\boldsymbol{\tau}} \le \boldsymbol{a}(s)z'(s) + \boldsymbol{b}(s)z(s) + \boldsymbol{c}(s) + \boldsymbol{d}_v(s)\sqrt{z(s)} \le \overline{\boldsymbol{\tau}}$$

$$0 \le \boldsymbol{\lambda}(s) \le 1$$

$$\boldsymbol{r}_E(\boldsymbol{\lambda}(s)) \in C_{r_E}$$

$$\boldsymbol{r}_E(s_B) = \boldsymbol{r}_{E,B}$$

$$\boldsymbol{r}_E(s_E) = \boldsymbol{r}_{E,E}.$$

The last two equality constraints indicate that the start point ($\boldsymbol{r}_{E,B}$) and end point $\boldsymbol{r}_{E,E}$ of the path are fixed.

# 6 TASK SPACE APPROACH

The following section introduces the task space approach. The discussion is limited to planar paths. First the method for general planar paths is introduced. Following, this section shows how this approach can be used to derive optimal trajectories when the initial path is described in terms of polygonal lines.

## 6.1 General Solution Strategy

The solution strategy of the proposed task space approach is based on a separation of path planning and the calculation of an optimal motion profile. The idea is similar to the one proposed in (Rajan, 1985). First an initial path for the EE of the robot $\boldsymbol{r}_{E,0}(s)$ is planned via the CPs $\boldsymbol{D} = [\boldsymbol{d}_0, \ldots, \boldsymbol{d}_j, \boldsymbol{d}_{n_D}]^T$ using (1). A time optimal solution for this PF problem is calculated. In a next step, a path optimization algorithm in the sense of variation of the CPs is used to get even faster overall trajectories. Therefore, with the new CPs, a new EE-path is calculated and again a time optimal PF problem for this path is solved. This is repeated till convergence (no further improvement of the execution time $t_E$) is achieved.

Figure 3 shows a flow-chart of this two-stage optimization strategy. An advantage of this method is, that, depending on the restrictions that have to be considered, any of the consisting and well tested time optimal PF-algorithms like (Bobrow et al., 1985), (Shin and McKay, 1986), (Verscheure et al., 2009b), (Verscheure et al., 2009a), (Reynoso-Mora et al., 2013), (Pham, 2013) can be used. The choice of the path optimization algorithm is not easy since the problem is non-convex.

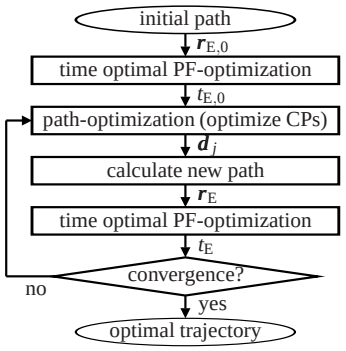


Figure 3: Solution strategy for the multistage optimization.

## 6.2 Task Space Approach for Planar Spline Paths

The following procedure can not only be used for planar robots, but also for multi-axis robots, when the EE-path is defined within a plane. In this plane the path, as well as the tube boundaries can be represented by splines. The CP locations can be defined as a convex combination

$$\boldsymbol{d}_j = \underline{\boldsymbol{d}}_j \alpha_j + \overline{\boldsymbol{d}}_j (1 - \alpha_j) \text{ for } j = 1 \ldots n_D - 1. \tag{16}$$

of the bounding CPs $\overline{\boldsymbol{d}}_j$ and $\underline{\boldsymbol{d}}_j$ of the tube curves $\overline{\boldsymbol{r}}_E$ and $\underline{\boldsymbol{r}}_E$, as shown in Figure 4. The parameters $0 < \alpha_j < 1$ represent the optimization variables for the path optimization. Thus, the optimization problem for the two-stage approach can be written as

$$\min_{\boldsymbol{\alpha}} J \tag{17}$$

$$s.t. \;\; 0 < \alpha_j < 1 \text{ for } j = 1 \ldots n_D - 1$$

$$\boldsymbol{r}_E(s, \boldsymbol{\alpha}) \in C_{r_E},$$

wherein the time optimal PF optimization (12) has to be solved in every iteration of the path optimization.

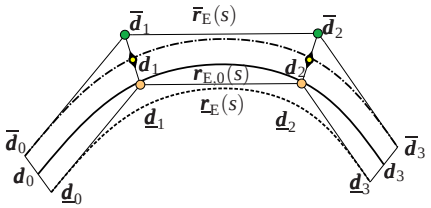


Figure 4: Task space approach: Planar tube and path.

## 6.3 Tube Following with Polygonal Lines

In (16) bounding CPs $\underline{\boldsymbol{D}} = [\underline{\boldsymbol{d}}_0, \ldots, \underline{\boldsymbol{d}}_j, \underline{\boldsymbol{d}}_{n_D}]^T$ and $\overline{\boldsymbol{D}} = [\overline{\boldsymbol{d}}_0, \ldots, \overline{\boldsymbol{d}}_j, \overline{\boldsymbol{d}}_{n_D}]^T$ are defined. Nevertheless, the tube restriction (13) has to be considered in the optimization problem (17). Now, one may ask how they should

be defined to guarantee that the resulting curve with the CPs $\boldsymbol{d}_j = \underline{\boldsymbol{d}}_j \alpha_j + \overline{\boldsymbol{d}}_j (1-\alpha_j)$ lies within the tube. Unfortunately it is not straight forward to calculate intersection points of splines (Mørken et al., 2009), and following it is hard to define the CPs boundings in a reasonable way for splines.

But, as mentioned in section 1, the usage of straight lines with subsequent edge combining is still a wide spread approach for path planning in industrial applications nowadays. Also the global path planner of mobile robots often provide polygonal lines, see e.g. (Zou et al., 2006). For such cases it is a manageable problem to define the position of the bounding CPs in a reasonable way. In the course of this section we will present an approach to achieve optimal trajectories along continuous paths, starting with polygonal lines surrounded by polygonal tube restrictions.

### 6.3.1 Local Convex Hull Property

To eliminate the possibility of an intersection between the resulting spline and the tube, we will use the local convex hull property, as for example proposed in (Mørken et al., 2009):

> If the unions of all local convex hulls of two spline curves are disjunct, so are the splines.

If we are able to define the bounding CPs in a way that this condition holds, the tube approximation (13) and the related restriction in the optimization problem (17) can be omitted.

The local convex hull of a spline curve with degree $d$ is the smallest convex set, that contains the points $\boldsymbol{d}_{j-d} \dots \boldsymbol{d}_j$. In Figure 5 the local convex hulls (exemplary for $d = 3$) of the bounding CPs $\underline{\boldsymbol{D}}$ and $\overline{\boldsymbol{D}}$ are shown. The presence of multiple CPs let the convex hull degenerate to lines or points in some cases. The spline curves $\underline{\boldsymbol{r}}(s)$ and $\overline{\boldsymbol{r}}(s)$ defined by $\underline{\boldsymbol{D}}$ and $\overline{\boldsymbol{D}}$ lie within the union of the local convex hulls. If an intersection between these two curves can be excluded, the resulting spline $\boldsymbol{r}_\mathrm{E}(s)$, whose CPs are calculated as a convex combination of $\underline{\boldsymbol{D}}$ and $\overline{\boldsymbol{D}}$, lies guaranteed within the $\underline{\boldsymbol{r}}(s)$ and $\overline{\boldsymbol{r}}(s)$ and following within the tube.

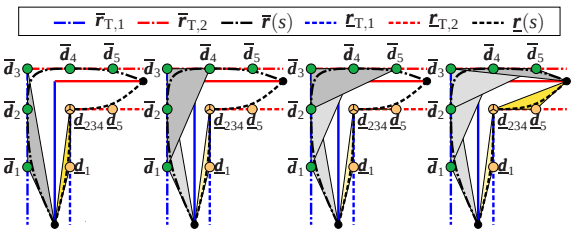


Figure 5: Local convex hulls for $\overline{\boldsymbol{r}}(s)$ (gray), $\underline{\boldsymbol{r}}(s)$ (yellow).

### 6.3.2 Control Point Placement

We will use the knowledge from section 6.3.1 to define the CPs positions in a way that the resulting curve guaranteed lies within the defined tube. For a reasonable definition following points have to be considered:

- The polygonal path is defined by the points $\boldsymbol{p}_j$ in task space coordinates. The angle between two lines is denoted as $\varphi$.
- The combination of two lines is called segment. Following a path consists of $n_\mathrm{L}$ lines, respectively $n_\mathrm{S} = n_\mathrm{L} - 1$ segments.
- The descriptions 'inner' and 'outer' corner refer to a movement along the path from $s_\mathrm{B}$ to $s_\mathrm{E}$.
- The local convex hulls of $\underline{\boldsymbol{r}}(s)$ and $\overline{\boldsymbol{r}}(s)$ must not intersect.
- The necessary number of CPs for one segment depends on the spline degree $d$.
- No overlaps between CPs are allowed.
- The projected tube lengths $\overline{l}_{T,i}$ and $\underline{l}_{T,i}$ denote the projection of $\underline{\boldsymbol{r}}_{T,i}$ on $\overline{\boldsymbol{r}}_{T,i}$ and conversely the projection of $\overline{\boldsymbol{r}}_{T,i}$ on $\underline{\boldsymbol{r}}_{T,i}$, as shown in Figure 6 right.

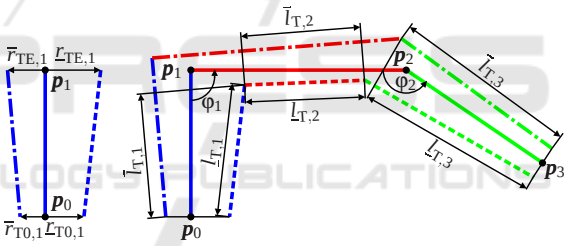


Figure 6: Left: Tube distance definition. Right: polygonal path with tube restrictions and projected tube lengths.

**Algorithm:**
For all lines $k = 1 \dots n_\mathrm{L}$

1. Treat special cases: For $\varphi = \pi$ the lines are combined to one, the case $\varphi = 0$ is excepted. Assuming long lines in comparison to the tube distances avoids the case that a line lies completely within the tube of an adjacent line.
2. Define the tubes: The tubes are defined as lines with orthogonal distances to the line $\underline{r}_{\mathrm{T0},k}$ and $\overline{r}_{\mathrm{T0},k}$ at the start and $\underline{r}_{\mathrm{TE},k}$ and $\overline{r}_{\mathrm{TE},k}$ at the end of the line as shown in Figure 6 left.
3. Generate the overall tube: Calculate the intersection points between $\overline{\boldsymbol{r}}_{\mathrm{T},k}$ and $\overline{\boldsymbol{r}}_{\mathrm{T},k+1}$ and also between $\underline{\boldsymbol{r}}_{\mathrm{T},k}$ and $\underline{\boldsymbol{r}}_{\mathrm{T},k+1}$. Complete the tube by extrapolating and cutting the particular tubes at intersection points as shown in Figure 6 right.
4. Place CPs at half of projected tube lengths: Bounding CPs $\underline{\boldsymbol{d}}$ are placed at $\underline{l}_{\mathrm{T},k}/2$, $\overline{\boldsymbol{d}}$ at $\overline{l}_{\mathrm{T},k}/2$.

5. Place CPs around the corners: Determine the connection line between the outer CPs (dotted and dotdashed in Figure 7 left). If this connection line lies within the tube (dotdashed), the outer CPs are also placed there. Otherwise (dotted), the line is parallel shifted to the inner corner and the outer CPs are placed at the intersection points of the line with the outer tube. The inner bounding CPs are placed at the inner tube corner.
6. Place CPs at the tube corners: At the tube corners bounding control points of multiplicity $d-1$ are placed. The number of optimization variables can be reduced by placing the outer CPs onto the inner tube corner, as shown in Figure 7 right. Then it is guaranteed that the path touches the inner tube corner. This should be an advantage in most cases, except if the inner corner lies near to a singularity.

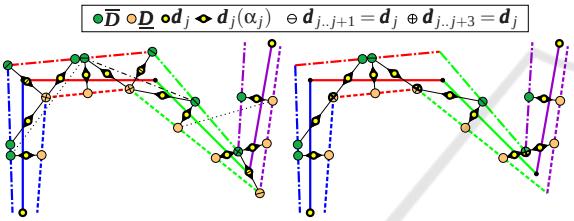


Figure 7: Left: Control point placement for $d=3$. Right: Reduction of optimization variables.

## 7 EXAMPLES AND RESULTS

At first this section gives a comparison of the proposed task space approach with the joint space approach in terms of a 2D-example defined in (Debrouwere et al., 2014). Additionally, an example with polygonal line paths for the industrial robot Stäubli RX130L is presented. The following examples are programmed in Matlab and run on a standard PC with a 3.4 GHz processor.

### 7.1 General Planar Paths

The parameters of the 2DOF serial manipulator in Figure 8 are: $l_1 = l_2 = 1$m, $r_{s,1} = r_{s,2} = 0.5$m and $m_1 = m_2 = 1$kg. With $\overline{\boldsymbol{\tau}} = [30, 10]$ Nm and $\underline{\boldsymbol{\tau}} = -\overline{\boldsymbol{\tau}}$, the motor torque restrictions are given, see (Debrouwere et al., 2014) for details. The initial joint motion is defined as

$$\boldsymbol{q}_0(s) = \begin{pmatrix} -4\pi(s^2 - s) \\ -\pi s \end{pmatrix}. \qquad (18)$$

The resulting EE path, see Figure 8, is discretized into $n = 100$ pieces. The joint angle restrictions are $\underline{\boldsymbol{q}}(s) = \frac{1}{2}\boldsymbol{q}_0(s)$ and $\overline{\boldsymbol{q}}(s) = 2\,\boldsymbol{q}_0(s)$, the spline $\boldsymbol{\lambda}(s)$ is linear (degree 1). With $r_T(s) = 0.1(1-s^4)$ the position dependent tube size is defined. For the task space approach the tube and the EE path have to be approximated by splines, which can be achieved in a satisfying manner with $n_D > 10$.

In Table 1 a listing of resulting execution and calculation times is given. It should be mentioned that the calculation times strongly depend on the chosen path discretization and chosen optimization criteria. To allow a fair comparison we defined them equally as much as possible for both approaches. The abbreviations are TF_TS for tube-following task space, and TF_JS for tube-following in the joint space. These results show, that both approaches perform nearly equally for this very simple example. The percentage savings in execution time (last column in Table 1) of the joint space approach are comparable to those presented in (Debrouwere et al., 2014). Since we use non-optimized Matlab code for the optimization, calculation times are slightly higher. With the number of CPs the quality of the solution can be modified. The decrease in execution time stands in contrast to a higher calculation time. A view on Figure 8 shows, that the optimal paths of the two approaches are quite different, in spite of the similar execution times. This is also an evidence on the non-convex characteristic of the problem. The resulting motor torques normalized on the maximum values as well as the optimal trends $z_{opt}(s)$ are shown in Figure 9.

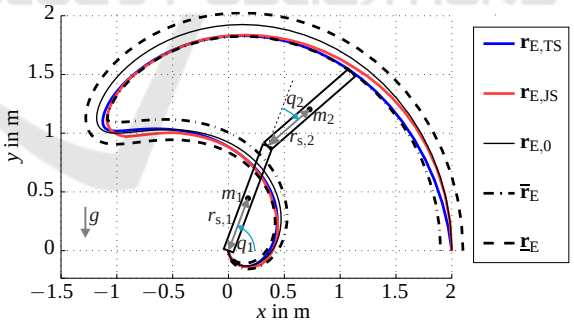


Figure 8: Optimal end-effector paths and tube constraints.

Table 1: Calculation $t_{CPU}$ and execution times $t_E$.

| Approach | $n_D$, $k_c$ | $t_E$ in s | $t_{CPU}$ in s | TF/PF |
|---|---|---|---|---|
| PF | - | 2.30 | 0.334 | - |
| TF_TS | 11 | 2.20 | 4.16 | 4.35% |
| TF_TS | 13 | 2.19 | 3.81 | 4.78% |
| TF_TS | 15 | 2.17 | 5.36 | 5.65% |
| TF_JS | 11 | 2.17 | 3.98 | 5.65% |
| TF_JS | 13 | 2.14 | 5.46 | 6.96% |

**Comparison.** Both approaches have their certain pros and cons. The task space approach does not re-

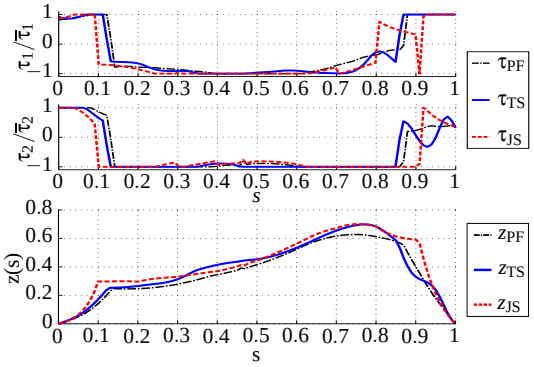


Figure 9: Top: Normalized torques $\boldsymbol{\tau}/\overline{\boldsymbol{\tau}}$. Bottom: Optimal trends $z_{\text{opt}}(s)$ for all approaches.

quire a joint angle parametrization, which drives up the optimization time if chosen too big, or additionally restrict the problem if chosen too small. The number of optimization variables for the path optimization depends on the path length, respectively the number of chosen CPs for the task space approach $n_{\text{opt}} \leq n_{\text{D}} - 2$. While the number of optimization variables for the path optimization at the joint space approach also depends on the number of degrees of freedom $n_{\text{opt}} = n_{\text{DOF}}\, k_{\text{c}}$. The necessity of the inverse kinematics may be seen as disadvantage of the task space approach. But, it empowers us to simply include desired EE-orientations $\boldsymbol{\varphi}_{\text{E}}(s)$ along the path. While an additional equality constraint has to be added to the optimization problem (15) for the joint space approach to consider a desired EE-orientation.

A big advantage of the task space approach is that optimal trajectories defined in terms of polygonal lines as shown in section 6.3 can be derived, where the tube approximation can be omitted due to an intelligent placement of the CPs, as shown in the following example.

## 7.2 Polygonal Lines

The following example is implemented for the industrial robot Stäubli RX130L in Figure 1. For simplicity, joints $q_4$, $q_5$ and $q_6$ are locked. In contrast to the simple example in the section before, several effects like viscous and Coulomb friction are considered. The time optimal PF optimization is done with a dynamic programming approach. Three lines, in the $y$ - $z$ plane in front of the robot with the points

$$\boldsymbol{p}_0 = [0.75\ -0.5\ 0.1]\ \text{m} \quad \boldsymbol{p}_1 = [0.75\ -0.5\ 1]\ \text{m}$$
$$\boldsymbol{p}_2 = [0.75\ 0.5\ 1]\ \text{m} \quad \boldsymbol{p}_3 = [0.75\ 0.7\ 0.1]\ \text{m}$$

represent the initial path, shown in Figure 10. This path is discretized into $n = 150$ equidistant pieces. The robot's hardware restrictions for the joint velocities and motor torques are given as

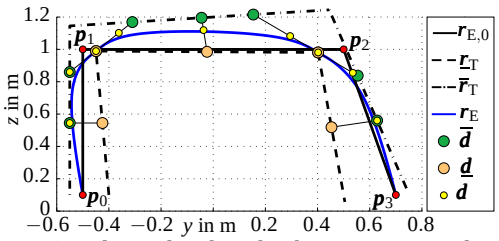


Figure 10: Polygonal path with tube restrictions and resulting optimal path with the associated CPs.

$$\overline{\dot{\boldsymbol{q}}} = -\underline{\dot{\boldsymbol{q}}} = [4.45, 4.45, 5.7]\text{rad/s} \tag{19}$$
$$\overline{\boldsymbol{\tau}} = -\underline{\boldsymbol{\tau}} = [9.05, 9.05, 6.4]\text{Nm}. \tag{20}$$

To show some special cases, different tube distances are chosen for the lines:

$$\underline{r}_{\text{T0}} = \{10, 1, 10\}\ \text{cm}, \quad \overline{r}_{\text{T0}} = \{5, 15\,, 1\}\ \text{cm}$$
$$\underline{r}_{\text{TE}} = \{5, 2, 20\}\ \text{cm}, \quad \overline{r}_{\text{TE}} = \{5, 25, 5\}\ \text{cm}.$$

The BOBYQA algorithm from NLOPT library (Johnson, 2011) was used via the Matlab interface for path optimization. For validation purposes several other algorithms from the NLOPT library were proved, resulting in nearly the same trajectory within different calculation times, strongly depending on the discretization. Resulting joint velocities and torques are shown in Figure 11. Execution and calculation times are given in Table 2. Where $\boldsymbol{r}_{\text{E},0}$, indicates the line path, $\boldsymbol{r}_{\text{E}}$ the optimal and $\boldsymbol{r}_{\text{E},\boldsymbol{\alpha}_0}$ the initial path with $\alpha_j = 0.5, j = 1..n_{\text{opt}}$. Fixed corner CPs, as described in section 6.3.2, reduce the number of optimization variables from $n_{\text{opt}} = 9$ to 7. This influences not only the calculation times, but also generates a better initial solution. The percentage time savings refer to the time optimal motion along the three lines, where the robot has to stop after each line.

Table 2: Execution and calculation times.

| | $\boldsymbol{r}_{\text{E},0}$ | $\boldsymbol{r}_{\text{E},\boldsymbol{\alpha}_0}$ | $\boldsymbol{r}_{\text{E}}$ | $\boldsymbol{r}_{\text{E},\boldsymbol{\alpha}_0}$ | $\boldsymbol{r}_{\text{E}}$ |
|---|---|---|---|---|---|
| | | $n_{\text{opt}} = 9$ | | $n_{\text{opt}} = 7$ | |
| $t_{\text{E}}$ in s | 1.46 | 1.46 | 1.14 | 1.13 | 1.05 |
| $t_{\text{CPU}}$ in s | 0.85 | 0.72 | 18 | 0.73 | 12 |
| save in % | — | 0 | 21.7 | 14.82 | 22.8 |

# 8 CONCLUSIONS

In the present paper we proposed a task space approach to solve the planar optimal tube following problem. The focus is on time optimality, nevertheless other optimality criteria like energy consumption or path length can also be considered with this

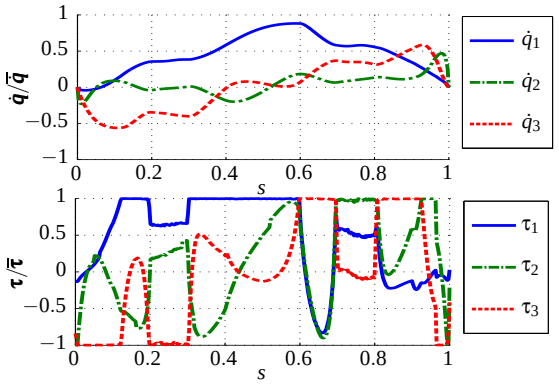


Figure 11: Top: Normalized joint velocities. Bottom: Normalized motor torques.

method. By means of numerical examples we have shown that the proposed approach performs nearly equally to an existing one, with the benefit that optimal trajectories from polygonal paths without an additional tube restriction can be derived. Future works will include among other things further improvements of the algorithm, an automatism for the tube generation and the extension to spatial paths.

## ACKNOWLEDGEMENTS

This work has been supported by the Linz Center of Mechatronics (LCM) in the framework of the Austrian COMET-K2 program.

## REFERENCES

Antonelli, G., Chiaverini, S., Palladino, M., Gerio, G. P., and Renga, G. (2005). Joint space point-to-point motion planning for robots: An industrial implementation. In Ztek, P., editor, *Proceedings of the 16th IFAC World Congress*.

Bobrow, J. E., Dubowsky, S., and Gibson, J. S. (1985). Time-optimal control of robotic manipulators along specified paths. *International Journal Robotics Research*, 4:3–17.

Bremer, H. (2008). *Elastic Multibody Dynamics: A Direct Ritz Approach*. Springer Verlag, Heidelberg.

DeBoor, C. (1978). *A practical guide to splines*. Springer.

Debrouwere, F., Van Loock, W., Pipeleers, G., and Swevers, J. (2014). Time-optimal tube following for robotic manipulators. In *Advanced Motion Control (AMC),2014 IEEE 13th International Workshop on*, pages 392–397.

Gattringer, H., Oberherber, M., and Springer, K. (2014). Extending continuous path trajectories to point-to-point trajectories by varying intermediate points. *International Journal of Mechanics and Control*, 15(01):35–43.

Geu Flores, F. and Kecskemthy, A. (2012). Time-optimal path palanning along specified trajectories. In Gattringer, H. and Gerstmayr, J., editors, *Multibody System Dynamics, Robotics and Control*, pages 1–15.

Johanni, R. (1988). *Optimale Bahnplanung bei Industrierobotern*. Technische Universität München.

Johnson, S. G. (2011). *The NLopt nonlinear-optimization package*.

Mørken, K., Reimers, M., and Schulz, C. (2009). Computing intersections of planar spline curves using knot insertion. *Comput. Aided Geom. Des.*, 26(3):351–366.

Pfeiffer, F. and Johanni, R. (1987). A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation*, 3(2):115 –123.

Pham, Q. (2013). A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *CoRR*, abs/1312.6533.

Piegl, L. A. and Tiller, W. (1997). *The NURBS book (2. ed.)*. Monographs in visual communication. Springer.

Rajan, V. (1985). Minimum time trajectory planning. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 759–764.

Reynoso-Mora, P., Chen, W., and Tomizuka, M. (2013). On the time-optimal trajectory planning and control of robotic manipulators along predefined paths. In *American Control Conference (ACC), 2013*, pages 371–377.

Shiller, Z. and Dubowsky, S. (1988). Global time optimal motions of robotic manipulators in the presence of obstacles. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 370–375 vol.1.

Shin, K. G. and McKay, N. D. (1986). A dynamic programming approach to trajectory planning of robotic manipulators. *Automatic Control, IEEE Transactions on*, 31(6):491–500.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics - Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing series. Springer.

Verscheure, D., Demeulenaere, B., Swevers, J., De Schutter, J., and Diehl, M. (2009a). Time-optimal path tracking for robots: A convex optimization approach. *Automatic Control, IEEE Transactions on*, 54(10):2318–2327.

Verscheure, D., Diehl, M., De Schutter, J., and Swevers, J. (2009b). Recursive log-barrier method for on-line time-optimal robot path tracking. In *American Control Conference, 2009. ACC '09.*, pages 4134 –4140.

Zaverucha, G. (2005). Approximating polylines by curved paths. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 2, pages 758–763 Vol. 2.

Zou, A.-M., Hou, Z.-G., Tan, M., and Liu, D. (2006). Path planning for mobile robots using straight lines. In *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pages 204–208.