Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Optimal Tube Following for Robotic Manipulators ⋆

Frederik Debrouwere * **Wannes Van Loock** * **Goele Pipeleers** *
Jan Swevers *

* *Division PMA, Department of Mechanical Engineering, KU Leuven, Belgium, e-mail:* <firstname>.<lastname>@mech.kuleuven.be

Abstract: Optimal *path following* for robots considers the problem of moving along a predetermined Cartesian geometric end effector path (which is transformed into a predetermined geometric joint path), while some objective is minimized: e.g. motion time or energy loss. In practice it is often not required to follow a path exactly but only within a certain tolerance. By deviating from the path, within the allowable tolerance, one could gain in optimality. In this paper, we define the allowable deviation from the path as a tube around the given geometric path. We then search for the optimal motion inside the tube. This transforms the path following problem to a *tube following* problem. In contrast to the (time or energy) optimal path following problem, the tube following problem is not convex. However, we propose a problem formulation that can still be solved efficiently, as will be illustrated by some numerical examples.

1. INTRODUCTION

Robot path following problems determine the motion of a robot along a predetermined geometric Cartesian end effector path without any preassigned timing information. Common practice is to transform the Cartesian path into a joint path using the inverse kinematics. Path following is often considered to be the low level stage in a decoupled motion planning approach (Bobrow et al., 1985; Shin and Mckay, 1985; Van Loock et al., 2013a), since the motion planning problem (path planning and following) is difficult and highly complex to solve in its entirety (von Stryk and Bulirsch, 1992; Diehl et al., 2005). First, a high level path planner determines a geometric path, ignoring the system dynamics but taking into account geometric path constraints. Second, an optimal trajectory along the geometric path is determined that takes the system dynamics and limitations into account. Since the dynamics along a geometric path can be described by a scalar path coordinate s and its time derivatives (Bobrow et al., 1985; Shin and Mckay, 1985; Van Loock et al., 2013a), the decoupled approach simplifies the motion planning problem to great extent. Furthermore, the path following problem in joint space for a robotic manipulator with simplified constraints can be cast as a *convex* optimization problem (Verscheure et al., 2009a,b). This guarantees efficient computation of globally optimal solutions.

In many applications, the Cartesian geometric end effector path planned by the path planner does not need to be followed exactly but within certain position and orientation tolerances. Typical examples are milling robots where some geometrical tolerance on the workpiece is given. By deviating from the predetermined path, within the allowable tolerance one could gain in optimality.

In (Van Loock et al., 2013b) an optimal path following formulation is presented that provides freedom on the joint path. This freedom on the joint path then results in freedom on the Cartesian path. However, for the applications considered in this paper, where the tolerances are specified in Cartesian space, this formulation should be extended because Cartesian tolerances cannot be transformed back into joint tolerances for robotic manipulators in general.

This paper presents a method that combines freedom on the joint paths, as proposed in (Van Loock et al., 2013b), with constraints on the end-effector Cartesian position and orientation which correspond to the given end effector tolerances. The Cartesian position constraints translate into a tube around the given geometric Cartesian path, to which the end effector is bounded. The freedom on the joint paths is taken sufficiently large such that it is not the restricting factor in the optimization.

The resulting *tube following* problem is nonconvex. In this paper we propose a problem formulation, starting from the path following formulation, that can still be solved efficiently using a standard interior point solver.

This paper is organised as follows. Section 2 reviews the path following problem formulation given in (Verscheure et al., 2009a). Then, Section 3 extends this path following problem to a tube following problem. Here we review the joint path parametrisation given in (Van Loock et al., 2013b) and we define tube constraints on the end effector position and constraints on the orientation of the end effector. Section 4 illustrates the proposed framework with some numerical examples of time-optimal and energy-optimal tube following respectively.

⋆ This work benefits from K.U.Leuven-BOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC), the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO), the European research project EMBOCON FP7-ICT-2009-4 248940, project G.0377.09 of the Research Foundation Flanders (FWO Vlaanderen), and K.U.Leuvens Concerted Research Action GOA/10/11. Goele Pipeleers is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen).

Throughout the paper we will use the following shorthand notations for the derivatives of a function $f(s(t))$: $\dot{f} = \frac{df}{dt}, \ddot{f} = \frac{d^2f}{dt^2}, f' = \frac{\partial f}{\partial s}, f'' = \frac{\partial^2 s}{\partial s^2}$ where t indicates time and s the path coordinate. Furthermore, we indicate scalars with a lower-case letter, e.g. n, vectors with a bold lower-case letter, e.g. $\boldsymbol{q}$, and matrices with an upper-case letter, e.g. M. $\boldsymbol{q}_i$ denotes the i-th element of $\boldsymbol{q}$.

2. OPTIMAL PATH FOLLOWING PROBLEM FORMULATION

Consider a robotic manipulator with n degrees of freedom and joint angles $\boldsymbol{q} \in \mathbb{R}^n$. The equations of motion are given by

$$\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}), \qquad (1)$$

where $\boldsymbol{\tau} \in \mathbb{R}^n$ are the joint torques, $M \in \mathbb{R}^{n\times n}$ is the mass matrix, $C \in \mathbb{R}^{n\times n}$ is a matrix, linear in $\dot{\boldsymbol{q}}$, accounting for Coriolis and centrifugal effects and, $\boldsymbol{g}$ is a vector accounting for gravity and other position dependent torques.

Consider a prescribed geometric path $\boldsymbol{q}(s)$ as a function of a scalar path coordinate s, given in joint space coordinates. The time dependence of the path is determined through $s(t)$. Without loss of generality it is assumed that the trajectory starts at $t = 0$, ends at $t = T$ and, $0 = s(0) \leq s(t) \leq s(T) = 1$. It is furthermore assumed that we always move forward along the path, i.e. $\dot{s}(t) \geq 0, \forall t \in [0, T]$.

Using the chain-rule we rewrite joint velocities and accelerations as

$$\dot{\boldsymbol{q}}(s) = \boldsymbol{q}'(s)\dot{s} \text{ and, } \ddot{\boldsymbol{q}}(s) = \boldsymbol{q}''(s)\dot{s}^2 + \boldsymbol{q}'(s)\ddot{s}.$$

Substitution of the above equations in (1) projects the equations of motion onto the path (Verscheure et al., 2009a):

$$\boldsymbol{\tau}(s(t)) = \boldsymbol{\psi}_s\left(s(t), \dot{s}(t)^2, \ddot{s}(t), \boldsymbol{q}(s(t)), \boldsymbol{q}'(s(t)), \boldsymbol{q}''(s(t))\right).$$

Now, by using the same transformation of variables as in (Verscheure et al., 2009a; Van Loock et al., 2013a) we transform the problem from a time t dependent problem into a path s dependent problem where we use s as an independent variable instead of time t.

$$\dot{s}^2 = b(s), \text{ where } \ddot{s} = \frac{1}{2}b'(s).$$

This results in the following dynamics

$$\boldsymbol{\tau}(s) = \boldsymbol{\psi}_b\left(s, b(s), b'(s), \boldsymbol{q}(s), \boldsymbol{q}'(s), \boldsymbol{q}''(s)\right).$$

2.1 Time-optimal path following

The total motion time is given by

$$T = \int_0^T 1dt = \int_0^1 \frac{1}{\dot{s}}ds = \int_0^1 \frac{1}{\sqrt{b(s)}}ds.$$

The time-optimal path following problem is then formulated as

$$\begin{aligned} \underset{b(\cdot),\boldsymbol{\tau}(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}}ds \\ \text{subject to} \quad & b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2, b(s) \geq 0 \\ & \boldsymbol{\tau}(s) = \boldsymbol{\psi}_b\left(s, b(s), b'(s), \boldsymbol{q}(s), \boldsymbol{q}'(s), \boldsymbol{q}''(s)\right) \\ & \boldsymbol{\tau}_- \leq \boldsymbol{\tau}(s) \leq \boldsymbol{\tau}_+ \\ & \text{for } s \in [0, 1]. \end{aligned} \qquad (2)$$

Once the optimal solution for $b(\cdot), \boldsymbol{\tau}(\cdot)$ is obtained, the relation between path coordinate and time can be obtained from the relation

$$t(s) = \int_0^s \frac{1}{\sqrt{b(\sigma)}}d\sigma.$$

Note that optimization problem (2) is a fixed end-time problem due to the transformation from time domain t to path domain s. In general this is much easier to solve than a free end-time problem due to the strongly non-linear dependence of the solution with varying end-times.

This time-optimal path following problem (hence fixed $\boldsymbol{q}(s)$) is *convex* for a simplified robot and simple task constraints (Verscheure et al., 2009a; Debrouwere et al., 2012). In the extension to tube following we will allow deviations from the fixed joint path, hence $\boldsymbol{q}(s)$ is free as in (Van Loock et al., 2013b) and the optimization problem is *nonconvex*. The proposed problem formulation, given in the following section, results in an numerical optimization problem which can be solved efficiently using standard nonconvex solvers.

2.2 Energy-optimal path following

In (Verscheure et al., 2009a) a trade-off is made between thermal energy losses and motion time. Hence energy loss minimization results in larger motion times. The thermal losses are dominated by the electrical resistive energy loss for each joint i which is proportional to integral of the square of the joint motor torque. The total thermal losses are then proportional to:

$$\sum_{i=1}^n \int_0^T \frac{\boldsymbol{\tau}_i^2}{\boldsymbol{\tau}_{+,i}^2}dt = \sum_{i=1}^n \int_0^1 \frac{\boldsymbol{\tau}_i^2}{\boldsymbol{\tau}_{+,i}^2\sqrt{b(s)}}ds, \qquad (3)$$

where $\boldsymbol{\tau}_{+,i}^2$ is used as a normalization factor.

It can be shown (Boyd and Vandenberghe, 2004) that $x^2/\sqrt{y}$ is a convex function of (x, y), for $y \geq 0$, hence the electrical energy losses are convex.

Another approach could be to only minimize the energy losses (3) while constraining the motion time to some maximal value T_m. The energy-optimal path following problem is similar to the time-optimal path following problem (2), however it has a different objective function (3), and an additional constraint $\int_0^1 b(s)^{-1/2}ds \leq T_m$. This energy-optimal approach extends easily to tube following. The extra freedom (deviation from the Cartesian path within the tube) allows the robot to deviate from the nominal path to minimize the energy losses while preserving the motion time T_0^* of the time-optimal path following problem. Hence $T_m = T_0^*$.

The following section extends the time-optimal path following problem to time-optimal tube following problems. The energy-optimal tube following problem derivation is straightforward.

3. FROM PATH FOLLOWING TO TUBE FOLLOWING

Generally a robot task is specified in Cartesian coordinates of the end effector pose $\boldsymbol{y}(s) = (x, y, z, \phi, \theta, \psi)^T =$

$(\boldsymbol{x}(s)^T, \boldsymbol{\phi}(s)^T)^T$, since this is the practical execution space of the robot, while the joint space coordinates $\boldsymbol{q}(s)$ are used to control the robot. Here $\boldsymbol{x} = (x, y, z)^T$ represents the Cartesian position of the end effector in the world frame, and $\boldsymbol{\phi} = (\phi, \theta, \psi)^T$ represents its roll, pitch and yaw angles (Spong, 1989). The relation between the end effector coordinates and joint coordinates is given by the *forward kinematics*

$$\boldsymbol{y}(s) = \boldsymbol{\chi}(\boldsymbol{q}(s)).$$

We can also define forward position and orientation kinematics as

$$\boldsymbol{x}(s) = \boldsymbol{\chi}_x(\boldsymbol{q}(s)) \text{ and } \boldsymbol{\phi}(s) = \boldsymbol{\chi}_\phi(\boldsymbol{q}(s)).$$

In a path following task a fixed geometric path for $\boldsymbol{q}(s)$ is determined from the desired Cartesian path $\boldsymbol{y}(s)$ by using the inverse relation of $\boldsymbol{\chi}(\cdot)$ (inverse kinematics). In a tube following task, the Cartesian path should lie inside a tube around some predetermined Cartesian path while deviations of the orientation from the desired orientation are constrained as well. Hence, we search for a new $\boldsymbol{q}(s)$ such that $\boldsymbol{y}(s)$ lies inside that tube and obeys the orientation constraints. Hence $\boldsymbol{q}(s)$ is a free optimization variable that has to obey constraints on $\boldsymbol{x}(s)$ and $\boldsymbol{\phi}(s)$: $\boldsymbol{x}(s) = \boldsymbol{\chi}_x(\boldsymbol{q}(s)) \in \mathcal{C}_x$ and $\boldsymbol{\phi}(s) = \boldsymbol{\chi}_\phi(\boldsymbol{q}(s)) \in \mathcal{C}_\phi$ for $s \in [0, 1]$, and hence constraints on $\boldsymbol{y}(s) = (\boldsymbol{x}(s)^T, \boldsymbol{\phi}(s)^T)^T$:

$$\boldsymbol{y}(s) = \boldsymbol{\chi}(\boldsymbol{q}(s)) \in \mathcal{C} = \mathcal{C}_x \times \mathcal{C}_\phi \text{ for } s \in [0, 1], \quad (4)$$

Since $\boldsymbol{\chi}(\cdot)$ is an analytical relation between joint space and Cartesian space it can easily be included in the tube following optimization problem:

$$\begin{aligned} \underset{b(\cdot),\boldsymbol{q}(\cdot),\boldsymbol{\tau}(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\ \text{subject to} \quad & b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2, b(s) \geq 0 \\ & \boldsymbol{\tau}(s) = \boldsymbol{\psi}_b\left(s, b(s), b'(s), \boldsymbol{q}(s), \boldsymbol{q}'(s), \boldsymbol{q}''(s)\right) \\ & \boldsymbol{\tau}_- \leq \boldsymbol{\tau}(s) \leq \boldsymbol{\tau}_+ \\ & \boldsymbol{\chi}(\boldsymbol{q}(s)) \in \mathcal{C} \\ & \text{for } s \in [0, 1]. \end{aligned} \quad (5)$$

We now adopt the joint parametrisations from (Van Loock et al., 2013b) to allow freedom in the joint path $\boldsymbol{q}(s)$ while we add constraints $\mathcal{C}$ on the Cartesian path. In the following Subsections we review this joint path formulation and derive tube and orientation constraints on the end effector Cartesian path that compose $\mathcal{C}$.

Since the resulting time-optimal tube following problem has an infinite number of optimization variables and constraints, it is discretized by adopting the direct transcription method from (Verscheure et al., 2009a) which presents a parametrisation for $b(s)$ and $\boldsymbol{\tau}(s)$. A discretized s-grid is assumed where we have s^k for $k = 0..K$ and $s^{k+1} = s^k + \Delta s^k$ and $s^{k+\frac{1}{2}} = s^k + \frac{1}{2}\Delta s^k$.

3.1 Parametrisation of $\boldsymbol{q}(s)$

The path $\boldsymbol{q}_i(s)$ of joint i is defined as a convex combination of predetermined upper and lower bounds, $\underline{\boldsymbol{q}}_i(s)$ and $\overline{\boldsymbol{q}}_i(s)$ respectively, for $\boldsymbol{q}_i(s)$ as in (Van Loock et al., 2013b). Here $\underline{\boldsymbol{q}}_i(s)$ and $\overline{\boldsymbol{q}}_i(s)$ need to be twice differentiable.

$$\boldsymbol{q}_i(s) = \boldsymbol{p}_i(s)\underline{\boldsymbol{q}}_i(s) + (1 - \boldsymbol{p}_i(s))\overline{\boldsymbol{q}}_i(s),$$

where $0 \leq \boldsymbol{p}_i(s) \leq 1 \quad \forall s \in [0, 1]$ and $\boldsymbol{q}'(s) = \boldsymbol{\alpha}(s, \boldsymbol{p}(s), \boldsymbol{p}'(s))$ and $\boldsymbol{q}''(s) = \boldsymbol{\beta}(s, \boldsymbol{p}(s), \boldsymbol{p}'(s), \boldsymbol{p}''(s))$ can be computed using the chain rule. By choosing the paths $\underline{\boldsymbol{q}}_i(s)$ and $\overline{\boldsymbol{q}}_i(s)$ the torque $\boldsymbol{\tau}(s)$ and end effector pose $\boldsymbol{y}(s)$ only depend on $\boldsymbol{p}(s)$, allowing us to write the tube following problem as:

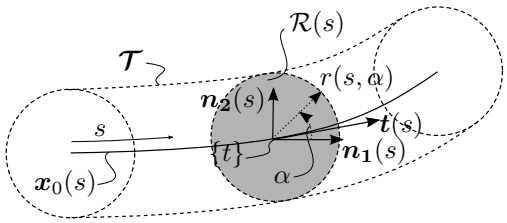

Fig. 1. Tube $\boldsymbol{\mathcal{T}}$ around a central path $\boldsymbol{x}_0(s)$

$$\begin{aligned} \underset{b(\cdot),\boldsymbol{p}(\cdot),\boldsymbol{\tau}(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\ \text{subject to} \quad & b(0) = \dot{s}_0^2, b(1) = \dot{s}_T^2, b(s) \geq 0 \\ & \boldsymbol{\tau}(s) = \boldsymbol{\psi}_p\left(s, b(s), b'(s), \boldsymbol{p}(s), \boldsymbol{p}'(s), \boldsymbol{p}''(s)\right) \\ & 0 \leq \boldsymbol{p}_i(s) \leq 1 \\ & \boldsymbol{\tau}_- \leq \boldsymbol{\tau}(s) \leq \boldsymbol{\tau}_+ \\ & \boldsymbol{\chi}(\boldsymbol{q}(\boldsymbol{p}(s))) \in \mathcal{C} \\ & \text{for } s \in [0, 1]. \end{aligned}$$

The optimization now only needs to determine a convex combination of already predefined bounds $\underline{\boldsymbol{q}}_i(s)$ and $\overline{\boldsymbol{q}}_i(s)$. The choice of $\underline{\boldsymbol{q}}_i(s)$ and $\overline{\boldsymbol{q}}_i(s)$ and the parametrisation for $\boldsymbol{p}_i(s)$ are design parameters. As mentioned before, the bounds $\underline{\boldsymbol{q}}_i(s)$ and $\overline{\boldsymbol{q}}_i(s)$ should be chosen sufficiently large, such that the end effector tube constraint is the restricting constraints and not the joint bounds. To allow for varying $\boldsymbol{p}(s)$ along the path, we adopt the polynomial spline parametrisation, with *B-spline* basis, of degree l with g internal knots presented in (Van Loock et al., 2011).

$$\boldsymbol{p}_i(s) = \sum_{j=-l}^{g} c_{i,j} B_{j,l+1}(s),$$

where the coefficients $c_{i,j}$ are now the optimization variables and $B_{j,l+1}(s)$ is the spline basis. $\boldsymbol{p}(s)$ depends linearly on the optimization variables. The resulting optimization problem can still be solved efficiently for reasons mentioned above and as indicated in Van Loock et al. (2013b).

3.2 Tube (Position) Constraints

The actual end effector path $\boldsymbol{x}(s) = (x(s), y(s), z(s))^T$ should lie within a tube around the central Cartesian path $\boldsymbol{x}_0(s) = (x_0(s), y_0(s), z_0(s))^T$ with radius $r(s)$ which represents the allowable deviation from the path. This tube surface $\boldsymbol{\mathcal{T}}$ is described by the following parametric equation:

$$\boldsymbol{\mathcal{T}}(s, \alpha) = \boldsymbol{x}_0(s) + r(s, \alpha)\left(cos(\alpha)\boldsymbol{n_1}(s) + sin(\alpha)\boldsymbol{n_2}(s)\right), \quad \text{for } \alpha \in [0, 2\pi] \text{ and } s \in [0, 1], \quad (6)$$

with $\boldsymbol{n_1}(s) = (\boldsymbol{t}(s) \times \boldsymbol{v}) / \left\|\boldsymbol{t}(s) \times \boldsymbol{v}\right\|_2$ the normal vector and $\boldsymbol{n_2}(s) = (\boldsymbol{t}(s) \times \boldsymbol{n_1}(s)) / \left\|\boldsymbol{t}(s) \times \boldsymbol{n_1}(s)\right\|_2$ the binormal vector to the tangent $\boldsymbol{t}(s)$ of the central path $\boldsymbol{x}_0(s)$ at s (where $\boldsymbol{v}$ is an arbitrary vector that determines the direction of $\boldsymbol{n_1}$). α represents the angle around the tangent $\boldsymbol{t}$ of the curve $\boldsymbol{x}_0(s)$ at s, starting from the normal vector $\boldsymbol{n_1}$. $r(s, \alpha)$ represents the radius of the tube as a function

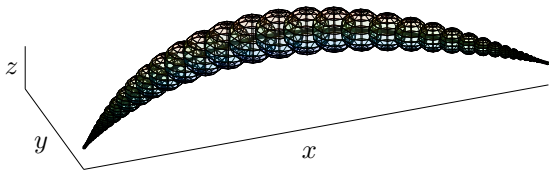

Fig. 2. Illustration of the quadratic tube approximation where $r(s)$ is a parabola where $r(0) = r(1) = 0$.

of s and α. In the following we assume r to be independent from α, hence, we assume a constant radius $r(s)$ at every s. Figure 1 clarifies the above definitions with an illustration.

We approximate[1] the tube $\boldsymbol{\mathcal{T}}(s^k, \alpha)$ at every s^k as a sphere $\boldsymbol{\mathcal{B}}^k$ with radius $r(s^k)$. This can be seen in Figure 2. Hence the constraint that $\boldsymbol{x}(s^k)$ should lie inside the tube for all s^k can be approximated to $\boldsymbol{x}(s^k)$ lying in $\boldsymbol{\mathcal{B}}^k$. This results in the following quadratic constraint of the form (4) where

$$\mathcal{C}_x = \{\boldsymbol{x}(s^k) | \boldsymbol{x}(0) = \boldsymbol{x}_0(0), \boldsymbol{x}(1) = \boldsymbol{x}_0(1), \\ \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right)^T \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right) \leq r(s^k)^2\}.$$

Where the first two terms request that the new Cartesian path should start and end at the start and end of the original path.

Extension to ellipsoidal tube approximation: We approximate the tube at every s^k as an ellipsoid. This results in a constraint of the form (4) where

$$\mathcal{C}_x = \\ \{\boldsymbol{x}(s^k) | \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right)^T E^k(s^k) \left(\boldsymbol{x}(s^k) - \boldsymbol{x}_0(s^k)\right) \leq 1\},$$

where $E^k(s^k)$ is a symmetric positive definite matrix. The eigenvectors of E^k define the principal directions of the ellipsoid while the inverse of the square root of the eigenvalues of E^k are the corresponding equatorial radii. Following the tube definition of (6) the eigenvectors of E^k should be the tangent vector $\boldsymbol{t}$ and normal vectors $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ to the central path $\boldsymbol{x}_0(s^k)$ with radii $r_{\boldsymbol{t}}$, $r_{\boldsymbol{n_1}}$ and $r_{\boldsymbol{n_2}}$. Here $r_{\boldsymbol{n_1}}$ and $r_{\boldsymbol{n_2}}$ define an ellipsoidal cross section $r(s^k, \alpha)$ while $r_{\boldsymbol{t}}$ represents the radius along $\boldsymbol{t}$.

Approximation error and grid size: When there is high curvature of $r(s^k)$, the sphere $\boldsymbol{\mathcal{B}}^k$ will not represent the tube surface well at e.g. $s^{k-\frac{1}{2}}$. This high approximation error could be reduced by placing the grid points closely together: $\Delta s^k \sim \left(\frac{d^2 r(s)}{ds^2}\Big|_{s^k} \right)^{-1}$ or by using an ellipsoidal approximation instead of a spherical. Then the tangential radius $r_{\boldsymbol{t}}$ can be chosen to be bigger when there is small curvature or smaller when there is larger curvature at s^k, to obtain a better approximation of the tube surface. This results in the following guideline: $r_{\boldsymbol{t}} \sim \left(\frac{d^2 r(s)}{ds^2}\Big|_{s^k} \right)^{-1}$.

Furthermore, since the tube at s^{k-1} and s^k is approximated by spheres with radius $r(s^{k-1})$ and $r(s^k)$, the grid points should be placed closely if the radius $r(s)$ is small to obtain a sufficient approximation of the interior of the tube: $\Delta s^k \sim r(s^k)$.

[1] Implementing the tube constraints (6) at each s^k imposes that $\boldsymbol{x}(s^k)$ should lie in the disc $\mathcal{R}(s^k)$ (see Figure 1), which would be to restrictive, since only a limited amount of parametrisations $\boldsymbol{q}(s)$ validate these constraints. Hence we extend the disc to a volume $\boldsymbol{\mathcal{B}}^k$.

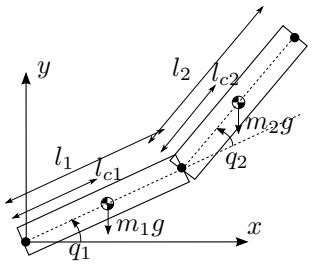

Fig. 3. The planar robot used in the examples.

Finally can be seen that for a coarser s-grid, the overlapping of the spheres can be insufficient to represent the interior of the tube. Hence, we could request $\Delta s^k < \frac{1}{2}\left(r(s^k) + r(s^{k+1})\right)$.

3.3 Orientation constraints

In certain applications the deviation of the orientation $\boldsymbol{\phi} = (\phi, \theta, \psi)^T$ along the new path in the tube with respect to the orientation $\boldsymbol{\phi}_0 = (\phi_0, \theta_0, \psi_0)^T$ along the original path should be constrained. This can be the case for a laser cutting task where deviation on the position is allowed within some tolerance, but where no deviation is allowed on the orientation of the laser.

Practical orientation constraints can be defined on the roll pitch and yaw angles of the frame $\{t\}$ with the axis $\boldsymbol{t}, \boldsymbol{n_1}, \boldsymbol{n_2}$ (see Figure 1). The roll angle ϕ is defined around $\boldsymbol{t}$, the pitch θ around $\boldsymbol{n_1}$ and the yaw ψ around $\boldsymbol{n_2}$. Constraints on the orientation angles can then be written as quadratic constraints of the form (4) where

$$\mathcal{C}_\phi = \{\boldsymbol{\phi} | (\phi - \phi_0)^2 \leq \epsilon_\phi^2, (\theta - \theta_0)^2 \leq \epsilon_\theta^2, (\psi - \psi_0)^2 \leq \epsilon_\psi^2, \\ \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0(0), \boldsymbol{\phi}(1) = \boldsymbol{\phi}_0(1)\}.$$

Where the last two terms request that the new orientation should start and end with the original orientation.

4. NUMERICAL EXAMPLES

To illustrate the efficiency of our approach, we consider two examples. For clarity of the graphs, we consider a 2 dof planar robotic manipulator in the vertical plane, which can be seen in Figure 3. The y axis is the vertical axis while the x axis is the horizontal axis. The system parameters we consider are the following: $m_1 = 1$ kg, $m_2 = 1$ kg, $l_1 = 1$ m, $l_2 = 1$ m, $l_{c1} = 0.5$ m, $l_{c2} = 0.5$ m, $\boldsymbol{\tau}_- = -(30, 15)^T$ Nm, $\boldsymbol{\tau}_+ = (30, 15)^T$ Nm and moments of inertia $I_1 = I_2 = 0.5$ kg/m^2. We do not consider orientation constraints.

The tube following problem (5) is implemented in Python using CasADi (Andersson et al., 2012) as modelling software and Ipopt (Wächter and Biegler, 2006) as non-linear solver. The problems are discretized with $K = 100$ and are solved on an Ubuntu Virtual Box with 1Gb RAM, running on a 2.4GHz Windows laptop.

4.1 Linear motion

We consider a line in Cartesian space, going from $(1.5, 0)$m to $(0, 1.5)$m, as the nominal path $\boldsymbol{x}_0(s)$. Using the inverse kinematics and a polynomial fit we get a nominal joint path $\boldsymbol{q}_0(s)$. The tube is chosen to be a tube with constant radius R: $r(s) = R$ m. Furthermore, we choose two bounds $\underline{\boldsymbol{q}}(s) = 0.5\boldsymbol{q}_0(s)$ and $\overline{\boldsymbol{q}}(s) = 2\boldsymbol{q}_0(s)$.

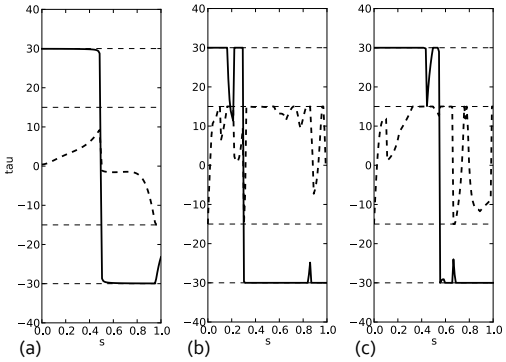

Fig. 4. Torques of joint 1 (full line) and joint 2 (dashed line) as a function of s: (a) path following, (b) joint freedom, (c) tube following.

Time-Optimal Path Following We first consider the time-optimal path following problem. Figure 4.a shows the torques as a function of s for both joints. We can see that it is a time-optimal motion since at any s, there is always one torque constraint active. Figure 5.a shows the robot moving along the given Cartesian path.

Time-Optimal with Joint Freedom Secondly we consider the problem were we allow freedom on the joints according to 3.1, where $\boldsymbol{p}(s)$ is a polynomial spline with $g = 10$ knots of degree $l = 2$, but without tube constraints. The spline coefficients of $\boldsymbol{p}(s)$ are initialized such that the initial joint path equals the nominal joint path $\boldsymbol{q}_0$ such that the optimization is initialized inside the tube. The result of the optimization is shown in Figure 5.b. We can see that the robot reduces its inertia by pulling back the second link at the beginning of the motion by using the joint freedom. Reducing this inertia allows for faster motion as can be seen in Table 1.

Time-Optimal Tube Following Finally, we consider the tube following problem with $R = 0.2$ m, where $\boldsymbol{p}(s)$ is a polynomial spline with $g = 10$ knots of degree $l = 2$. Figure 4.c shows the torques as a function of s for both joints. Due to the different Cartesian path, the torques are different than in the path following case. Figure 5.c shows the robot moving along the optimized Cartesian path within the tube. We can see that the robot again reduces its inertia by pulling back the second link, however the end effector needs to stay inside the tube. To reach the end position as fast as possible the robot extracts it second link again towards the end, again, hitting the tube constraint.

Table 1 compares the computation and motion time for the three time-optimal linear motion cases. The tube following problem is 12% faster than the path following problem. Furthermore, the computation time is only 3 times higher than for the convex time-optimal path following problem.

4.2 Circular motion

Time-Optimal Tube Following In a second example, we consider a circular counter clockwise Cartesian path with radius 1.5 m, for which we solve a tube following problem for a finite set of tube radii $R \in [0, 0.2]$ m. The robot base is located at the center of the circular path. We consider

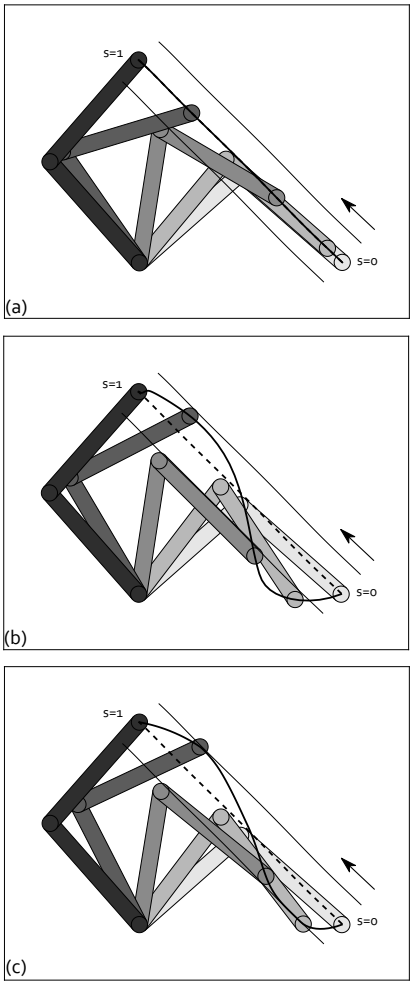

Fig. 5. Illustration of the robot (given at equidistant time instances in grey shades going from light to dark according to time from zero to the end), robot nominal path $\boldsymbol{x}_0$ (dashed line), tube edges where $R = 0.2$ m (thin full lines) and optimized path $\boldsymbol{x}$ (thick full line): (a) path following, (b) joint freedom, (c) tube following.

	Computation time	Motion time
Path Following	0.32s [1]	1.017s [1]
Joint Freedom	0.57s [1.78]	0.858s [0.844]
Tube Following	0.98s [3.06]	0.898s [0.883]

Table 1. Comparison of the computation time [normalized] and motion time [normalized] for the three linear motion cases described in Section 4.1.

$\boldsymbol{p}(s)$ as a polynomial spline with $g = 30$. Figure 6.a shows the robot for tube following problem with radius $R = 0.2$ m. The robot extends the second link to use the gravity to move faster in the descending part. Figure 7 shows the normalized motion time (tube following motion time divided by path following motion time) as a function of the tube radius R (full line). Allowing a small deviation from the nominal Cartesian path $\boldsymbol{x}_0$ already yields a significant reduction of the execution time (e.g. for this example $R = 0.01$ m results in a motion time reduction by

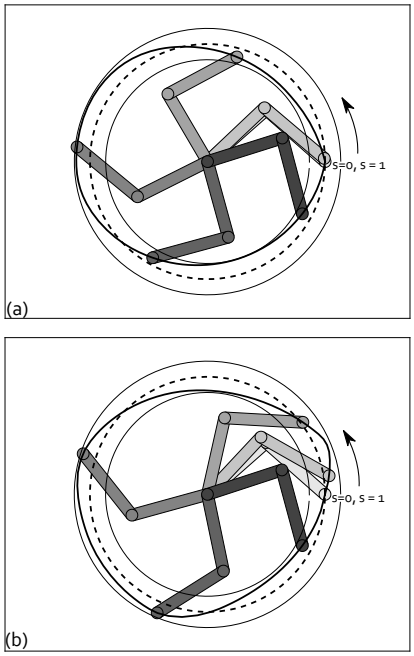

Fig. 6. Illustration of the robot (given at equidistant time instances in grey shades going from light to dark according to time from zero to the end), robot nominal path $\boldsymbol{x}_0$ (dashed line), tube edges where $R = 0.2$ m (thin full lines) and optimized path $\boldsymbol{x}$ (thick full line): (a) time-optimal tube following, (b) energy-optimal tube following.

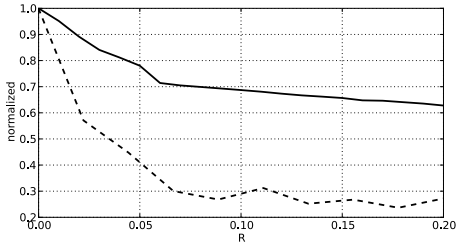

Fig. 7. Normalized (tube following case divided by path following case) trade-off between tube radius R and motion time (full line) and energy losses for $T_m = T_0^*$ (dashed line).

$\sim 5\%$ while $R = 0.1$ m results in a motion time reduction by $\sim 30\%$.

Energy-Optimal Tube Following We apply the energy-optimal tube following described in Section 2.2 to the circular path for a finite set of tube radii $R \in [0, 0.2]$ m. Figure 6.b shows the robot moving along the energy-optimal path for $R = 0.2$ m. The robot keeps the end effector as low as possible in the ascending part of the motion (acting against gravity as little as possible) while it extends the second link to use the gravity to consume less energy in the descending part. Figure 7 shows the normalized energy losses (tube following energy losses divided by path following energy losses) as a function of the tube radius R (dashed line). Allowing a small deviation from the nominal Cartesian path $\boldsymbol{x}_0$ already yields a significant reduction of the energy losses: e.g. for this example $R = 0.01$ m results in a energy loss reduction by $\sim 20\%$ while the motion time is the same as the time-optimal path following motion time. For $R = 0.1$ m the energy loss is reduced by $\sim 70\%$. For all radii the resulting motion time T^* does not exceed the time-optimal path following motion time T_0.

REFERENCES

Andersson, J., Åkesson, J., and Diehl, M. (2012). CasADi – A symbolic package for automatic differentiation and optimal control. In S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther (eds.), *Recent Advances in Algorithmic Differentiation*, Lecture Notes in Computational Science and Engineering. Springer, Berlin.

Bobrow, J., Dubowsky, S., and Gibson, J. (1985). Time-optimal control of robotic manipulators along specified paths. *The International Journal of Robotics Research*, 4(3), 3–17. doi:10.1177/027836498500400301.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Debrouwere, F., Van Loock, W., Pipeleers, G., Diehl, M., Swevers, J., and De Schutter, J. (2012). Time-optimal robot path following with cartesian acceleration constraints: a convex optimization approach. In *Mechatronics Forum, LINZ, Austria*, 469–475.

Diehl, M., Bock, H.G., Diedam, H., and Wieber, P.B. (2005). Fast Direct Multiple Shooting Algorithms for Optimal Robot Control. In *Fast Motions in Biomechanics and Robotics*. Heidelberg, Allemagne.

Shin, K. and Mckay, N. (1985). Minimum-time control of robotic manipulators with geometric path constraints. *Automatic Control, IEEE Transactions on*, 30(6), 531–541.

Spong, M.W. (1989). *Robot Dynamics and Control*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

Van Loock, W., Bellens, S., Pipeleers, G., De Schutter, J., and Swevers, J. (2013a). Time-optimal parking and flying: Solving path following problems efficiently. In *IEEE International Conference on Mechatronics, Vicenza, 27-28 February, 1 March*.

Van Loock, W., Pipeleers, G., and Swevers, J. (2011). Optimal input design for flat systems using b-splines. In *American Control Conference (ACC), 2011*, 3281–3282.

Van Loock, W., Pipeleers, G., and Swevers, J. (2013b). Time-optimal quadrotor flight. In *European Control Conference, Zurich*, 1788–1792.

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. doi:10.1109/TAC.2009.2028959.

Verscheure, D., Diehl, M., De Schutter, J., and Swevers, J. (2009b). On-line time-optimal path tracking for robots. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, 610–616. IEEE Press, Piscataway, NJ, USA.

von Stryk, O. and Bulirsch, R. (1992). Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37, 357–373.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106, 25–57. doi:10.1007/s10107-004-0559-y.