KU LEUVEN

Citation	Debrouwere, F., Van Loock, W., Pipeleers, G., Diehl, M., De Schutter, J., Swevers, J. (2013) **Time-Optimal Path Following for Robots with Object Collision Avoidance using Lagrangian Duality** Proceedings on the International Workshop on Robot Motion and Control. International Workshop on Robot Motion and Control. WAsowo, Poland, 3-5 July 2013 (pp. 186-191)
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	http://dx.doi.org/10.1109/RoMoCo.2013.6614606
Publisher homepage	
Author contact	E-mail: frederik.debrouwere@kuleuven.be Phone number: *+32 16 32 92 64*
IR	https://lirias.kuleuven.be/handle/123456789/398357

(article begins on next page)

Time-Optimal Path Following for Robots with Object Collision Avoidance using Lagrangian Duality

Frederik Debrouwere[1], Wannes Van Loock[1], Goele Pipeleers[1], Moritz Diehl[2], Joris De Schutter[1] and Jan Swevers[1]

Abstract— **Time-optimal path following considers the problem of moving along a predetermined geometric path in minimum time while respecting system constraints. This paper focusses on time-optimal path following problems in robotics where collision must be avoided with other robots or moving obstacles. The developed method is based on the convex reformulation of the time-optimal path following problem with simplified dynamics presented in [1]. The robot and the obstacles are modelled as unions of convex polyhedra and the collision avoidance constraints are derived using Lagrangian duality. These constraints render the optimization problem non-convex. However, numerical simulations show that the resulting non-convex optimization problem can still be solved efficiently using a non-linear solver, due to the time-optimal path following formulation [1] and the proposed formulation of the collision avoidance constraints.**

I. Introduction

Path following deals with the problem of following a geometric path without any preassigned timing information. Many industrial robot tasks, such as welding, glueing, laser cutting and milling can be cast as path following problems. In addition, path following is often considered to be the low level stage in a decoupled motion planning approach [2], [3]. First, a high level 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 accounts for the system dynamics and limitations. Since the dynamics along a geometric path can be described by a scalar path coordinate s_r and its time derivatives [2], [3], the decoupled approach simplifies the motion planning problem to great extent. Recently, it was shown that the path following problem for a robotic manipulator with simplified constraints can be cast as a *convex* optimization problem [1] by means of a non-linear transformation of optimization variables. This guarantees the efficient computation of globally optimal solutions.

The convexity of the resulting optimization problem only holds for simplified robot dynamics in free space. However, in many applications, robots are interacting with moving obstacles such as humans, automated guided vehicles (AGV's), pallets on a conveyor or other robots. This work focusses on the derivation of suitable constraints to avoid these moving obstacles, while moving time-optimally along the desired trajectory. It is assumed that the path of the object is known and the optimization problem is solved off-line. An extension to uncertain or unknown object trajectories could be made by sensing the object (position and shape), predicting its trajectory and solving the optimization problem on-line.

In this paper, the robots and objects are modelled as unions of convex sets, similar as in [4]. The collision avoidance constraints are written in an explicit formulation by using Lagrange duality [5] resulting in a non-convex optimization problem, which can still be solved efficiently due to the path following formulation [6]. Despite the fact that modelling a complex geometry as a union of convex polyhedrons can result in an excessive number of polyhedrons [7], this approach is used here, since the path following formulation allows for a significant reduction of the polyhedra that have to be considered (see Section IV-B).

Other popular collision avoidance techniques for static objects use algorithms to calculate the minimal distance between the convex sets [8]. This approach uses a good approximation of the real geometry but results in a computational inefficient optimization problem. Some model the objects as super-quadratics, cylinders, etc. and define distances between these objects and points on the robot. The advantage of this approach is that explicit constraints can be formulated. However these constraints are highly non-linear resulting in a hard-to-solve optimization problem. The approach used here combines the benefits of both approaches: a good approximation of the geometry is obtained by modelling it with convex sets and the collision avoidance is formulated efficiently using Lagrange duality.

The paper is organized as follows. Section II reviews the convex reformulation of the time-optimal path following problem of [1]. Section III discusses the modelling of the robot and object geometry and introduces the collision avoidance constraints into the optimization problem. In section IV extensions are presented that profit from the typical solutions of the collision avoidance problem along a path. Finally, the the numerical efficiency of the proposed method is validated in Section IV with some examples. In the remainder of the paper the indices r and o denote the robot and object 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.Leuven's Concerted Research Action GOA/10/11. Goele Pipeleers is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen).

[1]Department of Mechanical Engineering, Division PMA, KU Leuven, BE-3001 Heverlee, Belgium, Frederik.Debrouwere@mech.kuleuven.be

[2]Department of Electrical Engineering, Division SCD, KU Leuven, BE-3001 Heverlee, Belgium

II. Time-optimal path following

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

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

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

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

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

where $\boldsymbol{q}' = d\boldsymbol{q}(s_r)/ds_r$ and, $\boldsymbol{q}'' = d^2\boldsymbol{q}(s_r)/ds_r^2$. Substitution of the above equations in (1) projects the equations of motion onto the path [1]:

$$\boldsymbol{\tau}(s_r) = \boldsymbol{m}(s_r)\ddot{s}_r + \boldsymbol{c}(s_r)\dot{s}_r^2 + \boldsymbol{g}(s_r). \tag{3}$$

By introducing the variable

$$b(s_r) = \dot{s}_r^2, \tag{4}$$

it was shown in [1] that the time-optimal path following problem is transformed into a *convex* optimization problem. Indeed, since $b(s_r) \geq 0$, the total motion time

$$T = \int_0^T 1dt = \int_0^1 \frac{1}{\dot{s}_r} ds_r = \int_0^1 \frac{1}{\sqrt{b(s_r)}} ds_r,$$

is a convex function. Furthermore, since $b'(s_r) = 2\ddot{s}_r$, the torque constraints are linear in $b(s_r)$ and $b'(s_r)$. The time-optimal path following problem is then reformulated as

$$\begin{aligned} \underset{b(\cdot)}{\text{minimize}} \quad & \int_0^1 \frac{1}{\sqrt{b(s_r)}} ds_r \\ \text{subject to} \quad & b(0) = \dot{s}_{r,0}^2,\ b(1) = \dot{s}_{r,T}^2,\ b(s_r) \geq 0 \\ & \boldsymbol{\tau}_- \leq \boldsymbol{m}(s_r)\frac{b'(s_r)}{2} + \boldsymbol{c}(s_r)b(s_r) + \boldsymbol{g}(s_r) \leq \boldsymbol{\tau}_+ \\ & \text{for } s_r \in [0, 1]. \end{aligned} \tag{5}$$

In [1], [9] many constraints are described that maintain the convexity of the problem, such as velocity constraints and acceleration constraints.

III. Collision Avoidance

This section adds collision avoidance constraints to the time-optimal path following problem formulated above.

The robot and the object are modelled as a union of convex polyhedrons. Then, avoiding collision between the robot and the object is equivalent to the constraint that the distance between each of the the convex sets of the robot and the object is larger than some minimal value $\epsilon > 0$. In this section the models for the robot and the objects are derived. Furthermore, it is shown that the minimal distance requirement between the sets can be transformed into two non-convex and three convex constraints at every path coordinate $s \in [0, 1]$ using Lagrange duality [5].

For off-line optimization, it is assumed that the path of the *object* is known in space and that the evolution along that path is given by the path coordinate s_o. Furthermore, it is assumed that the time dependence of the object $s_o(t)$ is also known. Hence the path of the object is fully known as a function of time.

A. Modelling

To express collision avoidance between the robot an the object in a common world frame $\{w\}$ the models should be expressed in this world frame $\{w\}$. However, the robot and the object geometry are modelled as a union of closed convex rigid body polyhedrons (does not change shape), where each polyhedron represents a part k of the geometry defined in its own part frame $\{p_k\}$

$${}_{p_k}A^{[k]}\,{}_{p_k}\boldsymbol{x} \leq {}_{p_k}\boldsymbol{b}^{[k]}. \tag{6}$$

where ${}_{p_k}A^{[k]}$ and ${}_{p_k}\boldsymbol{b}^{[k]}$ do not depend on s since only rigid bodies are considered. Now its representation in the world frame $\{w\}$ needs to be obtained. Suppose the part frame is translated, with vector ${}_w\boldsymbol{x}^{w,p_k}(s)$, and rotated, with rotation matrix ${}^{p_k}_{w}R(s)$, with respect to the world frame. The homogeneous transformation [10] between the coordinates in the world frame ${}_w\boldsymbol{x}$ and the coordinates in the part frame ${}_{p_k}\boldsymbol{x}$ is given by

$${}_{p_k}\boldsymbol{x} = {}^{p_k}_{w}R(s)^T\,{}_w\boldsymbol{x} - {}^{p_k}_{w}R(s)^T\,{}_w\boldsymbol{x}^{w,p_k}(s). \tag{7}$$

Plugging in (7) into the part polyhedron (6) we get the polyhedron in the world frame

$${}_wA^{[k]}(s)\,{}_w\boldsymbol{x} \leq {}_w\boldsymbol{b}^{[k]}(s),$$

where

$$\begin{aligned} {}_wA^{[k]}(s) &= {}_{p_k}A^{[k]}\,{}^{p_k}_{w}R(s)^T, \\ {}_w\boldsymbol{b}^{[k]}(s) &= {}_{p_k}\boldsymbol{b}^{[k]} + {}_wA^{[k]}(s)\,{}_w\boldsymbol{x}^{w,p_k}(s). \end{aligned} \tag{8}$$

The robot is then modelled as the union of convex polyhedra expressed in some world frame $\{w\}$ (indicated by the leading subscript)

$${}_w\mathcal{R} = \bigcup_{i=1}^{n_r} {}_w\mathcal{R}^{[i]}(s_r),$$

where

$${}_w\mathcal{R}^{[i]}(s_r) = \{{}_w\boldsymbol{x} \in \mathbb{R}^3 | {}_wA_r{}^{[i]}(s_r)\,{}_w\boldsymbol{x} \leq {}_w\boldsymbol{b_r}{}^{[i]}(s_r)\}.$$

Here n_r is the number of polyhedra and for $i = 1..n_r$: ${}_wA_r{}^{[i]}(s_r) \in \mathbb{R}^{r_i\times 3}$ and ${}_w\boldsymbol{b_r}{}^{[i]}(s_r) \in \mathbb{R}^{r_i}$ where r_i is the number of sides of the polyhedron $\mathcal{R}^{[i]}$. An example of a robot model is given in Figure 1 where the link polyhedra are modelled as tubes with ten faces.

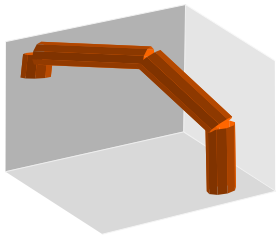


Fig. 1. Seven dof robot modelled as four polyhedra

The object is modelled analogously as the union of convex polyhedra expressed in the same world frame $\{w\}$

$$ {}_w\mathcal{O} = \bigcup_{j=1}^{n_o} {}_w\mathcal{O}^{[j]}(s_o), $$

where

$$ {}_w\mathcal{O}^{[j]}(s_o) = \{ {}_w\boldsymbol{x} \in \mathbb{R}^3 | {}_wA_o{}^{[j]}(s_o) {}_w\boldsymbol{x} \leq {}_w\boldsymbol{b_o}{}^{[j]}(s_o) \}. $$

Here n_o is the number of polyhedra and for $j = 1..n_o$: ${}_wA_o{}^{[j]}(s_o) \in \mathbb{R}^{r_j \times 3}$ and ${}_w\boldsymbol{b_o}{}^{[j]}(s_o) \in \mathbb{R}^{r_j}$ where r_j is the number of sides of the polyhedron $\mathcal{R}^{[j]}$. Extension to multiple objects is straightforward.

B. Constraint derivation via Lagrange duality

This section shows that the collision avoidance constraint between convex sets can be written as a minimization problem for each s_r, hence this minimization problem should be included as a constraint in the time-optimal path following problem. However, by using Lagrange duality this minimization problem constraint can be written as a set of regular equality and inequality constraints resulting in an efficient formulation of the collision avoidance requirement.

Avoiding collision means that the distance between the robot part ${}_w\mathcal{R}^{[i]}$ and the object part ${}_w\mathcal{O}^{[j]}$ has to be bigger than some minimal distance $\epsilon^{[i,j]} > 0$, for all i and j and for all time instants. This time dependence is transformed into an s_r dependence since ${}_w\mathcal{R}(s_r)$ is known if the path of the robot is known. By using the relation between time and the optimization variable $b(s)$

$$ t(s_r, b) = \int_0^{s_r} b(\sigma)^{-\frac{1}{2}} d\sigma, $$

and the fact that $s_o(t)$ is known, avoiding collision at all time instants means that the distance between ${}_w\mathcal{R}^{[i]}(s_r)$ and ${}_w\mathcal{O}^{[j]}(s_o(t(s_r, b)))$ has to be bigger than $\epsilon^{[i,j]}(s_r)$, for all i, j and $s_r \in [0, 1]$. The robot geometry model ${}_w\mathcal{R}^{[i]}(s_r)$ can be precomputed for all s_r. To simplify notation the dependence on s_r and s_o is omitted in the remainder of the section.

The distance between the two convex sets ${}_w\mathcal{R}^{[i]}$ and ${}_w\mathcal{O}^{[j]}$, $\textbf{dist}\left({}_w\mathcal{R}^{[i]}, {}_w\mathcal{O}^{[j]}\right)$, is defined as

$$ \begin{aligned} p^* = \underset{{}_w\boldsymbol{x}^{[i,j]}, {}_w\boldsymbol{y}^{[i,j]}}{\text{argmin}} \quad & \left\| {}_w\boldsymbol{x}^{[i,j]} - {}_w\boldsymbol{y}^{[i,j]} \right\|_2 \\ \text{subject to} \quad & {}_wA_r{}^{[i]} {}_w\boldsymbol{x}^{[i,j]} \leq {}_w\boldsymbol{b_r}{}^{[i]}, \\ & {}_wA_o{}^{[j]} {}_w\boldsymbol{y}^{[i,j]} \leq {}_w\boldsymbol{b_o}{}^{[j]}. \end{aligned} \tag{9} $$

Hence, a distance constraint can be formulated as $p^* \geq \epsilon^{[i,j]}$. The dual problem of (9) is given by [5]

$$ \begin{aligned} d^* = \underset{\boldsymbol{\lambda}^{[i,j]}, \boldsymbol{\mu}^{[i,j]}}{\text{argmax}} \quad & - {}_w\boldsymbol{b_r}{}^{[i]T} \boldsymbol{\lambda}^{[i,j]} - {}_w\boldsymbol{b_o}{}^{[j]T} \boldsymbol{\mu}^{[i,j]} \\ \text{subject to} \quad & {}_wA_r{}^{[i]T} \boldsymbol{\lambda}^{[i,j]} + {}_wA_o{}^{[j]T} \boldsymbol{\mu}^{[i,j]} = 0, \\ & \left\| {}_wA_r{}^{[i]T} \boldsymbol{\lambda}^{[i,j]} \right\|_2^2 \leq 1, \\ & \boldsymbol{\lambda}^{[i,j]} \geq 0 \ \text{ and } \ \boldsymbol{\mu}^{[i,j]} \geq 0. \end{aligned} \tag{10} $$

If the primal problem is considered for a fixed s_r it is convex in $\boldsymbol{x}$ and $\boldsymbol{y}$, hence strong duality holds:

$$ d^* = p^*. \tag{11} $$

We define the objective of the dual problem as $g = - {}_w\boldsymbol{b_r}{}^{[i]T} \boldsymbol{\lambda}^{[i,j]} - {}_w\boldsymbol{b_o}{}^{[j]T} \boldsymbol{\mu}^{[i,j]}$. If we find dual variables $\tilde{\boldsymbol{\lambda}}^{[i,j]}$ and $\tilde{\boldsymbol{\mu}}^{[i,j]}$ by only requiring the constraints of (10) we know that $d^* \geq g(\tilde{\boldsymbol{\lambda}}^{[i,j]}, \tilde{\boldsymbol{\mu}}^{[i,j]})$ since $d^* = g(\boldsymbol{\lambda}^{*[i,j]}, \boldsymbol{\mu}^{*[i,j]})$ is the maximal value of g for a dual feasible (10) with optimal values $\boldsymbol{\lambda}^{*[i,j]}$ and $\boldsymbol{\mu}^{*[i,j]}$, in contrast to $g(\tilde{\boldsymbol{\lambda}}^{[i,j]}, \tilde{\boldsymbol{\mu}}^{[i,j]})$ which is just a value of g for feasible $\tilde{\boldsymbol{\lambda}}^{[i,j]}$ and $\tilde{\boldsymbol{\mu}}^{[i,j]}$ that validate the constraints of (10). By requiring $g(\tilde{\boldsymbol{\lambda}}^{[i,j]}, \tilde{\boldsymbol{\mu}}^{[i,j]}) \geq \epsilon^{[i,j]}$ we ensure that

$$ d^* \geq g(\tilde{\boldsymbol{\lambda}}^{[i,j]}, \tilde{\boldsymbol{\mu}}^{[i,j]}) \geq \epsilon^{[i,j]}. $$

Using the strong duality (11) we have shown that

$$ \textbf{dist}\left({}_w\mathcal{R}^{[i]}, {}_w\mathcal{O}^{[j]}\right) = p^* \geq \epsilon^{[i,j]}, $$

when imposing the following constraints in the addition of optimization variables $\boldsymbol{\lambda}^{[i,j]}$ and $\boldsymbol{\mu}^{[i,j]}$

$$ \begin{aligned} & - {}_w\boldsymbol{b_r}{}^{[i]T} \boldsymbol{\lambda}^{[i,j]} - {}_w\boldsymbol{b_o}{}^{[j]T} \boldsymbol{\mu}^{[i,j]} \geq \epsilon^{[i,j]}, \\ & {}_wA_r{}^{[i]T} \boldsymbol{\lambda}^{[i,j]} + {}_wA_o{}^{[j]T} \boldsymbol{\mu}^{[i,j]} = 0, \\ & \left\| {}_wA_r{}^{[i]T} \boldsymbol{\lambda}^{[i,j]} \right\|_2^2 \leq 1, \\ & \boldsymbol{\lambda}^{[i,j]} \geq 0 \ \text{ and } \ \boldsymbol{\mu}^{[i,j]} \geq 0, \end{aligned} \tag{12} $$

where $\epsilon^{[i,j]} > 0$ is the minimal imposed distance between the sets $\mathcal{R}^{[i]}$ and $\mathcal{O}^{[j]}$. These constraints are equivalent with the existence of a separating hyperplane between the two convex sets [5] (feasible solution for $\boldsymbol{\lambda}^{[i,j]}$ and $\boldsymbol{\mu}^{[i,j]}$ satisfying the constraints of (10) and resulting in a positive objective of (10)). When considering all $s_r \in [0, 1]$, the top two constraints are generally non-convex due to the b-depending ${}_w\boldsymbol{b_o}{}^{[j]}$ and ${}_wA_o{}^{[j]}$ which are multiplied with $\boldsymbol{\mu}$, the others are convex.

The time-optimal path following problem with collision avoidance can then be formulated by adding the constraints (12) to the time-optimal path following problem (5) for $s_r \in [0, 1]$, $i = 1..n_r$ and $j = 1..n_o$ where $\epsilon^{[i,j]}(s_r) > 0$.

This problem is no longer convex due to the collision avoidance constraints.

IV. Extensions to the collision avoidance problem

This section presents some extensions to the time-optimal path following problem with collision avoidance that profit from the typical solutions of the collision avoidance problem along a path.

A. Regularisation

Without collision avoidance the result is of the bang-bang type in the robot motor torques. By avoiding collision and only minimizing the end-time of the motion, typically the result is not of the bang-bang type before the object passes by, resulting in some freedom in the solution (see Section V). In this section additions to the objective function are presented that exploit this freedom, either to speed up the convergence or to reduce the energy losses.

1) Cumulative-time-regularisation: The freedom in the solution may result in slow convergence of the solver. To reduce this freedom and hence, in an attempt to improve the convergence, a cumulative time regularisation is introduced. This will result in a bang-bang behaviour, which results in a restricted number of possible solutions. In cumulative time regularisation the objective function is no longer the end-time but the end-time added with the weighted cumulative time from the start until the end. The cumulative time from $s = 0$ to s_r is given by

$$t_{cum}([0, s_r]) = \int_{[0,s_r]} t(\rho)d\rho = \int_0^{s_r} \left(\int_0^{\rho} b(\sigma)^{-\frac{1}{2}} d\sigma \right) d\rho,$$

hence, the new objective becomes

$$\begin{aligned} T_{cum,\gamma} &= t(1) + \gamma_t t_{cum}([0, 1[), \\ &= \int_0^1 b(\sigma)^{-\frac{1}{2}} d\sigma + \gamma_t \int_{[0,1[} \left(\int_0^{\rho} b(\sigma)^{-\frac{1}{2}} d\sigma \right) d\rho, \end{aligned}$$

where γ_t is the weight for the cumulative time. Since $b(\sigma)^{-\frac{1}{2}}$ is convex in σ, the cumulative time function is convex. It should be noted that in the case of no collision avoidance, using the cumulative time as the objective, gives the same results as using the end-time.

2) Energy regularisation: The freedom in the solution could be used to reduce the energy loss by using less torque in the parts of the trajectory where there is some freedom. Because the optimal solution of the motion time is mainly determined by the parts of the trajectory where the collision avoidance constraints do not matter, and the robot torques are maximal, there is only a minimal loss in time-optimality. Analogously to [1], in this paper only thermal energy losses in the motor (copper losses) are considered, which are given by

$$\sum_{i=1}^{n} \int_0^1 R_i \boldsymbol{\tau}_i(\sigma)^2 b(\sigma)^{-\frac{1}{2}} d\sigma.$$

where R_i is the motor resistance and $\boldsymbol{\tau}_i$ is the motor torque of joint i. This convex function in the optimization variables b and $\boldsymbol{\tau}_i$ is added to the objective function, multiplied with a weighting factor γ_E, after normalizing with e.g. the maximal value for the torques, analogous to [1].

B. Using collision intervals to speed up the convergence

The collision avoidance constraints do not need to be added at every path coordinate between all the convex sets of the robot union ${}_w\mathcal{R}$ and object union ${}_w\mathcal{O}$. It can be precomputed where, and between which sets, the collision will happen. This is done by calculating the optimal path following problem without collision avoidance (5) and by calculating the distance between the sets for a coarse s_r-grid by solving the primal problem (9), or by using the widely used algorithm by Gilbert, Johnson and Keerthi [11]. Furthermore, only those sets (indicated as critical collision set pairs) need to be considered that make sure that, when avoiding collision between these sets, there is no collision between all the sets. Hence collision intervals $s_r \in \mathcal{S}_{col}^{[i,j]} = [s_{r,col.start}^{[i,j]}, s_{r,col.end}^{[i,j]}]$ can be identified between critical collision pairs $\{i, j\}$ stored in the critical collision pair set $\mathcal{C}_{col}$. Since this problem formulation omits the hard (collision avoidance) constraints at irrelevant s_r coordinates faster convergence is expected. The following optimization problem can be formulated

$$\begin{aligned} \underset{b,\boldsymbol{\lambda}^{[i,j]},\boldsymbol{\mu}^{[i,j]}}{\text{minimize}} \quad & \int_0^1 b(\sigma)^{-\frac{1}{2}} d\sigma + \gamma_t \int_{[0,1[} \int_0^{\rho} b(\sigma)^{-\frac{1}{2}} d\sigma d\rho \\ & + \gamma_E \sum_{i=1}^{n} \int_0^1 \frac{\boldsymbol{\tau}_i(\sigma)^2 b(\sigma)^{-\frac{1}{2}}}{\boldsymbol{\tau}_{+,i}^2} d\sigma \\ \text{subject to} \quad & b(0) = \dot{s}_{r,0}^2, b(1) = \dot{s}_{r,T}^2, b(s_r) \geq 0, \\ & \boldsymbol{\tau}_- \leq \boldsymbol{m}(s_r) \frac{b'(s_r)}{2} \\ & \quad + \boldsymbol{c}(s_r) b(s_r) + \boldsymbol{g}(s_r) \leq \boldsymbol{\tau}_+, \\ & \text{for } s_r \in [0, 1], \\ & - {}_w\boldsymbol{b_o}^{[j]} \left(s_o \left(\int_0^{s_r} b(\sigma)^{-\frac{1}{2}} d\sigma \right) \right)^T \boldsymbol{\mu}^{[i,j]}(s_r) \\ & \quad - {}_w\boldsymbol{b_r}^{[i]}(s_r)^T \boldsymbol{\lambda}^{[i,j]}(s_r) \geq \epsilon^{[i,j]}(s_r), \\ & {}_wA_o^{[j]} \left(s_o \left(\int_0^{s_r} b(\sigma)^{-\frac{1}{2}} d\sigma \right) \right)^T \boldsymbol{\mu}^{[i,j]}(s_r) \\ & \quad + {}_wA_r^{[i]}(s_r)^T \boldsymbol{\lambda}^{[i,j]}(s_r) = 0, \\ & ||{}_wA_r^{[i]}(s_r)^T \boldsymbol{\lambda}^{[i,j]}(s_r)||_2^2 \leq 1, \\ & \boldsymbol{\lambda}^{[i,j]}(s_r) \geq 0 \text{ and } \boldsymbol{\mu}^{[i,j]}(s_r) \geq 0, \\ & \text{for } s_r \in \mathcal{S}_{col}^{[i,j]} \text{ and } \{i, j\} \in \mathcal{C}_{col}. \end{aligned} \tag{13}$$

where $\epsilon^{[i,j]} > 0$.

The general collision avoidance problem is then formulated as in Algorithm I. Pre-computing the collision intervals and pairs can be done very efficiently (see Section V) since only convex problems (path following and distance calculation) have to be solved.

Algorithm I Collision interval avoidance

Collision pre-computation:

- Solve (5)
- Solve (9) for some $s_r \in [0,1]$ grid
- Determine $\mathcal{S}_{col}$ and $\mathcal{C}_{col}$

Collision avoidance:

- Solve (13) using the solution of (5) as initial guess
- Recheck collision and iterate if necessary

γ_t	γ_E	E	T	iter	time [s]
1	0	1	1	60	1.2
0	0	0.69	1	67	1.6
0	10^{-6}	0.68	1	130	2.9
0	10^{-5}	0.63	$1+5\cdot10^{-5}$	140	3.1
0	10^{-4}	0.51	$1+2\cdot10^{-3}$	160	3.8

TABLE I

COMPARISON BETWEEN THE SOLUTION FOR SEVERAL VALUES OF THE REGULARISATION WEIGHTS γ_t AND γ_E

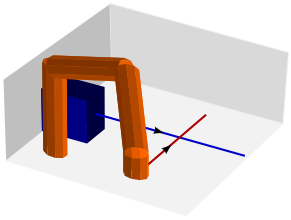


Fig. 2. Visualisation of the robot and object model and their trajectories.

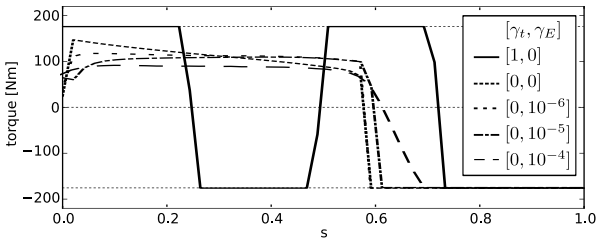


Fig. 3. Robot motor torque of joint one as a function of s_r for several values of γ_t and γ_E.

V. NUMERICAL VALIDATION

To illustrate the efficiency of the problem formulation (13), two examples are considered. In the first example a seven dof robot has to avoid hitting a moving box while moving along a trajectory. The second example considers two robots moving along a trajectory, while avoiding collision.

The time-optimal path following problem with collision avoidance (13) is implemented in Python using CasADi [12] as modelling software and Ipopt [13] as non-linear solver. All problems are discretized in 50 s_r-intervals and are solved on an Ubuntu Virtual Box with 1Gb RAM, running on a 2.4GHz Windows laptop.

A. Collision Avoidance between a robot and an object (box)

The following case is considered. A seven dof robot has to execute a linear trajectory time-optimally while avoiding a moving box on a conveyor that intersects with the robots trajectory. Figure 2 shows a visualisation of this set-up. We define each link, connected joint k, of the robot as a polyhedron tube with ten faces (see Figure 1) in the link frame $\{p_k\}$. Hence (8) can be used where ${}^{p_k}_{w}R(\boldsymbol{q}(s_r))^T$ and ${}_w\boldsymbol{x}^{w,p_k}$ are given by the robot kinematics. The box is modelled by a polyhedron with six faces and does not rotate on the conveyor, hence (8) can be used where ${}^{p_k}_{w}R^T$ is a constant matrix and ${}_w\boldsymbol{x}^{w,p_k}(s_o)$ progresses linearly with s_o and t.

Since collision is only possible between the end-effector of the robot and the box, the collision avoidance constraints are only added between these sets with $\epsilon = 0.01$ m for $s_r \in [0,1]$. Hence the collision interval approach is not used here. The time-optimal path following problem is solved for several values of γ_t and γ_E, with initial value $b(s_r) = 0.78\ \forall s_r \in [0,1]$. The number of iterations and computation times are listed in Table I. Furthermore, the energy loss E and total motion time T, both normalised to the case of $\gamma_t = 1$ and $\gamma_E = 0$, are also listed. Figure 3 shows the torque of joint one (dominant in the optimization) of the seven dof robot, for several values of γ_t and γ_E.

In the case of cumulative time regularisation ($\gamma_t = 1$), a bang bang behaviour is obtained. Due to the more restricted solution of the problem, the convergence is faster than when $\gamma_t = 0$. Two accelerate-decelerate bang-bang motions can be identified. One before the object passes by (move as fast as possible to the object) and one after (move as fast as possible again). Because of this bang-bang behaviour, much energy is lost which can already be decreased by setting $\gamma_t = 0$, even without energy regularisation ($\gamma_E = 0$), resulting in the same motion time. Here the robot makes sure that it does not hit the object, while not using its maximal allowed torque, and moves as fast as possible after the object is passed to obtain the minimal motion time. Due to more freedom in the solution this problem takes more iterations to solve. By adding energy-regularisation, even less energy is consumed (up to almost 50% when $\gamma_E = 10^{-4}$) with only minimal loss in time-optimality (up to $2 * 10^{-3}$ when $\gamma_E = 10^{-4}$). However, as can be seen from Table I, the number of iterations increases with the introduction of more energy regularisation.

It should be noted that these solutions are slightly depending on the initial value. Furthermore, in some cases with cumulative time regularisation it takes more iterations to converge than without. Hence, it can be concluded that this cumulative time regularisation can improve the convergence but this is not always the case.

Figure 4 shows the comparison between the torques of joint one and the distance between the robot end-effector and the box, as a function of s_r, with (for $\gamma_t = 0$ and $\gamma_E = 0$) and without collision avoidance. As can be seen from this figure, the distance to the object becomes zero (hence collision),

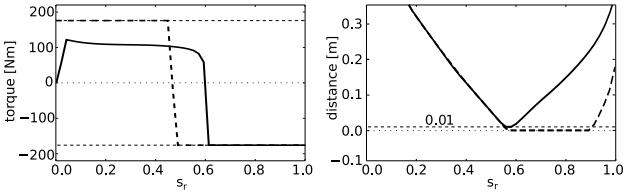


Fig. 4. Robot torque of joint one (left) and distance of the end effector to the object (right), as a function of s_r, without (dashed line) and with (full line) collision avoidance for $\epsilon = 0.01$ m, $\gamma_t = 0$ and $\gamma_E = 0$.

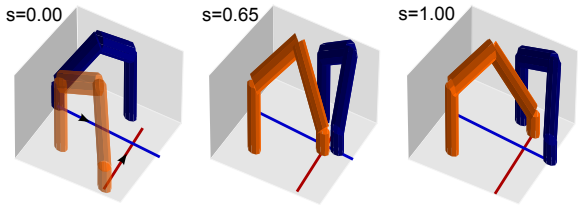


Fig. 5. Visualisation for two robots and their trajectories at $s_r = 0$, $s_r = 0.65$ and $s_r = 1$ with collision avoidance

from $s_r = 0.55$ to 0.95, without collision avoidance, and becomes equal to the minimal distance imposed (0.01 m), at $s_r = 0.55$ with collision avoidance. As already stated above, to avoid the object, the robot motor torque is below its maximal allowed value before $s_r = 0.55$.

B. Collision Avoidance between two robots

This section discusses the collision avoidance between two seven dof robots that have to execute linear trajectories that intersect, resulting in collision if the motions would be optimized without collision avoidance.

Since the path of the object is supposed to be known in the proposed method, the following approach is used. It is assumed that one robot has priority over the other, which means that the motion for that robot is optimized, time-optimally, disregarding the possibility for collision. The motion of the second robot is then optimized using the known trajectory, as a function of time, of the first robot while avoiding collision.

Figure 5 shows a visualisation of the set-up. Here the blue robot is the one with priority, while the orange robot has to avoid colliding into the blue robot. In this example, collision avoidance constraints are introduced between the upper arms and the end-effectors of both robots with $\epsilon = 0.02$ m. Solving the problem without collision intervals requires over 400 iterations in 22.3s, resulting in a suboptimal solution (not true optimal motion time). However, pre-computing the collision intervals and initialising the problem with the solution without collision avoidance requires only 59 iterations in 2.09s, resulting in the true optimal motion time. The pre-computation is done for a discretised trajectory s_r of 10 intervals and is solved in 1.2s. Figure 5 shows a visualisation at several s_r for the collision avoidance between the two robots. It should be noted that switching the priority of the robots could result in a different solution. Hence. calculating the true optimal motion of the two robots requires solving the optimal path following problem for two cases of priority. In this example the optimal motion time is the same if the orange robot has priority since the robots trajectories are identical.

VI. Conclusion

This paper presents an efficient approach to introduce collision avoidance constraints into the time-optimal path following framework of [1]. This renders the problem non-convex. However, due to the time-optimal path following formulation and the efficient formulation of the collision avoidance constraints, presented in this paper, the problem can still be solved efficiently, offline, as illustrated for collision avoidance between two seven dof robots. Future research should focus on the real-time implementation using estimations of the object trajectories since the computation time too high for on-line optimization. The computation time could be decreased by using code-generation and by running it on a faster computer.

References

[1] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *Automatic Control, IEEE Transactions on*, vol. 54, pp. 2318 –2327, oct. 2009.

[2] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[3] K. Shin and N. Mckay, "Minimum-time control of robotic manipulators with geometric path constraints," *Automatic Control, IEEE Transactions on*, vol. 30, no. 6, pp. 531–541, 1985.

[4] M. Gerdts, R. Henrion, D. Hömberg, and L. C., "Path planning and collision avoidance for robots," *Numerical algebra, control and optimization*, vol. 2, 2012.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

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

[7] K. Gupta and A. P. Pobil, *Practical Motion Planning in Robotics: Current Approaches and Future Directions*. New York, NY, USA: John Wiley & Sons, Inc., 1998.

[8] A. Dietrich, T. Wimböck, H. Täubig, A. Albu-Schäffer, and G. Hirzinger, "Extensions to reactive self-collision avoidance for torque and position controlled humanoids.," in *ICRA*, pp. 3455–3462, IEEE, 2011.

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

[10] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 1989.

[11] E. G. Gilbert, D. W. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in 3d space," *IEEE Journal on Robotics and Automation*, vol. 4, 1988.

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

[13] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.