Periodica Polytechnica Electrical Engineering and Computer Science, 62(1), pp. 16–23, 2018

Path Tracking Algorithms for Non-Convex Waiter Motion Problem

Ákos Nagy[1]*, Gábor Csorvási[1], István Vajk[1]

[1] Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics, H-1521 Budapest, P.O.B. 91, Hungary
* Corresponding author, email: akos.nagy@aut.bme.hu

Received: 20 October 2017, Accepted: 09 December 2017, Published online: 07 February 2018

Abstract

Originally, motion planning was concerned with problems such as how to move an object from a start to a goal position without hitting anything. Later, it has extended with complications such as kinematics, dynamics, uncertainties, and also with some optimality purpose such as minimum-time, minimum-energy planning. The paper presents a time-optimal approach for robotic manipulators. A special area of motion planning is the waiter motion problem, in which a tablet is moved from one place to another as fastas possible, avoiding the slip of the object that is placed upon it. The presented method uses the direct transcription approach for the waiter problem, which means a optimization problem is formed in order to obtain a time-optimal control for the robot. Problem formulation is extended with a non-convex jerk constraints to avoid unwanted oscillations during the motion. The possible local and global solver approaches for the presented formulation are discussed, and the waiter motion problem is validated by real-life experimental results with a 6-DoF robotic arm.

Keywords

motion planning, minimum-time control, time-optimal control, convex optimisation, robot control

1 Introduction

Time-optimal motion planning has been a topic of active research since the 1980. Minimum-time algorithms can maximize productivity, and reduce energy consumption of the robotic system. Direct approaches [1] or one step methods solve the entire problem in one step. In general it is a highly difficult task, since both the geometric constraints (including collision avoidance), and the timing along this geometric path (including dynamic limitation) have to be optimized.

As an alternative to direct approach, the motion planning problem is often decoupled [2, 3]. First, a high-level geometry path planner determines a path for the robot considering geometric constraints and ignoring system dynamics. In the next stage (path tracking), a velocity profile for a predefined path is generated, where all constraints of the robot are applied for the fixed path. Decoupled approach is preferred to direct approach for its lower computational time.

In this paper, we will focus on the second stage of the decoupled motion planning approach. Since the desired path of the robot is already defined, a scalar path coordinate ($\theta(t)$) can be used to represent robot position on the path [4-7]. The major advantage of the scalar path coordinate is that the high dimensional state-space model of the robotic system can be reduced.

Generally there are three different approaches to solve the decoupled minimum-time control problem. Indirect methods are based on the Pontryagin Maximum Principle: the optimal bang-bang control is obtained by integrating maximum and minimum accelerations, and finding optimal switching points, where the active constraints can be changed [8-10]. Dynamic programming approach divides the state space into a discrete grid, and the optimal solution is found in this plane [11, 12]. Direct transcription is the third velocity profile generation method. The optimal control problem can be cast as a convex optimization problem [5, 13]. Existing efficient convex optimisation solvers can be used in these methods in order to get minimum-time solution, and the convexity guarantees a globally optimal solution. Using direct transcription, the objective function can be extended with other criteria, i.e. thermal energy (integral of the square of the torque of

Cite this article as: Nagy, Á., Csorvási, G., Vajk, I. " Path Tracking Algorithms for Non-Convex Waiter Motion Problem", Periodica Polytechnica Electrical Engineering and Computer Science, 62(1), pp. 16–23, 2018. https://doi.org/10.3311/PPeecs.11606

joints), torque jumps (integral of the absolute value of the rate of change of the torque) [5]. However convex reformulation fails to address many practical applications (e.g. torque rate constraints, velocity-dependent torque constraints, jerk constraints) since these applications introduce terms that destroy the convexity.

In this paper, a direct transcription formulation for the time-optimal waiter problem with joint jerk constraints is presented. In the so-called waiter motion problem a tablet in the gripper of the robot manipulator is moved from one pose to the other such that the object resting on the tablet does not slide at any time during the movement.

Due to the bang-bang nature of time-optimal control, near infinite jerks can be occurred in the resulted trajectory, and this usually results unwanted oscillations. These vibrations can be eliminated by jerk constraint. Since the joint jerk limitation cannot be written as a convex constraint, the convex reformulation is not possible in this case. However these constraints can be written as a difference of two convex (DC) functions, and using existing DC solver methods, optimal or near-optimal control can be obtained.

The paper is organized as follows. In Section 2 the time-optimal waiter motion problem is presented with non-convex jerk constraint. The formulation is based on the work of Verscheure et al. [5], Debrouwere et al. [14]. A possible solver methods for the presented problem are discussed in Section 3. In Section 4, the generated velocity profile for a 6-DoF robotic manipulator is examined. Real-life experimental results are also presented in Section 4 with the 6-DoF robot.

2 Problem Formulation

In this section the non-convex time-optimal control problem is introduced using non-linear change of variables. In the presented approach, a mathematical optimisation problem is formed based on the work of Verscheure et al. [5], Debrouwere et al. [14].

2.1 General Time-Optimal Formulation

During problem formulation $q(t) \in \mathbb{R}^p$ indicates a p-dimensional generalized coordinate vector representing the configuration of the robot. The dynamic equation of a p-DoF robotic system is defined by using the second-order Lagrange equation [5]

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

where $\tau(t) \in \mathbb{R}^p$ is the generalized force vector, $\dot{(\cdot)}$ indicates time derivative, $\mathbf{M}(q(t)) : \mathbb{R}^p \to \mathbb{S}^p_{++}$ is the mass matrix, $\mathbf{C}(q(t),\dot{q}(t)) : \mathbb{R}^{2p} \to \mathbb{R}^{p\times p}$ is a matrix accounting for centrifugal, Coriolis effects (it is linear in $\dot{q}$), and $d(q(t)) : \mathbb{R}^p \to \mathbb{R}^p$ accounts for joint position dependent forces (e.g. gravity).

Consider the predefined geometric path $s(\theta(t))$ as a function of a scalar path coordinate ($\theta(t)$) given in joint space coordinates. Using the chain rule, the kinematic equations (including jerk) for the robot can be derived (from now on, the time dependency is omitted)

$$\begin{aligned} q &= s(\theta) \\ \dot{q} &= s'(\theta)\dot{\theta} \\ \ddot{q} &= s'(\theta)\ddot{\theta} + s''(\theta)\dot{\theta}^2 \\ \dddot{q} &= s'(\theta)\dddot{\theta} + 3s''(\theta)\ddot{\theta}\dot{\theta} + s'''(\theta)\dot{\theta}^3 \end{aligned} \tag{2}$$

where $s(\theta) : [0,1] \to \mathbb{R}^p$, $\theta : [0,T] \to [0,1]$, $s'(\theta) = \frac{ds(\theta)}{d\theta}$, $s''(\theta) = \frac{d^2 s(\theta)}{d\theta^2}$, $s'''(\theta) = \frac{d^3 s(\theta)}{d\theta^3}$ and T is the time when the robot reaches the end of the path. We also assume that $\dot{\theta} \geq 0$ holds everywhere, which is a general assumption in time-optimal control.

The position vector of the end-effector ($r(t) \in$ SE(3)) in the operational space is also considered. The conversion between the joint space and the operational space coordinates can be obtained by the following formula

$$q(t) \underset{\varphi^{-1}}{\overset{\varphi}{\rightleftarrows}} r(t), \tag{3}$$

where $\varphi(q)$ and $\varphi^{-1}(q)$ are direct and inverse kinematics of the robot respectively.

The analogous equations for the operational space coordinates (without Cartesian jerk) can be written as

$$r = p(\theta) \tag{4}$$

$$\dot{r} = p'(\theta)\dot{\theta} \tag{5}$$

$$\ddot{r} = p'(\theta)\ddot{\theta} + p''(\theta)\dot{\theta}^2, \tag{6}$$

where $p(\theta)$ is the fixed path of the end-effector.

A new variable is introduced for the problem formulation

$$b(\theta) := \dot{\theta}^2 \tag{7}$$

and using (7) the derivatives of the scalar path coordinate are defined by the derivatives of $b(\theta)$

$$\begin{aligned} \dot{\theta} &= \sqrt{b(\theta)} \\ \ddot{\theta} &= \frac{1}{2}b'(\theta) \\ \dddot{\theta} &= \frac{1}{2}b''(\theta)\sqrt{b(\theta)} \end{aligned} \tag{8}$$

From now on, $b(\theta)$, $b'(\theta)$ and $b''(\theta)$ are considered to be the variables of the optimization problem. To create a finite dimensional problem we will discretise the problem in $n + 1$ points in the $\theta = [0\ldots1]$ range. The i-th point is indicated by θ_i. The discretisation is based on the assumption that $b''(\theta)$ is constant between two discretisation points, therefore $b'(\theta)$ is linear, and $b(\theta)$ is quadratic in θ

$$\begin{gathered} b'(\theta) = b'(\theta_{i-1}) + b''(\bar{\theta}_i)(\theta - \theta_{i-1}) \\ b(\theta) = b(\theta_{i-1}) + b'(\theta_{i-1})(\theta - \theta_{i-1}) + \frac{1}{2} b''(\bar{\theta}_i)(\theta - \theta_{i-1})^2 \\ \theta \in [\theta_{i-1}, \theta_i] \quad \bar{\theta}_i = (\theta_i + \theta_{i-1})/2 \quad i = 1,\ldots,n, \end{gathered} \tag{9}$$

where $b''(\theta)$ is evaluated at the midpoint of the interval since it is discontinuous in the discretisation points. The other variables are evaluated at θ_i

$$b_i := b(\theta_i) \quad b'_i := b'(\theta_i) \quad b''_i := b''(\bar{\theta}_i) \tag{10}$$

The relationship between the variables in discrete time can be defined by

$$\begin{gathered} b''_i = \frac{b'_i - b'_{i-1}}{d\theta_i} \\ b_i = b_{i-1} + \frac{1}{2}(b'_i + b'_{i-1}) d\theta_i \\ d\theta_i = \theta_i - \theta_{i-1} \qquad i = 1,\ldots,n. \end{gathered} \tag{11}$$

The Lagrange dynamics of the robot is evaluated in every θ_i using (1), (2), (8)

$$\tau_i = \frac{1}{2} m_i b'_i + c_i b_i + d_i \quad i = 0,\ldots,n, \tag{12}$$

where the parameters of equation are written as

$$\begin{gathered} m_i := \mathbf{M}(s_i) s'_i \\ c_i := \mathbf{M}(s_i) s''_i + \mathbf{C}(s_i, s'_1) s'_1 \\ d_i := d(s_i), \end{gathered} \tag{13}$$

and these parameters are also evaluated in θ_i

$$\tau_i := \tau(\theta_i) \quad s_i := s(\theta_i) \quad s'_i := s'(\theta_i) \quad s''_i := s''(\theta_i) \tag{14}$$

The derivatives of the path should be specified for (12). When $s(\theta)$ is differentiable, s'_i and s''_i can be calculated analytically. Otherwise an approximation method is necessary for the derivatives (e.q.: numerical derivation, spline interpolation) [6, 14].

The acceleration and the jerk of the p-DoF robot are discretised similar to dynamics based on (2), (8)

$$\begin{gathered} \ddot{q}_i = \frac{1}{2} s'_i b'_i + s''_i b_i \\ \dddot{q}_i = \frac{1}{2} \bar{s}'_i b''_i \sqrt{\bar{b}_i} + \frac{3}{2} \bar{s}''_i \bar{b}'_i \sqrt{\bar{b}_i} + \bar{s}'''_i \bar{b} \sqrt{\bar{b}_i}, \end{gathered} \tag{15}$$

where $\dddot{q}_i$ is evaluated at the midpoints

$$\begin{gathered} \dddot{q}_i := \dddot{q}(\bar{\theta}_i) \quad \bar{s}'_i := s'(\bar{\theta}_i) \quad \bar{s}''_i := s''(\bar{\theta}_i) \quad \bar{s}'''_i := s'''(\bar{\theta}_i) \\ \bar{b}'_i := \frac{b'_i + b'_{i-1}}{2} \quad \bar{b}_i := b_{i-1} + \left(\frac{1}{8} b'_i + \frac{3}{8} b'_{i-1}\right) d\theta_i, \end{gathered} \tag{16}$$

where $\bar{b}_i$ can be derived using (9), (11)

$$\begin{gathered} b(\bar{\theta}_i) = \bar{b}_i = b_{i-1} + b'_{i-1} \frac{d\theta_i}{2} + \frac{1}{2} b''_i \frac{d\theta_i^2}{4} \rightarrow \\ \bar{b} = b_{i-1} + b'_{i-1} \frac{d\theta_i}{2} + (b'_i - b'_{i-1}) \frac{d\theta_i}{8} \rightarrow \\ \bar{b} = b_{i-1} + \left(\frac{1}{8} b'_i + \frac{3}{8} b'_{i-1}\right) d\theta_i \end{gathered} \tag{17}$$

It should be note that using the presented discretisation approach, $\dddot{q}(\theta)$, $\ddot{q}(\theta)$, $\dot{q}(\theta)$ are non-linear between two subsequent points.

The goal of the problem formulation is to generate time-optimal motion for the robot, therefore the objective function of the optimisation problem is the motion time (T), which can be expressed by variable $b(\theta)$:

$$T = \int_0^T 1 dt = \int_{\theta(0)}^{\theta(T)} \dot{\theta}^{-1} \, d\theta = \int_0^1 b(\theta)^{-1/2} \, d\theta, \tag{18}$$

which is a convex function of $b(\theta)$. Computing T in discrete time by numerical integration based on (9) would result a non-convex objective function. Therefore an another approximation is applied for T, in which $b(\theta)$ is linear in θ. This technique is also used in the work of Debrouwere et al. [14]. So the approximated T is a convex function of b_i

$$T \approx 2 \sum_{i=1}^{n} \left(\frac{d\theta_i}{b_i^{1/2} + b_{i-1}^{1/2}} \right). \tag{19}$$

2.2 Waiter Motion Constraint

Let us consider a waiter motion which consists of moving a tablet carrying a glass along a predefined path as fast as possible so that the objects placed on it do not slide. The problem is introduced in the work of Flores and Kecskeméthy [9] using an indirect approach. The optimal trajectory is constructed by integrating the system equations forward and backward in time from switching points. The drawback of this approach is that it is difficult to accurately determining these switching points.

In this paper a different method is presented, where a convex sliding constraint is defined in the problem formulation. Similar result to the presented work can be found in [15, 16].

Since the goal is that the object does not move on the tablet, a static friction model is used, and the no sliding condition can be written as

$$\sqrt{F_x^2 + F_y^2} \le \mu_0 F_z \tag{20}$$

$$F = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = m\left(\ddot{\bar{r}}(t) - g\right), \tag{21}$$

where F is the force applied to the object excluding the gravity, F_x, F_y and F_z are the components of F in the local coordinate frame of the tablet, m is the object mass, g is the gravitational acceleration vector, μ_0 is the dry friction coefficient between the contact surfaces and $\ddot{\bar{r}}$ indicates the translational part of object acceleration (first three coordinate of $\ddot{r}$). An illustration of the applied force (F) can be seen in Fig. 1.

In order to transform the sliding constraint to a convex form, rearrange (20) using Euclidean norm

$$|F|_2 \le \sqrt{\mu_0^2 + 1} \cdot F_z. \tag{22}$$

A friction angle (α) can be introduced to calculate μ_0

$$\mu_0 = \tan\alpha, \tag{23}$$

and F_z can be also expressed using the normal vector of the tablet plane (n)

$$F_z = F \cdot n. \tag{24}$$

Substituting (25), (26) into (24), we get the convex form of the sliding constraint

$$|F|_2 \le \frac{F \cdot n}{\cos\alpha} \tag{25}$$

Using (21), and simplify the formula with mass m, the obtained sliding constraint is

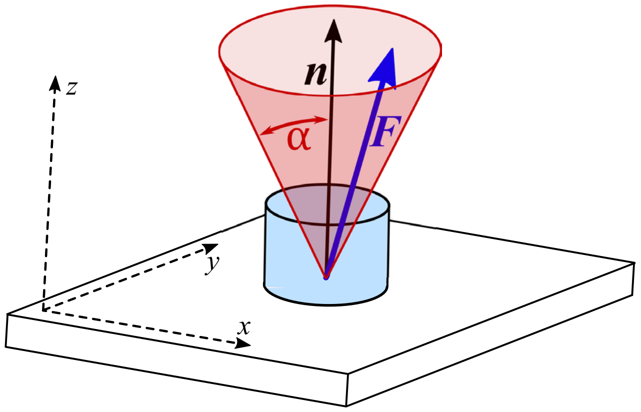

Fig. 1 The presented scenario for the waiter motion problem. The F force applied to the object placed on the tablet.

$$\left|\ddot{\bar{r}} - g\right|_2 \le \frac{\left(\ddot{\bar{r}} - g\right) \cdot n}{\cos\alpha} \tag{26}$$

2.3 The Whole Optimization Problem

The complete optimization problem in discrete time is formed based on the presented constraints and objective function. Besides the sliding constraint, velocity, acceleration, jerk and torque constraints are also considered in the model

$$\begin{aligned}
\min \quad & 2\sum_{i=1}^{n}\left(\frac{d\theta_i}{b_i^{1/2} + b_{i-1}^{1/2}}\right) & \\
\text{s. t.:} \quad & b_1'' = \frac{b_i' - b_{i-1}'}{d\theta_i} & i = 1,\dots,n \\
& b_i = b_{i-1} + \frac{1}{2}\left(b_i' + b_{i-1}'\right)d\theta_i & i = 1,\dots,n \\
& \bar{b}_1' := \frac{b_i' + b_{i-1}'}{2} & i = 1,\dots,n \\
& \bar{b}_i = b_{i-1} + \left(\frac{1}{8}b_i' + \frac{3}{8}b_{i-1}'\right)d\theta_i & i = 1,\dots,n \\
& \ddot{\bar{r}}_i = \frac{1}{2}p_i' b_i' + p_i'' b_i & i = 0,\dots,n \\
& \ddot{q}_i = \frac{1}{2}s_i' b_i' + s_i'' b_i & i = 0,\dots,n \\
& \dddot{q}_i = \frac{1}{2}\bar{s}_i' b_i'' \sqrt{\bar{b}_i} & \\
& \quad + \frac{3}{2}\bar{s}_i'' \bar{b}_i' \sqrt{\bar{b}_i} + \bar{s}_i''' \bar{b}_i \sqrt{\bar{b}_i} & i = 1,\dots,n \\
& \tau_i = \frac{1}{2}m_i b_i' + c_i b_i + d_i & i = 0,\dots,n \\
& 0 \le b_i \le b_i^{\max} & i = 0,\dots,n \\
& -\ddot{q}_i^{\max} \preceq \ddot{q}_i \preceq \ddot{q}_i^{\max} & i = 0,\dots,n \\
& -\dddot{q}_i^{\max} \preceq \dddot{q}_i \preceq \dddot{q}_i^{\max} & i = 1,\dots,n \\
& -\tau_i^{\max} \preceq \tau_i \preceq \tau_i^{\max} & i = 0,\dots,n \\
& \left|\ddot{\bar{r}} - g\right|_2 \le \frac{\left(\ddot{\bar{r}} - g\right) \cdot n}{\cos\alpha},
\end{aligned} \tag{27}$$

where $b_i^{\max} \ge 0$ is a velocity limit, $\ddot{q}_i^{\max} \succeq 0$ is an acceleration limit, $\dddot{q}_i^{\max} \succeq 0$ is a jerk limit, $\tau_i^{max} \succeq |d|$ is a torque limit, and '$\succeq, \preceq$' represent componentwise inequalities.

Problem (29) is a finite dimensional non-convex problem, where the optimisation variables are b_i, b_i' and b_i'', all other parameters are defined before the optimisation, or can be expressed using the equality constraints.

Without the jerk constraint, the problem would be a convex optimization problem, as other constraints are affine or quadratic (sliding constraint), and the objective function is convex. It would be namely a Second-Order Cone Programming (SOCP) problem, which could be solved efficiently using existing convex optimisation solvers.

3 Solver Methods for the Control Problem

In this section different solver methods for the presented problem are discussed. Due to non-convexity, local optimum points can be found in the feasible set of the problem. Therefore local and global solvers are discussed separately in the followings.

3.1 Global Solvers

Before global solver methods are presented, the non-convex jerk constraint is reformulated

$$-\dddot{q}_i^{\max} \preceq \sqrt{\overline{b}_i}\left(\overline{s}_i'\frac{1}{2}b_i''+3\overline{s}_i''\frac{1}{2}\overline{b}_i'+\overline{s}_i'''\overline{b}_i\right)\preceq \dddot{q}_i^{\max} \tag{28}$$

Since square of root function is non-negative, (30) can be written as

$$\begin{aligned}&\left(s\frac{1}{2}b_i''+3\overline{s}_i''\frac{1}{2}\overline{b}_i'+\overline{s}_i'''\overline{b}_i\right)-\dddot{q}_i^{\max}\overline{b}_i^{-\frac{1}{2}}\preceq 0\\&-\left(\overline{s}_i'\frac{1}{2}b_i''+3\overline{s}_i''\frac{1}{2}\overline{b}_i'+\overline{s}_i'''\overline{b}_i\right)-\dddot{q}_i^{\max}\overline{b}_i^{-\frac{1}{2}}\preceq 0\end{aligned} \tag{29}$$

Since the left sides of (29) are affine, and the right side ($\overline{b}_i^{-\frac{1}{2}}$) is a convex function of $\overline{b}_i$, (29) can be formed as a difference of two convex function (DC function). DC functions have some appealing properties [17, 18], which makes it possible to find the global optimum faster than a general non-convex solver.

A general DC optimisation problem can be defined using DC constraints

$$\begin{aligned}&\min_x \quad f(x)\\&\text{s. t.:}\quad u_i(x)-v_i(x)\le 0 \quad i=0,\ldots,l,\end{aligned} \tag{30}$$

where $f(x)$, $u_i(x)$ and $v_i(x)$ are convex functions. Problem (27) can be considered as a general DC program (30), and existing solvers for DC programs can be applied for the presented waiter motion problem. In general, existing DC global solvers are not so efficiently as the convex solvers regarding to runtime, convergence and numerical stability.

A cutting plane procedure for DC program was developed in [19]. To understand the cutting plane method, reformulate (30) to the following canonical form

$$\begin{aligned}&\min_x \qquad f(x)\\&\text{subject to}\quad x\in D\setminus \operatorname{int} C,\end{aligned} \tag{31}$$

where D and C are convex sets, and int C indicates the interior of C. Every DC optimization problem can be written in canonical form [20].

If the DC program satisfy the so-called regularity condition [20], then the following optimality condition can be made: x^* is global optimum of (31) if and only if

$$\begin{aligned}&D\left(x^*\right)\subset C\\&D\left(x^*\right)=D\cap\left\{x:f\left(x\right)\le f\left(x^*\right)\right\}\end{aligned} \tag{32}$$

An example for (32) can be seen in Fig. 2. Cutting plane methods iteratively converges to a x^* point from an initial, feasible point (x^0). At every step, a convex polytope (P_k) is defined satisfying

$$D\left(x^k\right)\subset P_k\subset\left\{x:f\left(x\right)\le f\left(x^k\right)\right\}, \tag{33}$$

If condition $D\left(x^k\right)\subset C$ is true, then the actual x^k is a global optimum. Otherwise a cutting plane is defined, and the polytope is refined using

$$P_{k+1}=P_k\cap\left\{x:l_k\left(x\right)\le 0\right\} \tag{34}$$

A new x^{k+1} should be also defined, and it can be proved that the presented method converges to a global optimum.

Branch and bound (B&B) based approach can be also used for (30) [17]. Due to the DC property of the constraints, lower and upper bounds of the objective function can be approximated. A B&B algorithm for non-convex time-optimal problem is presented in [21], where a non-convex velocity-dependent torque constraint is applied, which is very similar to the presented jerk constraint (29).

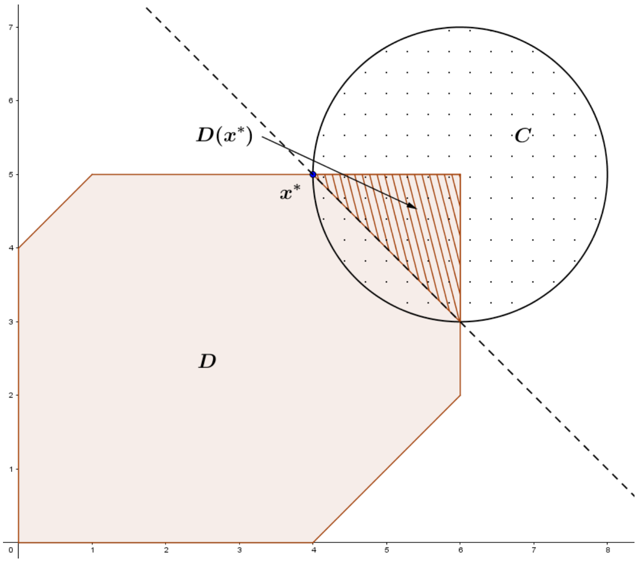

Fig. 2 Example for the DC optimality condition. x^* is a global optimum point, and $D\setminus\operatorname{int} C$ is the feasible set of the problem. The dashed line is the contour line of the objective function.

3.2 Local Solvers

Local solvers for the general DC problem can perform significantly better compared to global methods. In this section a Sequential Convex Programming (SCP) algorithm [14] is presented as a local solver for the waiter motion problem.

The main idea behind the SCP algorithm is to iteratively linearize the concave part of the DC constraints to transform the problem into a convex optimisation problem. Guaranteed convergence to a local optimum can be proven for this method in contrast to general non-convex solution methods.

Suppose that x^k is a given point, then the approximated problem is

$$\begin{aligned} \min_{x} \quad & f(x) + \beta \left\| x - x^k \right\|_2^2 \\ \text{s. t.:} \quad & u_i(x) - v_i\left(x^k\right) - \nabla_x v_i\left(x^k\right)\left(x - x^k\right) \le 0 \\ & i = 0, \ldots, l, \end{aligned} \tag{35}$$

where DC constraints are linearised around x^k , and β is a regularization factor added to the objective function to ensure proper convergence. The stopping criterion of the sequential algorithm can be the sufficiently low decrease of the objective function

$$\left\| T^{k+1} - T^k \right\| \le \epsilon, \tag{36}$$

where T^k indicates the travel time of the current iteration k.

4 Experimental Results

In this section the path tracking method is demonstrated in a real life scenario using a Mitsubishi RV-3SDB 6-DoF robot manipulator (see Fig. 3). A local solver (see Section 2) for the presented optimization problem isapplied to obtain a near time optimal trajectory for the waiter motion problem. The solver is implemented in C++ using the state-of-art Gurobi commercial SOCP, LP solver [22].

4.1 Offline Velocity Profile Generation

A tablet was placed in the gripper of the robotic arm, and a empty glass was put on the tablet. The path of the gripper has been defined manually in operational space, and using inverse kinematics the joint space path was calculated (see Fig. 4). The orientation of the table was fixed during the motion, and it is defined using Euler angles. We used the Robotics Toolbox for Matlab [23] to calculate the inverse kinematics for the path. Based on the algorithm presented in Section 3.2 time parameterized trajectory was generated for the waiter motion problem.

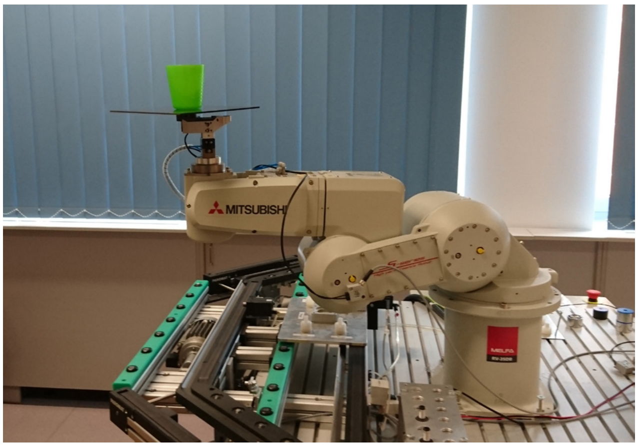

Fig. 3 Mitsubishi RV-3SDB robot.

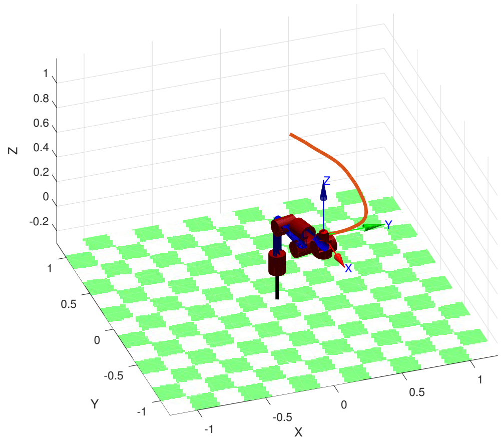

Fig. 4 The fixed path (orange line) of the gripper in Matlab using the Robotics Toolbox.

Velocity, acceleration, jerk, and a constraint accounting for sliding effect are used for the demonstration. Torque constraints were not used, because the dynamic parameters of the robot were not known at the time.

For sliding constraint of the problem formulation it was necessary to determine the friction angle (α) used in (25). We measured α by placing the glass on the table, and increased the angle of the tablet until the glass begun to slide. The actual value of α was 9° in our scenario.

Two generated velocity, acceleration and jerk profiles for the third joint of the robot can be seen in Fig. 5. Only the jerk constraint was different in the two profiles: symmetrical 100 rad/s^3 limit was used for the first case, and 5 rad/s^3 for the second profile.

4.2 Real-Time Path Tracking

The real-time path tracking method has been implemented in the following way: the desired joint positions were sent

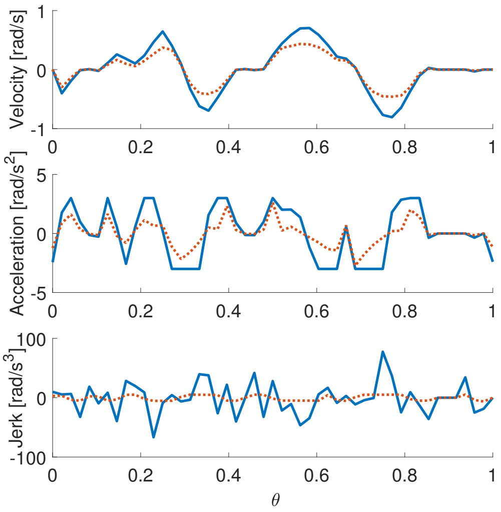

Fig. 5 The generated profiles for the third joint. For the solid profile higher jerk constraint (100 rad/s^3) is used compared to the dotted one (5 rad/s^3).

to the robot controller at the fixed cycle time of the controller (7.1 ms [24]) over a TCP/IP network connection. The controller send back the joint positions, so the actual positions were also measured during path tracking.

The desired joint space path was equal to the predefined path, but it also had a time parameter. The actual position of the path was selected based on this time coordinate. To make the motion smoother, interpolation was used between path points.

As we used the predefined path as the control signal, difference between the predefined and the traversed path was only depends on the accuracy of the robotic arm, which is negligibly small (max. ±0.02 mm in operational space [25]). However, the velocity profiles can have

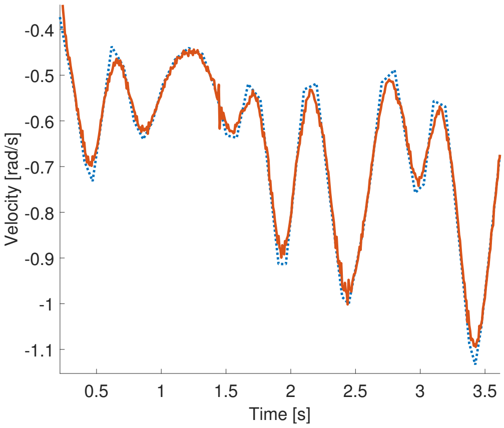

Fig. 6 The difference between the reference and the measured velocities for the first joint. Dotted line is the reference velocity, the solid line is the measured profile from the robot controller.

higher error due to the approximations of the path derivatives. The actual error in the velocity profile can be seen in Fig. 6 for the last joint.

5 Conclusion

In this paper, the well-known time-optimal control problem has been discussed using the direct transcription approach. The problem formulation was extended with two, special property. A waiter motion task was considered, which implied a convex sliding constraint in the problem formulation, and a non-convex jerk constraint was also taken into account. The possible global and local solver algorithms were discussed for the optimization problem. In the last section of the paper, experimental result has been presented using a 6-DoF robot manipulator based on a local method to the non-convex problem.

References

[1] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., Thrun, S. "Principles of Robot Motion: Theory, Algorithms, and Implementations", MIT Press, Cambridge, MA, USA, 2005.

[2] LaValle, S. M. "Planning Algorithms", Cambridge University Press, New York, NY, USA, 2006.

[3] Ardeshiri, T., Norrlöf, M., Löfberg, J., Hansson, A. "Convex Optimization approach for Time-Optimal Path Tracking of Robots with Speed Dependent Constraints", In: Proceedings of 18th IFAC World Congress, pp. 14648–14653, Milano, Italy, 2011.

[4] Shin, K., Mckay, N. D. "Minimum-time control of robotic manipulators with geometric path constraints", IEEE Transactions on Automatic Control, 30(6), pp. 531–541, 1985. https://doi.org/ 10.1109/TAC.1985.1104009

[5] Verscheure, D., Demeulenaere, B., Swevers, J., Schutter, J. D., Diehl, M. "Time-Optimal Path Tracking for Robots: A Convex Optimization Approach", IEEE Transactions on Automatic Control,54(10), pp. 2318–2327, 2009. https://doi.org/10.1109/TAC.2009.2028959

[6] Lipp, T., Boyd, S. "Minimum-time speed optimisation over a fixed path", International Journal of Control, 87(6), pp. 1297–1311, 2014. https://doi.org/10.1080/00207179.2013.875224

[7] Hauser, K. "Fast Interpolation and Time-Optimization on Implicit Contact Submanifolds", In: Proceedings of Robotics: Science and Systems, Berlin, Germany, 2013.

[8] Bobrow, J. E., Dubowsky, S., Gibson, J. S. "Time-optimal Control of Robotic Manipulators along Specified Paths", International Journal of Robotics Research, 4(3), pp. 3–17, 1985. https://doi.org/10.1177/027836498500400301

[9] Flores, F. G., Kecskeméthy, A. "Time-Optimal Path Planning Along Specified Trajectories", In: Gattringer, H., Gerstmayr, J. (eds.), Multibody System Dynamics, Robotics and Control, (pp. 1–16.), Springer, Vienna, 2013.
https://doi.org/10.1007/978-3-7091-1289-2_1

[10] Consolini, L., Locatelli, M., Minari, A., Piazzi, A. "An optimal complexity algorithm for minimumtime velocity planning", Systems & Control Letters, 103, pp. 50–57, 2017.
https://doi.org/10.1016/j.sysconle.2017.02.001

[11] Oberherber, M., Gattringer, H., Müller, A. "Successive dynamic programming and subsequent spline optimization for smooth time optimal robot path tracking", Mechanical Sciences, 6(2), pp. 245–254, 2015.
https://doi.org/10.5194/ms-6-245-2015

[12] Max, G., Lantos, B. "Time optimal control of four-in-wheel-motors driven electric cars", Periodica Polytechnica Electrical Engineering and Computer Science, 58(4), pp. 149–159, 2014.
https://doi.org/10.3311/PPee.7806

[13] Nagy, Á., Vajk, I. "LP-based Velocity Profile Generation for Robotic Manipulators", International Journal of Control, 2017.
https://doi.org/10.1080/00207179.2017.1286535

[14] Debrouwere, F., Loock, W. V., Pipeleers, G., Tran, D. Q., Diehl, M., Schutter, J. D., Swevers, J. "Time-Optimal Path Following for Robots With Convex-Concave Constraints Using Sequential Convex Programming", IEEE Transactions on Robotics, 29(6), pp. 1485–1495, 2013.
https://doi.org/10.1109/TRO.2013.2277565

[15] Duijkeren, N. V., Debrouwere, F., Pipeleers, G., Swevers, J. "Cartesian constrained time-optimal point-to-point motion planning for robots : the waiter problem", In: Benelux Meeting on Systems and Control, Lommel, Belgium, 2015.

[16] Csorvási, G., Nagy, Á., Vajk, I. "Near Time-Optimal Path Tracking Method for Waiter Motion Problem", In: Proceedings of 20th IFAC World Congress, pp. 5080–5085, Toulouse, France, 2017.

[17] Tuy, H. "Convex Analysis and Global Optimization", 2nd ed., Springer International Publishing, 2016.
https://doi.org/ 10.1007/978-3-319-31484-6

[18] Zhang, Q. "A new necessary and sufficient global optimality condition for canonical DC problems", Journal of Global Optimization, 55(3), pp. 559–577, 2013.
https://doi.org/10.1007/s10898-012-9908-1

[19] Tuy, H. "Canonical DC programming problem: Outer approximation methods revisited", Operations Research Letters, 18(2), pp. 99–106, 1995.
https://doi.org/10.1016/0167-6377(95)00037-X

[20] Horst, R. "Handbook of Global Optimization", Springer US, 1995.
https://doi.org/10.1007/978-1-4615-2025-2

[21] Nagy, Á., Vajk, I. "Minimum-Time Path Tracking for Robots with Non-Convex Constraints", In: Proceedings of the 15th International Symposium on Intelligent Systems and Informatics (SISY), pp. 163–168, 2017.

[22] Gurobi Optimization, Inc., Gurobi Optimizer Version 7.0.2. [Computer software], 2017.

[23] Corke, P. I. "Robotics, Vision and Control: Fundamental Algorithms in Matlab", Springer, 2011.

[24] Mitsubishi, CR750/CR751 Series Controller Instruction Manual, 2014.

[25] Mitsubishi, RV-3SD/3SDJ/3SDB/3SDBJ Series: Standard Specifications Manual, 2009.