Redundancy Resolution in Minimum-time Path Tracking of Robotic Manipulators

Alexander Reiter, Hubert Gattringer and Andreas Müller
Institute of Robotics, Johannes Kepler University Linz, Altenberger Straße 69, Linz, Austria

Keywords: Robotics, Optimal Control, Trajectory Planning, Redundant Robots, Inverse Kinematics.

Abstract: Minimum-time trajectories for applications where a geometric path is followed by a kinematically redundant robot's end-effector may yield economical improvements in many cases compared to conventional manipulators. While for non-redundant robots the problem of finding such trajectories has been solved, the redundant case has not been treated exhaustively. In this contribution, the problem is split into two interlaced parts: inverse kinematics and trajectory optimization. In a direct optimization approach, the inverse kinematics problem is solved numerically at each time point. Therein, the manupulator's kinematic redundancy is exploited by introducing scaled nullspace basis vectors of the Jacobian of differential velocities. The scaling factors for each time point are decision variables, thus the inverse kinematics is solved optimally w.r.t. the trajectory optimization goal, i.e. minimizing end time. The effectiveness of the presented method is shown by means of the example of a planar 4R manipulator with two redundant degrees of freedom.

1 INTRODUCTION

In industrial applications such as painting, welding or gluing a geometric end-effector path is defined leaving only the problem of finding a suitable time evolution of the joints of the executing robot. Introducing minimum-time trajectories may yield economical advantages as a shorter trajectory duration results in a lower task cycle time. Kinematically redundant manipulators provide favorable properties such as increased workspace dexterity and improved task-specific adaptiveness compared to conventional, non-redundant manipulators. The topic of time optimal trajectory planning has been studied in a large number of publications. While for non-redundant manipulators this problem has been solved, e.g. (Bobrow et al., 1985; Shin and McKay, 1985; Pfeiffer and Johanni, 1986), for redundant robots no satisfying methods have been proposed yet. Concepts of minimum-time trajectory planning for kinematically redundant manipulators can be largely separated into two method families: joint space and workspace-based techniques. Members of the former group assume, that the path following problem is solved as an equality constraint to the trajectory optimization process. Alternatively, a joint space parametrization is assumed to be available (Pham, 2014). In the latter group, methods incorporate inverse kinematics. Due to the mathematical representation of redundant robots' kinematics, solutions are often obtained numerically (Liégeois, 1977). Redundancy allows to augment such solutions by adding objectives such as maximizing performance measures, e.g. directional dynamic manipulability in (Chiacchio, 1990). However, the choice of a performance measure is crucial as it must act as a local proxy for the superseding time minimization. Methods relying on joint space decomposition (Wampler, 1987) offer computation of analytic inverse kinematics for certain manipulator structures. There are various joint space decomposition approaches, some have drawbacks such as the inability to process certain paths, or boundary conditions, c.f. (Ma and Watanabe, 2004). Others rely on diffeomorphisms that may be difficult to obtain, c.f. (Galicki, 2000). Summarizing, the unsolved problem is often a kinematic one, particularly the redundancy resolution in the inverse kinematics problem poses difficulties.

In Section 2 of this paper, the problem of computing minimum-time joint trajectories for tracking a kinematically redundant serial manipulator's prescribed end-effector path is formulated. Section 3 discusses methods to fulfill the path tracking constraints. The main contribution of this paper is presented in Section 4. A method is introduced wherein the path following requirement is treated using an inverse kinematics scheme underlying the trajectory

Reiter, A., Gattringer, H. and Müller, A.
Redundancy Resolution in Minimum-time Path Tracking of Robotic Manipulators.
DOI: 10.5220/0005975800610068
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 2*, pages 61-68
ISBN: 978-989-758-198-4

optimization problem. The numerical inverse kinematics scheme is augmented by optimally scaled nullspace basis vectors of the instantaneous differential velocity Jacobian in order to obtain time-optimal trajectories. This new approach is applied to the example of a planar manipulator with two redundant degrees of freedom in Section 5. Therein, the trajectory planning problem is formulated using direct multiple shooting (Bock and Plitt, 1984) and solved using a modern interior-point method. Unfavorable properties of time integration and appropriate countermeasures are also discussed. Section 6 concludes this contribution and gives insight in possible enhancements to the proposed method.

The algorithms are presented using an acceleration level inverse kinematics for the sake of brevity but can be similarly formulated for higher-order derivatives.

2 PROBLEM DESCRIPTION

2.1 Kinematically Redundant Manipulators

The configuration of a robotic manipulator is uniquely defined using coordinates $q_i, i = 1 \ldots n$ in the configuration space $\mathbb{V}^n$, i.e. $\mathbf{q}^\top = (q_1, \ldots, q_n) \in \mathbb{V}^n$. The end-effector pose $\mathbf{z}_\mathrm{E}$ can be described with its Cartesian position $\mathbf{r}_\mathrm{E} \in \mathbb{R}^3$ and its orientation, denoted by the rotation matrix $\mathbf{R}_\mathrm{E}$, i.e. $\mathbf{z}_\mathrm{E} = (\mathbf{R}_\mathrm{E}, \mathbf{r}_\mathrm{E}) \in \mathrm{SO}(3) \times \mathbb{R}^3$ instead of $\mathrm{SE}(3)$. The forward (or direct) kinematics mapping $f : \mathbb{V}^n \mapsto \mathrm{SO}(3) \times \mathbb{R}^3$ maps joint configurations to end-effector poses.

The robot's workspace W is given as the image of the direct kinematics mapping f, i.e. $W = \{C \in \operatorname{im} f | \mathbf{h}(\mathbf{q}) \leq 0\} \subset \mathrm{SO}(3) \times \mathbb{R}^3$ wherein only geometrically admissible configurations (inequality constraints $\mathbf{h}$) are considered. The workspace dimension is given as $m = \dim W$.

A serial manipulator is considered kinematically redundant if $n > m$, i.e. the configuration space is of higher dimension than the workspace. For non-redundant manipulators (where $n = m$ holds) the inverse mapping of the forward kinematics, f^{-1}, is well-defined, and in special cases even given in closed form, e.g. for standard 6R robots with a spherical wrist. In the case of redundant robots, one has to resort to different, mostly numerical or iterative, methods.

2.2 Optimal Trajectory Planning for Prescribed End-effector Paths

The goal of minimum-time trajectory optimization is to find the shortest possible time evolution of the considered manipulator's joint positions such that a given end-effector path is tracked while being restricted to technological limitations. In the following it is assumed that the path is given by means of a series of desired poses, continuously parametrized with a scalar path parameter $s \in [0,1]$, i.e. $\mathbf{z}_{\mathrm{E,d}}(s) = (\mathbf{R}_{\mathrm{E,d}}(s), \mathbf{r}_{\mathrm{E,d}}(s)) : \mathbb{R} \mapsto \mathrm{SO}(3) \times \mathbb{R}^3$. The index d denotes desired quantities. This yields a non-linear optimization problem (NLP) of the form

$$\min_{\mathbf{x}} \int_0^{t_\mathrm{f}} 1 \, \mathrm{d}t \tag{1}$$

$$\begin{aligned}
\text{s.t.} \quad & \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q} && (2)\\
& \mathbf{q}_\mathrm{min} \leq \mathbf{q} \leq \mathbf{q}_\mathrm{max} && (3)\\
& \dot{\mathbf{q}}_\mathrm{min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_\mathrm{max} && (4)\\
& \ddot{\mathbf{q}}_\mathrm{min} \leq \ddot{\mathbf{q}} \leq \ddot{\mathbf{q}}_\mathrm{max} && (5)\\
& \mathbf{Q}_\mathrm{min} \leq \mathbf{Q} \leq \mathbf{Q}_\mathrm{max} && (6)\\
& 0 \leq s \leq 1 && (7)\\
& s(0) = 0, s(t_\mathrm{f}) = 1 && (8)\\
& \dot{s} \geq 0 && (9)\\
& \mathbf{z}_{\mathrm{E,d}}(s) = f(\mathbf{q}) && (10)\\
& \mathbf{q}(0) = \mathbf{q}_0, \mathbf{q}(t_\mathrm{f}) = \mathbf{q}_\mathrm{f} && (11)\\
& \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0, \dot{\mathbf{q}}(t_\mathrm{f}) = \dot{\mathbf{q}}_\mathrm{f} && (12)
\end{aligned}$$

wherein $\mathbf{x}$ represents the vector of optimization variables describing the time evolution of the joint positions $\mathbf{q}$. Declarations of dependencies of $\mathbf{x}$ will be omitted below. The optimization problem is subjected to the manipulator's (in general non-linear) dynamics denoted as the equations of motion (2) with the vector of minimal coordinates $\mathbf{q}$ and its time derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$. $\mathbf{M}$ is the system's inertia matrix, $\mathbf{g}$ represents non-linear terms in the equations of motion, consisting of the Coriolis, centrifugal, gravitational and dissipative effects. $\mathbf{Q}$ indicates the vector of generalized forces and torques. Limitations of the manipulator's joint positions in (3), joint velocities (4) and possibly higher derivatives such as joint accelerations (5) may also be incorporated. Further bounds are applied to the generalized forces $\mathbf{Q}$ in (6). The desired end-effector path is prescribed using a monotonically increasing (9), bounded (7) path parameter. (10) represents the aforementioned path tracking requirement. Typically, there are also initial and final (11), (12) constraints of the robot's joint positions and their time derivatives. In addition, constraints for cyclic tasks can be formulated as $\mathbf{q}_0 = \mathbf{q}_\mathrm{f}$ and $\dot{\mathbf{q}}_0 = \dot{\mathbf{q}}_\mathrm{f}$.

This optimization problem can be decomposed into the trajectory optimization problem and an underlying path-tracking subproblem.

3 PATH FOLLOWING

In a minimum-time path tracking optimization problem, the path following constraint can be imposed as an equality constraint (10). Other approaches assume a known joint space parametrization to be known, e.g. (Pham, 2014). Alternatively, an inverse kinematics mapping $f^{-1} : W \mapsto \mathbb{V}^{n}$ can be applied to obtain joint quantities from workspace quantities. As mentioned in Section 2.1, no closed-form solution to the inverse kinematics problem exists for kinematically redundant manipulators. However, there are other approaches such as joint space decomposition (Wampler, 1987; Ma and Watanabe, 2004) or Jacobian-based numeric methods (Whitney, 1969; Liégeois, 1977).

3.1 Path Following and Inverse Kinematics

In order to resolve the path following requirement using inverse kinematics, divide-and-conquer as well as unite-and-conquer methods can be applied. Joint space decomposition can be used as a divide-and-conquer type approach. Therein, a manipulator's structure has to be explicitly separated into two or more parts. Then the inverse kinematics problem can be solved based on a loop closure condition. Joint space decomposition makes direct use of kinematic redundancy as operations are performed on joint level. The choice of decomposition may not be straight-forward and thus a result of an superseding integer program. The inverse kinematics solution can be performed analytically only in cases with suitable geometry but not in general. Also, the enforcement of the aforementioned loop closure condition is non-trivial.

In unite-and-conquer methods such as differential inverse kinematics, firstly introduced in (Whitney, 1969), a least-squares solution (w.r.t. an end-effector error quantity) yields all joint quantities at once. However, this family of methods needs to be augmented in order to exploit kinematical redundancy. In this paper, the latter type of inverse kinematics methods is used.

The derivations of differential inverse kinematics schemes below are well-known but reproduced here as an introduction to and a motivation for the main contribution of this paper presented in Section 4.

The most simple case is first-order differential inverse kinematics,

$$\dot{\mathbf{r}}_{\mathrm{E}} = \mathbf{J}\left(\mathbf{q}\right)\dot{\mathbf{q}}, \tag{13}$$

wherein $\mathbf{J} = \frac{\partial \dot{\mathbf{r}}_{\mathrm{E}}}{\partial \dot{\mathbf{q}}} \in \mathbb{R}^{m\times n}$ denotes the forward kinematics Jacobian, a non-square, wide matrix. Thus it is not invertible, but an approximate solution for the joint velocities $\dot{\mathbf{q}}$ can be computed minimizing the error in the least-squares sense, i.e.

$$\dot{\mathbf{q}} = \mathbf{J}^{+}\dot{\mathbf{r}}_{\mathrm{E,d}} \tag{14}$$

wherein $\mathbf{J}^{+} = \mathbf{J}^{\top}\left(\mathbf{J}\mathbf{J}^{\top}\right)^{-1}$ denotes the right Moore-Penrose pseudoinverse. Alternatively, the dynamically consistent pseudoinverse $\mathbf{J}^{+}_{\mathbf{M}} = \mathbf{M}^{-1}\mathbf{J}^{\top}\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\top}\right)^{-1}$ can be used (Khatib, 1988). Compared to (13), for (14) the index d was added to the end-effector velocity as it is now a given, desired quantity. For computing the matrix inverse of $\left(\mathbf{J}\mathbf{J}^{\top}\right)$ in singular configurations, a regularization term can be introduced, i.e. $\mathbf{J}^{+} = \mathbf{J}^{\top}\left(\mathbf{J}\mathbf{J}^{\top} + \kappa\mathbf{I}\right)^{-1}$ with a small $\kappa > 0$. In general, non-singular configurations the nullspace of $\mathbf{J}$ has dimension $n - m > 0$, i.e. the manipulator is capable of internal motion that does not affect the end-effector motion. This property can be exploited using an inverse kinematics scheme (Liégeois, 1977) that is augmented to pursue additional goals. Scalar performance measures w such as kinematic manipulability (Yoshikawa, 1985b) or dynamic (Yoshikawa, 1985a) manipulability can be maximized by adding a velocity term to (14). This velocity points in the direction of $\mathbf{v} = \frac{\partial w}{\partial \mathbf{q}}$ and is projected into the nullspace of the Jacobian, i.e.

$$\dot{\mathbf{q}} = \mathbf{J}^{+}\dot{\mathbf{r}}_{\mathrm{E,d}} + \mathbf{N}\mathbf{v} \tag{15}$$

with the nullspace projector $\mathbf{N} = \left(\mathbf{I} - \mathbf{J}\mathbf{J}^{+}\right)$. $\mathbf{I}$ denotes the identity matrix of appropriate size. Substituting (15) in (13) shows that no end-effector motion results from the additional term. Pose-dependent performance measures such as kinematic or dynamic manipulability suffer from the fact that they only represent a local, instantaneous property. As a result, they can hardly be exploited in the course of an superseding trajectory optimization problem minimizing a global property such as a trajectory time.

Similar inverse kinematics approaches can be set up for higher time derivatives simply by deriving (13) w.r.t. time and isolating the highest time derivative of the joint positions $\mathbf{q}$, e.g. an acceleration-level approach yields

$$\ddot{\mathbf{q}} = \mathbf{J}^{+}\left(\ddot{\mathbf{r}}_{\mathrm{E,d}} - \dot{\mathbf{J}}\dot{\mathbf{q}}\right) + \mathbf{N}\mathbf{v} \tag{16}$$

wherein $\mathbf{v}$ can again represent a performance measure gradient projected into the Jacobian nullspace.

3.2 Numerical Inverse Kinematics in Trajectory Planning

In trajectory planning tasks, constraints regarding derivatives of the joint positions $\mathbf{q}$ may be imposed, e.g. zero joint velocities at the end-effector final position. By close examination of (16), it can be found that such a constraint is not necessarily fulfilled as a vector $\mathbf{v}$ may not have the appropriate magnitude to cancel all internal accelerations. Thus, a scaling factor γ needs to be introduced to fulfill such requirements, i.e. for the case of an acceleration-level approach

$$\ddot{\mathbf{q}} = \mathbf{J}^{+}\left(\ddot{\mathbf{r}}_{\mathrm{E,d}} - \dot{\mathbf{J}}\dot{\mathbf{q}}\right) + \gamma\mathbf{N}\mathbf{v}. \tag{17}$$

The nullspace scaling factor γ introduced in (17) needs to vary over time in order to stick to multiple constraints across the trajectory, i.e. $\gamma = \gamma(t)$.

4 NULLSPACE BASIS SCALING

Using the above methods, the joint state is changed such that a performance measure w is maximized locally, i.e. following the instantaneous gradient $\frac{\partial w}{\partial \mathbf{q}}$, projected into the current Jacobian nullspace. Even if the step size γ is adjusted properly, this may not yield an optimal joint state evolution across the path. If a manipulator provides more than one redundant degree of freedom, the projection of the gradient will always lie in a subspace of the nullspace. To make use of remaining free nullspace directions, task priority-based methods (Nakamura et al., 1987) can be used to pursue additional (ideally non-conflicting) goals with lower priorities.

The key idea of the present approach is to compute a basis for the Jacobian nullspace, i.e. $\ker \mathbf{J} = \operatorname{span}\{\mathbf{a}_i\}, i = 1, \ldots, (n-m)$. The basis vectors $\mathbf{a}_i$ are then scaled by factors $\gamma_i, i = 1, \ldots, (n-m)$ and added to the inverse kinematics solution (17), i.e.

$$\ddot{\mathbf{q}} = \mathbf{J}^{+}\left(\ddot{\mathbf{r}}_{\mathrm{E,d}} - \dot{\mathbf{J}}\dot{\mathbf{q}}\right) + \sum_{i=1}^{n-m} \gamma_i(t)\,\mathbf{a}_i. \tag{18}$$

For manipulators with a kinematic redundancy of $n - m = 1$, there is only one basis vector of the nullspace. Thus, redundancy is fully exploited by both approaches, performance measure-based methods and nullspace basis scaling. However, for higher degrees of redundancy $n - m > 1$, exploiting the full nullspace as in (18) enables an superseding optimization process to directly modify the joint trajectories according to the criteria to be minimized. In contrast to other approaches, there is no need of a projected performance measure gradient acting as a proxy function to pursue the optimization goals. As for both, performance measure gradients and the nullspace basis vectors, symbolic expressions can be obtained beforehand by means of computer algebra systems, the difference in computational cost is negligible.

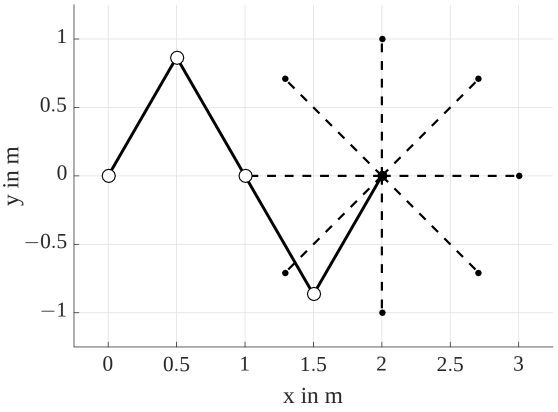

Figure 1: Planar manipulator with four revolute joints in initial configuration.

Section 5 shows that this approach can be easily applied to problems with readily available kinematic and dynamical models.

In the method development (13) to (18), only position coordinates were treated as workspace coordinates for simplicity. Adding a prescribed end-effector orientation increases the complexity of the problem. The differential inverse kinematics needs also to be computed for the end-effector's angular velocity or its time derivatives. Care has to be taken in order to establish consistency in the physical units of products of the Jacobian.

5 EXAMPLE

5.1 Kinematic and Dynamic Model

The method presented in Section 4 is illustrated using the simple example of the planar manipulator depicted in Figure 1, moving along straight line paths. The robot consists of four revolute joints (4R), its links have masses $m_{\mathrm{L},i}$ and moments of inertia $C_{\mathrm{L},i}$ (about their respective centers of mass). The joints are directly actuated by means of motors without mass and inertia. Damping coefficients d_i account for joint friction. The system is not influenced by gravity. Numerical values used for the simulation can be obtained from Table 1.

A minimal coordinate representation of the manipulator's configuration yields $\mathbf{q}^{\top} = (q_1, q_2, q_3, q_4) \in$

Table 1: Parameters of planar manipulator.

symbol	description	value
$m_{\mathrm{L},i}$	link mass	10 kg
l_i	link length	1 m
$C_{\mathrm{L},i}$	link moment of inertia	$m_{\mathrm{L},i} l_i^2/12$
d_i	damping coefficient	0.1 Nm/rad
$q_{i,\max/\min}$	joint position limits	$\pm\pi$ rad
$\dot{q}_{i,\max/\min}$	joint velocity limits	± 2 rad/s
$M_{i,\max/\min}$	joint torque limits	± 10 Nm

$V = S^4$ wherein S represents the 0-sphere of each joint's admissible range of positions $(q_{i,\min}, q_{i,\max})$. The forward kinematics mapping can be easily derived by hand and is given by f. The equations of motion $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}$ can be derived using well-known techniques such as the LAGRANGE formalism or the Projection Equation (Bremer, 1988). The vector of generalized forces $\mathbf{Q}^\top = (M_1, M_2, M_3, M_4)$ consists of the motor torques.

In this example, the end-effector position, but not its orientation is considered as workspace coordinates, i.e. $\mathbf{r}_{\mathrm{E}}^\top = (x, y) \in W = \{C \in \operatorname{im} f | \mathbf{q} \in V\} \subset \mathbb{R}^2$. Thus, the degree of kinematic redundancy is $n - m = 2$.

5.2 Inverse Kinematics

Applying (18) to the present example yields the inverse kinematics law

$$\ddot{\mathbf{q}} = \mathbf{J}^+ \left(\ddot{\mathbf{r}}_{\mathrm{E,d}} - \dot{\mathbf{J}}\dot{\mathbf{q}}\right) + \sum_{i=1}^{2} \gamma_i(t)\,\mathbf{a}_i. \tag{19}$$

The inverse kinematics laws (19) and above have only instantaneous, point-wise characteristics. In order to obtain joint trajectories for a given end-effector path, (19) has to be evaluated for all $\ddot{\mathbf{r}}_{\mathrm{E,d}}$ along the path and time integrated using numerical methods to obtain lower time derivatives, i.e.

$$\ddot{\mathbf{q}} \quad \rightarrow \quad \hat{\dot{\mathbf{q}}} = \int_0^{t_\mathrm{f}} \ddot{\mathbf{q}}\,\mathrm{d}t \quad \rightarrow \quad \hat{\mathbf{q}} = \int_0^{t_\mathrm{f}} \hat{\dot{\mathbf{q}}}\,\mathrm{d}t. \tag{20}$$

Numerical time integration introduces workspace drift errors $\mathbf{e} = \mathbf{r}_{\mathrm{E,d}} - f(\hat{\mathbf{q}})$, $\dot{\mathbf{e}} = \dot{\mathbf{r}}_{\mathrm{E,d}} - \mathbf{J}(\hat{\mathbf{q}})\,\hat{\dot{\mathbf{q}}}$. This issue can be avoided by adding stabilizing terms to (19) such that $\ddot{\mathbf{e}} + \mathbf{K}_1\dot{\mathbf{e}} + \mathbf{K}_0\mathbf{e} = \mathbf{0}$. Re-writing the error dynamics in terms of single-order ordinary differential equations by $\mathbf{e}_1 = \mathbf{e}$ and $\mathbf{e}_2 = \dot{\mathbf{e}}$ yields

$$\begin{pmatrix} \dot{\mathbf{e}}_1 \\ \dot{\mathbf{e}}_2 \end{pmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_0 & -\mathbf{K}_1 \end{bmatrix} \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \tag{21}$$

whose structure can be exploited for pole-placement. Incorporating the error dynamics scheme (21) into (19) yields

$$\ddot{\mathbf{q}} = \mathbf{J}^+ \left(\ddot{\mathbf{r}}_{\mathrm{E,d}} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{K}_1\dot{\mathbf{e}} + \mathbf{K}_0\mathbf{e}\right) + \sum_{i=1}^{2} \gamma_i(t)\,\mathbf{a}_i. \tag{22}$$

In the pseudoinverse $\mathbf{J}^+ = \mathbf{J}^\top \left(\mathbf{J}\mathbf{J}^\top + \kappa\mathbf{I}\right)^{-1}$ regularization is conducted with $\kappa = 10^{-10}$. The matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ were chosen such that all poles of the error dynamics are at -2. In this simple case, the nullspace basis vectors $\mathbf{a}_i \in \ker \mathbf{J}(\mathbf{q})$ can be computed analytically as functions of the system's parameters and the current joint configuration.

5.3 Task

For this example, the robot's task is to move its end-effector along straight line paths

$$\mathbf{r}_{\mathrm{E,d}} = \mathbf{r}_0 + sL \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} \tag{23}$$

of length $L = 1$ m wherein $s \in [0, 1]$ denotes the path parameter. The task is to be performed for slopes $\varphi = 0, \pi/4, \pi/2, \ldots, 7\pi/4$, c.f. Figure 1. At the initial point $\mathbf{r}_0^\top = (2, 0)$ m the manipulator's configuration is chosen to be $\overline{\mathbf{q}}_0^\top = (\pi/3, -2\pi/3, 0, 2\pi/3)$ rad. The task is performed as a minimum-time rest-to-rest maneuver, i.e. $\dot{\mathbf{q}}(t = 0) = \dot{\mathbf{q}}(t = t_\mathrm{f}) = \mathbf{0}$.

5.4 Direct Multiple Shooting Trajectory Optimization

In this section, the optimization problem posed in Section 2.2 is reformulated incorporating the specifics of the used manipulator from Section 5.1, its inverse kinematics scheme from Section 5.2 and the task defined in Section 5.3. Using direct multiple shooting, originally developed in (Bock and Plitt, 1984), the time domain is discretized into N uniform intervals and scaled with the final trajectory time t_f as a decision variable. In each interval $[t_k, t_{k+1}]$ the system's state is integrated using a fourth-order explicit Runge-Kutta scheme, denoted as the function $\mathbf{f}(\mathbf{x}_k(t_\mathrm{f}), \mathbf{u}_k)$. Further declaration of dependencies of t_f are omitted for the sake of brevity. Therein the optimization system's state consists of the path parameter and its time derivative as well as the manipulator's joint positions and velocities, i.e. $\mathbf{x}_k^\top = \left(s_k, \dot{s}_k, \mathbf{q}_k^\top, \dot{\mathbf{q}}_k^\top\right)$. $\mathbf{u}_k^\top = (\ddot{s}_k, \gamma_{k,1}, \gamma_{k,2})$ represents the control input of the optimization problem as piecewise constant functions.

The vector of decision variables $\mathbf{x}$ consists of concatenations of the intermediate states $\mathbf{x}_k, k = 0, \ldots, N$ and the controls $\mathbf{u}_k, k = 0, \ldots, N-1$, as well as of the trajectory's final time t_f, i.e.

$$\mathbf{x}^\top = \left(\mathbf{x}_0^\top, \mathbf{u}_0^\top, \ldots, \mathbf{x}_{N-1}^\top, \mathbf{u}_{N-1}^\top, \mathbf{x}_N^\top, t_\mathrm{f}\right). \tag{24}$$

The NLP posed in Section 2.2 can be thus reformulated as

$$\min_{\mathbf{x}} t_\mathrm{f} \tag{25}$$

$$
\begin{aligned}
\text{s.t.}\quad & 0 \le s_k \le 1 && (26)\\
& \dot{s}_k \ge 0 && (27)\\
& \mathbf{q}_\mathrm{min} \le \mathbf{q}_k \le \mathbf{q}_\mathrm{max} && (28)\\
& \dot{\mathbf{q}}_\mathrm{min} \le \dot{\mathbf{q}}_k \le \dot{\mathbf{q}}_\mathrm{max} && (29)\\
& s_0 = 0, s_N = 1 && (30)\\
& \dot{s}_0 = \dot{s}_N = 0 && (31)\\
& \mathbf{q}_0 = \overline{\mathbf{q}}_0 && (32)\\
& \dot{\mathbf{q}}_0 = \dot{\mathbf{q}}_N = \mathbf{0} && (33)\\
& \ddot{\mathbf{q}}_k = \mathbf{J}_k^+ \left(\ddot{\mathbf{r}}_{\mathrm{E,d}}(s_k) - \dot{\mathbf{J}}_k \dot{\mathbf{q}}_k + \mathbf{K}_1 \dot{\mathbf{e}}_k \right.\\
& \qquad \left. + \mathbf{K}_0 \mathbf{e}_k \right) + \sum_{i=1}^{2} \gamma_{i,k} \mathbf{a}_{i,k} && (34)\\
& \mathbf{Q}_\mathrm{min} \le \mathbf{M}(\mathbf{q}_k) \ddot{\mathbf{q}}_k + \mathbf{g}(\mathbf{q}_k, \dot{\mathbf{q}}_k) \le \mathbf{Q}_\mathrm{max} && (35)\\
& \mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0} && (36)
\end{aligned}
$$

wherein $\ker \mathbf{J}_k = \operatorname{span}\{\mathbf{a}_{i,k}\}$. The state is constrained by (26) to (29). There are also initial and final conditions of the state, (30) to (33), but the final joint positions $\mathbf{q}_N$ are free and obtained as a result of the NLP. Equation (34) describes the inverse kinematics resolution law used in the state integration and the computation of the inverse dynamics constrained by (35). As the NLP is implemented as direct multiple shooting, (36) is required to close the state integration gaps between adjacent shooting intervals.

For direct multiple shooting, the structure of the Jacobian matrix of the (in)equality constraints w.r.t. the decision variables $\mathbf{x}$ is blockdiagonal if the ordering of $\mathbf{x}_k$ and $\mathbf{u}_k$ is as described in (24). Block diagonal matrices enable efficient solution algorithms to be applied. However, in this special case where the final time t_f is also a decision variable, an additional dense column reduces the matrix sparsity as all constraints depend on this variable. The corresponding sparsity pattern for $N = 5$ is depicted in Figure 2.

As this NLP is non-convex, globally optimal solutions cannot be guaranteed. Solutions obtained from this problem depend on the initial guess. For the initial guess of the path parameter s_k, a linear time evolution was assumed, resulting in a constant velocity profile for $\dot{s}_k$. For these s_k and $\dot{s}_k$, an initial guess for the time evolution of the joint positions $\mathbf{q}_k$ is obtained using an numerical inverse kinematics scheme on velocity level. The controls $\mathbf{u}_k$ are initialized with zeros.

For efficient numerical solution the NLP was implemented using MATLAB interface to the optimization framework Casadi 3.0rc3 (Andersson, 2013) and solved with Ipopt 3.12.3 (HSL MA27 for linear subproblems).

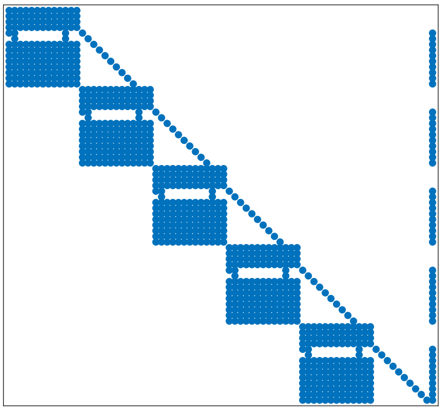

Figure 2: Sparsity pattern of the constraint Jacobian for $N = 5$ uniform time intervals.

The trajectories obtained by the method described above are continuously differentiable once w.r.t. time, i.e. $\mathbf{q}(t) \in \mathcal{C}^1$ as a second-order inverse kinematics resolution scheme with piecewise constant inputs $\mathbf{u}_k$ was used. The method can be easily generalized to higher levels of continuity by simply deriving the inverse kinematics scheme (22) and adding further states to the NLP. This is useful for system that require continuous torques ($\mathbf{q}(t) \in \mathcal{C}^2$), especially differentially flat elastic systems ($\mathbf{q}(t) \in \mathcal{C}^4$), c.f. (Springer et al., 2013).

5.5 Results

The optimization problem posed in the previous section was solved with a time discretization of $N = 100$ uniform intervals. As an example, the obtained trajectories for $\varphi = \pi/2$ are shown in Figure 3 for the path parameter, and in Figure 4 and Figure 5 for the joint positions and velocities, respectively. The results satisfy the constraints posed in (26) to (36) and provide a minimum for t_f. However, this minimum is only local as the optimization problem is non-convex. The resulting joint torques are depicted in Figure 6. It can be seen that at all times, at least two of the constrained joint velocities or joint torques are saturated, which is one of the characteristics of a minimum-time property of the obtained trajectories. Corresponding snapshots of the manipulator's optimal motion are depicted in Figure 7.

Iteration counts, computation times obtained using an INTEL XEON E3-1246 V3 processor as well as the resulting trajectory end times for all slopes φ can be obtained from Table 2.

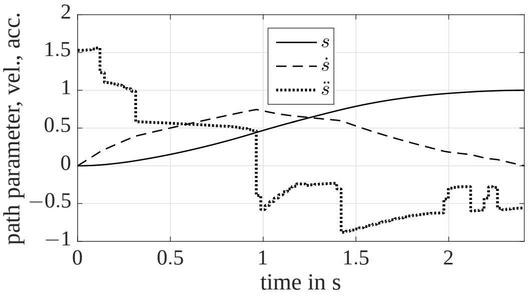

Figure 3: Path parameter s, velocity $\dot{s}$, acceleration $\ddot{s}$ ($\varphi = \pi/2$).

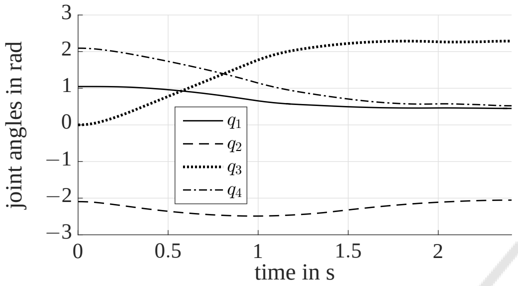

Figure 4: Joint positions $\mathbf{q}$ ($\varphi = \pi/2$).

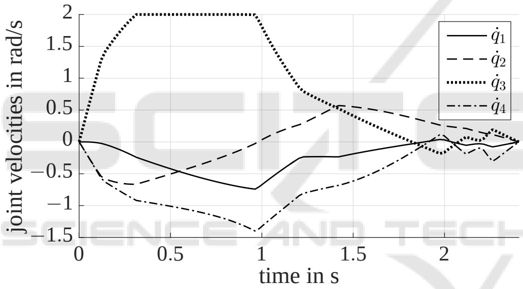

Figure 5: Joint velocities $\dot{\mathbf{q}}$ ($\varphi = \pi/2$).

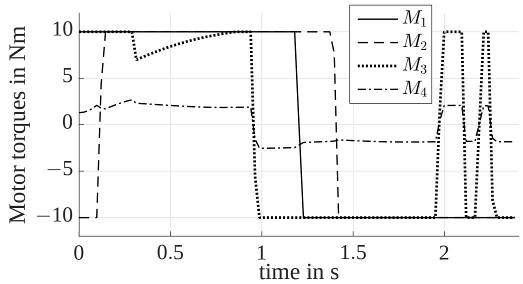

Figure 6: Motor torques $\mathbf{Q}$ ($\varphi = \pi/2$).

6 CONCLUSION

This study has presented a contribution to the solution of the time-optimal path following problem for kinematically redundant manipulators. In this approach, the problem is divided into the trajectory optimization and an underlying inverse kinematics problem.

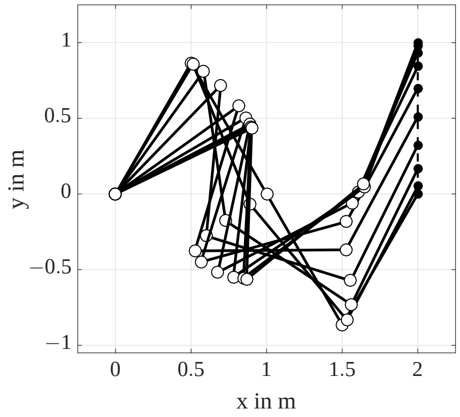

Figure 7: Time evolution along the path with $\varphi = \pi/2$ (10 snapshots equally distributed in time).

Table 2: Optimization results for all φ.

φ	t_f in s	#it.	CPU time in s
0	2.1262	206	29
$\pi/4$	2.4746	290	41
$\pi/2$	2.4073	206	30
$3\pi/4$	1.7304	213	30
π	1.7583	157	22
$5\pi/4$	2.5625	222	31
$3\pi/2$	1.5966	239	34
$7\pi/4$	1.1505	134	19

The former is solved using a numerical computation scheme, augmented to fully exploit redundancy in an optimal way such that the latter problem yields optimal results. It was discussed that this method is valuable for robots with multiple redundant degrees of freedom.

The method was successfully applied to a planar manipulator with two redundant joints moving its end-effector along prescribed straight line paths.

In future work, the method proposed in this paper will be applied to more complex, spatial examples also incorporating prescribed orientations. Regarding the implementation, the multiple shooting method from Section 5.4 can be refined to use non-uniform shooting intervals to be included as decision variables instead of varying t_f and using uniform time intervals. This would allow for local adjustment of the time resolution and will yield a purely blockdiagonal structure in the Jacobian of constraints, c.f. Figure 2.

ACKNOWLEDGEMENTS

This work has been supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria.

REFERENCES

Andersson, J. (2013). *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven.

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

Bock, H. G. and Plitt, K. J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the IFAC World Congress*, pages 242–247. Pergamon Press.

Bremer, H. (1988). *Dynamik und Regelung mechanischer Systeme*. Teubner Studienbücher.

Chiacchio, P. (1990). Exploiting redundancy in minimum-time path following robot control. In *Proc. American Control Conf.*, pages 2313–2318.

Galicki, M. (2000). Time-optimal controls of kinematically redundant manipulators with geometric constraints. *IEEE Transactions on Robotics and Automation*, 16(1):89–93.

Khatib, O. (1988). Augmented object and reduced effective inertia in robot systems. In *American Control Conference 1988*, pages 2140 – 2147. IEEE.

Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems Man and Cybernetics*, 12:868–871.

Ma, S. and Watanabe, M. (2004). Time optimal path-tracking control of kinematically redundant manipulators. *JSME International Journal*, 47(2):582–590.

Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *International Journal of Robotics Research*, 6(2):3–15.

Pfeiffer, F. and Johanni, R. (1986). A concept for manipulator trajectory planning. In *International Conference on Robotics and Automation*, pages 1399 – 1405. IEEE.

Pham, Q.-C. (2014). A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Rob*, 30(6):1533–1540.

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

Springer, K., Gattringer, H., and Staufer, P. (2013). On time-optimal trajectory planning for a flexible link robot. *Journal of Systems and Control Engineering*, 227(10):751–762.

Wampler, C. (1987). Inverse kinematic functions for redundant manipulators. In *IEEE International Conference on Robotics and Automation*, pages 610 – 617.

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, 10(2):47 – 53.

Yoshikawa, T. (1985a). Dynamic manipulability of robot manipulators. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1033–1038.

Yoshikawa, T. (1985b). Manipulability of robotic mechanisms. *International Journal of Robotics Research*, 4(2):3–9.