Spline Deformation of Locally Optimal Trajectories: Feasibility and Upper Bound on Control Inputs

Alexander Pekarovskiy, Thomas Nierhoff, Sandra Hirche and Martin Buss

Abstract— **Deformation of optimal trajectories has a great potential in various applications due to the ability of real-time recomputation of the overall trajectory when applying new boundary conditions. This paper presents a novel approach where optimal trajectories are created offline through numerical direct optimal control methods. Afterwards the trajectories are deformed online with a spline deformation approach, providing minimum acceleration deviation between optimal and deformed trajectories and considerably reducing the computational complexity of the algorithm during run time. A feasibility check based on upper bounds for the deformed trajectory, the controller tracking error and the resulting torque is provided. This guarantees correct task execution in the presence of bounded disturbances and unmodeled dynamics.**

I. INTRODUCTION

Existing numerical optimal control methods recalculate the entire optimal control problem (OCP) for every new set of boundary conditions [1], [2]. Direct optimal control methods provide the capability of solving numerical OCPs by discretizing both states and controls while simultaneously solving a finite set of nonlinear optimization problems (NLP). In contrast to indirect methods [1], they do not rely on the determination of first-order necessary conditions and the solution of a Hamiltonian boundary value problem, i.e. direct methods are less sensitive to boundary conditions. Solutions of direct methods readily incorporate constraints for a successful task execution, but are limited as they require a good initial guess in order to converge to a feasible solution and are too slow to find an optimal trajectory online.

Another approach is presented in [3] where a least-squares based approach named Laplacian Trajectory Editing (LTE) minimizes the acceleration deviation to a given reference path. When given feasible boundary points, the approach is able to retarget the entire trajectory online. As the method does not rely on any kernel/internal state, its resolution is solely limited by the number of trajectory sampling points.

In [4] the combination of a numerical optimal control method and LTE-based deformation is presented. This approach is experimentally verified using a robotic arm for constrained dynamic manipulation tasks. However, the paper does not provide theoretical bounds on control inputs with respect to the deformed trajectories which may lead to violation of constraints and torque saturation.

The design of trajectory following controllers plays a crucial role in robotics. Their robustness and trajectory feasibility depend on the boundedness of the system matrices and disturbance, the chosen control scheme and knowledge about the desired trajectory. Uniform boundedness and positive definiteness of the inertia matrix are necessary when designing adaptive controllers and ensuring global Lyapunov stability for various control laws [5], [6]. A uniform bounds analysis for the Coriolis/Centripetal matrix is provided in [7]. Kiefer *et al.* [8] proposed to use modeling errors as a joint acceleration disturbance to track a near time-optimal trajectory with prescribed accuracy by modification of controller gains.

A variety of path-tracking controllers dealing with kinematic and dynamic constraints are proposed in literature. Bobrow *et al.* [9] introduced the time-optimal path parametrization (TOPP) idea with a bang-bang controller and a switching law that depends on a maximum velocity curve to fulfil torque constraints. The approach provides both time optimal solutions and algorithm for online computation. However, there are some limitations such as admissible accelerations which reduce the maximum acceleration capabilities of the robot, especially when the motors have different power and thus different maximum torques. The method requires ideal knowledge of the robot dynamics. Shin *et al.* [10] introduced admissible regions and investigated at most quadratic torque-velocity dependence for non-simply connected admissible regions. Other works [10], [11] analyzed constraints on torque and its derivative and proposed time-scaling filters to modify the velocity profile accordingly. The major difference between the methods is that for TOPP the path is fixed and the time is optimized, whereas for LTE-based deformation the time is fixed and the trajectory is optimized.

The contribution of this paper is a feasibility verification for online deformation of previously calculated optimal trajectories. If the amount of deformation is small, the optimality properties related to the acceleration profile of original optimal trajectory in task space will be preserved in the least squares sense. As an example, curvature properties of the deformed motion are directly influenced. It is shown that for a specific type of spline deformation, it is possible to derive

The research leading to these results has received funding partly from the European Research Council under the European Union’s Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. [267877], partly from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement n. [643433], project “Robotic Assistant for MCI Patients at home (RAMCIP)“ and partly from the Technische Universität München - Institute for Advanced Study (www.tum-ias.de), funded by the German Excellence Initiative.

A. Pekarovskiy and M. Buss are with the Chair of Automatic Control Engineering and the TUM Institute for Advanced Study, Technische Universität München, Lichtenbergstr. 2a, 85748 Garching, Germany
{a.pekarovskiy, mb}@tum.de

T. Nierhoff and S. Hirche are with the Chair of Information-oriented Control, Technische Universität München, D-80333 Munich, Germany
{tn, hirche}@tum.de

tight boundaries of the deformed trajectory, making the approach an ideal choice for motion adaptation in constrained environments. Maximum norms on the deformed trajectory, feedback controller and the maximum possible torque are provided in a segment-wise manner.

The remainder of this paper is as follows: Sec. II provides detailed problem formulation. Sec. III discusses the OCP formulation. Sec. IV then describes the trajectory deformation process. Sec. V analyses the feasibility of trajectory tracking for kinematic and dynamic constraints. Finally, Sec. VI discusses the presented approach and suggests ideas for further expansion.

II. Problem formulation

The dynamic model of the robot is described by

$$\boldsymbol{\tau}_r = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}), \qquad (1)$$
$$\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}),$$

with $\boldsymbol{\tau}_r$ as a torque vector correspondent to a joint motion $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$. Given system matrices of a robot model $\mathbf{M}$, $\mathbf{C}$, $\mathbf{G}$ and $\mathbf{F}$, computed torque and outer loop PD controller scheme with $\mathbf{k}_p$, $\mathbf{k}_v$ gains, maximum torque disturbance $\boldsymbol{\tau}_d$, original optimal trajectory $\mathbf{x}_o$, $\dot{\mathbf{x}}_o$, $\boldsymbol{\tau}_o$ and a resulting deformed trajectory $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$, $\ddot{\mathbf{x}}_d$ satisfying given boundary conditions, the goal is to find out whether the corresponding upper bounds of the actuator torques $\|\boldsymbol{\tau}\|_\infty$ are feasible at every time step. It is done by performing feasibility check not along the entire trajectory but only at some specific segmentation points of the deformed trajectory, as in [4]. The problem is reformulated as

$$\|\boldsymbol{\tau}\|_\infty = f(\mathbf{M}, \mathbf{C}, \mathbf{G}, \mathbf{F}, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, \mathbf{k}_p, \mathbf{k}_v, \boldsymbol{\tau}_d, \mathbf{x}_o, \dot{\mathbf{x}}_o, \boldsymbol{\tau}_o), \quad (2)$$

where $\|\boldsymbol{\tau}\|_\infty$ has to be compared with maximum input motor torque τ_{bound}.

As an illustration, you can imagine a fully actuated 3-DoF planar robot moving in a 3-dimensional task space $\mathbf{x} = \begin{bmatrix} x & y & \phi \end{bmatrix}$, see Fig. 1. For each new motion, initial and end boundary points of the trajectory are assigned based on the task goal. After the end boundary points are determined in terms of task space position and velocity, a trajectory is calculated through spline deformation that moves the robot from its initial point to the end point. Here upper bounds on the joint torques and task space kinematic parameters $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$, $\ddot{\mathbf{x}}_d$ have to be provided to guarantee safe trajectory execution.

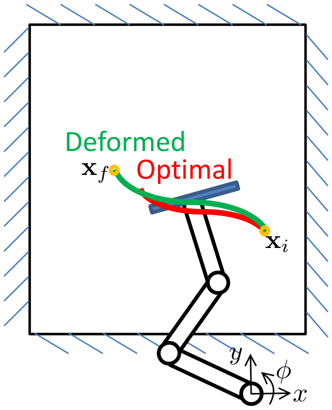

Fig. 1: Robotic manipulator operates within constrained task space or in presence of obstacles. The optimal trajectory is deformed using LTE with the new start and end point.

III. Optimal Trajectory Generation with Direct Collocation

For the presented approach all original optimal trajectories are first precomputed offline using a Direct Collocation method DirCol [2], resulting in piece-wise polynomial functions for states and controls. Even if the obtained results of numerical optimal control methods are in general only locally optimal, we will call them optimal for brevity for the remainder of this paper.

Most optimal control methods including DirCol [12] cannot guarantee a feasible solution for general nonlinear problems. And even if a solution is found, it still takes a considerable amount of time to calculate it. This time depends to a large extent on the complexity of the system, its constraints and tolerances. We gain advantage by precomputing several optimal trajectories for various initial and final points as well as different costs offline. These costs can represent basically anything: minimum time, energy, etc. What really matters afterwards, during the deformation procedure, is the acceleration profile of the optimal trajectory that possesses essential information for the dynamic task.

For example, we set an OCP in (3) for a robotic task with the robot dynamics as in [13] and choose an objective to provide the desired motion behavior. It is important to notice that the state space of the OCP consists of the task space coordinates.

$$\begin{aligned}
\underset{\mathbf{x_q}, \boldsymbol{\tau_o}, t_f}{\text{minimize}} \quad & J(\mathbf{x_q}, \boldsymbol{\tau_o}, t_f) \\
\text{subject to} \quad & \dot{\mathbf{x}}_\mathbf{q} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x_q} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}(\mathbf{x_q})^{-1} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \mathbf{0} \\ \mathbf{N}(\mathbf{x_q}) \end{bmatrix} \\
& h(\mathbf{x_q}(0, t_f), \boldsymbol{\tau_o}, t_f) = 0 \qquad (3) \\
& g(\mathbf{x_q}, \boldsymbol{\tau_o}, t) > 0
\end{aligned}$$

with

$$\mathbf{N}(\mathbf{x_q}) = \mathbf{M}(\mathbf{x_q})^{-1}(\mathbf{C}(\mathbf{x_q}) + \mathbf{G}(\mathbf{x_q}) + \mathbf{F}(\mathbf{x_q})), \qquad (4)$$

where $h(\mathbf{x_q}(0, t_f), \boldsymbol{\tau_o}, t_f)$ represents the initial and final states as equality constraints, $g(\mathbf{x_q}, \boldsymbol{\tau_o}, t) > 0$ incorporates kinematic and dynamic inequality constraints, $\mathbf{x_q} = (\mathbf{x_o}, \dot{\mathbf{x}}_\mathbf{o})^T$, $\mathbf{x_o} = \begin{bmatrix} x & y & \phi \end{bmatrix}$ is the state vector, $\mathbf{f} = \mathbf{J}^{-T}\boldsymbol{\tau_o}$ denotes the task space forces, $\mathbf{J}$ is the Jacobian matrix, $\mathbf{M}$ is the inertia matrix, $\mathbf{C}$ is the Coriolis and centrifugal forces, $\mathbf{G}$ accounts for all gravitational terms, and $\mathbf{F}$ is the Coulomb and viscous frictional force.

For this method, time is discretized as $0 = t_1^o < t_2^o ... < t_n^o = t_f$. The controls are approximated by piecewise linear interpolating functions between t_j^o and t_{j+1}^o as

$$\hat{u} = u(t_j^o) + \frac{t - t_j^o}{\Delta t_j^o}(u(t_{j+1}^o) - u(t_j^o)),$$
$$j \in [1, ..., n-1], t_j^o < t < t_{j+1}^o, \qquad (5)$$

where $\Delta t_j^o = t_{j+1}^o - t_j^o$. At the same time, the states $\hat{x}(t_j^o)$ for each segment are approximated with piece-wise cubic polynomials

$$\hat{x}(t_j^o) = \sum_{k=0}^{3} a_{j,k} \left(\frac{t - t_j^o}{\Delta t_j^o} \right)^k ,$$
$$j \in [1, ..., n-1], t_j^o < t < t_{j+1}^o, \tag{6}$$

using Hermite interpolation

$$\begin{aligned} a_{j,0} &= x(t_j^o), a_{j,1} = \Delta t_j^o f_j, \\ a_{j,2} &= -3x(t_j^o) - 2\Delta t_j^o f_j + 3x(t_{j+1}^0) - \Delta t_j^o f_{j+1}, \\ a_{j,3} &= 2x(t_j^o) + \Delta t_j^o f_j - 2x(t_{j+1}^o) + \Delta t_j^o f_{j+1}, \end{aligned} \tag{7}$$

where $f_j = f(x(t_j^o), u(t_j^0), t_j^o)$. The approximating functions of the states must satisfy the constraints at the grid points t_j^o and at the centers of the discretization interval. The chosen approximations of $x(t)$ already satisfy these constraints at t_j^o. The optimal control problem is split in a set of NLP problems with a number of collocation constraints, inequality constraints at the grid points t_j^o and the initial and end-point constraints at t_1^o and t_n^o. As result, we get a set of motions working for a specific set of boundary constraints.

IV. Deforming Optimal Trajectories Through Laplacian Trajectory Editing

Typically, and also including previous papers [12], [13] an optimal trajectory for every separate motion has been produced with direct collocation method, but its high computational complexity and sensitivity towards a good initial guess are major drawbacks. In contrast, this paper assumes that optimal trajectories have been calculated beforehand and will be deformed online to meet additional requirements. Although giving up the optimality of the direct collocation method, it results in minimum acceleration deviation between optimal and deformed trajectories, ensuring a certain amount of similarity between them. In addition, computational complexity is reduced considerably.

A generic method for deforming trajectories through a least-squares approach is Laplacian Trajectory Editing. The key idea of LTE during trajectory deformation is to calculate local trajectory properties, resulting in a linear system of equations. When adding boundary constraints, the resulting overdetermined system of equations can be solved using least squares.

It was shown in [4] that the least squares solution that requires matrix inversion can be decomposed in 2 parts. The first part is a sole translation of the original optimal trajectory. The second part is a cubic spline interpolation (e.g. using Bernstein polynomials) of the form

$$\mathbf{P}_j = \sum_{k=0}^{3} \mathbf{a}_{j,k} t^k, \quad t \in [0,1], \tag{8}$$

for the j-th trajectory interval.

Such representation possesses some useful properties. Upper bounds for optimal trajectories can be calculated offline with any segmentation. The same holds for pure translation of these bounds. Meanwhile the upper bounds for the second term can be found online using control points of Bernstein polynomial.

V. Feasibility of a Deformed Trajectory

By retargeting the trajectory in task space and not configuration space, the curse of dimensionality can be avoided. On the downside, multiple deformations are possible in case robot movement is infeasible. Thus a quick feasibility check is required.

A. Trajectory and torque feasibility

This section investigates the torque feasibility of the resulting trajectory for a given parametrization of the robot feedback controller. The estimation is based on a piecewise polynomial representation of the deformed trajectory by taking advantage of the special properties of Bernstein polynomials.

The following control structure, see Fig. 2, consists of an offline optimal trajectory generator and a reactive goal generator that sets new boundary states distinct from the original ones depending on the task. Afterwards, LTE based deformation generates the new trajectory $\mathbf{x_d}$, $\dot{\mathbf{x}}_\mathbf{d}$, $\ddot{\mathbf{x}}_\mathbf{d}$. This new trajectory is followed by a feedback controller. In this work we assume the feedback controller to consist of an internal computed-torque controller in combination with a PD controller. In many cases the robot must consider both kinematic and dynamic constraints. Such limitations originate from a finite applicable torque on each joint. LTE at the same time deforms trajectories in a task space. Thus, this section develops upper bounds for the joint torques based on the LTE deformation. These bounds are easily verified and allow predictions whether the deformed trajectory can be executed.

1) Segment-wise representation: An important feature of the method is that it is sufficient to check feasibility only at the segmentation points. Checking the required torques only at the boundary segment points reduces the overall computations. As the position, velocity and acceleration constraints are equitemporally discretized with Δt, and the optimal trajectory and control are calculated with different but known time grid $\Delta t_j^o, j \in [1, ..., n-1]$, we superimpose those two grids, see Fig. 3.

Motion time is reduced to fit the time of optimal trajectory. The minimum acceleration deviation is calculated on the original not scaled time grid.

For each interval of the superimposed grid, it holds that x_d is cubic, $\dot{x}_d$ is quadratic and $\ddot{x}_d$ is linear. The motion may not be feasible when the robot position, velocity or acceleration constraints are violated for the required motion. Thus, each of these terms should be checked on each time interval as well.

2) Boundedness of Deformed Trajectory: Upper bounds on the deformed trajectory for each time interval allow the robot to suffice dynamic constraints in joint space. Explicit bounds on the position, velocity and acceleration deviation result in deviation in joint torques that are bounded

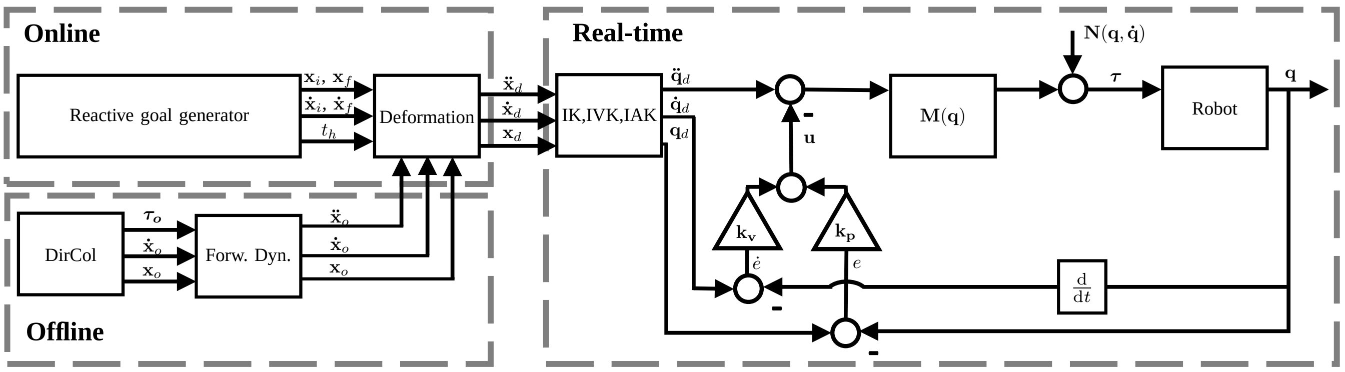

Fig. 2: Overall motion planning and control scheme. Setting of new boundary states $\mathbf{x}_i$, $\mathbf{x}_f$, $\dot{\mathbf{x}}_i$, $\dot{\mathbf{x}}_f$, and overall motion time t_h is carried out based on sensory data and provided goal. The optimal trajectory $\mathbf{x}_o$, $\dot{\mathbf{x}}_o$ together with optimal control $\boldsymbol{\tau}_o$ is generated through a direct collocation method. Using forward dynamics $\ddot{\mathbf{x}}_o$ is calculated. The optimal trajectory is deformed using LTE with the new start and end point. The obtained task space trajectory is converted to joint space $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$, $\ddot{\mathbf{q}}_d$ through inverse kinematics (IK), inverse velocity (IVK) and inverse acceleration kinematics (IAK). It is further tracked using the control structure with computed torque and PD controller.

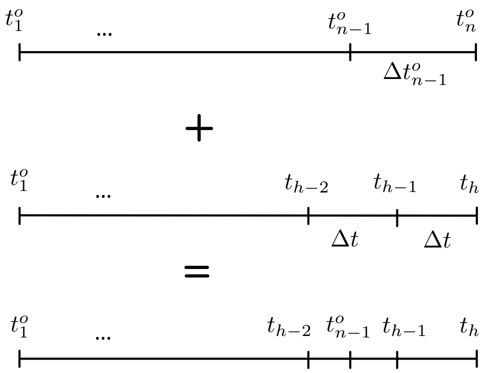

Fig. 3: Optimal time grid (top) remains untouched with fixed motion time of OCP t_n^o. Then time grid superposition (bottom) is obtained as a sum of original optimal and LTE reduced time grids.

from above. The vectors $\mathbf{q}$ and $\dot{\mathbf{q}}$ are related to the task space coordinates through the inverse and differential inverse kinematics schemes. After derivation we obtain

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{J}\dot{\mathbf{q}}, \\ \ddot{\mathbf{x}} &= \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\ddot{\mathbf{q}}.\end{aligned} \tag{9}$$

In case of redundancy and another prioritized tasks, presence of unbounded secondary tasks projected into the kernel of $\mathbf{J}$ can lead to (theoretically) arbitrarily large joint velocities and accelerations. In this case an upper bound of the resulting joint angle velocity of the secondary task must be determined as well. A pseudoinverse-based scheme can be used to calculate the differential inverse kinematics as

$$\begin{aligned}\ddot{\mathbf{q}} &= \mathbf{J}_1^+\mathbf{h}_1 + (\mathbf{I} - \mathbf{J}_1^+\mathbf{J}_1)\mathbf{J}_2^+\mathbf{h}_2, \\ \mathbf{h}_i &= \ddot{\mathbf{x}}_\mathbf{i} - \dot{\mathbf{J}}_\mathbf{i}\dot{\mathbf{q}}, \\ \dot{\mathbf{q}} &= \mathbf{J}_1^+\dot{\mathbf{x}}_\mathbf{1} + (\mathbf{I} - \mathbf{J}_1^+\mathbf{J}_1)\mathbf{J}_2^+\dot{\mathbf{x}}_\mathbf{2},\end{aligned} \tag{10}$$

where $\mathbf{J_i}$ is the Jacobian of i-th manipulation variable, $\dot{\mathbf{x}}_\mathbf{2}$, $\ddot{\mathbf{x}}_\mathbf{2}$ are the desired velocity and acceleration of the 2-nd manipulation variable.

Operator $\mathbf{I} - \mathbf{J}_1^+\mathbf{J}_1$ projects a generic velocity vector in the null space of the Jacobian matrix. The null space projection can be used to perform various optimizations [14], without affecting the given tracking task. For instance, one can consider an optimization procedure for torque and acceleration minimization as a desired additional task for the manipulator null-space [15]. If the secondary task is specified similarly to the primary task in terms of a Jacobian $\mathbf{J}$ and task space coordinates $\dot{\mathbf{x}}$, it is bounded from above according to

$$\begin{aligned}\|\ddot{\mathbf{q}}_d\|_\infty \le{}& \left\|\mathbf{J}_1^+\right\|_\infty (\|\ddot{\mathbf{x}}_{1d}\|_\infty + \left\|\dot{\mathbf{J}}_1\right\|_\infty \|\dot{\mathbf{q}}_{1d}\|_\infty) + \\ & (\|\mathbf{I}\|_\infty + \left\|\mathbf{J}_1^+\right\|_\infty \|\mathbf{J}_1\|_\infty) \\ & \left\|\mathbf{J}_2^+\right\|_\infty (\|\ddot{\mathbf{x}}_{2d}\|_\infty + \left\|\dot{\mathbf{J}}_2\right\|_\infty \|\dot{\mathbf{q}}_{2d}\|_\infty), \\ \|\dot{\mathbf{q}}_d\|_\infty \le{}& \left\|\mathbf{J}_1^+\right\|_\infty \|\dot{\mathbf{x}}_{1d}\|_\infty + \\ & (\|\mathbf{I}\|_\infty + \left\|\mathbf{J}_1^+\right\|_\infty \|\mathbf{J}_1\|_\infty) \left\|\mathbf{J}_2^+\right\|_\infty \|\dot{\mathbf{x}}_{2d}\|_\infty ,\end{aligned} \tag{11}$$

where $\dot{\mathbf{x}}_\mathbf{2}$, $\ddot{\mathbf{x}}_\mathbf{2}$ are bounded from above by the secondary task design.

For the non-redundant case, e.g. as in Fig. 1, or in redundant case with a single tracking task we get respectively both the manipulator Jacobian and its norms uniquely defined as

$$\ddot{\mathbf{q}} = \mathbf{J}^+(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}), \tag{12}$$

which links the task space acceleration $\ddot{\mathbf{x}}$ to joint space velocity and acceleration $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$. Whereas the Moore-Penrose pseudoinverse ensures that a unique solution with minimal joint velocity can be found, the singular-robust inverse provides upper bounds of the matrix norm also in a singular configuration.

Without loss of generality, in this work we consider a single tracking task case and use pseudo-inverse as a primary solution. Using (9) and (12), the norms of $\dot{\mathbf{q}}_d$, $\ddot{\mathbf{q}}_d$ are bounded by

$$\|\ddot{\mathbf{q}}_d\|_\infty \le \left\|\mathbf{J}^+\right\|_\infty (\|\ddot{\mathbf{x}}_d\|_\infty + \left\|\dot{\mathbf{J}}\right\|_\infty \|\dot{\mathbf{q}}_d\|_\infty),$$
$$\|\dot{\mathbf{q}}_d\|_\infty \le \left\|\mathbf{J}^+\right\|_\infty \|\dot{\mathbf{x}}_d\|_\infty\,, \tag{13}$$

where $\|\cdot\|_\infty$ is the infinity norm (∞-norm) of a continuous function on a closed interval. As $\dot{\mathbf{q}}_d$, $\ddot{\mathbf{q}}_d$ depend both on the Jacobian $\mathbf{J}$ and the task space velocity/acceleration $\dot{\mathbf{x}}_d$, $\ddot{\mathbf{x}}_d$, an upper bound has to be determined for them as well. The norm of the manipulator Jacobian is bounded from above by a constant scalar, due to the finite extent of each robot. If the Jacobian is Lipschitz continuous and the joint velocities are bounded, the norm of $\dot{\mathbf{J}}$ is bounded by a scalar factor $j_{\max}$ as

$$\left\|\dot{\mathbf{J}}\right\|_\infty \le j_{\max}. \tag{14}$$

An upper bound of the norm $\mathbf{J}^+$ can be derived through the singular values $\sigma_{\min}, ..., \sigma_{\max}$ of $\mathbf{J}$. It is

$$\left\|\mathbf{J}^+\right\|_\infty \le \sqrt{m}\left\|\mathbf{J}^+\right\|_2 \le \frac{\sqrt{m}}{\sigma_{\min}} = s_{\max}, \tag{15}$$

where m is the rank of the Jacobian and $s_{\max}$ is a finite scalar if the robot does not enter a singular configuration, which is ensured through a workspace analysis.

In most cases the upper bounds of $\dot{\mathbf{x}}_d$, $\ddot{\mathbf{x}}_d$ cannot be calculated in closed form and must be determined by iterating along the deformed trajectory. An exception is the spline-based deformation as in Sec. IV. When splitting the deformed trajectory up in two parts, the sole translation of optimal trajectory enables one to calculate an upper bound of the norm of $\dot{\mathbf{x}}_o$, $\ddot{\mathbf{x}}_o$ independent of the deformation as $\dot{x}_{\max}$, $\ddot{x}_{\max}$. Since the additive deformation term is represented as Bernstein polynomials $\mathbf{B}_k^l$, its derivative is calculated based on

$$\dot{\mathbf{B}}_k^l = l(\mathbf{B}_{k-1}^{l-1} - \mathbf{B}_k^{l-1}), \tag{16}$$

where l is a degree of polynomial. For a spline curve described by

$$\mathbf{x}_s = \sum_{k=0}^{l} \mathbf{a}_k \mathbf{B}_k^l, \tag{17}$$

its velocity is a Bézier spline of degree $l-1$ defined by l control points $l(\mathbf{a}_1 - \mathbf{a}_0), \ldots, l(\mathbf{a}_l - \mathbf{a}_{l-1})$ calculated as

$$\dot{\mathbf{x}}_s = l\sum_{k=0}^{l-1}(\mathbf{a}_{k+1} - \mathbf{a}_k)\mathbf{B}_k^{l-1}, \tag{18}$$

and the acceleration as

$$\ddot{\mathbf{x}}_s = l(l-1)\sum_{k=0}^{l-2}(\mathbf{a}_{k+2} - 2\mathbf{a}_{k+1} + \mathbf{a}_k)\mathbf{B}_k^{l-2}. \tag{19}$$

Thanks to non-negativity of each Bernstein polynomial, the norm of $\dot{\mathbf{x}}$, $\ddot{\mathbf{x}}$ is bounded by

$$\|\dot{\mathbf{x}}_d\|_\infty \le \dot{x}_{\max} + l \max_{k=0,\ldots,l-1} |\mathbf{a}_{k+1} - \mathbf{a}_k|\,, \tag{20}$$
$$\|\ddot{\mathbf{x}}_d\|_\infty \le \ddot{x}_{\max} + l(l-1) \max_{k=0,\ldots,l-2} |\mathbf{a}_{k+2} - 2\mathbf{a}_{k+1} + \mathbf{a}_k|\,.$$

Thus, (13) is rewritten as

$$\|\ddot{\mathbf{q}}_d\|_\infty \le \frac{\sqrt{m}}{\sigma_{\min}}(\|\ddot{\mathbf{x}}_d\|_\infty + j_{\max}\frac{\sqrt{m}}{\sigma_{\min}}\|\dot{\mathbf{x}}_d\|_\infty), \tag{21}$$
$$\|\dot{\mathbf{q}}_d\|_\infty \le \frac{\sqrt{m}}{\sigma_{\min}}\|\dot{\mathbf{x}}_d\|_\infty\,.$$

The upper bound on the joint velocities and accelerations guarantees that the trajectory deformed in task space does not violate any joint space constraints. Still, for a successful trajectory execution, the tracking error of the feedback controller and torque bounds are required.

3) Boundedness of Feedback Controller: With the imposed dynamic constraints, feasibility of the control scheme for a given trajectory and bounded external disturbance needs to be provided. The feedback controller is characterized by a tracking error and its dynamics

$$\begin{aligned}\mathbf{e} &= \mathbf{q_d} - \mathbf{q},\\ \dot{\mathbf{e}} &= \dot{\mathbf{q}}_\mathbf{d} - \dot{\mathbf{q}},\\ \ddot{\mathbf{e}} &= \mathbf{u} + \mathbf{w_{dist}},\end{aligned} \tag{22}$$

where the applied controller torque and disturbance term are represented as

$$\begin{aligned}\mathbf{u} &= \ddot{\mathbf{q}}_\mathbf{d} + \mathbf{M}^{-1}(\mathbf{N} - \boldsymbol{\tau}),\\ \mathbf{u} &= \mathbf{k_v}\dot{\mathbf{e}} + \mathbf{k_p}\mathbf{e},\\ \mathbf{w_{dist}} &= \mathbf{M}^{-1}\boldsymbol{\tau}_d,\end{aligned} \tag{23}$$

where $\boldsymbol{\tau}_d$ is a disturbance torque applied to the motors that takes into account the unmodeled dynamics and a maximum disturbance force applied to the robot during operation. The upper bound for this torque can be assigned for each time interval independently. This is especially useful for robotic tasks where the disturbance related to contacts occur only at some specific intervals, for instance, neighboring with t_h. The tracking error dynamics can be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{e}\\ \dot{\mathbf{e}}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{e}\\ \dot{\mathbf{e}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}\mathbf{u} + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}\mathbf{w_{dist}}. \tag{24}$$

The transfer function of the disturbance to error and to error derivative is as follows [16]

$$e_i(s) = \frac{1}{s^2 + k_{vi}s + k_{pi}}w_i(s) = G_i(s)w_i(s), \tag{25}$$
$$\dot{e}_i(s) = \frac{s}{s^2 + k_{vi}s + k_{pi}}w_i(s) = sG_i(s)w_i(s).$$

However, for obtaining the ∞-norm of the error, the 2-norm of the transfer function and the 2-norm of the input are required [17, Chap. 2]. Thus, the ∞-norms of the error functions $e_i(t)$ and $\dot{e}_i(t)$ are found using

$$\|\mathbf{w}(s)\|_2 \le \left\|\mathbf{M}^{-1}\right\|_2 \|\boldsymbol{\tau}_d\|_2 \le \bar{m}_{\text{inv}}\bar{d},$$
$$\|e_i(t)\|_\infty \le \|G_i(s)\|_2 \|w_i(s)\|_2 \le \|G_i(s)\|_2\, \bar{m}_{\text{inv}}\bar{d}, \tag{26}$$
$$\|\dot{e}_i(t)\|_\infty \le \|sG_i(s)\|_2 \|w_i(s)\|_2 \le \|sG_i(s)\|_2\, \bar{m}_{\text{inv}}\bar{d},$$

where the norm of the transfer function is defined as a 2-norm over frequency of $G(s)$ from (25)

$$\|G(s)\|_2 = \left(\frac{1}{2\pi}\int\limits_{-\infty}^{+\infty} |G(j\omega)|^2 \,\mathrm{d}\omega\right)^{\frac{1}{2}}. \tag{27}$$

For the definite integral from (27) we find

$$\begin{aligned} G(j\omega) &= \frac{-j\omega k_v + (k_p - \omega^2)}{(k_p - \omega^2)^2 + (\omega k_v)^2}, \\ j\omega G(j\omega) &= \frac{2\omega^2\sqrt{k_p} + j(k_p\omega - \omega^3)}{(k_p - \omega^2)^2 + (\omega k_v)^2}, \end{aligned} \tag{28}$$

and by choosing critically damped PD gains $k_v = 2\sqrt{k_p}$

$$\begin{aligned} |G(j\omega)|^2 &= \frac{1}{(k_p + \omega^2)^2}, \\ |j\omega G(j\omega)|^2 &= \frac{\omega^2}{(k_p + \omega^2)^2} = \frac{1}{k_p + \omega^2} - \frac{k_p}{(k_p + \omega^2)^2}, \end{aligned} \tag{29}$$

we calculate the definite integral of the proper rational function for which the denominator contains repeated irreducible quadratic factors, and thus

$$\begin{aligned} \int\limits_{-\infty}^{+\infty} \frac{1}{k_p + \omega^2}\mathrm{d}\omega &= \left(\frac{1}{\sqrt{k_p}}\tan^{-1}\frac{\omega}{\sqrt{k_p}}\right)\Bigg|_{-\infty}^{+\infty} = \frac{\pi}{\sqrt{k_p}}, \\ \int\limits_{-\infty}^{+\infty} \frac{1}{(k_p + \omega^2)^2}\mathrm{d}\omega &= \left(\frac{\omega}{2k_p(k_p + \omega^2)} + \frac{1}{2k_p}\frac{\pi}{\sqrt{k_p}}\right)\Bigg|_{-\infty}^{+\infty} \\ &= \frac{\pi}{2k_p\sqrt{k_p}}, \\ \int\limits_{-\infty}^{+\infty} \frac{\omega^2}{(k_p + \omega^2)^2}\mathrm{d}\omega &= \int\limits_{-\infty}^{+\infty} \frac{1}{k_p + \omega^2}\mathrm{d}\omega - k_p \int\limits_{-\infty}^{+\infty} \frac{1}{(k_p + \omega^2)^2}\mathrm{d}\omega \\ &= \frac{\pi}{2\sqrt{k_p}}. \end{aligned} \tag{30}$$

Finally, the 2-norm for i-th joint

$$\begin{aligned} \|G_i(s)\|_2 &= \sqrt{\frac{1}{(4k_{pi}\sqrt{k_{pi}})}}, \\ \|sG_i(s)\|_2 &= \sqrt{\frac{1}{(4\sqrt{k_{pi}})}}, \end{aligned} \tag{31}$$

results in

$$\begin{aligned} \|e_i(t)\|_\infty &\leq \|G_i(s)\|_2 \|w_i(s)\|_2 \leq \sqrt{\frac{1}{(4k_{pi}\sqrt{k_{pi}})}}\bar{m}_{\mathrm{inv}}\bar{d}, \\ \|\dot{e}_i(t)\|_\infty &\leq \|sG_i(s)\|_2 \|w_i(s)\|_2 \leq \sqrt{\frac{1}{(4\sqrt{k_{pi}})}}\bar{m}_{\mathrm{inv}}\bar{d}, \end{aligned} \tag{32}$$

and subsequently the acceleration error from (22) is bounded by

$$\|\ddot{e}_i(t)\|_\infty \leq k_{pi}\|\dot{e}_i(t)\|_\infty + k_{vi}\|\dot{e}_i(t)\|_\infty + \bar{m}_{\mathrm{inv}}\bar{d}. \tag{33}$$

As it can be seen from the structure of the error norms (32), an increase of the controller gains will lead to a better error suppression, but it will also result in higher required torques.

4) Torque Boundedness: In general the joint torques are limited by the hardware used, thus the goal is to keep the ∞-norm of all torque values below a specific constant $\boldsymbol{\tau}_{\max}$ to prevent torque saturation. This is done by evaluating the worst case scenario, i.e. by deriving upper bounds for each factor involved in the torque calculation.

Theorem. *Given the kinematic and dynamic models of the robot (1)-(12), the feedback control structure (22), and a bounded external disturbance (23), then the ∞-norm of the joint torques is bounded from above for each segment of the superimposed time grid from Fig. 3 by a scalar value*

$$\begin{aligned} \tau_{\mathrm{bound}} =& \bar{m}\|\ddot{\mathbf{q}}_d\|_\infty + \bar{m}(\mathbf{k_v}\|\dot{\mathbf{e}}\|_\infty + \mathbf{k_p}\|\mathbf{e}\|_\infty) + \\ & \bar{c}m(\|\dot{\mathbf{q}}_d\|_\infty + \|\dot{\mathbf{e}}\|_\infty)^2 + \bar{g} + \\ & v_b\sqrt{m}(\|\dot{\mathbf{q}}_d\|_\infty + \|\dot{\mathbf{e}}\|_\infty) + k_b \end{aligned} \tag{34}$$

for LTE deformation decomposed in transformation and spline interpolation expressed by Bernstein polynomials.

Proof. The norms of the matrices $\mathbf{M}$, $\mathbf{C}$, $\mathbf{G}$ [18] are limited above by factors as

$$\begin{aligned} \mathbf{M}(\mathbf{q}) &\leq \bar{m}\mathbf{I}, \\ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) &\leq \bar{c}\|\dot{\mathbf{q}}\|_2 \leq \bar{c}\sqrt{m}\|\dot{\mathbf{q}}\|_\infty, \\ \mathbf{G}(\mathbf{q}) &\leq \bar{g}, \\ \mathbf{F}(\mathbf{q}) &\leq v_b\|\dot{\mathbf{q}}\|_2 + k_b \leq v_b\sqrt{m}\|\dot{\mathbf{q}}\|_\infty + k_b, \end{aligned} \tag{35}$$

thus the bound on the overall torque from computed torque with PD controller scheme is rewritten using upper bounds on joint space velocities and accelerations (21), errors (32) and system matrices (35) in a following way:

$$\begin{aligned} \|\boldsymbol{\tau}\|_\infty &= \|\mathbf{M}(\ddot{\mathbf{q}}_d - \mathbf{u}) + \mathbf{N}\|_\infty \\ &\leq \|\mathbf{M}\|_\infty \|(\ddot{\mathbf{q}}_d - \mathbf{u})\|_\infty + \|\mathbf{N}\|_\infty \\ &\leq \bar{m}\|\ddot{\mathbf{q}}_d\|_\infty + \bar{m}\|\mathbf{u}\|_\infty + \bar{c}m\|\dot{\mathbf{q}}\|_\infty^2 + \\ &\quad \bar{g} + v_b\sqrt{m}\|\dot{\mathbf{q}}\|_\infty + k_b \\ &\leq \bar{m}\|\ddot{\mathbf{q}}_d\|_\infty + \bar{m}(\mathbf{k_v}\|\dot{\mathbf{e}}\|_\infty + \mathbf{k_p}\|\mathbf{e}\|_\infty) + \\ &\quad \bar{c}m(\|\dot{\mathbf{q}}_d\|_\infty + \|\dot{\mathbf{e}}\|_\infty)^2 + \bar{g} + \\ &\quad v_b\sqrt{m}(\|\dot{\mathbf{q}}_d\|_\infty + \|\dot{\mathbf{e}}\|_\infty) + k_b = \tau_{\mathrm{bound}} \end{aligned} \tag{36}$$

with the scalar value τ_{bound} accounting for the maximum motor torque boundary for the deformed trajectory. □

One should consider to perform a torque feasibility check for each joint independently to make norms tighter, especially in cases when different motors are used in the robot.

Segment-wise torque estimation is carried out for the scenario presented in Fig. 1. Initial boundary point is fixed for optimal and deformed trajectories $x_i = 0\,m$, $y_i = 0.3\,m$, $\phi_i = \pi/4\,rad$. Final optimal boundary point is $x_o = -0.18\,m$, $y_o = 0.46\,m$, $\phi_o = \pi/4\,rad$, $V_{xo} = -0.4\,m/s$, $V_{yo} = 0.4\,m/s$, $\omega_o = 0.08\,rad/s$, and deformed final boundary is $x_f = -0.23\,m$, $y_f = 0.56\,m$, $\phi_f = 0.64\,rad$, $V_{xf} = -0.53\,m/s$, $V_{yf} = 0.66\,m/s$, $\omega_f = 0.03\,rad/s$. Upper boundaries on torque for deformed trajectories with $\mathbf{k_p} = [1369,\ 1089,\ 841]$, $\mathbf{k_v} = [74,\ 66,\ 58]$ are shown in Fig. 5 and Fig. 6.

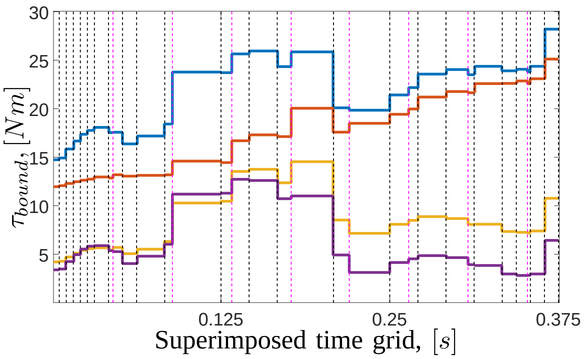

Fig. 4: Upper boundary for ∞-norm of the torques with $\bar{d} = 0.01Nm$ for all joints τ_{bound} (blue), and for every joint separately $\tau_{\text{i,bound}}$ (red, orange and violet) on the superimposed grid. Optimal time grid (black) and LTE-based time grid (magenta) are depicted with vertical dotted lines.

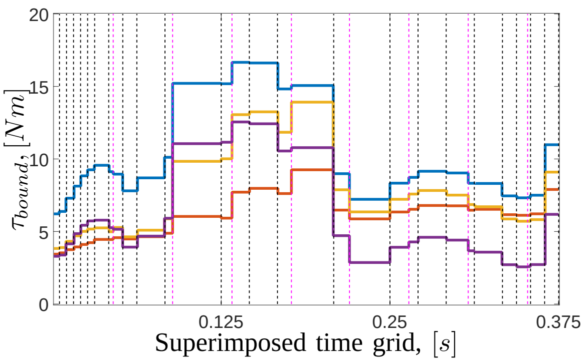

Fig. 5: Upper boundary for ∞-norm of the torques with $\bar{d} = 0Nm$ for all joints τ_{bound} (blue), and for every joint separately $\tau_{\text{i,bound}}$ (red, orange and violet) on the superimposed grid. Optimal time grid (black) and LTE-based time grid (magenta) are depicted with vertical dotted lines.

In case if the feasibility check fails, the following steps can be taken to lower torque bounds.

As a first option, optimization with minimax objective is carried out that modifies controller gains prior to the robot movement.

$$\begin{aligned} \underset{\mathbf{k}_p}{\text{minimize}} \quad & \max_i \tau_{i,\text{bound}} \\ \text{subject to} \quad & \tau_{i,\text{bound}} \leq \tau_{max} \\ & \mathbf{k}_{p,\min} \leq \mathbf{k}_p \leq \mathbf{k}_{p,\max}. \end{aligned} \tag{37}$$

The cost is responsible for torque minimization. This search algorithm is consistent with torque constraints and helps to choose the best controller parametrization. The second option is to increase number of segments for spline deformation. As overall time is fixed, this will lead to smaller bounds on norms of $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$, $\ddot{\mathbf{x}}_d$ and thus will result in tighter torque bounds, however additional computational effort is required. As a last option, one can deform another precomputed optimal trajectory to fit boundary conditions.

If all three procedures fail to find suitable upper bound on torque, the robot motion should be terminated.

VI. Conclusion

This paper discusses a method for retargeting and tracking previously calculated optimal trajectories online. The approach is executed together with the feasibility check on kinematic and dynamic constraints of the generated trajectories. Analytical upper bounds on the manipulator joint torques depending on the amount of deformation are derived. Even though calculation of ∞-norm is computationally efficient, it is still quite conservative and can be replaced with tighter bounds for a specific robot. Better knowledge about the system matrices, external forces and disturbances results in a smaller disturbance bounds and thus improves performance of the control scheme and decreases maximum torque boundaries. Future work will focus on applicability of this method to other numerical optimal control schemes.

References

[1] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences, 135(1).*, pp. 497–528, 2009.

[2] O. von Stryk, "User's guide for dircol (version 2.1): a direct collocation method for the numerical solution of optimal control problems," *Report, Lehrstuhl M2 Höhere Mathematik und Numerische Mathematik, Technische Universität München*, 1999.

[3] T. Nierhoff and S. Hirche, "Fast trajectory replanning using laplacian mesh optimization," in *IEEE Int. Conf. on Control, Automation, Robotics and Vision*, 2012.

[4] A. Pekarovskiy, T. Nierhoff, J. Schenek, Y. Nakamura, S. Hirche, and M. Buss, "Online deformation of optimal trajectories for constrained nonprehensile manipulation," in *IEEE Int. Conf. on Robotics and Automation*, 2015.

[5] F. Ghorbel, B. Srinivasan, and M. W. Spong, "On the uniform boundedness of the inertia matrix of serial robot manipulators," *Journal of Robotic Systems*, vol. 15, no. 1, pp. 17–28, 1998.

[6] J. Moreno and S. González, "An adaptive output feedback tracking controller for manipulators subject to constrained torques," in *Proceedings of IEEE Conf. of Decision and Control*, pp. 2026–2031, 2006.

[7] J. Mulero-Martinez, "Uniform bounds of the coriolis/centripetal matrix of serial robot manipulators," *Robotics, IEEE Transactions on*, vol. 23, pp. 1083–1089, Oct 2007.

[8] J. Kieffer, A. J. Cahill, and M. R. James, "Robust and accurate time-optimal path-tracking control for robot manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 6, pp. 880–890, 1997.

[9] J. E. 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.

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

[11] O. Gerelli and C. Bianco, "Real-time path-tracking control of robotic manipulators with bounded torques and torque-derivatives," in *In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 532–537, Sept 2008.

[12] O. von Stryk and M. Schlemmer, "Optimal control of the industrial robot manutec r3," *Computational optimal control, International series of Numerical Mathematics*, vol. 115, pp. 367–382, 1994.

[13] A. Pekarovskiy and M. Buss, "Optimal control goal manifolds for planar nonprehensile throwing," *In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013.

[14] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1990.

[15] H. Seraji, "Task options for redundancy resolution using configuration control," in *Decision and Control, Proceedings of the 30th IEEE Conference on*, pp. 2793–2798 vol.3, Dec 1991.

[16] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot manipulator control: theory and practice*. CRC Press, 2003.

[17] J. C. Doyle, B. A. Francis, and A. Tannenbaum, *Feedback control theory*, vol. 1. Macmillan Publishing Company, New York, 1992.

[18] W. Chung, L.-C. Fu, and S.-H. Hsu, "Motion control," in *Springer Handbook of Robotics*, pp. 133–159, 2008.