2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Predictive Online Inverse Kinematics for Redundant Manipulators

Christoph Schuetz, Thomas Buschmann, Joerg Baur, Julian Pfaff and Heinz Ulbrich

Abstract— **Determining the optimal solution for the inverse kinematics of redundant robots has been the focus of much previous research. Instantaneous approaches are computationally efficient, but may cause high joint velocities due to their local character. In this paper, we present an efficient implementation of a global approach following Pontryagin's Minimum Principle for online calculation. Within a moving horizon, we exploit the decoupled structure of the resulting optimal control problem by using the conjugate gradient method for solving the nonlinear dynamic problem. Different examples of cost functionals are presented and the real-time capability is shown by applying this approach to a 9-DOF redundant manipulator.**

I. INTRODUCTION AND BACKGROUND

With the decreasing cost of actuators and electronic components, the development of robotic systems with many degrees-of-freedom (DOFs) is becoming more and more viable. This makes it increasingly likely that a given task is described by fewer equations than could be satisfied with the system's number of DOFs. In such redundant configurations there are (usually) an infinite number of solutions for a given task. This enables the controller to choose a solution which is optimal with respect to some user-defined criterion.

A common approach to resolve the inverse kinematics (IK) problem of redundant manipulators using a local optimization technique was first proposed in [1], [2]. The task is defined in task coordinates $\boldsymbol{x}_d(t)$, which is often chosen as the end effector position and orientation for robotic manipulators. Formulating the task constraint $\boldsymbol{x}_d(t) = \boldsymbol{x}(\boldsymbol{q}(t))$ at the velocity level leads to an underdetermined set of linear equations in the joint velocities $\dot{\boldsymbol{q}}$

$$\dot{\boldsymbol{x}}_d(t) = \dot{\boldsymbol{x}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{1}$$

where the Jacobian $\boldsymbol{J}(\boldsymbol{q})$ is defined as $\frac{\partial \boldsymbol{x}(\boldsymbol{q})}{\partial \boldsymbol{q}}$. The desired $\dot{\boldsymbol{q}}$ is determined by minimizing the cost $\frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{W} \dot{\boldsymbol{q}} + L(\boldsymbol{q})$ over the duration of the next timestep Δt in the task's nullspace. Joint velocities are weighted by $\boldsymbol{W}$ and $L(\boldsymbol{q})$ are problem-dependent costs. With the increment $\Delta L \approx \frac{\partial L(\boldsymbol{q})}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} \Delta t$, the explicit solution is given by

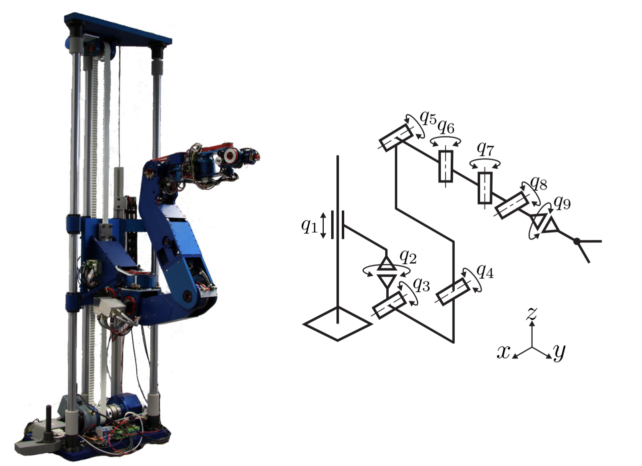
$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\dot{\boldsymbol{x}} + \alpha \boldsymbol{N} \left(-\frac{\partial L}{\partial \boldsymbol{q}}\right)^T \tag{2}$$

$$\boldsymbol{J}^{\#} = \boldsymbol{W}^{-1}\boldsymbol{J}^T \left(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T\right)^{-1} \tag{3}$$

$$\boldsymbol{N} = \left(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}\right) \tag{4}$$

This research was partly funded by the European Commission in the 7th Framework Programme (CROPS GA no 246252).

C. Schuetz, T. Buschmann, J. Baur, J. Pfaff and H. Ulbrich are with the Institute of Applied Mechanics, Technische Universität München, Boltzmannstr. 15, 85748 Garching, Germany {schuetz, buschmann, baur, pfaff, ulbrich}@amm.mw.tum.de



(a) CROPS Manipulator (b) Kinematic scheme

Fig. 1: Modular 9-DOF agricultural robot developed within the CROPS project [4]

where $\boldsymbol{J}^{\#}$ is the *Moore-Penrose* pseudoinverse, $\boldsymbol{N}$ the *null-space projection matrix*, $\boldsymbol{I}$ the identity matrix and α a scalar weighting factor. The trajectory $\boldsymbol{q}(t)$ is obtained by integrating $\dot{\boldsymbol{q}}(t)$. A comprehensive overview of different pseudoinverse-based approaches can be found in [3].

Since the approach described above is formulated for every time instance at the level of joint velocities, it can only modify the robot's configuration along the null-space directions $\boldsymbol{N}(\boldsymbol{q})$ resulting from the current configuration $\boldsymbol{q}$. It cannot take into account how $\dot{\boldsymbol{x}}_d(t)$, collision avoidance, joint limit avoidance or joint velocities will be affected by the way the chosen $\dot{\boldsymbol{q}}$ influences the *future configuration* $\boldsymbol{q}$. This may be considered to be the main drawback of the algorithm, since future constraints are often known in advance. Fulfilling these constraints using the local approach can cause severe peaks in the joint velocities which are not necessary from a global point of view of the given task (see Section II-A).

This problem can be addressed using optimal control methods with integral cost functions. One approach is the formulation of the Hamiltonian following Pontryagin's Minimum Principle (PMP) [5], which defines necessary conditions that can be solved using several algorithms [6]:

- Boundary conditions of the PMP differential equations define a two-point boundary value problem [7].
- Exploring a local search space at a quadratic approximation of the value function (*cost-to-go*) yields to a

feedback law, which is used by the differential dynamic programming (DDP) approach [8].
- Based on the partial derivative of the Hamiltonian w.r.t. the input vector, optimal trajectories are found by the conjugate gradient method [9].

A. Related Work

With CHOMP, a planning framework, suited for highly redundant robot systems, has been presented by [10], [11]. Optimal trajectories are obtained using covariant gradient techniques. Thereby, a cost formulation invariant to time prevents, that e.g. passing obstacles very fast and close is considered more optimal than keeping a safe distance. Collisions are avoided based on a precomputed distance field. Another approach has been presented by [12]: The problem is solved employing sequential convex optimization, i.e. quadratic, convex approximations of the problem are solved iteratively. Infeasible constraints are turned into penalty functions to guide the algorithm to a feasible solution area. A two-stage approach has been presented by [13]: First, a feasible solution is found using a RRT algorithm which serves as an initial guess for the subsequent optimization. In contrast to ours, the aforementioned approaches, as well as other sampling-based [14], [15] or global methods [16], [17] are generally not used online for high-dimensional systems due to the high computational costs.

TASSA ET AL. have presented a variant of the DDP approach [18]. Thank to its computational efficiency and the implementation of a moving horizon scheme, it is almost suited for real-time use. Since this approach is related to ours, a comparison with our following scheme will be in the focus of our upcoming work.

B. Contribution

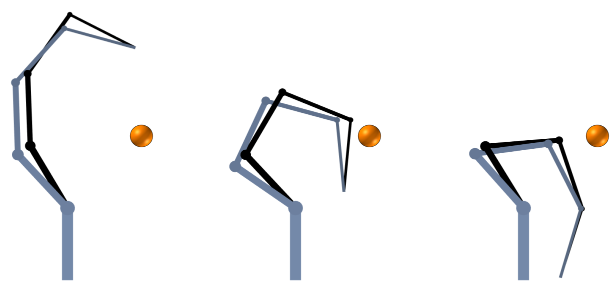
In this paper, we propose to use the numerically efficient conjugate gradient method to solve the PMP as formulated by [7]. Since the optimization input is projected to the nullspace of the redundant manipulator, workspace contraints $\dot{\boldsymbol{x}}_d(t)$ are preserved. Real-time optimization is enabled by applying a moving horizon approach (comparable to Model Predictive Control). We implemented this approach for a 9-DOF modular agricultural robot (Fig. 1a), that has been developed at our institute for harvesting of single crops (e.g. sweet pepper or apples) or precision spraying of grapes [4]. It has 1 prismatic and 8 rotational joints, the kinematic scheme is shown in Fig. 1b. Since the workspace is 6-dimensional (3 cartesian positions and 3 rotations), the degree of redundancy is 3. For the inverse kinematics solution, an local inverse kinematics approach was presented in [4], [19], which will be the reference for our new approach.

In Section II, we present the basic idea of our approach including the problem formulation, optimality criteria and an efficient numerical solution. Section III deals with different examples of cost functionals that we have used in our application. The implementation for the 9-dof CROPS manipulator and results are shown in Section IV, while Section V gives a summary and an outlook for future work.

II. A PREDICTIVE APPROACH FOR INVERSE KINEMATICS

A. Motivational Example

A motivational example is given in Fig. 2. The planar $4-$DOF robot performs a predefined movement on a straight line from top to bottom. While the workspace trajectory $\boldsymbol{x}_d(t)$ is supposed to be collision-free, a collision may occur between link 4 and the obstacle (*orange* sphere). The local optimization starts avoiding the collision as soon as the obstacle enters a certain activation distance d_a (*black* robot). By this time, however, the local optimization has led to an unfavorable configuration close to the obstacle. The globally optimal solution chooses a better configuration during the first period of the motion (*blue* robot), significantly reducing peak joint speeds. The evolution of the pseudoenergy $E_{ps} = \dot{\boldsymbol{q}}_k^T \dot{\boldsymbol{q}}_k$ at each timestep t_k (Fig. 3) shows that the *black* robot has to set high joint velocities to avoid the collision while the *blue* robot using the global solution moves smoothly.



(a) $t = 0.5$s (b) $t = 3.0$s (c) $t = 4.5$s

Fig. 2: Collision avoidance in the null-space of the robot. *Black robot:* Local Optimization. *Blue robot:* Global Optimization

In a similiar way, e.g. joint limits, high joint velocities or other constraints can be avoided by using a global solution of the inverse kinematics problem for a redundant robot. In

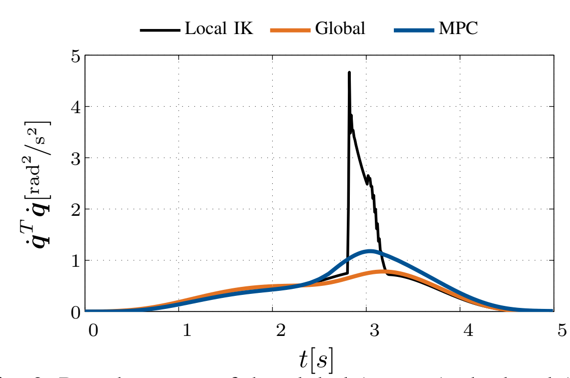


Fig. 3: Pseudoenergy of the global (*orange*), the local (*black*) and the MPC (*blue*) optimization

the following section we present a predictive approach which takes advantage of a global optimal solution within a certain time window by formulating the optimization problem in an efficient way and solving it online.

B. Pontryagin's Minimum Principle

Since the method described by (2)–(4) is local, constraints are detected locally by the algorithm even if all necessary boundary conditions are known for the entire trajectory. It is therefore advantageous to use the knowledge of these constraints to take them into account within the inverse kinematics algorithm. Nakamura [7] proposed the following global optimization problem for solving the inverse kinematics for redundant manipulators with the input $\boldsymbol{u}$ to the nullspace:

$$\min_{\boldsymbol{q}(\tau)} L(\boldsymbol{q},\boldsymbol{u},t) = \int_{\mathcal{T}} l(\boldsymbol{q},\boldsymbol{u},t)\,\mathrm{d}t \tag{5}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q},\boldsymbol{u},t) = \boldsymbol{J}^{\#}\dot{\boldsymbol{x}} + \alpha\left(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}\right)\boldsymbol{u} \tag{6}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{7}$$

$$t \in [t_0, t_{\text{end}}]$$

From Pontryagin's Minimum Principle we obtain the following set of optimality conditions

$$H(\boldsymbol{q},\boldsymbol{u},\boldsymbol{\lambda},t) = l(\boldsymbol{q},\boldsymbol{u},t) + \boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{q},\boldsymbol{u},t) \tag{8}$$

$$\left(\frac{\partial H}{\partial \boldsymbol{\lambda}}\right)^T = \dot{\boldsymbol{q}} \tag{9}$$

$$-\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T = \dot{\boldsymbol{\lambda}} = -\left(\frac{\partial l}{\partial \boldsymbol{q}}\right)^T - \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\right)^T\boldsymbol{\lambda} \tag{10}$$

$$\boldsymbol{x}(\boldsymbol{q}(t_0)) = \boldsymbol{x}_0 \tag{11}$$

$$\boldsymbol{\lambda}(t_{\text{end}}) = \boldsymbol{0} \tag{12}$$

$$\boldsymbol{N}\boldsymbol{\lambda}(0) = \boldsymbol{0} \tag{13}$$

$$\boldsymbol{u}^\star = \operatorname*{argmin}_{\boldsymbol{u}} H(\boldsymbol{q}^\star,\boldsymbol{\lambda}^\star,\boldsymbol{u},t), \tag{14}$$

where starred versions of the variables denote the optimal solution. The boundary conditions on $\boldsymbol{x}$ (11) are self-explanatory and those on $\boldsymbol{\lambda}$ (12) follow from the free final state and the fact that the problem has *Lagrange* form. The transversality condition (13) ensures that the adjoint variables lie within the normal cone to the tangent cone of the constraint set defined by $\boldsymbol{x}(\boldsymbol{q}(t_0)) = \boldsymbol{x}_0$. Requiring that the prescribed task $\boldsymbol{x}_0$ be satisfied at the beginning enables the calculation of a globally optimal solution, but also leads to a coupled two-point boundary value problem (BVP) due to the transversality condition.

If we intend to solve the problem continuously in the spirit of MPC as proposed above, we must ensure that $\boldsymbol{q}(t)$ is continuous, i.e. that $\boldsymbol{q}(t_0) = \boldsymbol{q}_0$. This modified optimal control problem leads to optimality criteria with *decoupled boundary conditions* for $\boldsymbol{\lambda}$ and $\boldsymbol{q}$:

$$H(\boldsymbol{q},\boldsymbol{u},\boldsymbol{\lambda},t) = l(\boldsymbol{q},\boldsymbol{u},t) + \boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{q},\boldsymbol{u},t) \tag{15}$$

$$\left(\frac{\partial H}{\partial \boldsymbol{\lambda}}\right)^T = \dot{\boldsymbol{q}} \tag{16}$$

$$-\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T = \dot{\boldsymbol{\lambda}} = -\left(\frac{\partial l}{\partial \boldsymbol{q}}\right)^T - \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\right)^T\boldsymbol{\lambda} \tag{17}$$

$$\boldsymbol{q}(t_0) = \boldsymbol{q}_0 \tag{18}$$

$$\boldsymbol{\lambda}(t_{\text{end}}) = \boldsymbol{0} \tag{19}$$

$$\boldsymbol{u}^\star = \operatorname*{argmin}_{\boldsymbol{u}} H(\boldsymbol{q}^\star,\boldsymbol{\lambda}^\star,\boldsymbol{u},t) \tag{20}$$

Conditions (16)-(19) allow us to calculate $\boldsymbol{\lambda}(t), \boldsymbol{q}(t)$ for any given $\boldsymbol{u}(t)$ by simply integrating (16) forward in time and (17) backward in time. We may therefore conceptually substitute this into (20), leading to an unconstrained optimization problem for $\boldsymbol{u}^\star$:

$$\boldsymbol{u}^\star = \operatorname*{argmin}_{\boldsymbol{u}} H(\boldsymbol{q}(\boldsymbol{u}),\boldsymbol{\lambda}(\boldsymbol{u}),\boldsymbol{u}) \tag{21}$$

$$= \operatorname*{argmin}_{\boldsymbol{u}} \hat{H}(\boldsymbol{u}) \tag{22}$$

Such problems are easily solved using the gradient method[1] characterized by an iterative calculation of the unknowns according to the basic rule

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \beta\frac{\partial \hat{H}}{\partial \boldsymbol{u}}^T, \tag{23}$$

where β is calculated from some step-size rule. See [20] for a review of the mathematical background.

C. Efficient Numerical Solution

While simple, the gradient method is known for its slow convergence. We therefore use the conjugate gradient method as described by [9] to solve the dynamic optimal control problem (15)-(20). This method provides a computationally efficient solution of the nonlinear problem in real time. Thereby, the basic algorithm, applied to the inverse kinematics problem of the redundant manipulator performs an iterative sequential forward integration of the system ODE (16) and backward integration of the adjoint variable ODE (17). The computational scheme is shown in algorithm (1). Direct kinematics of the robot are calculated by the recursive approach as presented in [21]. The step size β can either be chosen as an adequately small constant or by using some sort of step-size control algorithm, e.g. using the backtracking method [22]. The conjugate gradient method has shown faster convergence compared to a steepest descent algorithm as formerly reported in [9] (See Fig. 4 for the example in II-A).

D. Predictive Approach

Depending on the number of DOFs and the length of the trajectory, the global approach may not be solvable in real time. We therefore propose an approach which follows the idea of model-predictive control [23] to consider future limits

[1] If $\boldsymbol{u}$ is constrained ($\boldsymbol{u} \in \mathcal{U}$), the projected gradient method can be used.

Algorithm 1 Conjugate Gradient Method

$j \leftarrow 0$
$\boldsymbol{q}^0(\tau) \leftarrow \int_\tau \dot{\boldsymbol{q}}(\boldsymbol{u}_0, t)\mathrm{d}t$
repeat
  $\boldsymbol{\lambda}^j(\tau) \leftarrow \int_\tau \dot{\boldsymbol{\lambda}}(\boldsymbol{q}_j, \dot{\boldsymbol{q}}_j \boldsymbol{u}_j, t)\mathrm{d}t$
  $\boldsymbol{g}^j \leftarrow \frac{\partial H}{\partial u}$
  if $j \neq 0$ **then**
    $\beta^j \leftarrow \frac{(\boldsymbol{g}^j, \boldsymbol{g}^j)}{(\boldsymbol{g}^{(j-1)}, \boldsymbol{g}^{(j-1)})}$
    where $(\boldsymbol{g}^j, \boldsymbol{g}^j) = \int_\tau (\boldsymbol{g}^j(t))^T \boldsymbol{g}(\boldsymbol{t})^j \mathrm{d}t$
    $\boldsymbol{s}^j \leftarrow -\boldsymbol{g}^j + \beta^j \boldsymbol{s}^{(j-1)}$
  else
    $\boldsymbol{s}^j \leftarrow -\boldsymbol{g}^j$
  end if
  $\boldsymbol{u}^{(j+1)} \leftarrow \boldsymbol{u}^j + \alpha^j \boldsymbol{s}^j$
  $j \leftarrow j + 1$
  $\boldsymbol{q}^j \leftarrow \int_\tau \dot{\boldsymbol{q}}(\boldsymbol{u}^j, t)\mathrm{d}t$
  exit $\leftarrow (j > j_{\max}) \vee \left(\frac{L^j - L^{j-1}}{L^{j-1}} < \epsilon\right) \vee \left(L^j > L^{j-1}\right)$
until (exit $=$ **true**)
return $\boldsymbol{u}^j$

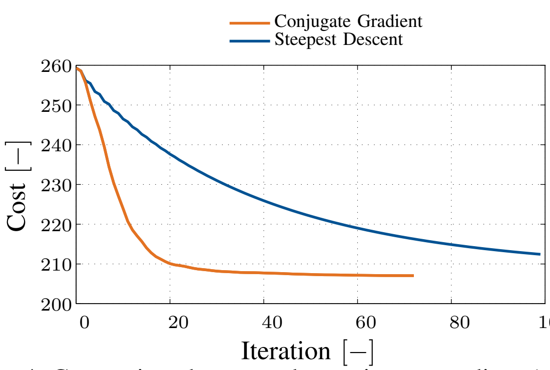


Fig. 4: Comparison between the conjugate gradient (red) and the steepest descent (black) method for the planar 4-DOF manipulator.

by predicting the behavior of the system as a function of the input trajectory $\boldsymbol{u}(\tau)$ within a moving horizon $\tau \in [t_k, t_k + T_{\text{horizon}}]$. The basic scheme is shown in Algorithm (2).

Calculating the optimal trajectory using a 1st order approach for the system equation as shown in (6) may lead to *non-zero* initial joint velocities $\dot{\boldsymbol{q}}_k$ since joint acceleration in the *nullspace* is not considered within the cost functional. The resulting trajectory using the MPC approach may be non-smooth as shown in Fig. 5.

We propose therefore to use an adapted system equation for optimization on acceleration level. A formulation of the kinematics equation on acceleration level is considered computational disadvantageous since additional gradients of e.g. $\boldsymbol{J}, \boldsymbol{J}^{\#}$ are needed. The augmentation of the state space in

Algorithm 2 Model Predictive Control Approach

$T_{\text{start}} = t_0,\ T_{\text{end}} = t_n,\ \Delta t_{\text{mpc}} = t_{k+1} - t_k$
for $k = 0$ to n **do**
  $\tau \leftarrow t_k + T_{\text{horizon}}$
  $\tau' \leftarrow t_k + \Delta t_{\text{mpc}}$
  determine $\boldsymbol{u}^*(\tau)$ using $\boldsymbol{u}^0(\tau) = \boldsymbol{u}^*_{\text{previous}}(\tau)$
  apply $\boldsymbol{u}^*(\tau')$
end for

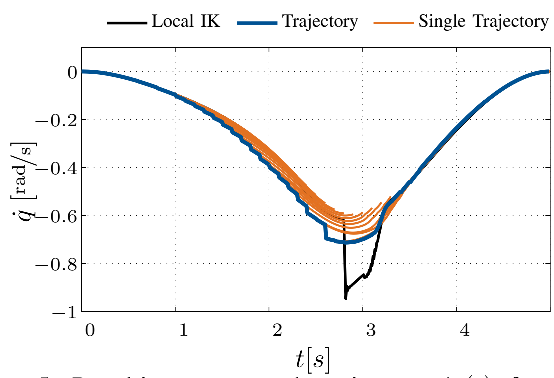


Fig. 5: Resulting nonsmooth trajectory $\dot{q}_2(t)$ from MPC optimization on velocity level (*blue*) of the example given in Section II-A and single MPC solution trajectories (*orange*) in comparison to the local inverse kinematics approach (*black*).

(6) by $\boldsymbol{u}$ and its derivative $\hat{\boldsymbol{u}}$ as the input vector is an obvious approach. (6)-(7) are replaced by the following equations:

$$\min_{\boldsymbol{q}(\tau)} L(\boldsymbol{q}, \hat{\boldsymbol{u}}, t) = \int_\tau l(\boldsymbol{q}, \hat{\boldsymbol{u}}, t)\,\mathrm{d}t \tag{24}$$

$$\begin{aligned}\boldsymbol{f}_{\text{acc}}(\boldsymbol{q}, \hat{\boldsymbol{u}}, t) &= \begin{bmatrix}\boldsymbol{f}_1\\ \boldsymbol{f}_2\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\boldsymbol{u}}\end{bmatrix} = \dots\\ &= \begin{bmatrix}\boldsymbol{J}^{\#}\dot{\boldsymbol{x}} + \alpha\left(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}\right)\boldsymbol{u}\\ \hat{\boldsymbol{u}}\end{bmatrix}\end{aligned} \tag{25}$$

Hence, Eq. (15)-(20) become:

$$\boldsymbol{\lambda}_{\text{acc}} = \begin{bmatrix}\boldsymbol{\lambda}_1\\ \boldsymbol{\lambda}_2\end{bmatrix} \tag{26}$$

$$\begin{aligned}H(\boldsymbol{q}, \hat{\boldsymbol{u}}, \boldsymbol{\lambda}_{\text{acc}}, t) = l(\boldsymbol{q}, \hat{\boldsymbol{u}}, t) + \dots\\ \boldsymbol{\lambda}_1^T \boldsymbol{f}_1(\boldsymbol{q}, \hat{\boldsymbol{u}}, t) + \boldsymbol{\lambda}_2^T \boldsymbol{f}_2(\hat{\boldsymbol{u}})\end{aligned} \tag{27}$$

$$\left(\frac{\partial H}{\partial \boldsymbol{\lambda}_{\text{acc}}}\right)^T = \begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\boldsymbol{u}}\end{bmatrix} \tag{28}$$

$$-\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T = \dot{\boldsymbol{\lambda}}_1 = -\left(\frac{\partial l}{\partial \boldsymbol{q}}\right)^T - \left(\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{q}}\right)^T \boldsymbol{\lambda}_1 \tag{29}$$

$$-\left(\frac{\partial H}{\partial \boldsymbol{u}}\right)^T = \dot{\boldsymbol{\lambda}}_2 = -\left(\frac{\partial l}{\partial \boldsymbol{u}}\right)^T - \left(\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{u}}\right)^T \boldsymbol{\lambda}_1 \tag{30}$$

$$\boldsymbol{q}(t_0) = \boldsymbol{q}_0, \quad \boldsymbol{u}(t_0) = \boldsymbol{u}_0 \tag{31}$$

$$\boldsymbol{\lambda}_{\text{acc}}(t_{\text{end}}) = \boldsymbol{0} \tag{32}$$

$$\hat{\boldsymbol{u}}^\star = \underset{\hat{\boldsymbol{u}}}{\operatorname{argmin}}\, H(\boldsymbol{q}^\star, \boldsymbol{\lambda}^\star_{\text{acc}}, \hat{\boldsymbol{u}}, t) \tag{33}$$

The numerical solution is analogous to the optimization on velocity level in *nullspace* as presented in Section II-C.

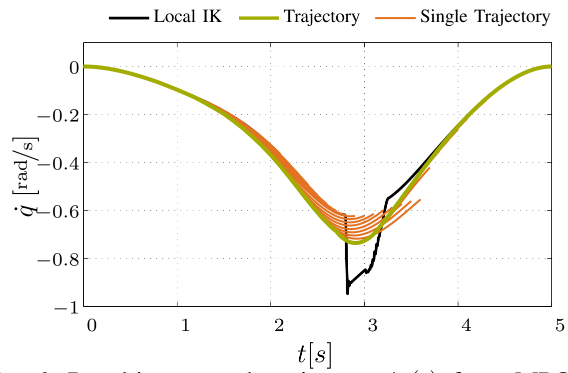


Fig. 6: Resulting smooth trajectory $\dot{q}_2(t)$ from MPC optimization on acceleration level (*green*) of the example given in Section II-A and single MPC solution trajectories (*orange*) in comparision to the local inverse kinematics approach (*black*).

Required gradients are given in the APPENDIX. Using this approach, we obtain smooth trajectories as shown in Fig. 6.

III. EXAMPLE COST FUNCTIONS

Optimality of the trajectory $\boldsymbol{q}(t)$ is defined by the cost function L in (5) and (24). L can be formulated as an integral over time of a sum of n_c sub cost functions l_i weighted by scalar factors c_i. The required gradients $\frac{\partial l}{\partial \boldsymbol{q}}$ and $\frac{\partial l}{\partial \boldsymbol{u}}$ or $\frac{\partial l}{\partial \hat{\boldsymbol{u}}}$ for solving the optimal control problem following the proposed approach in Section II are given in the APPENDIX.

$$L(\boldsymbol{q}, \boldsymbol{u}, t) = \int_{\tau} \sum_{i=1}^{n_c} l_i(\boldsymbol{q}, \boldsymbol{u}, t) \mathrm{d}t \tag{34}$$

The desired behavior of the system such as minimal joint velocities, avoidance of collisions and joint limits is expressed by these sub cost functions l_i. Examples are given in the following section.

A. Joint Velocity and Acceleration in Nullspace

For minimizing joint velocities and joint acceleration in *nullspace*, the sum of squares of $\dot{\boldsymbol{q}}$ and $\hat{\boldsymbol{u}}$ can be chosen as cost functionals, respectively.

$$l_v = \dot{\boldsymbol{q}}^T \dot{\boldsymbol{q}} \tag{35}$$

$$l_{na} = \hat{\boldsymbol{u}}^T \hat{\boldsymbol{u}} \tag{36}$$

B. Joint Limit Avoidance

According to [4], [24] a piecewise defined, continuous polynomial function penalizes joint positions $\boldsymbol{q}$ near the maximum (minimum) joint limits $q_{M,i}$ ($q_{m,i}$) within a threshold $\bar{q}_{M,i}$ ($\bar{q}_{m,i}$).

$$l_l = \begin{cases} \sum_{i=1}^{n} \frac{(q_i - \bar{q}_{M,i})^2}{(q_{M,i} - \bar{q}_{M,i})^2} & \text{if } q_i > \bar{q}_{M,i} \\ \sum_{i=1}^{n} \frac{(q_i - \bar{q}_{m,i})^2}{(q_{m,i} - \bar{q}_{m,i})^2} & \text{if } q_i < \bar{q}_{m,i} \\ 0 & \text{else} \end{cases} \tag{37}$$

C. Collision Avoidance

For collision avoidance, we use the algorithms developed for self-collision avoidance of the humanoid robot LOLA [25] and applied to the CROPS robot [19]. Thereby, complex geometries of the robot model are simplified by using *swept-sphere volumes* (SSV). The cost function of the closest distance d_k between two bodies is given by Eq. (38).

$$l_c = \begin{cases} \frac{s}{3} (d_a - d_k)^3 & \text{if } d_k < d_a \\ 0 & \text{else} \end{cases} \tag{38}$$

The parameter s determines the maximum value of the cost function when collision occurs and d_a the activation distance. For the calculation of the gradients refer to [25].

IV. EXPERIMENTAL RESULTS

The proposed computational scheme has been implemented for a 9-DOF modular robot (see Section I). For preliminary tests, a self-collision scenario has been chosen, similiar to the setup in [19]. Increasing the length of the moving horizon T_{horizon} improves the result, as shown in Fig. 7. Regarding the computational times, the smaller horizon was already feasible to calculate on a MATLAB/SIMULINK real time control unit with a sampling rate of 10 ms and one iteration step. A larger horizon of 2s takes currently up to 7-times real-time computation. While a small horizon of 0.15s slightly improves the avoidance of velocity peaks, a larger horizon (2s) leads to significant better results.

Preliminary tests have shown the real-time capability of our approach. However, these tests have been performed using a relatively small moving horizon. Extending the moving prediction time window by improving the current implementation is currently on-going.

V. CONCLUSION

In this paper, a novel approach for determining a predictive solution for inverse kinematics of redundant manipulators, which can be calculated online, has been presented. Thereby, an optimal control problem following the formulation of

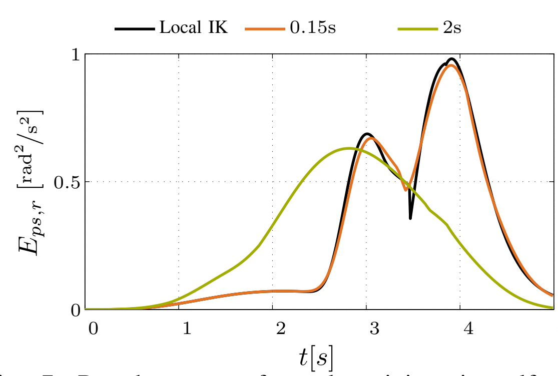


Fig. 7: Pseudoenergy of revolute joints in self-collision scenario for the harvesting robot calculated by local inverse kinematics and MPC approach with a moving horizon of 0.15s and 2s.

Pontryagin's Minimum Principle as proposed in [7] has been solved within a moving horizon. Facing the large computational costs, we proposed to exploit the decoupled structure of the nonlinear problem applying the conjugate gradient method. The proposed approach has shown its advantages compared to instantaneous approaches by responding to future constraints. It has been proven by the implementation for a 9-DOF redundant robot that this method is suitable for real-time application. Especially the enlargement of the moving time window by optimizing the implementation will be in the focus of our future research. Furthermore we intend to perform quantitative comparisons to other related approaches, such as variations of the DDP approach as well as investigations on robustness and stability.

APPENDIX

Gradients for Optimization on Velocity Level

$$\frac{\partial f}{\partial \boldsymbol{q}} = \frac{\partial \dot{\boldsymbol{q}}}{\partial \boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{q}}^{\#} \dot{\boldsymbol{w}} - \boldsymbol{J}_{\boldsymbol{q}}^{\#} \boldsymbol{J}_{\boldsymbol{u}} - \boldsymbol{J}^{\#} \boldsymbol{J}_{\boldsymbol{q}} \boldsymbol{u} \tag{39}$$

$$\frac{\partial f}{\partial \boldsymbol{u}} = \boldsymbol{I} - \boldsymbol{J}^{\#} \boldsymbol{J} \tag{40}$$

$$\frac{\partial l_v}{\partial \boldsymbol{q}} = \frac{\partial}{\partial \boldsymbol{q}} \left(\dot{\boldsymbol{q}}^T \dot{\boldsymbol{q}} \right) = 2 \left(\frac{\partial f}{\partial \boldsymbol{q}} \right)^T \dot{\boldsymbol{q}} \tag{41}$$

$$\frac{\partial l_v}{\partial \boldsymbol{u}} = 2 \left(\frac{\partial f}{\partial \boldsymbol{u}} \right)^T \dot{\boldsymbol{q}} \tag{42}$$

$$\frac{\partial l_l}{\partial \boldsymbol{q}} = \begin{cases} \sum\limits_{i=1}^{n} 2 \frac{\left(q_i - \bar{q}_{M,i}\right)}{\left(q_{M,i} - \bar{q}_{M,i}\right)^2} & \text{if } q_i > \bar{q}_{M,i} \\ \sum\limits_{i=1}^{n} 2 \frac{\left(q_i - \bar{q}_{m,i}\right)}{\left(q_{m,i} - \bar{q}_{m,i}\right)^2} & \text{if } q_i < \bar{q}_{m,i} \\ 0 & \text{else} \end{cases} \tag{43}$$

$$\boldsymbol{J}_{\boldsymbol{q}}{}^{\#} = \frac{\partial \boldsymbol{J}^{\#}}{\partial \boldsymbol{q}} = \partial_{\boldsymbol{q}} \boldsymbol{J}^T \left(\boldsymbol{J} \boldsymbol{J}^T \right)^{-1} + \boldsymbol{J}^T \Big[- \left(\boldsymbol{J} \boldsymbol{J}^T \right)^{-1} \dots \\ \Big((\partial_{\boldsymbol{q}} \boldsymbol{J}) \boldsymbol{J}^T + \boldsymbol{J} (\partial_{\boldsymbol{q}} \boldsymbol{J}^T) \Big) \left(\boldsymbol{J} \boldsymbol{J}^T \right)^{-1} \Big] \tag{44}$$

Gradients for Optimization on Acceleration Level

$$\frac{\partial f_1}{\partial \boldsymbol{q}} = \frac{\partial f}{\partial \boldsymbol{q}}, \quad \frac{\partial f_1}{\partial \boldsymbol{u}} = \boldsymbol{I} - \boldsymbol{J}^{\#} \boldsymbol{J} \tag{45}$$

$$\frac{\partial f_2}{\partial \boldsymbol{q}} = \frac{\partial f_2}{\partial \boldsymbol{u}} = \boldsymbol{0} \tag{46}$$

$$\frac{\partial l_{na}}{\partial \hat{\boldsymbol{u}}} = 2 \hat{\boldsymbol{u}}, \quad \frac{\partial f_2}{\partial \hat{\boldsymbol{u}}} = \boldsymbol{I} \tag{47}$$

REFERENCES

[1] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 2, pp. 47 –53, June 1969.

[2] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, pp. 868–871, 1977.

[3] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, pp. 201–212, 1990, 10.1007/BF00126069.

[4] J. Baur, J. Pfaff, H. Ulbrich, and T. Villgrattner, "Design and development of a redundant modular multipurpose agricultural manipulator," in *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, Kaohsiung, Taiwan, July 2012, pp. 823–830.

[5] L. S. Pontryagin, *The mathematical theory of optimal processes*. New York [u.a.]: Interscience Publ., 1962.

[6] E. Todorov and W. Li, "Optimal control methods suitable for biomechanical systems," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, vol. 2. IEEE, 2003, pp. 1758–1761.

[7] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," *Int. J. Rob. Res.*, vol. 6, pp. 32–42, March 1987.

[8] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. New York, New York, USA: American Elsevier Publishing Company, Inc., 1970.

[9] L. Lasdon, S. Mitter, and A. Waren, "The conjugate gradient method for optimal control problems," *IEEE Transactions on Automatic Control*, vol. 12, no. 2, pp. 132–138, Apr. 1967.

[10] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, May 2009, pp. 489–494.

[11] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, Sept. 2013.

[12] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," in *Proceedings of Robotics: Science and Systems*, 2013.

[13] A. El Khoury, F. Lamiraux, and M. Taix, "Optimal motion planning for humanoid robots," in *2013 IEEE International Conference on Robotics and Automation*, vol. 0, no. 1. Ieee, May 2013, pp. 3136–3141.

[14] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[15] M. Vande Weghe, D. Ferguson, and S. S. Srinivasa, "Randomized path planning for redundant manipulators without inverse kinematics," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Nov. 2007, pp. 477–482.

[16] H. Kim, K. Lee, S. Lee, and M. Park, "Obstacle avoidance navigation using a local path planning method," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 1. IEEE, 1996, pp. 869–874.

[17] S. Seereeram and J. Wen, "A global approach to path planning for redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 152–160, 1995.

[18] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2012, pp. 4906–4913.

[19] J. Baur, J. Pfaff, C. Schuetz, and H. Ulbrich, "Dynamic modeling and realization of an agricultural manipulator," in *Proceedings of XV International Symposium on Dynamic Problems of Mechanics, DINAME*, 2013.

[20] E. Polak, "An Historical Survey of Computational Methods in Optimal Control," *SIAM Review*, vol. 15, pp. 553–584, 1973.

[21] T. Buschmann, S. Lohmeier, H. Ulbrich, and F. Pfeiffer, "Dynamics simulation for a biped robot: modeling and experimental verification," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, May 2006, pp. 2673 –2678.

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[23] M. Diehl, H. Ferreau, and N. Haverbeke, "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation," in *Nonlinear Model Predictive Control SE - 32*, ser. Lecture Notes in Control and Information Sciences, L. Magni, D. Raimondo, and F. Allgöwer, Eds. Springer Berlin Heidelberg, 2009, vol. 384, pp. 391–417.

[24] F. Chaumette and T. Marchand, "A redundancy-based iterative approach for avoiding joint limits: application to visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 719–730, 2001.

[25] M. Schwienbacher, T. Buschmann, S. Lohmeier, V. Favot, and H. Ulbrich, "Self-collision avoidance and angular momentum compensation for a biped humanoid robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 581 –586.