IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

An Inverse Kinematics Algorithm with Smooth Task Switching for Redundant Robots

Hannes Gamper[1,2], Laura Rodrigo Pérez[1], Andreas Mueller[2],
Alejandro Díaz Rosales[1,3] and Mario Di Castro[1]

***Abstract*—This paper presents an inverse kinematics approach that combines two well-known Jacobian based methods, the task-priority framework and an optimization-based approach, such that tracking and optimization tasks can be executed simultaneously. The novelty of the proposed algorithm lies in the ability to smoothly switch between different tasks and thus to allow for a seamless and safe transition during robot operation. This has shown to improve the efficiency and user experience, especially during tele-operated interventions in complex environments.**

***Index Terms*—Kinematics, Redundant Robots, Motion Control.**

I. Introduction

THE teleoperation control of redundant robots requires dedicated inverse kinematic resolutions, that support the operators during operations in order to increase the efficiency of the task execution and at the same time decrease the psychological stress of the operators. As an example, the robotic fleet at CERN is used for remote interventions and maintenance tasks throughout the entire CERN complex. Many robotic interventions are unplanned and still require human perception and cognition to assess the situation and act accordingly, hence many operations are conducted using teleoperated robots. With the increasing complexity of interventions, the deployed manipulators also increase in dexterity, leading to highly redundant topologies. The desired support features of the inverse kinematics algorithm, besides the tracking of a desired trajectory for the end effector (EE), were requested by CERN robot operators after difficult and complex interventions. The requirements can be assigned to either tracking or optimization tasks, in the following referred to as $\mathbf{z}$ and τ, respectively. Examples of requested features that resemble tracking tasks $\mathbf{z}$ are:

- Following a desired trajectory under the constraint of one or more locked joints.
- Taking joint limits into account.
- Avoiding collisions with the environment.

Examples for optimization tasks τ are:

- Minimizing the error propagation throughout the manipulator in certain directions.
- Generating cyclic trajectories.
- Minimizing the motor torques.

Operators also requested to control a mobile robot by either moving only the base or only the robotic arm, thus splitting the interventions in prepositioning and fine manipulation sequences. Additionally, it was important for our operators to seamlessly switch between these tasks and not being forced to stop the movement between the tasks. This requires a seamless task switching using continuous switching profiles. In this paper an algorithm for smooth task switching is proposed, that combines two widely used techniques for solving the inverse kinematics of redundant robots, the optimization-based approach [1] and the task prioritization that relies on the augmented Jacobian [3] in a closed-loop inverse kinematics (CLIK) tracking problem. It further facilitates smooth task switching and adjustment of the agility of joints. The algorithm can be used for online resolution of the inverse kinematics but is of course applicable offline or to preplanned trajectories. A discussion on related work is presented in Section II. In Section III the proposed algorithm is presented and the stability proof is shown in Section IV. In Section V the materials, methods and the testing environment is discussed. The experimental results validate what we claim in the theoretical part and are presented in Section VI. Conclusions are given in Section VII.

II. Related Work

Computationally efficient and robust inverse kinematics algorithms have been an active field of research for decades and are still being investigated by researchers today. Extensive publications on reviews of existing methods (analytic, Jacobian-based, Newton-based, quadratic programming, data-driven, statistical and heuristic) have been published in [4], [5], [6] and [1]. Analytic approaches are very limited in terms of degree of freedom (DoF $\leq$ 6) whereas heuristic algorithms do not allow for secondary objectives. Data-driven approaches, quadratic programming and statistical methods do facilitate secondary objectives, but they are more complex and computationally expensive, which is problematic in real-time applications. Especially for redundant kinematics, the Jacobian based methods allow the implementation of practical techniques of redundancy resolution, and reviews on closed-loop inverse kinematics techniques [4] have shown that the damped least squares pseudo-inverse is still one of the most

Manuscript received: July, 15th, 2023; Accepted: March, 4th, 2024.
This paper was recommended for publication by Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments.
[1]European Organization for Nuclear Research (CERN), Espl. des Particules 1, 1211 Meyrin, Switzerland.
[2]Institute of Robotics, Johannes Kepler University Linz, Altenbergerstraße 69, 4040 Linz, Austria.
[3]Department of Cognitive Robotics, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.
Digital Object Identifier (DOI): see top of this page.

simple, computationally efficient and robust approach. Therefore, the proposed inverse kinematics algorithm with smooth task switching for redundant robots, is based on the Jacobian method in a closed-loop, damped, least squares pseudo-inverse form.

Resolving the redundancy of robotic manipulators in useful ways has been an active field of research on its own. Capable tools like the augmented or extended Jacobian [3], quadratic programming frameworks [7], the task-priority [8] and set-based task-priority [9] framework are taking additional objectives or measures into account, to exploit redundancy. However, none of these techniques allow for seamless enabling/disabling of the additional objectives, which is especially useful during teleoperated interventions in complex environments. An algorithm is presented in Section III that is simple, yet capable of toggling additional objectives or in other words, introduces smooth task switching.

III. Smooth Task Switching Algorithm

The algorithm implements a combination of two common Jacobian-based inverse kinematics approaches, the augmented Jacobian and an optimization-based approach, to implement our operator's requirements for tracking and optimization tasks. The task tracking problem is addressed in [8] [10] and the task optimization is a result of the Jacobian-based inverse kinematics resolution [1]. Additionally, switching parameters were introduced to implement the smooth task switching as well as the adjustment of the joints agility. In Section III-A the developed inverse kinematics algorithm will be derived and Section III-B presents the definition of the switching parameters.

A. Closed-Loop Inverse Kinematics

Denote with $\mathbf{q}(t) \in \mathbb{R}^\delta$ the vector of time dependent joint coordinates (generalized coordinates), where δ is the DoF of the robotic manipulator and $r \leq \delta$ the dimension of the task space. The tracking error is introduced as

$$\boldsymbol{\theta}(t, \mathbf{q}(t)) = \mathbf{z}_{ref}\left(t\right) - \mathbf{z}\left(\mathbf{q}\left(t\right)\right) \in \mathbb{R}^r, \tag{1}$$

where $\mathbf{z}_{ref}$ and $\mathbf{z}$ describe the reference and the actual tracking task. If the task describes the EE pose, it should be noticed that $\mathbf{z}$ are local coordinates, which is valid and singularity-free for small tracking errors. In full generality, the error is defined on the Lie group $SE\left(3\right)$, representing EE pose by means of homogenous transformations. As discussed in [2, Sec. 3.7.3 - Angle and axis], the above formulation is exact for small errors, and will be used throughout the paper. The time derivative of the tracking error is

$$\dot{\boldsymbol{\theta}} = \frac{\partial \boldsymbol{\theta}}{\partial t} + \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{q}}\dot{\mathbf{q}} = \dot{\mathbf{z}}_{ref} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \tag{2}$$

with the Jacobian $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{r\times\delta}$. To ensure the stability of the tracking task an error feedback is applied, designing the asymptotically stable error dynamics

$$\mathbf{0} = \dot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta}, \tag{3}$$

for the diagonal matrix $\mathbf{K} > 0$. Substituting (3) into (2) leads to the well-known CLIK architecture for the inverse problem

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\left(\dot{\mathbf{z}}_{ref} + \mathbf{K}\boldsymbol{\theta}\right). \tag{4}$$

In order to deal with non-square matrices, avoid singularities and set a priority for the movement of the different joints, the weighted and damped, least-squares pseudo inverse $\mathbf{J}^\dagger$, as a result of the optimization problem, see Appendix VII-B, will be applied for the Jacobian inverse, leading to

$$\dot{\mathbf{q}} = \underbrace{\mathbf{W}^{-1}\mathbf{J}^T\left(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T + \alpha(\mathbf{q})\mathbf{I}\right)^{-1}}_{\mathbf{J}^\dagger}\left(\dot{\mathbf{z}}_{ref} + \mathbf{K}\boldsymbol{\theta}\right), \tag{5}$$

with the weighing matrix

$$\mathbf{W} = \mathrm{diag}\{\sigma_{W1},\ \sigma_{W2},\ \ldots,\ \sigma_{W\delta}\}. \tag{6}$$

The definition of the weights $\boldsymbol{\sigma}_W = [\sigma_{W1},\ \sigma_{W2},\ \ldots,\ \sigma_{W\delta}]^T$ are described in Section III-B. The choice of the damping factor $\alpha(\mathbf{q})$ is shown in the Appendix VII-A. In the case of a redundant kinematic system, an artificial potential $\nabla\tau^T\dot{\mathbf{q}}$ can be added to the cost function of the optimization problem, leading to

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger\left(\dot{\mathbf{z}}_{ref} + \mathbf{K}\boldsymbol{\theta}\right) + \underbrace{\left(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J}\right)}_{\mathbf{N}}\mathbf{W}^{-1}\frac{\partial\tau}{\partial\mathbf{q}}, \tag{7}$$

with the null-space projector $\mathbf{N}$. In order to assign a priority to the tracking tasks, the augmented Jacobian and its corresponding null-space projector

$$\mathbf{J}_{1,2} = \begin{bmatrix}\mathbf{J}_1\\ \mathbf{J}_2\end{bmatrix} \in \mathbb{R}^{n\times\delta} \text{ and } \mathbf{N}_{1,2} = (\mathbf{I} - \mathbf{J}_{1,2}^\dagger\mathbf{J}_{1,2}) \in \mathbb{R}^{\delta\times\delta}, \tag{8}$$

with $n \leq \delta$ have to be introduced [1]. This allows to incorporate the two tracking tasks with the corresponding references $\mathbf{z}_{ref,1}$, $\mathbf{z}_{ref,2}$ and actual tasks $\mathbf{z}_1, \mathbf{z}_2$. The solution for the velocity simultaneously minimizing τ is

$$\begin{aligned}\dot{\mathbf{q}} = {}& \mathbf{J}_1^\dagger\left(\dot{\mathbf{z}}_{ref,1} + \mathbf{K}_1\boldsymbol{\theta}_1\right)\\ &+ \sigma_\theta(t)\underbrace{\left(\mathbf{I} - \mathbf{J}_1^\dagger\mathbf{J}_1\right)}_{\mathbf{N}_1}\mathbf{J}_2^\dagger\left(\dot{\mathbf{z}}_{ref,2} + \mathbf{K}_2\boldsymbol{\theta}_2\right)\\ &+ \sigma_\tau(t)\underbrace{\left(\mathbf{I} - \mathbf{J}_{1,2}^\dagger\mathbf{J}_{1,2}\right)}_{\mathbf{N}_{1,2}}\mathbf{W}^{-1}\frac{\partial\tau}{\partial\mathbf{q}},\end{aligned} \tag{9}$$

with the switching parameters $\sigma_\theta(t)$ and $\sigma_\tau(t)$, as defined in Section III-B, for smooth enabling/disabling of the tracking and optimization tasks.

B. Switching Parameter

To provide smooth switching of the tasks and weights, any trajectory from differentiability class C^3 can be applied. Here, a base profile $\sigma_B(t) \in \mathbb{R}$ will be defined on $t \in \mathbb{R}$ such that $\sigma_B(t) \in [0, 1]$ and $\sigma_B(t)$ is strictly increasing. Applying a $\sin^2$ function satisfies all requirements and can be piece-wise defined in acceleration as

$$\ddot{\sigma}_B(t) = \begin{cases}0 & t < 0\\ a_{max}\sin(\frac{2\pi}{t_S}t)^2 & 0 \leq t < \frac{t_S}{2}\\ -a_{max}\sin(\frac{2\pi}{t_S}t)^2 & \frac{t_S}{2} \leq t < t_S\\ 0 & t_S \leq t,\end{cases} \tag{10}$$

with the maximum acceleration $a_{max} > 0$ and the switching time t_S. This leads to the final switching profile

$$\sigma(t) = \sigma_S + \sigma_B(t)(\sigma_E - \sigma_S), \tag{11}$$

with the start and end weights $\sigma_S,\ \sigma_E \in \mathbb{R}$, which will be used for each $\sigma_{W1}(t) \dots \sigma_{W\delta}(t)$ in (6) and for $\sigma_\theta(t)$ and $\sigma_\tau(t)$ in (9). This enables the user to switch tasks on and off by only providing the duration for the switching process.

IV. Stability Analysis

The stability proofs for both the tracking and the optimization task have been shown in several papers before [1][8][10]. Following them, we are demonstrating the stability proof for the combined system and especially taking into account the newly introduced switching parameters. In Section IV-A, it will be demonstrated that the error dynamics of the tracking tasks are asymptotically stable and in Section IV-B an upper bound for the gain $\mathbf{K}$ for discrete systems is given.

A. Stability of the Tracking Tasks

First, equation (9) can be simplified, by omitting the third term resembling the optimization task. This step is justified because of the following two points: First, the optimization task always needs to be set up as the lowest priority task and thus does not influence the error dynamics of the higher priority tasks. Second, the optimization task has no reference and thus it is impossible to define the error dynamics for this term. However, to guarantee a feasible optimization task it needs to be linearly independent (compatible) [1, Sec. 10.5.3] from higher priority tasks, that is satisfying

$$\operatorname{rank}\left(\mathbf{J}_{1,2}^{\dagger}\right) + \operatorname{rank}\left(\frac{\partial \tau}{\partial \mathbf{q}}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{J}_{1,2}^{\dagger} & \frac{\partial \tau}{\partial \mathbf{q}}\end{bmatrix}\right). \tag{12}$$

In order to prove the stability of the tracking tasks in (9) we rewrite the task errors in the form

$$\bar{\boldsymbol{\theta}} = \begin{bmatrix}\boldsymbol{\theta}_1^T & \boldsymbol{\theta}_2^T\end{bmatrix}^T \tag{13}$$

and the partial time derivatives of the error in the form

$$\frac{\partial \bar{\boldsymbol{\theta}}}{\partial t} = \begin{bmatrix}\frac{\partial \boldsymbol{\theta}_1}{\partial t}^T & \frac{\partial \boldsymbol{\theta}_2}{\partial t}^T\end{bmatrix}^T = \dot{\bar{\mathbf{z}}}_{ref} \tag{14}$$

With this notation, we can now rewrite (9) in the form

$$\dot{\bar{\boldsymbol{\theta}}} = -\underbrace{\begin{bmatrix}\mathbf{K}_1 & \mathbf{0} \\ \mathbf{J}_2\mathbf{J}_1^{\dagger}\mathbf{K}_1 & \sigma_\theta(t)\mathbf{J}_2\mathbf{N}_1\mathbf{J}_2^{\dagger}\mathbf{K}_2\end{bmatrix}}_{\mathbf{A}(t)} \bar{\boldsymbol{\theta}} + \underbrace{\begin{bmatrix}\mathbf{I} - \mathbf{J}_1\mathbf{J}_1^{\dagger} & \mathbf{0} \\ -\mathbf{J}_2\mathbf{J}_1^{\dagger} & \sigma_\theta(t)\left(\mathbf{I} - \mathbf{J}_2\mathbf{N}_1\mathbf{J}_2^{\dagger}\right)\end{bmatrix}}_{\mathbf{B}(t)} \dot{\bar{\mathbf{z}}}_{ref} \tag{15}$$

yielding a non-autonomous error system. The lower right term in matrix $\mathbf{B}(t)$ disappears if and only if the task $\mathbf{J}_2$ is fully represented in the nullspace $\mathbf{N}_1$ and thus we assume

$$\mathbf{J}_2\mathbf{N}_1\mathbf{J}_2^{\dagger} = \mathbf{J}_2\mathbf{J}_2^{\dagger}, \tag{16}$$

leading to $\mathbf{N}_1 = \mathbf{I}$ in case $\mathbf{J}_2$ has full rank [10]. Then, the diagonal elements of matrix $\mathbf{B}(t)$ disappear because $\mathbf{J}_1^{\dagger}$ and $\mathbf{J}_2^{\dagger}$ are the right inverse of $\mathbf{J}_1$ and $\mathbf{J}_2$, respectively. The off diagonal elements in $\mathbf{A}(t)$ and $\mathbf{B}(t)$ disappear due to the annihilation condition $\mathbf{J}_2\mathbf{J}_1^{\dagger} = \mathbf{0}$ [8].

In order to test the error system for stability, we define a scalar Lyapunov candidate function

$$V\left(t, \bar{\boldsymbol{\theta}}\right) = \frac{1}{2}\bar{\boldsymbol{\theta}}^T\bar{\boldsymbol{\theta}} \tag{17}$$

which is bounded from below by definition and its time derivative is

$$\dot{V}\left(t, \bar{\boldsymbol{\theta}}\right) = \bar{\boldsymbol{\theta}}^T\dot{\bar{\boldsymbol{\theta}}} = -\bar{\boldsymbol{\theta}}^T\mathbf{A}(t)\bar{\boldsymbol{\theta}}. \tag{18}$$

The upper left submatrix of $\mathbf{A}(t)$ needs to be positive semidefinite and thus satisfy $\mathbf{K}_1 > 0$. The lower right submatrix is positive definite if the gains satisfy $\mathbf{K}_2 > 0$, $\sigma_\theta(t) > 0$, $\boldsymbol{\sigma}_W(t) > 0$ and if the first two tasks are linearly independent

$$\operatorname{rank}\left(\mathbf{J}_1^{\dagger}\right) + \operatorname{rank}\left(\mathbf{J}_2^{\dagger}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{J}_1^{\dagger} & \mathbf{J}_2^{\dagger}\end{bmatrix}\right). \tag{19}$$

and not annihilating [8], hence

$$\mathbf{J}_1\mathbf{J}_2^{\dagger} \neq \mathbf{0}. \tag{20}$$

In case $\sigma = 0$ the task is disabled, will not be tracked and thus is obviously not asymptotically stable. The proof can then be similarly shown with only the first tracking task. If the two sub-matrices of the main diagonal of $\mathbf{A}(t)$ are positive definite, then the matrix $\mathbf{A}(t)$ itself is positive definite and thus,

$$\dot{V}\left(t, \bar{\boldsymbol{\theta}}\right) < 0. \tag{21}$$

Since (15) resembles a non-autonomous system we employ Barbalat's Lemma to test for stability. Thus, we need to show that $\dot{V}\left(t, \bar{\boldsymbol{\theta}}\right)$ is uniformly continuous, which can be derived from the sufficient condition of

$$\ddot{V}\left(t, \bar{\boldsymbol{\theta}}\right) = -\bar{\boldsymbol{\theta}}^T\left(\mathbf{A}(t)^T\mathbf{A}(t) + \dot{\mathbf{A}}(t) + \mathbf{A}(t)^2\right)\bar{\boldsymbol{\theta}}, \tag{22}$$

being bounded. This condition holds if $\sigma(t)$, $\mathbf{A}(t)$ and $\dot{\mathbf{A}}(t)$ are bounded, which are satisfied as $\sigma(t) \in [0, 1]$ and $\sigma(t)$ is C^3 continuous, while $\mathbf{A}(t)$ and $\dot{\mathbf{A}}(t)$ are bounded, because $\mathbf{J}_1$, $\mathbf{J}_2$ contain only C^1 continuous functions and the pseudo inverses are damped. Thus, $\lim_{t\to\infty}\dot{V}(t, \bar{\boldsymbol{\theta}}) = 0$ holds and therefor also

$$\lim_{t\to\infty}\bar{\boldsymbol{\theta}} = \mathbf{0}. \tag{23}$$

B. Discrete Systems

An extensive discussion on the stability for discrete systems is presented in [11]. For our simple case, some bounds for the gains $\mathbf{K} = \operatorname{diag}([k_1, k_2, ..., k_n])$ of the discrete system corresponding to (3) can be found via the steps shown below. The discrete system

$$\frac{\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}}{\Delta t} = \mathbf{K}\boldsymbol{\theta}, \tag{24}$$

which can be rewritten as a discrete dynamic system

$$\boldsymbol{\theta}_{k+1} = \left(\mathbf{I} - \Delta t\mathbf{K}\right)\boldsymbol{\theta}_k, \tag{25}$$

that is asymptotically stable if the condition

$$||\operatorname{eig}\left\{\mathbf{I} - \Delta t\mathbf{K}\right\}|| < 1 \tag{26}$$

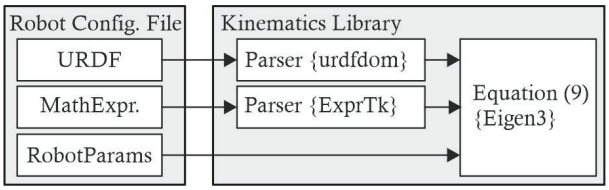

Fig. 1. The architecture of the Kinematics Library. The robot configuration file holds the geometric description of the mechanical system in either the URDF format or hard coded mathematical expressions, that will be parsed using the urdfdom or ExprTk library. The block that implements (9) uses Eigen3. The configuration file also holds all other parameters that are relevant for the motion controller of the CRF.

holds, which leads to lower and upper bounds for the gain $\mathbf{K}$ in the form

$$0 < k_i < \frac{2}{\Delta t}. \tag{27}$$

V. Test Environment

The presented algorithm is part of a new robot motion controller for the CERN Robotic Framework (CRF) [12]. The kinematics library was developed in C++ and the only external libraries used, are *Eigen3* [13], *urdfdom* [15] and *ExprTk* [14], see Figure 1. The code architecture was designed, taking into account expandability for future implementations and improvements, while still ensuring robustness with proper memory handling and a low computational cost during the main robot control loop. The inverse kinematics algorithm was deployed on a real-time system in a main control loop with a loop time of 2 ms. The computational complexity of the algorithm can not be stated in general, since it depends on the specific tasks one chooses to consider. However, in Section VI the computation times for the most complex scenario are presented.

All tests were done using a Ubuntu system with a Real-Time Linux Preempt Kernel patch [16] launching the process with the highest priority in the scheduler. It is worth mentioning that the system features an 11$^{\text{th}}$ Gen Intel® Core™ i5-1145G7E processor. The algorithm was tested in three scenarios, exploiting different tasks on different robotic systems, described in the following Sections V-A - V-C.

A. Test Scenario 1 - Train Inspection Monorail - 9 DoF

This test was performed using the Train Inspection Monorail (TIM) [17] [18] in the Large Hadron Collider (LHC) main tunnel at CERN. TIM is a robotic system driving along the LHC tunnel on a single rail that is mounted on the ceiling. The first two joints of the manipulator are rotational and the third one creates a translational movement. On the third, linear axis, a 6 DoF Kinova Jaco2 arm is mounted resulting in a total of 9 DoF, see Figure 2. This robotic arm is used for remote inspections and maintenance activities, like sensor testing and calibration or radiation measurements. For this test, the EE of the robot will track a six-dimensional reference $\mathbf{z}_{ref,1}$, leading to a redundancy with degree three. The tracking of the EE will be encoded as the highest priority task $\mathbf{z}_1$ in the first term in (9). The lower priority tracking task $\mathbf{z}_2$ will be used to block

Fig. 2. The Train Inspection Monorail (TIM) in the LHC tunnel.

joint q_5, while the robot is tracking the desired EE position. The optimization task τ will drive the robot away from the limits of joints q_1, q_2, q_3 and q_4 and will be enabled throughout the entire test. The test will show the behavior of the robotic system during the switching process, that enables/disables the tracking task $\mathbf{z}_2$ (in other words, blocks or unblocks joint q_5) using the switching parameter $\sigma_\theta(t)$.

B. Test Scenario 2 - Universal Robot UR10e - 6 DoF

The second test will be executed with the 6 DoF robot UR10e from Universal Robot. In order to demonstrate the capabilities of the presented algorithm, the redundancy is created by a reduction of the task space dimensions. Specifically, the EE will track a three-dimensional reference $\mathbf{z}_{ref,1}$ (only including the desired position), such that the robot resembles a redundant system with degree three. The tracking of the EE will be encoded as the highest priority task $\mathbf{z}_1$ in the first term in (9). The optimization task τ will be used to block certain joints, while the robot is tracking the desired EE position. The test will also show the behavior of the robotic system during the switching process, that enables/disables the optimization task (in other words, blocks or unblocks certain joints) using the switching parameter $\sigma_\tau(t)$.

C. Test Scenario 3 - Mobile Platform with Arm - 9 DoF

In this scenario, the robotic system is a mobile base with four mecanum wheels and a 6 DoF Kinova Jaco 2 arm mounted on top, resulting in a system with 9 velocity DoF and 10 joints to be controlled, see Figure 3. The EE will track a six-

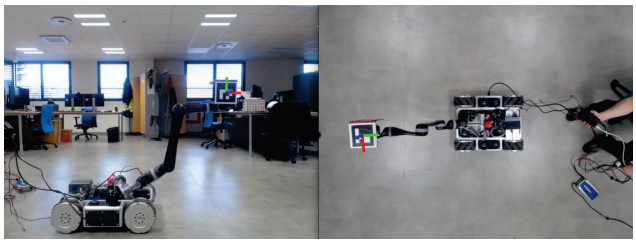

Fig. 3. Omnidirectional base with a Kinova Jaco2 with spherical wrist

dimensional reference, which will be encoded as the highest priority task $\mathbf{z}_1$ in the first term in (9). In this test, no additional tracking or optimization tasks are considered, but instead, the smooth switching of the weighted mobility of the single joints

will be demonstrated. This is especially useful when a mobile platform with an attached arm has to be prepositioned to a point of interest (that lies outside the workspace of the attached arm) using the wheels of the robot and once this point is reached, will handle the fine manipulation of an object, using the joints of the robotic arm only. The test will show the behavior of the robotic system while switching the mobility of the joints using the parameter $\boldsymbol{\sigma}_W(t)$. The measurement of the actual task $\mathbf{z}_1$ was done, using two cameras and two ArUco markers [19], see Figure 3. The marker planes were held parallel to the camera sensor plane throughout the entire test. This was done because the precision of the cameras is higher for measurements parallel to the camera sensor as opposed to measurements perpendicular to the camera sensor. The combination of the data from both cameras then allowed to reconstruct the actual task $\mathbf{z}_1$.

VI. Experimental Results

The acquired data during the tests, described in the previous Sections V-A - V-C, are presented and discussed in the following Sections VI-A - VI-C for the Train Inspection Monorail, the Universal Robot UR10e and the mobile platform with arm, respectively. The average computation time of the inverse kinematics algorithm in test scenario 3 was 82.5 µs, with a minimum of 41.3 µs and a maximum of 819.6 µs. As expected, test scenario 3 showed the highest computation time, since it implemented the most complex setup.

A. Test Scenario 1 - Train Inspection Monorail - 9 DoF

The desired EE pose $\mathbf{z}_{ref,1} \in \mathbb{R}^6$ for this test was set constant and represents the reference to the highest priority tracking task. The tracking error $\boldsymbol{\theta}_1$ is shown in Figure 4a. The lower priority task will block joint q_5 at a desired position $\mathbf{z}_{ref,2}$ if $\sigma_\theta = 1$ holds. The tracking error θ_2 is shown Figure 4b. Additionally, the optimization task

$$\tau = \sum_i -p_i \left(e^{c_i(q_i - q_{max,i})} + e^{-c_i(q_i - q_{min,i})} \right), \qquad (28)$$

with $i = 1 \dots 4$, will drive the robot away from the limits $q_{max,i}$ and $q_{min,i}$ of joints q_i, if $\sigma_\tau = 1$ holds. The exponential terms in (28) allow the robot to freely move within the joint limits and generate a strong, repelling force close to the joint limits. The effect of τ can be tuned with the gains p_i and c_i. The gradient of the optimization task is shown in Figure 4e and the switching parameters σ_θ and σ_τ are shown in Figure 4f. The parameter σ_τ will be set to a constant value of 1 and thus the optimization task will be active during the entire test in order to avoid collisions. The tracking task, and thus the parameter σ_θ, will be switched on and off twice, blocking the joint q_5, first at $\mathbf{z}_{ref,2}(t) = \frac{\pi}{2}$ for $t \leq 44$ and then at $\mathbf{z}_{ref,2}(t) = \frac{\pi}{4}$ for $t > 44$. The dashed lines in Figure 4 indicate the start and end of the switching sequences.

In Figure 4a it is clearly visible that the EE follows the task reference $\mathbf{z}_{ref,1}$ with minor errors, which mainly occur when the secondary task is being enabled. The tracking error θ_2, see Figure 4b, is higher during the switching sequence as long as $\sigma_\theta < 1$, but disappears as soon as $\sigma_\theta = 1$. This is also visible

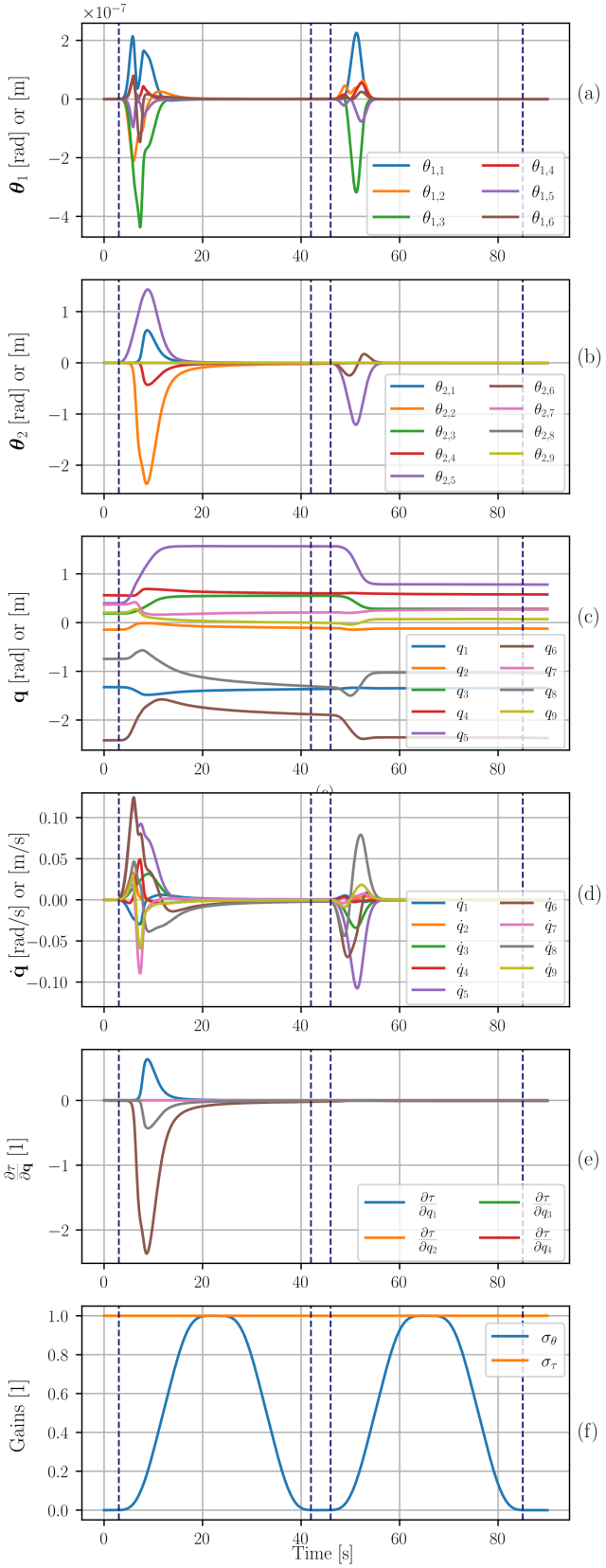

Fig. 4. Test data recorded with TIM in the LHC tunnel. Showing tracking error $\boldsymbol{\theta}_{1,i}$ for the i-th dimension (a), tracking error θ_2 (b), the inverse kinematics solution in joint angles $\mathbf{q}$ (c) and joint velocities $\dot{\mathbf{q}}$ (d), the gradient of the optimization task τ (e) and the switching parameters σ_θ and σ_τ (f).

in the plot of the joint angles (result of the inverse kinematics), see Figure 4c, for joint q_5 which is first blocked at $\frac{\pi}{2}$ rad and then at $\frac{\pi}{4}$ rad and in Figure 4d the joint velocities are shown. Since the robot starts in a relatively compact configuration, the optimization task is driving the robot away from the joint limits at the beginning of the sequence, as it is clearly visible in the plot of the gradient of the optimization task, see Figure 4e.

B. Test Scenario 2 - Universal Robot UR10e - 6 DoF

The desired EE trajectory in position $\mathbf{z}_{ref,1} \in \mathbb{R}^3$ is shown in Figure 5a. The trajectory resembles a linear movement in the three dimensions x, y and z. In this test, $\mathbf{z}_{ref,1}$ represents the reference for the highest priority tracking task. The tracking error $\boldsymbol{\theta}_1$ is shown in Figure 5b. The results of the redundancy resolution in joint space, are shown in Figure 5c and 5d with joint angles $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$, respectively. The gradient of the optimization task

$$\frac{\partial \tau}{\partial \mathbf{q}} = \mathbf{C}(\mathbf{q}_{des} - \mathbf{q}) \tag{29}$$

is shown in Figure 5e. $\mathbf{q}_{des} \in \mathbb{R}^{DoF}$ represents the vector with desired joint angles and $\mathbf{C}$ is a diagonal matrix with weighing gains in the main diagonal corresponding to the robot joints. Figure 5f shows the switching parameter σ_τ. The red and blue areas visualize the time during which the optimization task (blocking of joints) is being enabled/disabled ($0 < \sigma_\tau \leq 1$). In between the dashed lines in either of the red or blue areas, the optimization task is enabled ($\sigma_\tau = 1$). During the red area the optimization task with the desired joint angles $\mathbf{q}_{des,1} = \left[\sim, \sim, -\frac{pi}{2}, \sim, 2.5, \sim\right]$ will be enabled and during the blue area the desired joint angles are $\mathbf{q}_{des,2} = \left[\sim, \sim, \sim, \sim, \sim, 0\right]$. The corresponding joints with $\sim$ entries are free and will not be blocked.

In the red area, it can be observed that at the beginning of the switching process, the joints q_3 and q_5 start to move towards the desired angles as stated in $\mathbf{q}_{des,1}$. Then, in between the dashed lines, when the optimization task is enabled, the joints q_3 and q_5 hold their desired position. At the time the switching parameter starts disabling the task, the joints are free and start to move again according to the highest priority task reference $\mathbf{z}_{ref,1}$ with minimal joint velocities.

In the blue area, only joint q_6 will be blocked according to $\mathbf{q}_{des,2}$. At the beginning it is clearly visible that the joint is being pulled towards the zero line and is constant during the time, the task τ is enabled. Other than during the first sequence (red area), now the joint q_6 stays constant even after the joint has been unblocked again. This is first of all, because the desired EE trajectory $\mathbf{z}_{ref,1}$ is of dimension three and does not include the orientations and second, because the redundancy resolution is done with minimal joint velocities.

The task error $\boldsymbol{\theta}_1$ is shown in Figure 5b. It is clearly visible that the task switching is causing spikes in the position error. This is because the maximum accelerations of the single joints were not high enough to support a sufficiently fast repositioning of the robot into the new configuration. This effect can be avoided by increasing the switching time of the switching parameter σ_τ.

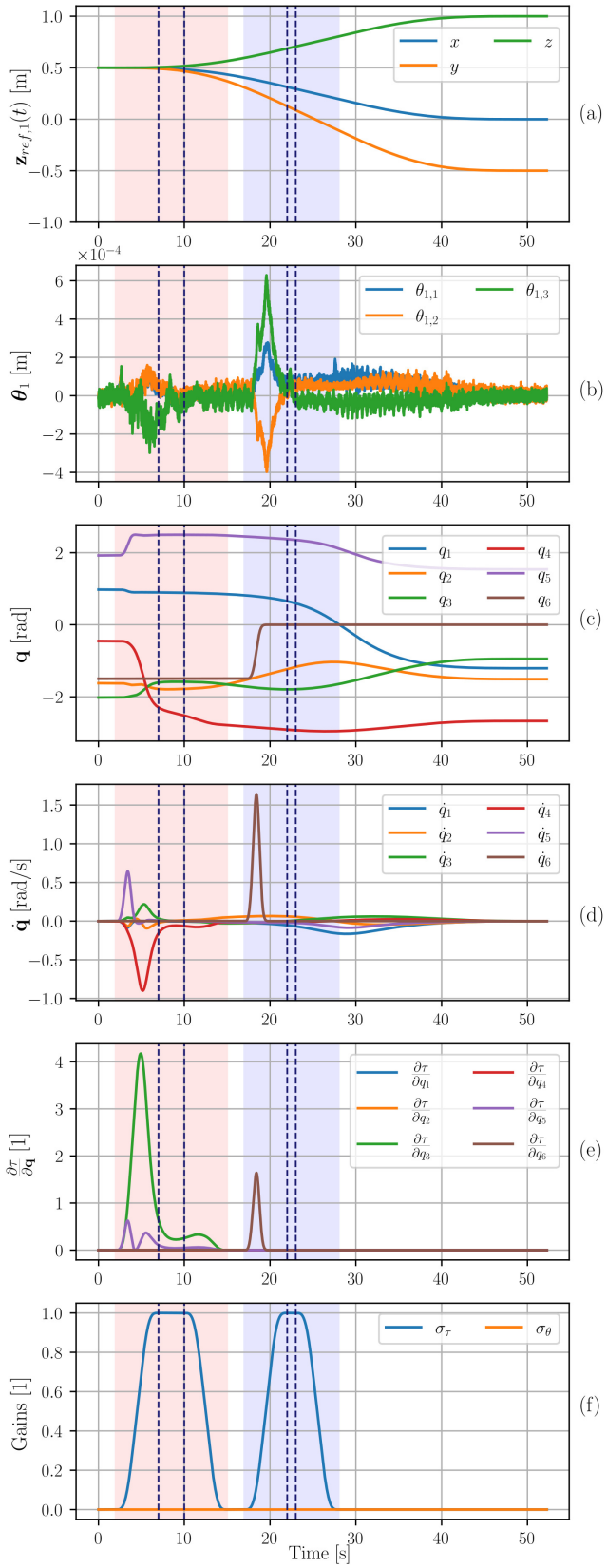

Fig. 5. Test data recorded with UR10e. Showing the desired task reference $\mathbf{z}_{ref,1}$ (a), tracking error $\boldsymbol{\theta}_1$ (b), the inverse kinematics solution in joint angles $\mathbf{q}$ (c) and joint velocities $\dot{\mathbf{q}}$ (d), the gradient of the optimization task τ (e) and the switching parameters σ_τ and σ_θ(f).

C. Test Scenario 3 - Mobile Platform with Arm - 9 DoF

The desired EE trajectory $\mathbf{z}_{ref,1} \in \mathbb{R}^6$ resembles a circle in the x-y plane with a constant orientation of the EE and lies at a constant height in the z-direction. This trajectory represents the reference for the highest priority tracking task. Figure 6a shows the tracking error $\boldsymbol{\theta}_1$, calculated with the actual EE position $\mathbf{z}_1$ from the data of two ArUco markers. The noise in these measurements is due to some false positive detections in the testing environment. A video of this experiment is available at http://ieee.org.

The results of the redundancy resolution in joint space are shown in Figures 6b and 6c for the joint angles of the mobile base $q_1 \dots q_4$ (wheels) and the manipulator $q_5 \dots q_{10}$. In Figures 6d and 6e the joint velocities of the mobile base and the manipulator are shown. The diagonal elements of matrix $\mathbf{W}$ in (5) are shown in Figure 6f and the blue area indicates the time during which the switching of the weights is taking place.

At the start of the trajectory the diagonal elements of $\mathbf{W}$ defined by $\boldsymbol{\sigma}_W$, see (11), are set to $\sigma_{W1..W4} = 1000$ and $\sigma_{W5..W10} = 1$, thus the desired EE trajectory will be followed by the 6 DoF manipulator exclusively. This is clearly visible in the plot of the wheel velocities which are all 0 before the blue area. In the blue area the weights are being switched to $\sigma_{W1..W4} = 1$ and $\sigma_{W5..W10} = 1000$. As shown in the plots of the joint velocities, Figure 6d and 6e, the wheels and the manipulator's joints are both moving during the switching process and after the blue area only the wheels are now creating the motion to follow the circular trajectory with the EE.

This feature increased the efficiency of operations during which a large workspace has to be covered and at the same time, fine manipulation tasks have to be performed. The base can then be used as a fast, less precise prepositioning system and the manipulator can later on be used for fine manipulation tasks while the wheels of the robot are blocked. All this can be done without stopping the movement or reconstructing objects in C++. At the time these tests were performed, no odometry has been implemented in the new motion controller yet and the inverse kinematics algorithm was launched without feedback regarding the actual position. Thus, all the accumulated errors like jerky motions of the Kinova at the beginning and end of the trajectory, or uneven floors are visible in the recorded EE position in Figure 6a.

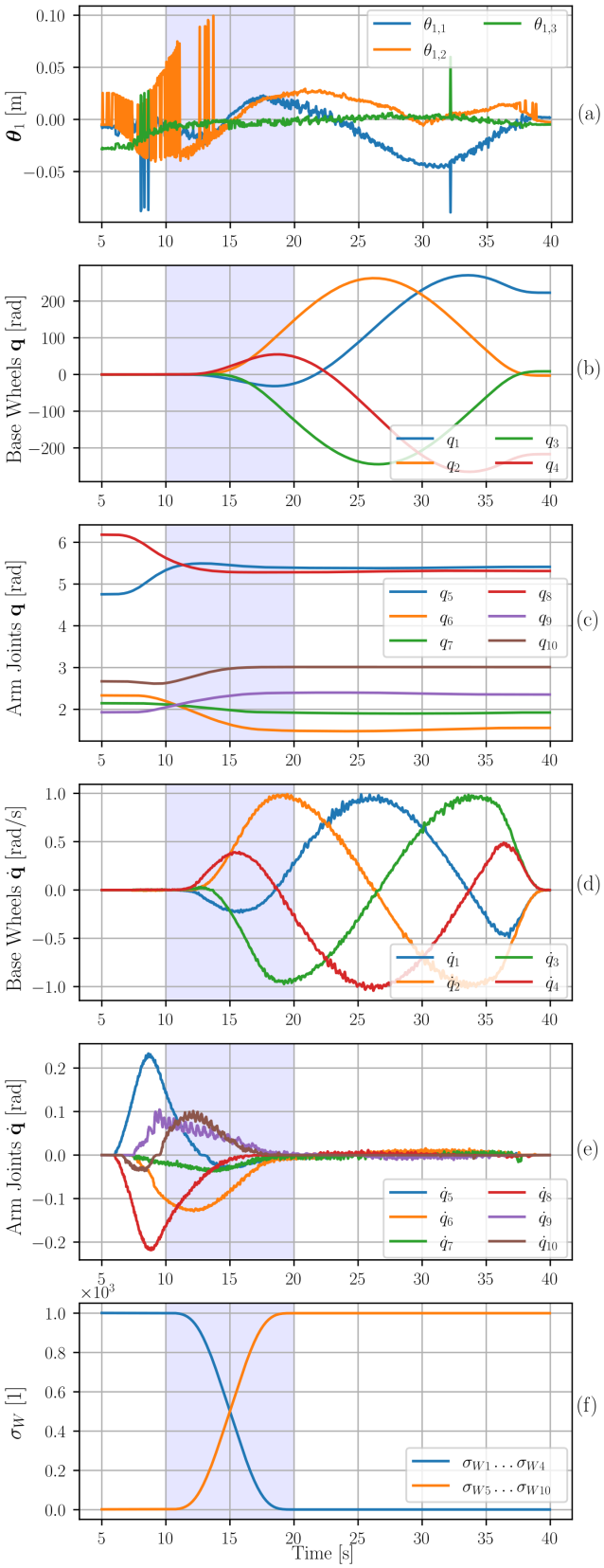

Fig. 6. Test data recorded with the mobile platform with arm. Showing tracking error $\boldsymbol{\theta}_1$ (a), the inverse kinematics solution in joint angles $\mathbf{q}$ (b) and (c) and joint velocities $\dot{\mathbf{q}}$ (d) and (e) and the weighing gains σ_{Wi} for the i-th dimension (f).

VII. Conclusions

The presented inverse kinematic approach combines the augmented Jacobian approach with an optimization based approach in order to incorporate tracking and optimization tasks. Furthermore, a smooth adjustment of the weights in the damped, least squares pseudo-inverse was implemented to control mobile robotic platforms, either moving only the base or only the robotic arm. The presented algorithm seems very promising to increase the efficiency and reduce the stress on operators, though quantified bench-mark human-study tests are still missing. The test results of the three experiments with a 6 DoF, and two 9 DoF systems are as

expected and showed stable behavior as desired. Nevertheless, there are some shortcomings with the presented algorithm: currently, only two tracking tasks can be taken into account. Disabling the second tracking task will result in this task not being tracked, but the nullspace will not be increased for the optimization task. In order to achieve this, the optimization task can be disabled as well and then the nullspace can be adjusted without problems. The maximum computation time for the most complex system (considering the presented hardware), lies below 1 ms, showing that the algorithm is well suited for real-time applications.

In summary, the presented algorithm is of great value for practical implementations and complex teleoperated interventions. Currently, efforts are being made to publish these developments as a standalone, open-source C++ library. Future work will focus on the formulation of optimization tasks that generate reliable cyclic trajectories and the computational complexity will be investigated in more detail for common use cases and presented in future publications.

APPENDIX

A. The Damping Factor

The damping factor $\alpha(\mathbf{q}) \in \mathbb{R}$ is set to

$$\alpha(\mathbf{q}) = \begin{cases} \alpha_0 \left(1 - \frac{w_{kin}(\mathbf{q})}{w_0}\right)^2 & w_{kin}(\mathbf{q}) < w_0 \\ 0 & w_{kin}(\mathbf{q}) \geq w_0 \end{cases} \tag{30}$$

where $\alpha_0, w_0 \in \mathbb{R}$ and the kinematic manipulability measure

$$w_{kin}(\mathbf{q}) = \sqrt{\det\{\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q})\}} \in \mathbb{R}. \tag{31}$$

B. The Optimization Based Redundancy Resolution

A solution to the inverse kinematics problem can be found formulating the constrained linear optimization problem,

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \tfrac{1}{2}\dot{\mathbf{q}}^T\mathbf{W}\dot{\mathbf{q}} + \nabla\tau^T\dot{\mathbf{q}} \\ \text{s.t.} \quad & \mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{z}}_{ref} = 0, \end{aligned} \tag{32}$$

which drives the redundant joints s.t. the joint velocities are minimal, exploits the redundancy according to the artificial potential energy $\tau(\mathbf{q})$ and defines the mobility of each joint by the positive definite weighing matrix $\mathbf{W}$. Formulating the corresponding Lagrangian function leads to the solution

$$\begin{aligned} \dot{\mathbf{q}} = {} & \mathbf{W}^{-1}\mathbf{J}^T\left(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T\right)^{-1}\dot{\mathbf{z}}_{ref} \\ & + \left(\mathbf{I} - \mathbf{W}^{-1}\mathbf{J}^T\left(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T\right)^{-1}\mathbf{J}\right)\mathbf{W}^{-1}\frac{\partial\tau}{\partial\mathbf{q}}, \end{aligned} \tag{33}$$

with the second term representing the redundancy resolution with respect to the optimization task τ. A detailed demonstration of the optimization problem and its solution can be found in [2, Sec. 3.5].

REFERENCES

[1] Siciliano, B. & Khatib Oussama, Handbook of Robotics *Springer Handbooks*. pp. 232-242 (2008,4)

[2] Bruno Siciliano & Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, Robotics - Modelling, Planning and Control *Springer London*. (2008)

[3] Siciliano, B. & Slotine, J. A general framework for managing multiple tasks in highly redundant robotic systems. (Institute of Electrical,2002,12)

[4] Colome, A. & Torras, C. Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements. *IEEE/ASME Transactions On Mechatronics*. **20** pp. 944-955 (2015,4)

[5] Aristidou, A., Lasenby, J., Chrysanthou, Y. & Shamir, A. Inverse Kinematics Techniques in Computer Graphics: A Survey. *Computer Graphics Forum*. **37**, 35-58 (2018,9)

[6] Singh, R., Kukshal, V. & Yadav, V. A Review on Forward and Inverse Kinematics of Classical Serial Manipulators.

[7] Rocchi, A., Hoffman, E., Caldwell, D. & Tsagarakis, N. OpenSoT: A whole-body control library for the compliant humanoid robot COMAN. *Proceedings - IEEE International Conference On Robotics And Automation*. **2015-June** pp. 6248-6253 (2015,6)

[8] Antonelli, G. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions On Robotics*. **25**, 985-994 (2009)

[9] Moe, S., Antonelli, G., Teel, A., Pettersen, K. & Schrimpf, J. Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results. *Frontiers Robotics AI*. **3** (2016,4)

[10] Arbo, M. & Gravdahl, J. Stability of the Tracking Problem with Task-Priority Inverse Kinematics. *IFAC-PapersOnLine*. **51** pp. 121-125 (2018,1)

[11] Falco, P. & Natale, C. On the stability of closed-loop inverse kinematics algorithms for redundant robots. *IEEE Transactions On Robotics*. **27**, 780-784 (2011,8)

[12] Mario Di Castro, Manuel Ferre, and Alessandro Masi. "CERNTAURO: A Modular Architecture for Robotic Inspection and Telemanipulation in Harsh and Semi-Structured Environments". In: IEEE Access 6 (2018), pp. 37506–37522. doi:10.1109/ACCESS.2018.2849572

[13] Guennebaud, G., Jacob, B. & Others. Eigen v3. (http://eigen.tuxfamily.org, 2010)

[14] Partow, A. C++ Mathematical Expression Toolkit Library (ExprTk). (https://www.partow.net/programming/exprtk/index.html)

[15] Sucan, I., Moulard, T. & Others. URDF DOM Development Files (https://github.com/ros/urdfdom)

[16] The Linux Foundation. Real-Time Linux Preempt RT. (https://wiki.linuxfoundation.org/realtime/start)

[17] M. Di Castro et al. "i-TIM: A Robotic System for Safety, Measurements, Inspection and Maintenance in Harsh Environments". In: 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). 2018, pp. 1–6. doi:10.1109/SSRR.2018.8468661

[18] Gamper, H., Forkel, D. Diaz Rosales, A., Playan Garai, J., Almagro, C., Buonocore, L., Matheson, E. & Di Castro, M. Adaptive Radiation Survey Using an Autonomous Robot Executing LiDAR Scans in the Large Hadron Collider. *Robotics Research*. pp. 295-303 (2023) doi:10.1007/978-3-031-25555-7_20

[19] Romero-Ramirez, Francisco & Muñoz-Salinas, Rafael & Medina-Carnicer, Rafael. (2018). Speeded Up Detection of Squared Fiducial Markers. Image and Vision Computing. 76. 10.1016/j.imavis.2018.05.004.