Torque-Bounded Task-Space Admittance Control for Redundant Manipulators

Ryo Kikuuwe

Abstract—**This paper presents a task-space admittance controller applicable to redundant manipulators equipped with torque sensors. It extends Kikuuwe's (2019) torque-bounded admittance controller (TBAC), which allows for imposing explicit limits on the joint actuator torques without causing unsafe behaviors such as oscillation and overshoots. The proposed controller enforces the end-effector to follow predefined task-space dynamics as long as the joint torques are unsaturated and the configuration is away from singularities. The behavior in the nullspace, which arises from the redundant degrees of freedom and singular configurations, is governed by predefined joint-space dynamics. The task-space and joint-space dynamics are combined through the use of a continualized pseudoinverse employing the singular value decomposition. Results of experiments using a seven-degree-of-freedom Kinova Gen3 robot illustrate the validity of the proposed admittance controller in various scenarios, including the case where the robot is fully stretched.**

Index Terms—**Admittance Control, Null-space, Redundancy, Singular Configuration**

I. Introduction

Admittance control is a control technique to regulate robots' reactions to external forces. It is one form of impedance control and is often referred to as position-based impedance control. A typical implementation of admittance control is illustrated in Fig. 1(a). The admittance controller is composed of an internal position controller and a virtual object referred to as a 'proxy,' which simulates simple dynamics such as a spring-mass-damper system. The proxy position q_x is adjusted based on the force τ_s measured by a force sensor or a torque sensor and a reference force τ_r given from a higher-level controller. The proxy position q_x acts as the target for the robot's internal position controller, which enforces the robot's actual position q_s to track the proxy position q_x. This controller structure is suitable for robots with complicated dynamics because its internal position controller suppresses the influence of the hardware dynamics such as inertia and joint friction. Its applications include haptic interfaces [1], [2], manual guidance of industrial manipulators [3], human-robot collaboration [4], [5], assembly [6], robotic orthoses [7], [8] and surgical robots [9].

One concern of the admittance control structure is that it is not straightforward to impose explicit limits on the actuator torques. This is because the torque limits imposed on the internal position controllers can lead to a separation between the proxy position q_x and the actual position q_s, which may result in undesirable snapping-back and overshooting behaviors of the robot. This property poses significant inconvenience in ensuring the safety of admittance-controlled robots in practical applications, especially those involving physical contact with humans.

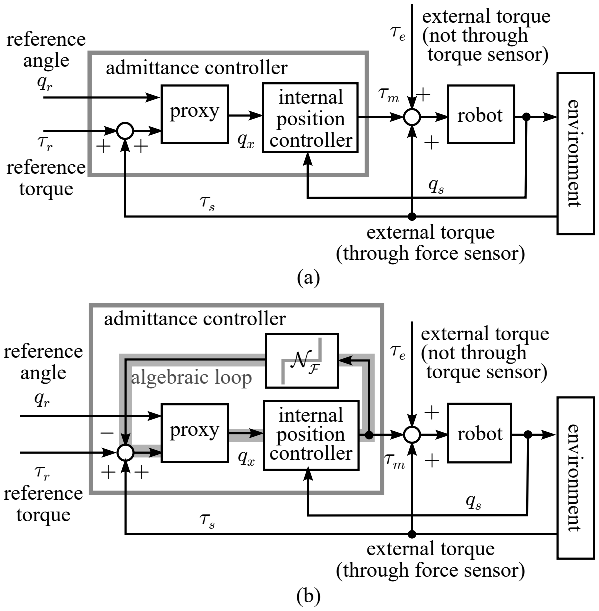


Fig. 1. Systems controlled with admittance controllers. (a) Typical implementation. (b) Torque-Bounded Admittance Control (TBAC) [10].

Another set of technical challenges arises when admittance control should be performed in the task space, i.e., the Cartesian space of the end-effector position and attitude. Such controllers would be needed in, e.g., human-robot collaboration and assembly tasks. In human-robot collaboration, the robot would need to move its end-effector in the direction in which it is pushed or twisted by the human. In assembly tasks, the contact force and compliance should be appropriately regulated in the task space, irrespective of the robot configuration. In this case, one needs to carefully manage the singular configurations and the redundant degrees of freedom (DOFs). When the robot is close to the singular configuration, the proxy velocity in the task space may be mapped to an excessively large joint velocity, which results in unsafe behavior of the robot. The singular configurations have been mainly handled by avoiding them by additional control algorithms [11], [12], or by using approximate pseudoinverses of Jacobians [13], [14]. Such approximations, however, would result in a certain level of inaccuracy in the task-space motion, as discussed

Manuscript compiled February 2, 2025, 14:06.

This work was supported by Kawasaki Heavy Industries, Ltd., Japan.

Ryo Kikuuwe is with Machinery Dynamics Laboratory, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan (e-mail: kikuuwe@ieee.org).

in [13], [15]. The redundancy has been treated by injecting additional damping [16]–[19] and compliance [17], [20], [21] in the nullspace motion[1].

In a previous paper, Kikuuwe [10] proposed a controller named a torque-bounded admittance control (TBAC), which incorporates explicit torque limit to the internal position controller while maintaining the consistency between the proxy position and the actual position. It acts as a standard admittance controller as long as the actuator torque is not saturated, but once it is saturated, the controller yields to external forces without inducing overshoot or oscillations. The controller incorporates an algebraic loop, as illustrated in Fig. 1(b), which algebraically constrains the proxy position and the actuator torque. The entire controller is formulated as a differential algebraic inclusion (DAI), with a discrete-time implementation derived through the implicit (backward) Euler discretization. This method was originally designed as a one-dimensional controller and thus can be implemented independently in each joint. That is, it is usable only in the joint space.

This paper presents a task-space extension of TBAC that tolerates singular configurations and can be used with redundant manipulators. It imposes explicit torque limits to each joint without impairing the safety, while it enforces the end-effector to follow predefined task-space proxy dynamics as long as the joint torques are unsaturated and the robot is out of the singularity configurations. The dynamics of the robot in the nullspace is determined by predetermined joint-space proxy dynamics. The task- and joint-space dynamics are combined as a lexicographic minimization problem, and its computational implementation is realized with an alternative approximation of the pseudoinverse, which is named a continualized pseudoinverse. The continualized pseudoinverse is equal to the original pseudoinverse as long as the singular value is zero or larger than a predetermined threshold, but it is continuous even when it has small singular values. The controller was validated with experiments using a seven-DOF Kinova Gen3 robot, in which the controller is shown to tolerate singular configurations, such as those where the manipulator is fully stretched.

One feature of the proposed controller is its incorporation of full admittance (inertia, viscosity, and stiffness) in both the task space and the joint space, unlike the methods in [16], [18], [19], [21]. Another feature is that it realizes the nullspace admittance control without explicitly computing the nullspace basis or the nullspace velocities, in contrast to the approach in [17], [19], [22]. It may also be an important point that the proposed controller employs a concise representation of the task-space rotational dynamics based on quaternions and quaternion derivatives, which are similar to but simpler than those in [23], [24]. In addition to the above features, the proposed controller inherits the main feature of TBAC, which is the safety under the actuator torque saturation.

This paper is organized as follows. Section II shows some mathematical preliminaries. Section III provides an overview and reinterpretation of the previously-proposed one-dimensional TBAC [10]. Section IV proposes a new admittance controller. Section V provides some theoretical analyses. Section VI shows the results of experiments employing a seven-DOF manipulator. Section VII provides some concluding remarks.

II. Mathematical Preliminaries

A. Notations

In this paper, $\mathbb{R}$ denotes the set of all real numbers, and $\mathbb{H}$ denotes the set of all unit quaternions. This paper treats a quaternion as a four-dimensional vector, which means that $\mathbb{H} \subset \mathbb{R}^4$. We also use $\mathbb{P} = \mathbb{R}^3 \times \mathbb{H} \subset \mathbb{R}^7$, which is used to express the position and the attitude of the end-effector by a 7-dimensional single vector. We use the addition operator $\oplus$, the subtraction operator $\ominus$, and the time-derivative operator $\circ$ for $\mathbb{P}$, which are defined in Appendix A.

This paper uses the following function:

$$\mathrm{proj}_{\boldsymbol{\mathcal{C}}}(\boldsymbol{x}) \triangleq \underset{\boldsymbol{\xi}\in\boldsymbol{\mathcal{C}}}{\mathrm{argmin}}\, \|\boldsymbol{\xi} - \boldsymbol{x}\| \tag{1}$$

where $\boldsymbol{\mathcal{C}} \subset \mathbb{R}^n$ is a convex set. The function $\mathrm{proj}_{\boldsymbol{\mathcal{C}}}$ can be said to be a projection operator onto the set $\boldsymbol{\mathcal{C}}$. This paper also uses the notation $\mathrm{co}\,\boldsymbol{\mathcal{C}}$ to denote the convex hull of a set $\boldsymbol{\mathcal{C}}$. With two scalars A and B, $\mathrm{co}\{A, B\} = \mathrm{co}\{B, A\} = [\min(A, B), \max(A, B)]$.

We also use the function $\mathrm{sat}_1 : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and the function $\mathrm{sat}_3 : \mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}^6$, which are respectively defined as follows:

$$\mathrm{sat}_1(\boldsymbol{F}, \boldsymbol{x}) \triangleq \begin{bmatrix} F_1 x_1 / \max(F_1, |x_1|) \\ \vdots \\ F_n x_n / \max(F_n, |x_n|) \end{bmatrix} \tag{2}$$

$$\mathrm{sat}_3(\boldsymbol{F}, \boldsymbol{x}) \triangleq \begin{bmatrix} F_1 \boldsymbol{x}_{1:3} / \max(F_1, \|\boldsymbol{x}_{1:3}\|) \\ F_2 \boldsymbol{x}_{4:6} / \max(F_2, \|\boldsymbol{x}_{4:6}\|) \end{bmatrix} \tag{3}$$

where x_i stands for the i-th element of $\boldsymbol{x}$ and $\boldsymbol{x}_{i:j}$ stands for the $(j-i+1)$-dimensional vector composed of the i-th to j-th elements of $\boldsymbol{x}$.

Derivations in Section III-B utilize a normal-cone function [10], which is the set-valued function defined as follows:

$$\mathcal{N}_{[A,B]}(x) = \begin{cases} \emptyset & \text{if } x > B \ \wedge \ x < A \\ [0, \infty) & \text{if } x = B \neq A \\ 0 & \text{if } x \in (A, B) \\ (-\infty, 0] & \text{if } x = A \neq B \\ (-\infty, \infty) & \text{if } x = A = B \end{cases} \tag{4}$$

where $A < B$. The following relation exists between the normal cone and the projection [25, Section A.3]:

$$x + \mathcal{N}_{[A,B]}(x) \ni y \iff x = \mathrm{proj}_{[A,B]}(y). \tag{5}$$

Here, addition and subtraction between a set $\mathcal{B} \subset \mathbb{R}$ and a single value $x \in \mathbb{R}$ should be understood as

$$\mathcal{B} \pm x = \bigcup_{\eta\in\mathcal{B}} (\eta \pm x). \tag{6}$$

This implies that, if $\mathcal{B} = [A, B] \subset \mathbb{R}$, $\mathcal{B} + x = [A + x, B + x]$.

[1] This paper uses the term ‘nullspace’ to mean the set of joint-space velocities that do not affect the end-effector velocity.

This paper uses the 'argument of the lexicographic minimum' operator, which should be read as follows:

$$\operatorname*{arglexmin}_{\boldsymbol{x}\in\mathcal{C}}\{f(\boldsymbol{x}), g(\boldsymbol{x})\} = \operatorname*{argmin}_{\boldsymbol{x}\,\in\,\operatorname{argmin}_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x})} g(\boldsymbol{x}). \quad (7)$$

Here, $\operatorname{argmin}_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x})$ is the set of the arguments of the minimum of $f(\boldsymbol{x})$, which can be set-valued. The return value of the $\operatorname{arglexmin}$ operator can also be set-valued, but this paper only deals with cases where it is single-valued.

With a symmetric positive semidefinite matrix $\boldsymbol{A}$, we use $\boldsymbol{A}^h$ to mean one of square matrices that satisfy $\boldsymbol{A} = \boldsymbol{A}^h\boldsymbol{A}^{hT}$ where $\boldsymbol{A}^{hT} = (\boldsymbol{A}^h)^T$. A matrix $\boldsymbol{A}^h$ can be the principal (symmetric) square root matrix $\boldsymbol{A}^{1/2}$ if all elements of $\boldsymbol{A}$ have the same physical dimensions. We use $\boldsymbol{A}^h$ to preserve the physical consistency because $\boldsymbol{A}$ comprising elements with different physical dimensions may result in $\boldsymbol{A}^{1/2}$ whose non-diagonal elements do not have consistent physical dimensions. If $\boldsymbol{A}^{-1}$ exists, one has $\boldsymbol{A}^{-1} = \boldsymbol{A}^{-hT}\boldsymbol{A}^{-h}$ where $\boldsymbol{A}^{-hT} = (\boldsymbol{A}^{hT})^{-1}$ and $\boldsymbol{A}^{-h} = (\boldsymbol{A}^h)^{-1}$. With such a matrix $\boldsymbol{A}^h$, the following relation holds:

$$\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}} = \|\boldsymbol{A}^{hT}\boldsymbol{x}\|, \quad (8)$$

which means that a weighted vector norm can be rewritten as an unweighted norm using $\boldsymbol{A}^{hT}$.

B. Pseudoinverse and Lexicographic Minimization

Recall that, for a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, there always exists a matrix $\boldsymbol{A}^+ \in \mathbb{R}^{n\times m}$ that satisfies the following conditions:

$$\begin{aligned}\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}, \quad \boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+,\\ (\boldsymbol{A}\boldsymbol{A}^+)^T = \boldsymbol{A}\boldsymbol{A}^+, \quad (\boldsymbol{A}^+\boldsymbol{A})^T = \boldsymbol{A}^+\boldsymbol{A}.\end{aligned} \quad (9)$$

The matrix $\boldsymbol{A}^+$ is referred to as the pseudoinverse of $\boldsymbol{A}$.

This paper uses the following fact:

Theorem 1. *With $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, the following statement holds true:*

$$\operatorname*{arglexmin}_{\boldsymbol{x}\in\mathbb{R}^n}\{\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2, \|\boldsymbol{x}\|^2\} = \boldsymbol{A}^+\boldsymbol{b}. \quad (10)$$

The proof is given in Appendix B. The author does not consider it a new result, but the expression (10), in its concise form, cannot be found in the literature as far as the author is aware.

As an extension of Theorem 1, we have the following:

Corollary 1. *The following statement holds true:*

$$\begin{aligned}&\operatorname*{arglexmin}_{\boldsymbol{x}\in\mathbb{R}^n}\{\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2, \|\boldsymbol{C}\boldsymbol{x}-\boldsymbol{d}\|^2\}\\ &= \boldsymbol{C}^{-1}\bar{\boldsymbol{A}}^+\boldsymbol{b} + \boldsymbol{C}^{-1}(\boldsymbol{I}_n - \bar{\boldsymbol{A}}^+\bar{\boldsymbol{A}})\boldsymbol{d}\end{aligned} \quad (11)$$

where $\bar{\boldsymbol{A}} \triangleq \boldsymbol{A}\boldsymbol{C}^{-1} \in \mathbb{R}^{m\times n}$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{d} \in \mathbb{R}^n$, and $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is a regular matrix.

Its proof is also given in Appendix B.

III. Overview and Reinterpretation of TBAC

In a previuos paper, Kikuuwe [10] proposed a controller referred to as a torque-bounded admittance control (TBAC). The controller in [10] is a one-dimensional controller and was validated only through joint-space experiments, where the controller was implemented independently to each joint of a robot. This section provides a quick overview of TBAC from a somewhat different perspective.

A. Conventional Admittance Control

Here we consider a one-DOF robot with a single actuated joint equipped with a joint position (angle) sensor and a force (torque) sensor. Let q_s denote the measured position, τ_m denote the commanded actuator force, and τ_s be the measured external force.

A typical and conventional idea of admittance control is that the robot should be position-controlled to follow a virtual object, hereinafter referred to as a proxy, such as those illustrated in Fig. 2. The proxy in Fig. 2 is an inerter-damper-spring[2] system connected to a reference position q_r and subject to the measured force τ_s and a reference force τ_r. When the robot is in contact with an environment surface, the force from the environment to the robot is measured as the force τ_s, and it is used as a force acting on the proxy. When the robot is statically in contact with the environment, the proxy is in equilibrium, and $\tau_s + \tau_r = K(q_x - q_r)$ holds true. This means that $-\tau_r$ can be interpreted as the desired value of τ_s if the spring constant K is set to be zero.

The equation of motion of the proxy illustrated in Fig. 2 can be given as follows:

$$M(\ddot{q}_x - \alpha_r) + B(\dot{q}_x - u_r) + K(q_x - q_r) = \tau_s + \tau_r \quad (12)$$

where $\alpha_r \triangleq \dot{u}_r$ and $u_r \triangleq \dot{q}_r$. Here, $M > 0$ is the inertia of the inerter, $B > 0$ is the viscosity of the damper, and $K > 0$ is the stiffness of the spring. The proxy position $q_x \in \mathbb{R}$ and velocity $\dot{q}_x$ are updated according to (12), and the position q_x is used as the command position to the position controller, which can be typically written as follows:

$$\tau_m = K_c(q_x - q_s) + B_c(\dot{q}_x - \dot{q}_s) + L_c\int(q_x - q_s)dt. \quad (13)$$

Here, K_c, B_c, and L_c are the proportional, derivative, and integral gains. A typical algorithmic structure of an admittance controller can be illustrated in Fig. 3(a). Here, T is the sampling interval and the subscript prv indicates the value of the associated variable in the previous timestep. Based on the Euler discretization, we have $q_x = q_{x,\mathrm{prv}} + Tu_x$ where $u_x \triangleq \dot{q}_x$. In the structure of Fig. 3(a), the proxy position q_x is sent to the position controller. The proxy position q_x and velocity u_x are reused in the next timestep to update them to new values.

B. Basic Version of TBAC

Let us consider the case where we need to impose the constraint $\tau_m \in [-F_c, F_c]$ on the actuator torque τ_m where

[2] An inerter is a mechanical element that produces the force proportional to the acceleration difference between its two ends, such as those appearing in [26].

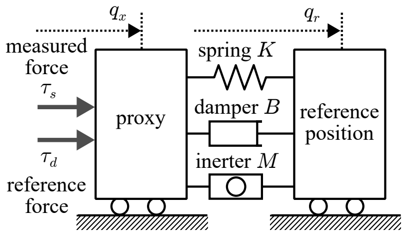
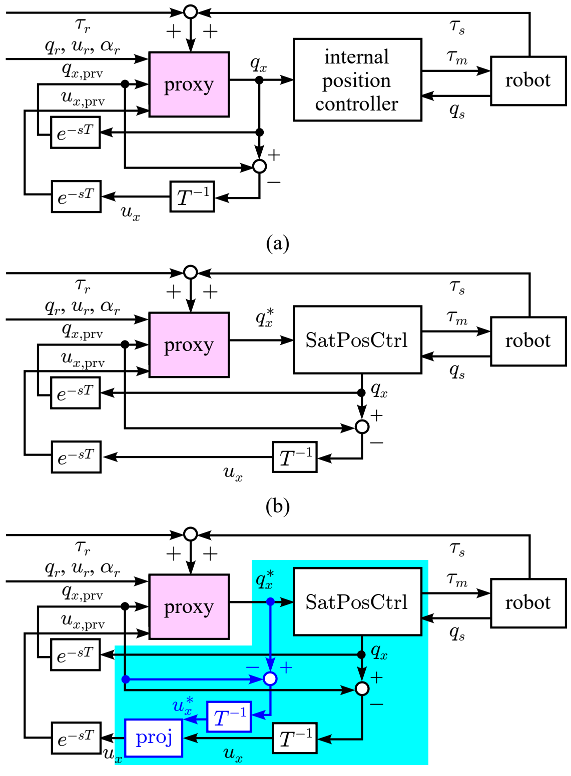


Fig. 2. A physical interpretation of the proxy.



(a)

(b)

(c)

Fig. 3. Alternative representations of admittance controllers; (a) the typical conventional admittance controller, (b) the basic version of TBAC [10, Sec. III], and (c) the velocity-projected version of TBAC [10, Sec. V.B]. The function SatPosCtrl is defined in (16). For simplicity, the error integral b_{xs} is omitted here.

$F_c > 0$. The TBAC [10] is a controller to deal with such a case, and it is described as follows:

$$M(\ddot{q}_x - \alpha_r) + B(\dot{q}_x - u_r) + K(q_x - q_r) \in \tau_s + \tau_r - \mathcal{N}_{[-F_c,F_c]}(\tau_m) \tag{14a}$$

$$\tau_m = K_c(q_x - q_s) + B_c(\dot{q}_x - \dot{q}_s) + L_c \int (q_x - q_s)dt. \tag{14b}$$

This set of equations can be seen as a differential-algebraic inclusion (DAI) with respect to q_x. Because the normal-cone term $+\mathcal{N}_{[-F_c,F_c]}(\tau_m)$ in (14a) forbids τ larger than F_c, the proxy's acceleration $\ddot{q}_x$ is determined so that $|\tau| \leq F_c$ is satisfied. As long as $|\tau| < F_c$, the controller (14) is equivalent to the ordinary admittance controller described by (12) and (13). The structure of TBAC (14) is shown in Fig. 1(b), in which the term $\mathcal{N}_{[-F_c,F_c]}(\tau_m)$ in (14a) appears as an algebraic feedback loop.

The discrete-time representation of the controller (14) can be obtained by its implicit Euler discretization, which can be obtained by replacing the derivatives by the finite differences, e.g., $\dot{u}_x := (u_x - u_{x,\mathrm{prv}})/T$. Through some derivations detailed in [10], which include the use of (5) to treat the set-valuedness, the discrete-time algorithm for solving (14) can be given as follows:

$$\hat{\tau}_s := \tau_s + \tau_r + M\alpha_r + Bu_r + K(q_r - q_{x,\mathrm{prv}}) \tag{15a}$$

$$q_x^* := \frac{Mu_{x,\mathrm{prv}} + T\hat{\tau}_s}{M + BT + KT^2} \tag{15b}$$

$$\{\tau_m, q_x\} := \mathrm{SatPosCtrl}(q_x^*, q_{x,\mathrm{prv}}, q_s, u_s, b_{xs,\mathrm{prv}}) \tag{15c}$$

$$u_x = (q_x - q_{x,\mathrm{prv}})/T \tag{15d}$$

$$b_{xs} := b_{xs,\mathrm{prv}} + T(q_x - q_s) \tag{15e}$$

where the function SatPosCtrl, which is a 'saturated position controller,' is defined as follows:

$$\text{Function } \mathrm{SatPosCtrl}(q_x^*, q_{x,\mathrm{prv}}, q_s, u_s, b_{xs,\mathrm{prv}}) \tag{16a}$$

$$\tau_m^{**} := L_c b_{xs,\mathrm{prv}} - B_c(u_s - (q_x^* - q_{x,\mathrm{prv}})/T) \tag{16b}$$

$$\tau_m^* := (K_c + L_cT)(q_x^* - q_s) + \tau_m^{**} \tag{16c}$$

$$\tau_m := \mathrm{proj}_{[-F_c,F_c]}(\tau_m^*) \tag{16d}$$

$$q_x := q_s + \frac{\tau_m - \tau_m^{**}}{B_c/T + K_c + L_cT} \tag{16e}$$

$$\text{Return } \{\tau_m, q_x\}. \tag{16f}$$

Note that this algorithm (15) does not involve any set-valued functions or non-closed-form equations, in spite of the fact that its original continuous-time representation (14) involves set-valuedness and differential-algebraic constraints.

The structure of the algorithm (15) can be illustrated as in Fig. 3(b). It can be interpreted as a combination of the proxy dynamics, represented by (15a) and (15b), and a saturated position controller, represented by the function SatPosCtrl defined in (16). The quantity q_x^* calculated by (15b) can be seen as a 'tentative' proxy position without considering the actuator saturation. This is provided to the function SatPosCtrl, which provides the necessary torque τ_m and a 'corrected' proxy position q_x, which corresponds to the maximal permissible velocity under the torque limitation. If the actuator is not saturated, the tentative proxy position q_x^* is just adopted as the proxy velocity q_x. The corrected proxy position q_x is used in the next timestep.

C. Velocity-Projected Version of TBAC

The previous paper [10] pointed out a flaw of the algorithm (15), which is that the actuator saturation may inject kinetic energy into the robot. When the robot is accelerated by an external force and the actuator torque is saturated, the proxy velocity u_x is also increased, and it affects the proxy position q_x in the next timestep. This means that, when the actuator is saturated, the work done by the external force is stored as the kinetic energy of the proxy, as well as that of the robot. This feature is undesirable for safety reasons.

To eliminate this flaw, the paper [10] has proposed a velocity projection method, which is referred to as a 'Modification B' in [10, Section V.B]. It modifies u_x so that it satisfies $u_x \in \mathrm{co}\{0, u_x^*\}$, which indicates that the saturation only shrinks the

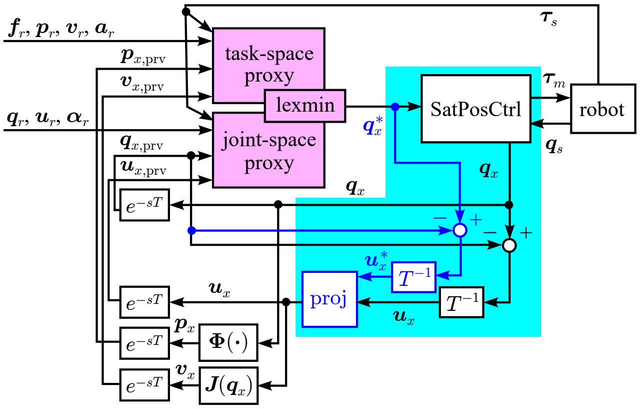


Fig. 4. Proposed controller. Note that the highlighted part has a structure identical to that of Fig. 3(c). The error integral $\boldsymbol{b}_{xs}$ is omitted here for simplicity.

proxy velocity u_x^* without altering its direction. This idea can be realized by adding the line

$$u_x := \text{proj}_{\text{co}\{0,u_x^*\}}(u_x) \tag{17}$$

in the end of the algorithm (15). The modified controller (15)+(17), which can be said to be a 'velocity-projected version' of TBAC, is illustrated in Fig. 3(c). It should be emphasized that this modification does not alter the current proxy position q_x, but does modify only the proxy velocity u_x, which influences the proxy position in the next timestep. The underlying idea is that the proxy velocity, which is a state variable independent from the proxy position, does not need to satisfy $u_x = (q_x - q_{x,\text{prv}})/T$, although the proxy position q_x needs to be consistent with the torque τ_m.

IV. Proposed Controller

This section proposes a multi-dimensional extension of TBAC applicable to redundant manipulators. The overall structure of the controller is illustrated in Fig. 4. The desired dynamics of the robot are defined by two proxies: the task-space proxy and the joint-space proxy. The block SatPosCtrl in Fig. 4 is an element-wise, multidimensional version of the one defined in (16), which acts on each joint independently. We assume that the robot is equipped with torque sensors attached to the joints.

A. Structure

We here consider an n-DOF rigid-link manipulator with a single end-effector. We use $\boldsymbol{q} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^n$, and $\boldsymbol{\alpha} \in \mathbb{R}^n$ to denote the joint-space position, velocity, and acceleration vectors, respectively, and they satisfy $\boldsymbol{u} = \dot{\boldsymbol{q}}$ and $\boldsymbol{\alpha} = \dot{\boldsymbol{u}}$. We also use $\boldsymbol{p} \in \mathbb{P}$, $\boldsymbol{v} \in \mathbb{R}^6$, and $\boldsymbol{a} \in \mathbb{R}^6$ to denote the task-space position, velocity, and acceleration vectors, respectively. It should be noted that $\boldsymbol{p} \in \mathbb{P}$ is a seven-dimensional vector composed of the three-dimensional position-vector part and the four-dimensional attitude-quaternion part. Each of the vectors $\boldsymbol{v} \in \mathbb{R}^6$ and $\boldsymbol{a} \in \mathbb{R}^6$ is composed of a three-dimensional translational part and a three-dimensional rotational part. They are associated as $\boldsymbol{v} = \overset{\circ}{\boldsymbol{p}}$ and $\boldsymbol{a} = \dot{\boldsymbol{v}}$ where $\circ$ is the time-derivative operator defined in Appendix A. We also use $\boldsymbol{\tau} \in \mathbb{R}^n$ and $\boldsymbol{f} \in \mathbb{R}^6$ to denote a joint torque vector and a task-space force vector, respectively. With these symbols, the subscripts s, x, and r denote the measured values, the proxy values, and the reference values, respectively.

With the forward kinematics $\boldsymbol{\Phi} : \mathbb{R}^n \to \mathbb{P}$ of the manipulator, we have $\boldsymbol{p} = \boldsymbol{\Phi}(\boldsymbol{q})$, and using the Jacobian matrix $\boldsymbol{J} : \mathbb{R}^n \to \mathbb{R}^6$, we have $\boldsymbol{v} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{u}$. We also use the Jacobian rate-of-change matrix $\boldsymbol{H} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^6$ defined as $\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \triangleq d\boldsymbol{J}(\boldsymbol{q})/dt$, which satisfies $\dot{\boldsymbol{v}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{u}} + \boldsymbol{H}(\boldsymbol{q}, \boldsymbol{u})\boldsymbol{u}$.

As can be seen in Fig. 4, we assume that the manipulator is equipped with joint angle sensors and joint torque sensors, which provide the measured angle vector $\boldsymbol{q}_s \in \mathbb{R}^n$ and the measured torque vector $\boldsymbol{\tau}_s \in \mathbb{R}^n$, respectively. The proposed controller involves two different proxies, which are the task-space proxy and the joint-space proxy. They are governed by certain dynamics coupled with a lexicographic prioritization, indicated as 'lexmin' in the figure. The positions and the velocities of the joint-space proxy and the task-space proxy are kept consistent with each other through the relations $\boldsymbol{p}_x = \boldsymbol{\Phi}(\boldsymbol{q}_x)$ and $\boldsymbol{v}_x = \boldsymbol{J}(\boldsymbol{q}_x)\boldsymbol{u}_x$ at every timestep. The output of the controller is the command torque $\boldsymbol{\tau}_m \in \mathbb{R}^6$ sent to the joint actuators.

Recall that the one-dimensional velocity-projected version of TBAC is written as (15)+(17). As its direct extension, the algorithm of the proposed controller is given as follows:

$$\boldsymbol{q}_x^* := \text{TwoProxies}(\boldsymbol{p}_{x,\text{prv}}, \boldsymbol{v}_{x,\text{prv}}, \boldsymbol{q}_{x,\text{prv}}, \boldsymbol{u}_{x,\text{prv}}, \boldsymbol{\tau}_s) \tag{18a}$$

$$\{\boldsymbol{\tau}_m, \boldsymbol{q}_x\} := \text{SatPosCtrl}(\boldsymbol{q}_x^*, \boldsymbol{q}_{x,\text{prv}}, \boldsymbol{q}_s, \boldsymbol{u}_s, \boldsymbol{b}_{xs,\text{prv}}) \tag{18b}$$

$$\boldsymbol{u}_x := (\boldsymbol{q}_x - \boldsymbol{q}_{x,\text{prv}})/T \tag{18c}$$

$$\boldsymbol{b}_{xs} := \boldsymbol{b}_{xs,\text{prv}} + T(\boldsymbol{q}_x - \boldsymbol{q}_s) \tag{18d}$$

$$\boldsymbol{u}_x := \text{proj}_{\text{co}\{\boldsymbol{0}_n, \boldsymbol{u}_x^*\}}(\boldsymbol{u}_x) \tag{18e}$$

$$\boldsymbol{p}_x := \boldsymbol{\Phi}(\boldsymbol{q}_x) \tag{18f}$$

$$\boldsymbol{v}_x := \boldsymbol{J}(\boldsymbol{q}_x)\boldsymbol{u}_x. \tag{18g}$$

Here, the function TwoProxies represents the combined proxy dynamics that will be detailed in the next Section IV-B. It determines the tentative joint-space proxy velocity $\boldsymbol{u}_x^*$, and it is used as the input to SatPosCtrl, the saturated position controller defined in (16), to obtain the joint torque command $\boldsymbol{\tau}_m$. As has been detailed in Section III-B, the function SatPosCtrl provides the corrected proxy velocity $\boldsymbol{u}_x$ as another output. The velocity is then 'projected' in (18e), resulting in a re-corrected velocity $\boldsymbol{u}_x$. This projection operation is discussed in Section III-C, and its multidimensional version is given as

$$\text{proj}_{\text{co}\{\boldsymbol{0}_n, \boldsymbol{u}_x^*\}}(\boldsymbol{u}_x) \triangleq \text{proj}_{[0,1]}\left(\frac{\boldsymbol{u}_x^{*T}\boldsymbol{u}_x}{\boldsymbol{u}_x^{*T}\boldsymbol{u}_x^*}\right)\boldsymbol{u}_x^*. \tag{19}$$

Finally, the joint-space proxy position $\boldsymbol{q}_x$ and velocity $\boldsymbol{u}_x$ are converted into the task-space proxy position $\boldsymbol{p}_x$ and velocity $\boldsymbol{v}_x$, respectively, through the forward kinematics.

Note that the lines (18b) to (18e) of the proposed controller exactly correspond to the lines (15c) to (15e) plus (17) of the one-dimensional TBAC.

B. Combined Proxy Dynamics

Now the combined proxy dynamics, which appear as the function $\mathrm{TwoProxies}$ in the line (18a) of the proposed controller (18), is presented. It is composed of the task-space proxy dynamics and the joint-proxy dynamics. The task-space proxy dynamics is given in the following continuous-time representation:

$$\mathring{\boldsymbol{p}}_x = \boldsymbol{v}_x,\ \dot{\boldsymbol{v}}_x = \boldsymbol{a}_x \tag{20a}$$

$$\begin{aligned}\boldsymbol{M}_\mathrm{T}(\boldsymbol{a}_x - \boldsymbol{a}_r) + \boldsymbol{B}_\mathrm{T}(\boldsymbol{v}_x - \boldsymbol{v}_r) + \mathrm{sat}_3(\boldsymbol{F}_\mathrm{T}, \boldsymbol{K}_\mathrm{T}(\boldsymbol{p}_x \ominus \boldsymbol{p}_r))& \\ = \boldsymbol{f}_s + \boldsymbol{f}_r&\end{aligned} \tag{20b}$$

where $\boldsymbol{f}_s$ and $\boldsymbol{f}_r \in \mathbb{R}^6$ are the force acting on the end-effector and the reference force, respectively, and $\boldsymbol{a}_r \triangleq \dot{\boldsymbol{v}}_r$ and $\boldsymbol{v}_r \triangleq \mathring{\boldsymbol{p}}_r$. The matrices $\boldsymbol{M}_\mathrm{T}$, $\boldsymbol{B}_\mathrm{T}$, and $\boldsymbol{K}_\mathrm{T} \in \mathbb{R}^{6\times 6}$ are the inertia, viscosity, and stiffness matrices, respectively, and they are symmetric positive-definite matrices. In the third term of the left-hand side of (20b), the function sat_3, defined in (3), is used to allow for setting upperbounds to the task-space proxy spring force. Here, $\boldsymbol{F}_\mathrm{T}$ is a vector composed as $\boldsymbol{F}_\mathrm{T} = [F_\mathrm{tra}, F_\mathrm{rot}]^T \in \mathbb{R}^2$ where F_tra and F_rot are the limits for the translational and rotational components, respectively.

In the dynamics (20), $\boldsymbol{f}_s$ is not directly available. It however can be assumed to satisfy the following relation with the measured torque $\boldsymbol{\tau}_s \in \mathbb{R}^n$:

$$\boldsymbol{\tau}_s \approx \boldsymbol{J}(\boldsymbol{q}_s)^T \boldsymbol{f}_s. \tag{21}$$

A discrete-time approximation of (20) can be given as follows:

$$\boldsymbol{p}_x = \boldsymbol{p}_{x,\mathrm{prv}} \oplus T\boldsymbol{v}_x,\ \boldsymbol{v}_x = \boldsymbol{v}_{x,\mathrm{prv}} + T\boldsymbol{a}_x \tag{22a}$$

$$\begin{aligned}\boldsymbol{M}_\mathrm{T}\,(\boldsymbol{a}_x - \boldsymbol{a}_r) + \boldsymbol{B}_\mathrm{T}(\boldsymbol{v}_{x,\mathrm{prv}} + T\boldsymbol{a}_x - \boldsymbol{v}_r)& \\ + \mathrm{sat}_3(\boldsymbol{F}_\mathrm{T}, \boldsymbol{K}_\mathrm{T}(\boldsymbol{p}_{x,\mathrm{prv}} \ominus \boldsymbol{p}_r)) = \boldsymbol{f}_s + \boldsymbol{f}_r&.\end{aligned} \tag{22b}$$

One can rewrite (22b) as follows:

$$\hat{\boldsymbol{C}}_\mathrm{T}\boldsymbol{a}_x - \hat{\boldsymbol{b}}_\mathrm{T} - \boldsymbol{f}_s = \boldsymbol{0}_6 \tag{23}$$

where

$$\hat{\boldsymbol{C}}_\mathrm{T} \triangleq \boldsymbol{M}_\mathrm{T} + T\boldsymbol{B}_\mathrm{T} \tag{24}$$

$$\hat{\boldsymbol{b}}_\mathrm{T} \triangleq -\boldsymbol{B}_\mathrm{T}\boldsymbol{v}_{x,\mathrm{prv}} + \boldsymbol{f}_r + \boldsymbol{f}_{re} \tag{25}$$

$$\boldsymbol{f}_{re} \triangleq \boldsymbol{M}_\mathrm{T}\boldsymbol{a}_r + \boldsymbol{B}_\mathrm{T}\boldsymbol{v}_r + \mathrm{sat}_3(\boldsymbol{F}_\mathrm{T}, \boldsymbol{K}_\mathrm{T}(\boldsymbol{p}_r \ominus \boldsymbol{p}_{x,\mathrm{prv}})). \tag{26}$$

The expression (23) can be seen as an algebraic problem with respect to the task-space acceleration $\boldsymbol{a}_x \in \mathbb{R}^6$, but to incorporate it into the framework of Fig. 4, it should be reformulated as a problem with respect to the joint-space acceleration $\boldsymbol{\alpha}_x \in \mathbb{R}^n$. Moreover, because the robot is assumed to have torque sensors, we need to replace $\boldsymbol{f}_s$ by $\boldsymbol{\tau}_s$ through the relation (21). Note that $\boldsymbol{a}_x$ and $\boldsymbol{\alpha}_x$ are related by

$$\boldsymbol{a}_x = \boldsymbol{J}(\boldsymbol{q}_{x,\mathrm{prv}})\boldsymbol{\alpha}_x + \boldsymbol{H}(\boldsymbol{q}_{x,\mathrm{prv}}, \boldsymbol{u}_{x,\mathrm{prv}})(\boldsymbol{u}_{x,\mathrm{prv}} + T\boldsymbol{\alpha}_x). \tag{27}$$

Here, $\boldsymbol{H}(\boldsymbol{q}, \boldsymbol{u})$ is the Jacobian rate-of-change, i.e., $\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \triangleq d\boldsymbol{J}(\boldsymbol{q})/dt$, which can be computed by conventional methods such as [27, Sec. 5.5] and [28]. Therefore, by substituting (27) into (23) and left-multiplying its both sides by $\boldsymbol{J}(\boldsymbol{q}_s)^T$, one can obtain the following:

$$\begin{aligned}&\boldsymbol{J}(\boldsymbol{q}_s)^T\hat{\boldsymbol{C}}_\mathrm{T}(\boldsymbol{J}(\boldsymbol{q}_{x,\mathrm{prv}}) + T\boldsymbol{H}(\boldsymbol{q}_{x,\mathrm{prv}}, \boldsymbol{u}_{x,\mathrm{prv}}))\boldsymbol{\alpha}_x \\ &-\boldsymbol{J}(\boldsymbol{q}_s)^T\big(\hat{\boldsymbol{b}}_\mathrm{T} - \hat{\boldsymbol{C}}_\mathrm{T}\boldsymbol{H}(\boldsymbol{q}_{x,\mathrm{prv}}, \boldsymbol{u}_{x,\mathrm{prv}})\boldsymbol{u}_{x,\mathrm{prv}}\big) - \boldsymbol{\tau}_s = \boldsymbol{0}_n.\end{aligned} \tag{28}$$

The solution to this problem is $\boldsymbol{\alpha}_x$, which is consistent with the task-space proxy dynamics (20).

Meanwhile, the joint-space proxy dynamics is given in the following continuous-time representation:

$$\dot{\boldsymbol{q}}_x = \boldsymbol{u}_x,\ \dot{\boldsymbol{u}}_x = \boldsymbol{\alpha}_x \tag{29a}$$

$$\boldsymbol{M}(\boldsymbol{\alpha}_x - \boldsymbol{\alpha}_r) + \boldsymbol{B}(\boldsymbol{u}_x - \boldsymbol{u}_r) + \mathrm{sat}_1(\boldsymbol{F}, \boldsymbol{K}(\boldsymbol{q}_x - \boldsymbol{q}_r)) = \boldsymbol{\tau}_s \tag{29b}$$

where the matrices $\boldsymbol{M}$, $\boldsymbol{B}$, and $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ are the inertia, viscosity, and stiffness matrices, respectively, for the joint-space proxy dynamics, and $\boldsymbol{u}_r \triangleq \dot{\boldsymbol{q}}_r$ and $\boldsymbol{\alpha}_r \triangleq \dot{\boldsymbol{u}}_r$. They are diagonal positive-definite matrices. In the third term of the left-hand side of (29b), the function sat_1 is the one defined in (2), and $\boldsymbol{F} \in \mathbb{R}^n$ is the vector of the limits for the forces of the joint-space proxy springs. A discrete-time approximation of (29) can be given as follows:

$$\boldsymbol{q}_x = \boldsymbol{q}_{x,\mathrm{prv}} + T\boldsymbol{u}_x,\ \boldsymbol{u}_x = \boldsymbol{u}_{x,\mathrm{prv}} + T\boldsymbol{\alpha}_x \tag{30a}$$

$$\begin{aligned}\boldsymbol{M}\,(\boldsymbol{\alpha}_x - \boldsymbol{\alpha}_r) + \boldsymbol{B}(\boldsymbol{u}_{x,\mathrm{prv}} + T\boldsymbol{\alpha}_x - \boldsymbol{u}_r)& \\ + \mathrm{sat}_1(\boldsymbol{F}, \boldsymbol{K}(\boldsymbol{q}_{x,\mathrm{prv}} - \boldsymbol{q}_r)) = \boldsymbol{\tau}_s&.\end{aligned} \tag{30b}$$

One can rewrite (30b) as follows:

$$\hat{\boldsymbol{C}}_\mathrm{J}\boldsymbol{\alpha}_x - \hat{\boldsymbol{b}}_\mathrm{J} - \boldsymbol{\tau}_s = \boldsymbol{0}_n \tag{31}$$

where

$$\hat{\boldsymbol{C}}_\mathrm{J} \triangleq \boldsymbol{M} + T\boldsymbol{B} \tag{32}$$

$$\hat{\boldsymbol{b}}_\mathrm{J} \triangleq -\boldsymbol{B}\boldsymbol{u}_{x,\mathrm{prv}} + \boldsymbol{\tau}_{re} \tag{33}$$

$$\boldsymbol{\tau}_{re} \triangleq \boldsymbol{M}\boldsymbol{\alpha}_r + \boldsymbol{B}\boldsymbol{u}_r + \mathrm{sat}_1(\boldsymbol{F}, \boldsymbol{K}(\boldsymbol{q}_r - \boldsymbol{q}_{x,\mathrm{prv}})). \tag{34}$$

The problem (31) can be seen as a problem with respect to $\boldsymbol{\alpha}_x$ that is consistent with the joint-space proxy dynamics (29).

The task-space proxy dynamics (28) and the joint-space proxy dynamics (31) are combined in the following manner:

$$\boldsymbol{\alpha}_x^* = \underset{\boldsymbol{\alpha}_x \in \mathbb{R}^6}{\mathrm{arglexmin}}\,\{\|\text{(28)'s lhs}\|_{\boldsymbol{M}^{-1}}, \|\text{(31)'s lhs}\|_{\boldsymbol{M}^{-1}}\} \tag{35}$$

where ‘lhs’ stands for the left-hand side. The expression (35) means that the task-space proxy dynamics (28) is prioritized over the joint-space proxy dynamics (31), and that the joint-space proxy dynamics is used only in the nullspace. The use of the weight matrix $\boldsymbol{M}^{-1}$ in (35) is for the asymptotic stability of the combined dynamics, which will be discussed in Section V-A. By using Corollary 1, an algorithm to solve the problem (35) can be given as follows:

$$\text{Function } \mathrm{TwoProxies}(\boldsymbol{p}_{x,\mathrm{prv}}, \boldsymbol{v}_{x,\mathrm{prv}}, \boldsymbol{q}_{x,\mathrm{prv}}, \boldsymbol{u}_{x,\mathrm{prv}}, \boldsymbol{\tau}_s) \tag{36a}$$

$$\boldsymbol{J}_s := \boldsymbol{J}(\boldsymbol{q}_s) \tag{36b}$$

$$\boldsymbol{H}_{x,\mathrm{prv}} := \boldsymbol{H}(\boldsymbol{q}_{x,\mathrm{prv}}, \boldsymbol{u}_{x,\mathrm{prv}}) \tag{36c}$$

$$\hat{\boldsymbol{J}}_x := \boldsymbol{J}(\boldsymbol{q}_{x,\mathrm{prv}}) + T\boldsymbol{H}_{x,\mathrm{prv}} \tag{36d}$$

$$\boldsymbol{C}_\mathrm{T} := \boldsymbol{M}^{-h}\boldsymbol{J}_s^T(\boldsymbol{M}_\mathrm{T} + T\boldsymbol{B}_\mathrm{T})\hat{\boldsymbol{J}}_x \tag{36e}$$

$$\boldsymbol{C}_\mathrm{J} := \boldsymbol{M}^{-h}(\boldsymbol{M} + T\boldsymbol{B}) \tag{36f}$$

$$\boldsymbol{f}_{re} := \boldsymbol{M}_\mathrm{T}\boldsymbol{a}_r + \boldsymbol{B}_\mathrm{T}\boldsymbol{v}_r + \mathrm{sat}_3(\boldsymbol{F}_\mathrm{T}, \boldsymbol{K}_\mathrm{T}(\boldsymbol{p}_r \ominus \boldsymbol{p}_{x,\mathrm{prv}})) \tag{36g}$$

$$\boldsymbol{\tau}_{re} := \boldsymbol{M}\boldsymbol{\alpha}_r + \boldsymbol{B}_\mathrm{J}\boldsymbol{u}_r + \mathrm{sat}_1(\boldsymbol{F}, \boldsymbol{K}(\boldsymbol{q}_r - \boldsymbol{q}_{x,\mathrm{prv}})) \tag{36h}$$

$$\boldsymbol{b}_\mathrm{T} := \boldsymbol{M}^{-h}\big(\boldsymbol{J}_s^T\big(- \boldsymbol{B}_\mathrm{T}\boldsymbol{v}_{x,\mathrm{prv}}$$

$$-(\boldsymbol{M}_{\mathrm{T}} + T\boldsymbol{B}_{\mathrm{T}})\boldsymbol{H}_{x,\mathrm{prv}}\boldsymbol{u}_{x,\mathrm{prv}}+\boldsymbol{f}_r+\boldsymbol{f}_{re}\big) + \boldsymbol{\tau}_s\big) \tag{36i}$$

$$\boldsymbol{b}_{\mathrm{J}} := \boldsymbol{M}^{-h}(-\boldsymbol{B}\boldsymbol{u}_{x,\mathrm{prv}} + \boldsymbol{\tau}_{re} + \boldsymbol{\tau}_s) \tag{36j}$$

$$\boldsymbol{C}_{\mathrm{TJ}} := \boldsymbol{C}_{\mathrm{T}}\boldsymbol{C}_{\mathrm{J}}^{-1} \tag{36k}$$

$$\boldsymbol{\alpha}_x^* := \boldsymbol{C}_{\mathrm{J}}^{-1}\boldsymbol{C}_{\mathrm{TJ}}^{+}\boldsymbol{b}_{\mathrm{T}} + \boldsymbol{C}_{\mathrm{J}}^{-1}\left(\boldsymbol{I}_n - \boldsymbol{C}_{\mathrm{TJ}}^{+}\boldsymbol{C}_{\mathrm{TJ}}\right)\boldsymbol{b}_{\mathrm{J}} \tag{36l}$$

$$\boldsymbol{u}_x^* := \boldsymbol{u}_{x,\mathrm{prv}} + T\boldsymbol{\alpha}_x^* \tag{36m}$$

$$\boldsymbol{q}_x^* := \boldsymbol{q}_{x,\mathrm{prv}} + T\boldsymbol{u}_x^* \tag{36n}$$

$$\text{Return } \boldsymbol{q}_x^*. \tag{36o}$$

It should be noted that the solution of (35) is $\boldsymbol{\alpha}_x^*$ obtained by (36l) and that the output of this function is its resultant proxy position $\boldsymbol{q}_x^*$. This function is to be used in the first line of the proposed controller algorithm (18). Recall that $\boldsymbol{M}^{-h}$ is a matrix satisfying $\boldsymbol{M}^{-1} = \boldsymbol{M}^{-hT}\boldsymbol{M}^{-h}$ as defined in Section II-A. Specifically, if $\boldsymbol{M}$ is a diagonal positive-definite matrix, $\boldsymbol{M}^{-h}$ can be simply chosen as $\boldsymbol{M}^{-h} = \boldsymbol{M}^{-1/2}$, which is the diagonal matrix whose diagonal elements are the reciprocals of the square roots of the diagonal elements of $\boldsymbol{M}$.

C. Computation of the Pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{+}$

We now discuss the computation of the pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{+}$ in the algorithm (36), i.e., the function TwoProxies. The matrix $\boldsymbol{C}_{\mathrm{TJ}}$ is an $n \times n$ square matrix, but it is not invertible because its rank is at most 6 and can be less in the singular configurations. The matrix $\boldsymbol{C}_{\mathrm{TJ}}$ can be decomposed as $\boldsymbol{C}_{\mathrm{TJ}} = \boldsymbol{C}_s\boldsymbol{C}_x$ where

$$\boldsymbol{C}_s \triangleq \boldsymbol{M}^{-h}\boldsymbol{J}_s^T(\boldsymbol{M}_{\mathrm{T}} + T\boldsymbol{B}_{\mathrm{T}}) \in \mathbb{R}^{n\times 6} \tag{37}$$

$$\boldsymbol{C}_x \triangleq \hat{\boldsymbol{J}}_x(\boldsymbol{M} + T\boldsymbol{B})^{-1}\boldsymbol{M}^h \in \mathbb{R}^{6\times n}. \tag{38}$$

Therefore, if both $\boldsymbol{C}_s$ and $\boldsymbol{C}_x$ are full-rank, we have $\boldsymbol{C}_{\mathrm{TJ}}^{+} = \boldsymbol{C}_x^{+}\boldsymbol{C}_s^{+}$. Thus, $\boldsymbol{C}_{\mathrm{TJ}}^{+}$ can be approximately obtained as follows:

$$\boldsymbol{C}_{\mathrm{TJ}}^{+} \approx \boldsymbol{C}_x^T(\boldsymbol{C}_x\boldsymbol{C}_x^T + \varepsilon_x^2\boldsymbol{I}_6)^{-1}(\boldsymbol{C}_s^T\boldsymbol{C}_s + \varepsilon_s^2\boldsymbol{I}_6)^{-1}\boldsymbol{C}_s^T. \tag{39}$$

Here, ε_x and ε_s are small positive values to deal with the case where $\boldsymbol{J}_x$, $\boldsymbol{J}_s$, or both are rank-deficient. This approximation can be seen as an application of the damped pseudoinverses (cf., e.g., [15], [29], [30]). It requires careful choice of the values of ε_x and ε_s, and can cause a certain level of inaccuracy even when the robot is away from the singular configuration.

An alternative approximate computational method for $\boldsymbol{C}^{+}$ follows. Recall that the matrix $\boldsymbol{C}_{\mathrm{TJ}} \in \mathbb{R}^{n\times n}$ can be decomposed as $\boldsymbol{C}_{\mathrm{TJ}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ where $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose diagonal elements are the singular values of the matrix $\boldsymbol{C}_{\mathrm{TJ}}$. The pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{+} \in \mathbb{R}^{n\times n}$ is obtained as $\boldsymbol{C}_{\mathrm{TJ}}^{+} = \boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^T$. Here, the matrix $\boldsymbol{\Sigma}^{+}$ is the $n \times n$ diagonal matrix whose i-th diagonal element is

$$\sigma_i^{+} \triangleq \begin{cases} 1/\sigma_i & \text{if } \sigma_i \neq 0 \\ 0 & \text{if } \sigma_i = 0 \end{cases} \tag{40}$$

where σ_i is the i-th diagonal element of $\boldsymbol{\Sigma}$. This definition injects the discontinuity around $\sigma_i = 0$, and also causes an excessively large value of σ_i^{+} when σ_i is non-zero but very small. One alternative is to replace σ_i^{+} by the following:

$$\sigma_i^{\oplus} \triangleq \frac{\sigma_i}{\max(\sigma_i^2, \varepsilon^2)} = \begin{cases} 1/\sigma_i & \text{if } |\sigma_i| > \varepsilon \\ \sigma_i/\varepsilon^2 & \text{if } |\sigma_i| \leq \varepsilon \end{cases} \tag{41}$$

where $\varepsilon > 0$ is a parameter appropriately chosen. With this replacement, one can obtain the matrix $\boldsymbol{C}_{\mathrm{TJ}}^{\oplus} \triangleq \boldsymbol{U}\boldsymbol{\Sigma}^{\oplus}\boldsymbol{V}^T$ where $\boldsymbol{\Sigma}^{\oplus}$ is the $n \times n$ diagonal matrix with the diagonal elements defined in (41). Hereafter, we refer to $\boldsymbol{C}_{\mathrm{TJ}}^{\oplus}$ as a *continualized pseudoinverse* of $\boldsymbol{C}_{\mathrm{TJ}}$.

One advantage of the presented continualized pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{\oplus}$ is that it exactly coincides with the original pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{+}$ as long as the singular values are either the exact zero or above the threshold ε. As for $\boldsymbol{C}_{\mathrm{TJ}}$, it has six non-zero singular values when $\boldsymbol{J}_x$ and $\boldsymbol{J}_s$ are full-rank, i.e., when the robot is not in the singular configurations. Therefore, as long as its largest six singular values are above the threshold ε, $\boldsymbol{C}_{\mathrm{TJ}}^{\oplus} = \boldsymbol{C}_{\mathrm{TJ}}^{+}$ is satisfied, and thus $\boldsymbol{\alpha}_x^*$ obtained by (36l) is the accurate solution of the lexicographic minimization problem (35).

It should be noted that all elements of $\boldsymbol{C}_{\mathrm{TJ}}$ are dimensionless quantities, and consequently, the singular values of $\boldsymbol{C}_{\mathrm{TJ}}$ are also dimensionless. This property is not generally true for arbitrary matrices; a matrix whose elements have different physical dimensions may have singular values having no consistent physical dimensions. Because the singular values of $\boldsymbol{C}_{\mathrm{TJ}}$ are dimensionless and physically consistent, the threshold ε used in (41) can also be understood as a physically consistent dimensionless quantity. This is not the case with (39), where neither ε_s nor ε_x does not have a consistent physical dimension.

Although the threshold ε is physically consistent, its choice still requires careful consideration. The threshold ε determines whether the task-space or the joint-space admittance is active in each direction; the joint-space admittance is active in directions in which the singular value of $\boldsymbol{C}_{\mathrm{TJ}}$ is below ε, and the task-space admittance is active otherwise. If T is small enough, the matrix $\boldsymbol{C}_{\mathrm{TJ}}$ reduces to the following:

$$\boldsymbol{C}_{\mathrm{TJ}} \approx \boldsymbol{M}^{-h}\boldsymbol{J}_s^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}_x\boldsymbol{M}^{-hT}. \tag{42}$$

The matrix $\boldsymbol{J}_s^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}_x \in \mathbb{R}^{n\times n}$ appearing here can be interpreted as *the task-space inertia evaluated in the joint space.* Thus, $\boldsymbol{C}_{\mathrm{TJ}}$ can be interpreted to represent the ratios of the task-space inertia $\boldsymbol{J}_s^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}_x$ to the joint-space inertia $\boldsymbol{M}$ evaluated in the joint space. In fact, in the vicinity of the singular configuration, the task-space inertia $\boldsymbol{J}_s^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}_x$ becomes lower in a particular direction in the joint space, which jeopardizes the stability. In this light, ε should be chosen considering *the lowest permissible task-space inertia in comparison to the joint-space inertia.*

If the joint inertia matrix $\boldsymbol{M}$ is chosen as small as the stability is maintained, $\boldsymbol{J}_s^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}_x \succeq \boldsymbol{M}$ is needed to ensure the stability. From this perspective, a conservative choice for the threshold ε would be around $\varepsilon = 1$. Unfortunately, some experimental observations revealed that the setting $\varepsilon = 1$ was overly conservative, disabling the task-space admittance even when the robot was far from the singular configuration. Although we currently do not have a definitive guideline for the choice of ε, we chose $\varepsilon = 0.03$ in our experimental setup comprising a 7-DOF manipulator, detailed in Section VI, through some trial and errors.

V. Some Analyses

A. Stability of Combined Proxy Dynamics

We here analyze the stability properties of the combined proxy dynamics, which are defined in Section IV-B and realized by the algorithm (36) in the discrete-time domain. For simplicity, we restrict the discussion to the case where $\boldsymbol{v}_r$, $\boldsymbol{a}_r$, $\boldsymbol{u}_r$, $\boldsymbol{\alpha}_r$, and $\boldsymbol{f}_r$ are all zeros, i.e., to the case where the reference positions are constant and the reference force is zero. We also assume that the position controller is not saturated and that the position controller is perfectly accurate. Then, we have $\boldsymbol{J}_x = \boldsymbol{J}_s$. For the notational brevity, we omit the subscript x within this section. Then, the continuous-time representation of the combined proxy dynamics realized by the algorithm (36) can be written as follows:

$$\dot{\boldsymbol{q}} = \boldsymbol{u} \tag{43a}$$

$$\begin{aligned}\dot{\boldsymbol{u}} = \operatorname*{arglexmin}_{\boldsymbol{\alpha}\in\mathbb{R}^n} \big(\{&\|\boldsymbol{J}^T\boldsymbol{M}_\mathrm{T}\boldsymbol{J}\boldsymbol{\alpha} \\ &+\boldsymbol{J}^T(\boldsymbol{B}_\mathrm{T}\boldsymbol{J}+\boldsymbol{M}_\mathrm{T}\boldsymbol{H})\boldsymbol{u} + \boldsymbol{J}^T\boldsymbol{\phi}_\mathrm{T}(\boldsymbol{\Phi}(\boldsymbol{q})\ominus\boldsymbol{p}_r)\|^2_{\boldsymbol{M}^{-1}}, \\ &\|\boldsymbol{M}\boldsymbol{\alpha}+\boldsymbol{B}\boldsymbol{u}+\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r)\|^2_{\boldsymbol{M}^{-1}}\}\big).\end{aligned} \tag{43b}$$

where $\boldsymbol{\phi}_\mathrm{T} : \mathbb{R}^6 \to \mathbb{R}^6$ and $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^n$ are functions defined as follows:

$$\boldsymbol{\phi}_\mathrm{T}(\Delta\boldsymbol{p}) \triangleq \mathrm{sat}_3(\boldsymbol{F}_\mathrm{T}, \boldsymbol{K}_\mathrm{T}\Delta\boldsymbol{p}) \tag{44}$$

$$\boldsymbol{\phi}(\Delta\boldsymbol{q}) \triangleq \mathrm{sat}_1(\boldsymbol{F}, \boldsymbol{K}\Delta\boldsymbol{q}). \tag{45}$$

Here, note that the state vector of the system is $\boldsymbol{x} \triangleq [\boldsymbol{q}^T, \boldsymbol{u}^T]^T \in \mathbb{R}^{2n}$. With some tedious but straightforward derivations using Corollary 1, one can rewrite (43b) in the following form:

$$\begin{aligned}\dot{\boldsymbol{u}} = &-\boldsymbol{J}^\#\boldsymbol{M}_\mathrm{T}^{-1}\left((\boldsymbol{B}_\mathrm{T}\boldsymbol{J}+\boldsymbol{M}_\mathrm{T}\boldsymbol{H})\boldsymbol{u}+\boldsymbol{\phi}_\mathrm{T}(\boldsymbol{\Phi}(\boldsymbol{q})\ominus\boldsymbol{p}_r)\right) \\ &-(\boldsymbol{I}_n-\boldsymbol{J}^\#\boldsymbol{J})\boldsymbol{M}^{-1}(\boldsymbol{B}\boldsymbol{u}+\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r))\end{aligned} \tag{46}$$

where

$$\boldsymbol{J}^\# \triangleq \boldsymbol{M}^{-hT}(\boldsymbol{M}_\mathrm{T}^{hT}\boldsymbol{J}\boldsymbol{M}^{-hT})^+\boldsymbol{M}_\mathrm{T}^{hT} \in \mathbb{R}^{n\times 6}. \tag{47}$$

Recall that $\boldsymbol{M}_\mathrm{T} = \boldsymbol{M}_\mathrm{T}^h\boldsymbol{M}_\mathrm{T}^{hT}$ and $\boldsymbol{M}^{-1} = \boldsymbol{M}^{-hT}\boldsymbol{M}^{-h}$. The matrix $\boldsymbol{J}^\#$ satisfies the following relations:

$$\boldsymbol{J}^\#\boldsymbol{J}\boldsymbol{J}^\# = \boldsymbol{J}^\# \tag{48a}$$

$$\boldsymbol{J}\boldsymbol{J}^\#\boldsymbol{J} = \boldsymbol{J} \tag{48b}$$

$$\boldsymbol{J}\boldsymbol{J}^\#\boldsymbol{M}_\mathrm{T}^{-1} = \boldsymbol{M}_\mathrm{T}^{-1}\boldsymbol{J}^{\#T}\boldsymbol{J}^T = \boldsymbol{J}\boldsymbol{J}^\#\boldsymbol{M}_\mathrm{T}^{-1}\boldsymbol{J}^{\#T}\boldsymbol{J}^T \tag{48c}$$

$$\boldsymbol{J}^\#\boldsymbol{J}\boldsymbol{M}^{-1} = \boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{J}^{\#T} = \boldsymbol{J}^\#\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{J}^{\#T}. \tag{48d}$$

Note that $\boldsymbol{J}^\#$ has similar properties to those of the dynamically consistent Jacobian inverse [31], [32].

Left-multiplying the both sides of (46) by $\boldsymbol{J}^T\boldsymbol{M}_\mathrm{T}\boldsymbol{J}$ yields the following:

$$\boldsymbol{J}^T\boldsymbol{M}_\mathrm{T}\boldsymbol{J}\dot{\boldsymbol{u}} = -\boldsymbol{J}^T((\boldsymbol{B}_\mathrm{T}\boldsymbol{J}+\boldsymbol{M}_\mathrm{T}\boldsymbol{H})\boldsymbol{u}+\boldsymbol{\phi}_\mathrm{T}(\boldsymbol{q}-\boldsymbol{p}_r)). \tag{49}$$

Here we used (48c) and (48b). Let us define a task-space energy function as follows:

$$V_\mathrm{T} \triangleq \frac{1}{2}\big(\boldsymbol{u}^T\boldsymbol{J}^T\boldsymbol{M}_\mathrm{T}\boldsymbol{J}\boldsymbol{u}+\Theta_\mathrm{T}(\boldsymbol{\Phi}(\boldsymbol{q})\ominus\boldsymbol{p}_r)\big). \tag{50}$$

Here, $\Theta_\mathrm{T} : \mathbb{R}^6 \to \mathbb{R}$ is a function satisfying

$$\boldsymbol{\phi}_\mathrm{T}(\Delta\boldsymbol{p}) = \frac{\partial\Theta_\mathrm{T}(\Delta\boldsymbol{p})}{\partial\Delta\boldsymbol{p}^T}. \tag{51}$$

Then, from (49), we have

$$\begin{aligned}\dot{V}_\mathrm{T} &= -\boldsymbol{u}^T\boldsymbol{J}^T\left((\boldsymbol{B}_\mathrm{T}\boldsymbol{J}+\boldsymbol{M}_\mathrm{T}\boldsymbol{H})\boldsymbol{u}+\boldsymbol{\phi}_\mathrm{T}(\boldsymbol{\Phi}(\boldsymbol{q})\ominus\boldsymbol{p}_r)\right) \\ &\quad+(\boldsymbol{J}\boldsymbol{u})^T\boldsymbol{\phi}_\mathrm{T}(\boldsymbol{\Phi}(\boldsymbol{q})\ominus\boldsymbol{p}_r)+\boldsymbol{u}^T\boldsymbol{J}^T\boldsymbol{M}_\mathrm{T}\boldsymbol{H}\boldsymbol{u} \\ &= -\boldsymbol{u}^T\boldsymbol{J}^T\boldsymbol{B}_\mathrm{T}\boldsymbol{J}\boldsymbol{u} \leq 0.\end{aligned} \tag{52}$$

The condition $\dot{V}_\mathrm{T} = 0$ holds when $\boldsymbol{J}\boldsymbol{u} = \boldsymbol{0}_6$. From (49), the invariance set under the condition $\boldsymbol{J}\boldsymbol{u} = \boldsymbol{0}_6$ can be obtained as

$$\boldsymbol{\mathcal{X}}_\mathrm{T} \triangleq \{[\boldsymbol{q}^T, \boldsymbol{u}^T]^T \in \mathbb{R}^{2n} \mid \boldsymbol{q} \in \boldsymbol{\mathcal{Q}}_\mathrm{T} \wedge \boldsymbol{J}\boldsymbol{u} = \boldsymbol{0}_6\} \tag{53}$$

where

$$\boldsymbol{\mathcal{Q}}_\mathrm{T} \triangleq \{\boldsymbol{q} \in \mathbb{R}^n \mid \boldsymbol{J}^T\boldsymbol{\phi}_\mathrm{T}(\boldsymbol{\Phi}(\boldsymbol{q})\ominus\boldsymbol{p}_r) = \boldsymbol{0}_n\}. \tag{54}$$

Therefore, from LaSalle's Invariance Principle, the set $\boldsymbol{\mathcal{X}}_\mathrm{T}$ is positively invariant and asymptotically stable. That is, we have $\boldsymbol{v} \triangleq \boldsymbol{J}\boldsymbol{u} \to \boldsymbol{0}_6$ and $\boldsymbol{J}^T\boldsymbol{\phi}_\mathrm{T}(\boldsymbol{\Phi}(\boldsymbol{q})\ominus\boldsymbol{p}_r) \to \boldsymbol{0}_n$ as $t \to \infty$.

Meanwhile, by using (48) and (49), (46) can be equivalently rewritten as follows:

$$\begin{aligned}\dot{\boldsymbol{u}} = &-(\boldsymbol{I}_n-\boldsymbol{J}^\#\boldsymbol{J})\boldsymbol{M}^{-1}(\boldsymbol{B}\boldsymbol{u}+\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r)) \\ &+\boldsymbol{J}^\#\left(\dot{\boldsymbol{v}}-\boldsymbol{H}\boldsymbol{u}\right)\end{aligned} \tag{55}$$

where we used the fact that $\dot{\boldsymbol{v}} = \boldsymbol{J}\dot{\boldsymbol{u}} + \boldsymbol{H}\boldsymbol{u}$. By using (48), (55) can be equivalently rewritten as follows:

$$\begin{aligned}\dot{\boldsymbol{u}} = &-\boldsymbol{M}^{-1}(\boldsymbol{I}_n-\boldsymbol{J}^T\boldsymbol{J}^{\#T})(\boldsymbol{B}\boldsymbol{u}+\boldsymbol{K}(\boldsymbol{q}-\boldsymbol{q}_r)) \\ &+\boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{J}^{\#T}\boldsymbol{M}\boldsymbol{J}^\#\left(\dot{\boldsymbol{v}}-\boldsymbol{H}\boldsymbol{u}\right).\end{aligned} \tag{56}$$

We also have the following expressions:

$$(\boldsymbol{I}_n-\boldsymbol{J}^T\boldsymbol{J}^{\#T})(\boldsymbol{M}\dot{\boldsymbol{u}}+\boldsymbol{B}\boldsymbol{u}+\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r)) = \boldsymbol{0}_n \tag{57}$$

$$(\boldsymbol{I}_n-\boldsymbol{J}^\#\boldsymbol{J})(\dot{\boldsymbol{u}}+\boldsymbol{M}^{-1}(\boldsymbol{B}\boldsymbol{u}+\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r))) = \boldsymbol{0}_n, \tag{58}$$

which are directly derived from (56). Let us define a joint-space energy function as follows:

$$V_\mathrm{J} = \frac{1}{2}(\boldsymbol{u}^T\boldsymbol{M}\boldsymbol{u}+\Theta(\boldsymbol{q},\boldsymbol{q}_r)) \tag{59}$$

where $\Theta : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a function satisfying

$$\boldsymbol{\phi}(\Delta\boldsymbol{q}) = \frac{\partial\Theta(\Delta\boldsymbol{q})}{\partial\Delta\boldsymbol{q}^T}. \tag{60}$$

Then, its time derivative is given as follows:

$$\begin{aligned}\dot{V}_\mathrm{J} &= \boldsymbol{u}^T(\boldsymbol{M}\dot{\boldsymbol{u}}+\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r)) \\ &= -\boldsymbol{u}^T(\boldsymbol{I}_n-\boldsymbol{J}^T\boldsymbol{J}^{\#T})\boldsymbol{B}\boldsymbol{u}+\boldsymbol{u}^T\boldsymbol{J}^T\boldsymbol{J}^{\#T}\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r) \\ &\quad+\boldsymbol{u}^T\boldsymbol{M}\boldsymbol{J}^\#(\dot{\boldsymbol{v}}-\boldsymbol{H}\boldsymbol{u}) \\ &= -\boldsymbol{u}^T\boldsymbol{B}\boldsymbol{u}+\boldsymbol{v}^T\boldsymbol{J}^{\#T}\big(\boldsymbol{B}\boldsymbol{u}+\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r) \\ &\quad+\boldsymbol{M}\boldsymbol{J}^\#(\dot{\boldsymbol{v}}-\boldsymbol{H}\boldsymbol{u})\big).\end{aligned} \tag{61}$$

This expression implies that $\dot{V}_\mathrm{J} \leq 0$ is satisfied when $\boldsymbol{v} = \boldsymbol{0}_6$, and $\dot{V}_\mathrm{J} = 0$ is satisfied when $\boldsymbol{u} = \boldsymbol{0}_n$. In addition, (56) implies that the invariance set under the condition $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_\mathrm{T} \wedge \boldsymbol{u} = \boldsymbol{0}_n$ can be written as

$$\boldsymbol{\mathcal{X}}_\mathrm{J} \triangleq \{[\boldsymbol{q}^T, \boldsymbol{u}^T]^T \in \mathbb{R}^{2n} \mid \boldsymbol{q} \in \boldsymbol{\mathcal{Q}}_\mathrm{J} \wedge \boldsymbol{u} = \boldsymbol{0}_n\} \tag{62a}$$

where

$$\boldsymbol{\mathcal{Q}}_\mathrm{J} \triangleq \boldsymbol{\mathcal{Q}}_\mathrm{T} \cap \{\boldsymbol{q} \in \mathbb{R}^n \mid (\boldsymbol{I}_n-\boldsymbol{J}^T\boldsymbol{J}^{\#T})\boldsymbol{\phi}(\boldsymbol{q}-\boldsymbol{q}_r) = \boldsymbol{0}_n\}. \tag{62b}$$

Therefore, from LaSalle's Invariance Principle, as long as $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_{\mathrm{T}}$, the set $\boldsymbol{\mathcal{X}}_{\mathrm{J}} \subset \boldsymbol{\mathcal{X}}_{\mathrm{T}}$ is asymptotically stable. Because $\boldsymbol{\mathcal{X}}_{\mathrm{T}}$ is asymptotically reached by $\boldsymbol{x}$, one can see that the set $\boldsymbol{\mathcal{X}}_{\mathrm{J}} \subset \boldsymbol{\mathcal{X}}_{\mathrm{T}}$ is also asymptotically asymptotically stable with the combined proxy dynamics (43). This means that the task-space position $\boldsymbol{p} = \boldsymbol{\Phi}(\boldsymbol{q})$ eventually converges to the position satisfying $\boldsymbol{J}^T \boldsymbol{\phi}_{\mathrm{T}}(\boldsymbol{p} \ominus \boldsymbol{p}_r) = \boldsymbol{0}_n$ and the nullspace motion also eventually settles, bringing the entire system into a steady state.

It should be noted that the set $\boldsymbol{\mathcal{Q}}_{\mathrm{J}}$ of configurations can be said to be the set of the equilibrium configurations of the combined proxy dynamics (43). If $\boldsymbol{p}_r$ is set within the reachable workspace of the robot except for the singular configurations, i.e., so that there exists $\boldsymbol{q}$ such that $\boldsymbol{p}_r = \boldsymbol{\Psi}(\boldsymbol{q})$ and $\operatorname{rank} \boldsymbol{J}(\boldsymbol{q}) = 6$, it reduces to

$$\begin{aligned} \boldsymbol{\mathcal{Q}}_{\mathrm{J}} = \big\{ \boldsymbol{q} \in \mathbb{R}^n \mid \boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{p}_r \, \wedge \\ (\boldsymbol{I}_n - \boldsymbol{J}^T \boldsymbol{J}^{\#T}) \boldsymbol{\phi}(\boldsymbol{q} - \boldsymbol{q}_r) = \boldsymbol{0}_n \big\}. \end{aligned} \tag{63}$$

Moreover, if $\boldsymbol{q}_r$ and $\boldsymbol{p}_r$ are set as $\boldsymbol{p}_r = \boldsymbol{\Psi}(\boldsymbol{q}_r)$ and $\operatorname{rank} \boldsymbol{J}(\boldsymbol{q}_r) = 6$, the set of equilibrium configurations reduces to $\boldsymbol{\mathcal{Q}}_{\mathrm{J}} = \{\boldsymbol{q}_r\}$. This means that, if $\boldsymbol{p}_r$ is set within the reachable workspace of the robot and $\boldsymbol{q}_r$ is set consistent with $\boldsymbol{p}_r$, the combined proxy dynamics (43) drives $\boldsymbol{q}$ to $\boldsymbol{q}_r$, resulting in the task-space position satisfying $\boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{p}_r$.

Remark 1. *It should be noted that, when $\boldsymbol{J}$ is full-rank, $\boldsymbol{J}^{\#}$ in (47) reduces to $\boldsymbol{J}^{\#} = \boldsymbol{M}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}$, which coincides with the dynamically consistent Jacobian inverse [31], [32]. The original definition of the dynamically consistent Jacobian inverse $\boldsymbol{J}^{\#}$ is chosen so that the following is satisfied:*

$$\forall \boldsymbol{\tau} \in \mathbb{R}^n, \quad \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau} = \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{J}^{\#T}\boldsymbol{\tau}, \tag{64}$$

which means that $\boldsymbol{J}^{\#T}$ maps any torque vector $\boldsymbol{\tau}$ to a task-space force vector $\boldsymbol{f}$ that results in the same task-space acceleration $\boldsymbol{a}$ as $\boldsymbol{\tau}$ results in according to the joint-space dynamics. The matrix $\boldsymbol{J}^{\#}$ defined in (47) satisfies (64) and also the following:

$$\forall \boldsymbol{\tau} \in \mathbb{R}^n, \ \exists \boldsymbol{\alpha} \in \mathbb{R}^n \text{ s.t. } \boldsymbol{J}\boldsymbol{\alpha} = \boldsymbol{M}_{\mathrm{T}}^{-1}\boldsymbol{J}^{\#T}\boldsymbol{\tau}, \tag{65}$$

which means that $\boldsymbol{J}^{\#T}$ maps any torque vector $\boldsymbol{\tau}$ to a task-space force vector $\boldsymbol{f}$ that results in a kinematically-realizable task-space acceleration $\boldsymbol{a}$ according to the task-space dynamics. In conclusion, $\boldsymbol{J}^{\#}$ in (47) acts as a Jacobian inverse that is consistent both in the joint-space and the task-space dynamics.

Remark 2. *The discrete-time algorithm (36) involves $\boldsymbol{C}_{\mathrm{TJ}}^{+}$ instead of $\boldsymbol{J}^{\#}$. Recall that, by setting T small enough, $\boldsymbol{C}_{\mathrm{TJ}}$ can be written as (42). From this, its pseudoinverse can be given as follows:*

$$\begin{aligned} \boldsymbol{C}_{\mathrm{TJ}}^{+} &= (\boldsymbol{M}_{\mathrm{T}}^{hT}\boldsymbol{J}_x\boldsymbol{M}^{-hT})^{+}(\boldsymbol{M}^{-h}\boldsymbol{J}_s^T\boldsymbol{M}_{\mathrm{T}}^{h})^{+} \\ &= \boldsymbol{M}^{hT}\boldsymbol{J}_x^{\#}\boldsymbol{M}_{\mathrm{T}}^{-1}\boldsymbol{J}_s^{\#T}\boldsymbol{M}^{h}. \end{aligned} \tag{66}$$

This implies that the proposed algorithm (36) implicitly and approximately involves the matrices $\boldsymbol{J}_x^{\#}$ and $\boldsymbol{J}_s^{\#}$. In addition, the term $\boldsymbol{C}_{\mathrm{TJ}}^{+}\boldsymbol{C}_{\mathrm{TJ}}$, which appears in (36l), is related to $\boldsymbol{J}_x^{\#}$ and $\boldsymbol{J}_s^{\#}$ by the following:

$$\begin{aligned} \boldsymbol{C}_{\mathrm{TJ}}^{+}\boldsymbol{C}_{\mathrm{TJ}} &= \boldsymbol{M}^{hT}\boldsymbol{J}_x^{\#}\boldsymbol{M}_{\mathrm{T}}^{-1}\boldsymbol{J}_s^{\#T}\boldsymbol{J}_s^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}_x\boldsymbol{M}^{-hT} \\ &= \boldsymbol{M}^{hT}\boldsymbol{J}_x^{\#}\boldsymbol{J}_s\boldsymbol{J}_s^{\#}\boldsymbol{J}_x\boldsymbol{M}^{-hT}, \end{aligned} \tag{67}$$

and, by assuming $\boldsymbol{J}_s \approx \boldsymbol{J}_x = \boldsymbol{J}$, it reduces to

$$\boldsymbol{C}_{\mathrm{TJ}}^{+}\boldsymbol{C}_{\mathrm{TJ}} \approx \boldsymbol{M}^{hT}\boldsymbol{J}^{\#}\boldsymbol{J}\boldsymbol{M}^{-hT} = \boldsymbol{M}^{-h}\boldsymbol{J}^T\boldsymbol{J}^{\#T}\boldsymbol{M}^{h}. \tag{68}$$

This expression allows for the interpretation of (36l) in relation to the so-called nullspace projector [19], [21], which may be helpful for further theoretical investigation.

B. Effect of the Weight Matrices

Here we show that the weight matrix $\boldsymbol{M}^{-1}$ appearing in the discrete-time representation (35) and the continuous-time one (43b) is necessary to guarantee the stability of the task-space dynamics. Let us consider the case where (43b) is replaced by

$$\begin{aligned} \dot{\boldsymbol{u}} = \operatorname*{arglexmin}_{\boldsymbol{\alpha} \in \mathbb{R}^n} \big(\big\{ \| & \boldsymbol{J}^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}\boldsymbol{\alpha} \\ & + \boldsymbol{J}^T(\boldsymbol{B}_{\mathrm{T}}\boldsymbol{J} + \boldsymbol{M}_{\mathrm{T}}\boldsymbol{H})\boldsymbol{u} + \boldsymbol{J}^T\boldsymbol{\phi}_{\mathrm{T}}(\boldsymbol{\Phi}(\boldsymbol{q}) \ominus \boldsymbol{p}_r)\|_{\boldsymbol{X}^T\boldsymbol{X}}^2, \\ & \|\boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\phi}(\boldsymbol{q} - \boldsymbol{q}_r)\|_{\boldsymbol{Y}^T\boldsymbol{Y}}^2 \big\} \big) \end{aligned} \tag{69}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are square matrices. Then, from Corollary 1, it results in

$$\dot{\boldsymbol{u}} = (\boldsymbol{Y}\boldsymbol{M})^{-1}\boldsymbol{C}^{+}\boldsymbol{X}\boldsymbol{J}^T\boldsymbol{c}_{\mathrm{T}} + (\boldsymbol{Y}\boldsymbol{M})^{-1}(\boldsymbol{I} - \boldsymbol{C}^{+}\boldsymbol{C})\boldsymbol{Y}\boldsymbol{c}_{\mathrm{J}} \tag{70}$$

where

$$\boldsymbol{C} \triangleq \boldsymbol{X}\boldsymbol{J}^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}(\boldsymbol{Y}\boldsymbol{M})^{-1} \tag{71}$$

$$\boldsymbol{c}_{\mathrm{T}} \triangleq (\boldsymbol{B}_{\mathrm{T}}\boldsymbol{J} + \boldsymbol{M}_{\mathrm{T}}\boldsymbol{H})\boldsymbol{u} + \boldsymbol{\phi}_{\mathrm{T}}(\boldsymbol{\Phi}(\boldsymbol{q}) \ominus \boldsymbol{p}_r) \tag{72}$$

$$\boldsymbol{c}_{\mathrm{J}} \triangleq \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\phi}(\boldsymbol{q} - \boldsymbol{q}_r). \tag{73}$$

Left-multiplying the both sides of (70) by $\boldsymbol{J}^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}$ yields the following:

$$\boldsymbol{J}^T\boldsymbol{M}_{\mathrm{T}}\boldsymbol{J}\dot{\boldsymbol{u}} = -\boldsymbol{X}^{-1}\boldsymbol{C}\boldsymbol{C}^{+}\boldsymbol{X}\boldsymbol{J}^T\boldsymbol{c}_{\mathrm{T}}. \tag{74}$$

Note that the obtained (74) is equivalent to (49) only if

$$\boldsymbol{X}^{-1}\boldsymbol{C}\boldsymbol{C}^{+}\boldsymbol{X}\boldsymbol{J}^T = \boldsymbol{J}^T \tag{75}$$

is satisfied, and the satisfaction of (75) is needed to guarantee the stability of the set $\boldsymbol{\mathcal{X}}_{\mathrm{T}}$ defined in (53) by satisfying $\dot{V}_{\mathrm{T}} \leq 0$. In order to satisfy (75), one sufficient condition is $\boldsymbol{X} = \boldsymbol{Y} = \boldsymbol{M}^{-h}$. This means that the weight matrices need to be chosen to guarantee the stability of the task-space dynamics, and $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{Y}^T\boldsymbol{Y} = \boldsymbol{M}^{-1}$ is an appropriate choice.

VI. Experiments

A. Experimental Setup.

The proposed method was tested with an experimental setup shown in Fig. 5, which is a Kinova Gen3 robot. It has seven joints equipped with torque sensors. The robot was connected to a PC running Windows OS, and the controllers for the robots were constructed with Microsoft Visual C++.

The proposed controller, which is the algorithm (18) with the function $\mathrm{TwoProxies}$ in (36) and the function $\mathrm{SatPosCtrl}$ in (16), was implemented with the sampling interval $T =$

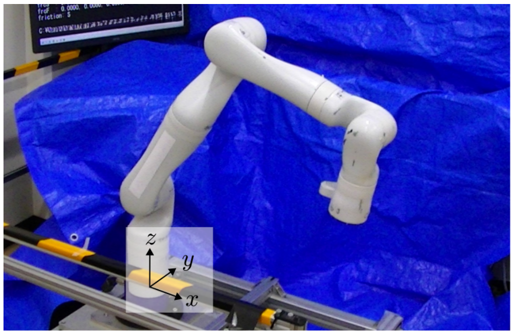


Fig. 5. Experimental setup (Kinova Gen 3).

0.001 s. The basic settings for the controller parameters were chosen in the following procedure. First, the torque limits F_c for the joints were set as 80 % of the maximum torque [33]; $F_c = 43.2$ Nm for the first four joints and $F_c = 27.2$ Nm for the other joints. Second, the proxy dynamics were disabled (i.e., $\boldsymbol{q}_x^*$ in (18) was kept constant), and the gain values (K_c, B_c, and L_c) for the function $\mathrm{SatPosCtrl}$ were chosen so that the joint angles were stiffly maintained at $\boldsymbol{q}_x^*$. Through some trial and errors, they were chosen as $\{K_c, B_c, L_c\} = \{1500 \text{ Nm}, 30 \text{ Nms}, 300 \text{ Nm/s}\}$ for the first four joints and $\{1000 \text{ Nm}, 20 \text{ Nms}, 200 \text{ Nm/s}\}$ for the other joints. Third, the task-space proxy dynamics was disabled by overwriting $\boldsymbol{C}_{\mathrm{TJ}}^{+} := \boldsymbol{O}_{7\times 7}$ in the function $\mathrm{TwoProxies}$, and the joint-space proxy parameters were chosen to realize stable joint-independent admittance control, resulting in:

$$\boldsymbol{M} = \mathrm{diag}\,[1.2, 1.2, 0.8, 0.8, 0.2, 0.2, 0.2] \text{ kg}{\cdot}\text{m}^2 \tag{76a}$$

$$\boldsymbol{B} = (1 \text{ s}^{-1}) \times \boldsymbol{M} \tag{76b}$$

$$\boldsymbol{K} = (0.25 \text{ s}^{-2}) \times \boldsymbol{M}, \tag{76c}$$

which realizes critical damping with the time constant of 2 s. The force limits for the joint-space proxy springs were set as

$$\boldsymbol{F} = [30, 30, 30, 30, 20, 20, 20]^T \text{ Nm}. \tag{77}$$

The reference position for the joint-space proxy was set as follows:

$$\boldsymbol{q}_r \equiv [0, 0, 0, 0, 0, 0, 0]^T \text{rad}. \tag{78}$$

Finally, the task-space proxy parameters were chosen to realize stable task-space admittance control. Some preliminary experiments resulted in:

$$\boldsymbol{M}_{\mathrm{T}} = \mathrm{blockdiag}\,[(2 \text{ kg}) \times \boldsymbol{I}_3, (0.2 \text{ kg}{\cdot}\text{m}^2) \times \boldsymbol{I}_3] \tag{79a}$$

$$\boldsymbol{B}_{\mathrm{T}} = (4 \text{ s}^{-1}) \times \boldsymbol{M}_{\mathrm{T}} \tag{79b}$$

$$\boldsymbol{K}_{\mathrm{T}} = (4 \text{ s}^{-2}) \times \boldsymbol{M}_{\mathrm{T}}, \tag{79c}$$

which realizes critical damping with the time constant of 0.5 s. The force limit for the task-space proxy spring was set as

$$\boldsymbol{F}_{\mathrm{T}} = [100 \text{ N}, \ 10 \text{ Nm}]^T. \tag{80}$$

Unless otherwise noted, the reference position for the task-space proxy was set as

$$\boldsymbol{p}_r \equiv [0.5 \text{ m}, 0 \text{ m}, 0.4 \text{ m}, 0, 0, 1, 0]^T, \tag{81}$$

and the reference force was set as $\boldsymbol{f}_r \equiv \boldsymbol{0}_6$.

In the function $\mathrm{TwoProxies}$, the pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{+}$ was replaced by the continualized pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{\oplus}$, detailed in Section IV-C, where the parameter ε was chosen as $\varepsilon = 0.03$. It was chosen as small as possible without causing oscillations when the robot was fully stretched.

The gravity compensation was performed by computing the gravity torques from the mass parameters of the links provided in [33]. The computed gravity torques were superposed to the output $\boldsymbol{\tau}_m$ of the controller (18).

The proposed controller was compared to a variant of the controller in which the function $\mathrm{SatPosCtrl}$ was disabled, to validate its capability of handling actuator torque saturation. Controllers depending on the robot's dynamics model, such as those in [17], [19], [21], were not included in the empirical comparison because the performance of such controllers would depend on the accuracy of the model. Being free from an explicit dynamics model can count as an advantage of the proposed controller, particularly for robots with highly frictional joints and complex dynamics.

Comparisons among different approximations of the pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{+}$ were not performed either in the experiments. This is because it would heavily depend on the choice of parameters of each approximation scheme. This paper does not argue the superiority of the continualized pseudoinverse $\boldsymbol{C}_{\mathrm{TJ}}^{\oplus}$ although it is a reasonable choice among various possible approximations as long as the singular value decomposition is allowed.

B. Experiments IA to ID: Moved by Hand

The first set of experiments was conducted to validate the task-space admittance control realized by the proposed method. As an Experiment IA, the experimenter performed the maneuver shown by the snapshots in Fig. 6(a), which is divided into the following periods:

- **Period A** ($t \in [0 \text{ s}, 5 \text{ s}]$): The experimenter lightly pushed the end-effector left and right, in the y direction, by his palm, without holding it.
- **Period B** ($t \in [5 \text{ s}, 7 \text{ s}]$): The experimenter pulled the end-effector in the $+x$ direction.
- **Period C** ($t \in [7 \text{ s}, 15 \text{ s}]$): The experimenter released his hand from the end-effector.

Fig. 6(b) shows the singular values of the matrix $\boldsymbol{C}_{\mathrm{TJ}}$ during the maneuver.

As can be seen in Fig. 6(a), in Period A, the end-effector moved left and right almost along a straight line without changing the attitude, although the experimenter did not attempt to hold it at a fixed attitude. It indicates the validity of the task-space dynamics realized by the proposed controller. In Period B, the robot came into a singular configuration, and some singular values became smaller as in Fig. 6(b). It indicates that the controller tolerates the singular configuration. In Period C, the robot returned to the original position, which was the reference position $\boldsymbol{p}_r$, indicating the spring term in the task-space proxy dynamics (20) worked properly.

Additional trials were performed with different parameter settings. As an Experiment IB, the rotational components of $\{\boldsymbol{M}_{\mathrm{T}}, \boldsymbol{B}_{\mathrm{T}}, \boldsymbol{K}_{\mathrm{T}}\}$ were scaled by 1000 from the values in (79),

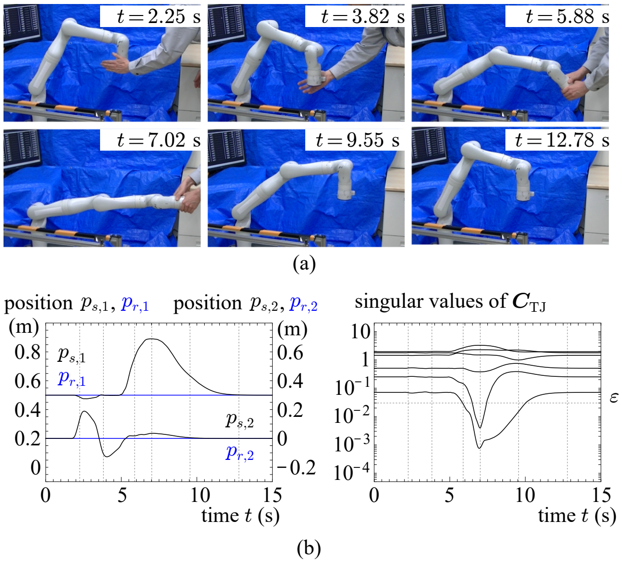


Fig. 6. Experiment IA. (a) Snapshots. (b) End-effector position in x and y directions and the singular values of $\boldsymbol{C}_{\mathrm{TJ}}$. (The vertical lines in the graphs correspond to the snapshots.)

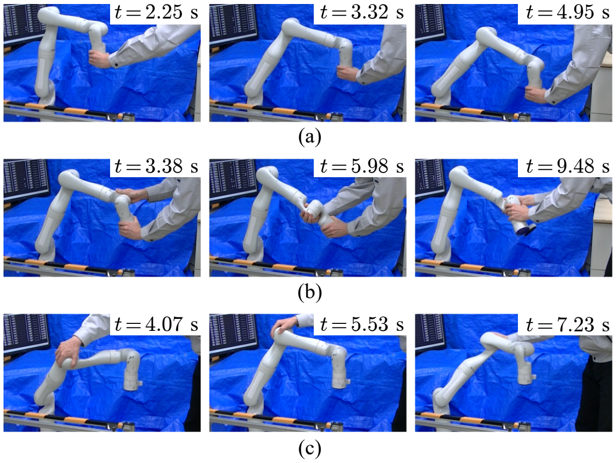


Fig. 7. Snapshots of (a) Experiment IB, (b) Experiment IC, and (c) Experiment ID.

and the experimenter moved the end-effector randomly around the initial position. As seen in Fig. 7(a), the end-effector was moved without changing its attitude, indicating the validity of the task-space dynamics realized by the proposed controller.

As an Experiment IC, the translational components of $\{\boldsymbol{M}_{\mathrm{T}}, \boldsymbol{B}_{\mathrm{T}}, \boldsymbol{K}_{\mathrm{T}}\}$ were scaled by 1000 from the values in (79), and the experimenter attempted to rotate the end-effector randomly. As seen in Fig. 7(b), the end-effector changed its attitude without changing its position. It should be emphasized that it even passed through the wrist singular configuration, as can be seen in $t = 5.98$ s of Fig. 7(b). This behavior indicates that the controller properly realizes the rotational dynamics and maintains validity even in singular configurations.

As an Experiment ID, all elements of $\{\boldsymbol{M}_{\mathrm{T}}, \boldsymbol{B}_{\mathrm{T}}, \boldsymbol{K}_{\mathrm{T}}\}$ were scaled by 1000 from the values in (79), and the experimenter pushed the elbow joint of the robot. As seen in Fig. 7(c), the robot moved only its elbow part without moving the end-effector, indicating the validity of the nullspace dynamics realized by the proposed controller.

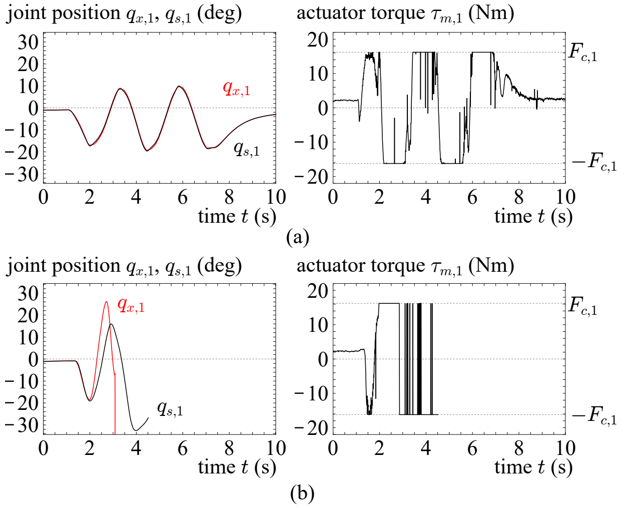


Fig. 8. Results of Experiment II. (a) Proposed controller. (b) Controller with the function $\mathrm{SatPosCtrl}$ disabled, with (16e) being replaced with $q_x := q^*$.

C. Experiment II: Behavior Under Torque Saturation

Another set of experiments was performed to test the effect of torque saturation. In this experiment, The parameters were set as in Section VI-A except F_c being set to be 30 % of the values indicated in Section VI-A. The robot was initially set at $\boldsymbol{p}_r$ in (81), and the experimenter moved the end-effector left and right, in the y direction, by holding it by hand.

Fig. 8 (a) shows a part of the results; the measured angle $q_{s,1}$, the joint-space proxy position $q_{x,1}$, and the command torque $\tau_{m,1}$ of the base (first) joint. It is shown that the torque $\tau_{m,1}$ was saturated but the proxy position $q_{x,1}$ remained close to the measured position $q_{s,1}$, resulting in stable behavior. In fact, after $t = 7$ s when the experimenter left the hand from the robot, the robot reached a settled state.

For the comparison, another trial was performed with a variant of the proposed controller in which the function $\mathrm{SatPosCtrl}$ was replaced by an ordinary torque-saturated PID controller. Specifically, in the algorithm of the function $\mathrm{SatPosCtrl}$, the line (16e) was replaced by $q_x := q^*$, which means that its internal position controller was a PID controller that saturates the torque without maintaining consistency with the proxy position $\boldsymbol{q}_x$. With this controller, the experimenter held the end-effector and attempted to move it left and right, in the y direction. The results of this case are shown in Fig. 8(b). It can be seen that, once the torques were saturated, the proxy position $\boldsymbol{q}_x$ deviated from the actual position $\boldsymbol{q}_s$. In fact, the experimenter's hand was pulled by the force swinging around unpredictably, and he had to grasp the end-effector tightly to prevent unsafe motion. The experimenter at last had to stop the robot at $t = 4.7$ s to prevent further violent behaviors.

These results suggest that the proposed controller properly inherits the beneficial properties of TBAC as seen in Fig. 8(a), and it is effective in preventing such undesirable behaviors seen in Fig. 8(b).

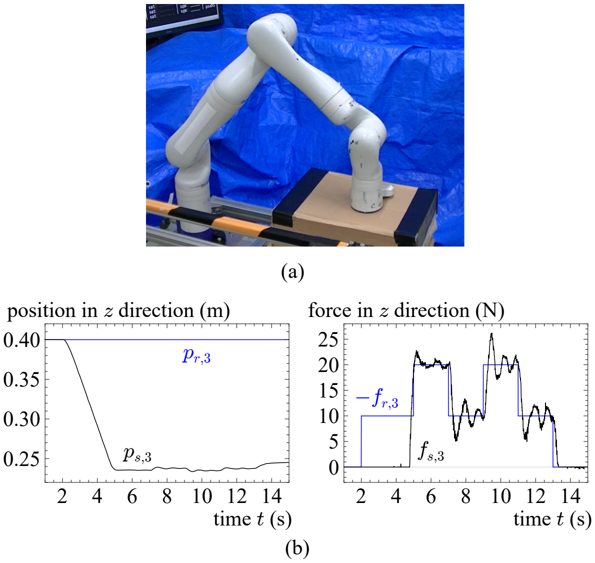


Fig. 9. Experiment III. (a) Robot pushing the contact environment. (b) Contact force and reference force in z direction.

D. Experiment III: Contact Force Control

Experiment III was performed to test the capability of the proposed controller in the contact force control. As a contact environment, a cardboard box was placed below the end-effector as shown in Fig 9(a). The reference force was set as $\boldsymbol{f}_r = [0, 0, f_{r,3}, 0, 0, 0]^T$ and $f_{r,3}$, which is the z component of the translational reference force, was varied in a step-like manner among 0 N, 10 N, and 20 N. To prevent oscillation that happened in preliminary experiments involving the environment contact, all elements of $\{\boldsymbol{M}_\mathrm{T}, \boldsymbol{B}_\mathrm{T}, \boldsymbol{K}_\mathrm{T}\}$ were scaled by 20 from the values in (79). The gain matrices were set as (79) except the $(3,3)$-th element of $\boldsymbol{K}_\mathrm{T}$ being set zero.

The results are shown in Fig. 9(b). The graph shows the third element (z component) of the force vector $\boldsymbol{f}_s = \boldsymbol{J}(\boldsymbol{q}_s)^{+}\boldsymbol{\tau}_s$. The results show that the contact force $f_{s,3}$ appropriately tracked the negative of the reference force $-f_{r,3}$, showing the applicability of the proposed admittance controller as a contact force controller.

E. Experiment IV: Unreachable Reference Position

Experiment IV was performed to test the controller's response to the reference position $\boldsymbol{p}_r$ set outside the reachable workspace. The reference position $\boldsymbol{p}_r$ was initial set as (81) and was linearly moved to $\boldsymbol{p}_r = [0\ \mathrm{m}, 0\ \mathrm{m}, 1.5\ \mathrm{m}, 1, 0, 0, 0]^T$ from $t = 0$ s to $t = 5$ s. This $\boldsymbol{p}_r$ represents a position right above the robot base with an upward-pointing attitude, located outside the robot's reachable workspace. After $t = 10$ s, the experimenter applied some forces on the end-effector and the links.

Fig. 10(a) shows some snapshots of this experiment and Fig. 10(b) shows the temporal changes of the end-effector positions and the singular values of $\boldsymbol{C}_\mathrm{TJ}$. It can be seen that the robot reached an equilibrium at a configuration pointing straight upward around $t = 4.5$ s. After this, two of the singular values of $\boldsymbol{C}_\mathrm{TJ}$ were below the threshold ε. Between $t = 9$ s and $t = 12$ s, the experimenter was able to twist the middle links, indicating the validity of the joint-space dynamics realized in the nullspace. Then, the experimenter moved the end-effector by lightly holding it, and the end-effector moved without much changes in the end-effector attitude, exhibiting the validity of the task-space admittance control. After the experimenter left his hand from the robot, the robot returned to the upward-pointing attitude, indicating the proper realization of the effect of the task-space proxy spring.

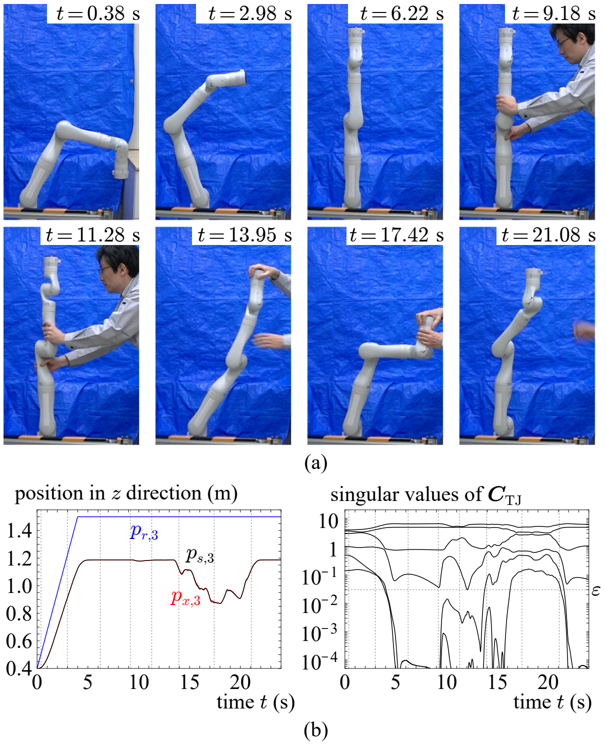


Fig. 10. Experiment IV. (a) Snapshots. (b) End-effector position in z direction and the singular values of $\boldsymbol{C}_\mathrm{TJ}$. (The vertical lines in the graphs correspond to the snapshots.

VII. Conclusions

This paper has proposed a task-space version of a torque-bounded admittance controller (TABC) that tolerates singular configurations and can be used with redundant manipulators. It imposes an explicit torque limit to each joint without causing undesirable or unsafe side effects, while it enforces the end-effector to follow predefined task-space proxy dynamics as long as the joint torques are unsaturated and the robot is out of the singular configurations. The behavior of the robot in the nullspace is determined by predefined joint-space proxy dynamics. The task- and joint-space proxy dynamics are combined in a lexicographic manner, with thresholding on singular values of a particular matrix derived from Jacobian matrices. The controller was validated through experiments with a seven-degree-of-freedom (DOF) Kinova Gen3 robot. The results illustrate that the proposed controller acts as a proper task-space admittance controller that safely handles actuator force saturation and singular configurations.

Appendix A
Quaternion Calculus

The attitude of a rigid body in the Cartesian space can be represented by a 3×3 rotation matrix ($\in \mathrm{SO}(3)$) whose column vectors represent the axes of the body-fixed frame. A unit quaternion can also represent the attitude of a rigid body. A unit quaternion $\boldsymbol{a} = [a_w, a_x, a_y, a_z]^T \in \mathbb{H}$ to its correspondent 3×3 rotation matrix can be associated by the following map:

$$\mathrm{q2m}(\boldsymbol{a}) \triangleq \begin{bmatrix} a_w^2 + a_x^2 - a_y^2 - a_z^2 & 2(a_x a_y - a_w a_z) & 2(a_x a_z + a_w a_y) \\ 2(a_x a_y + a_w a_z) & a_w^2 - a_x^2 + a_y^2 - a_z^2 & 2(a_y a_z - a_w a_x) \\ 2(a_x a_z - a_w a_y) & 2(a_y a_z + a_w a_x) & a_w^2 - a_x^2 - a_y^2 + a_z^2 \end{bmatrix}.$$

We define the time derivative operator $\circ$ for a unit quaternion $\boldsymbol{a} = [a_w, a_x, a_y, a_z]^T$ as follows:

$$\mathring{\boldsymbol{a}} \triangleq 2 \begin{bmatrix} -a_x & a_w & -a_z & a_y \\ -a_y & a_z & a_w & -a_x \\ -a_z & -a_y & a_x & a_w \end{bmatrix} \begin{bmatrix} \dot{a}_w \\ \dot{a}_x \\ \dot{a}_y \\ \dot{a}_z \end{bmatrix}. \tag{82}$$

Note that $\boldsymbol{a} \in \mathbb{H}$ and $\mathring{\boldsymbol{a}} \in \mathbb{R}^3$. With this operator, the angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ and the attitude $\boldsymbol{a} \in \mathbb{H}$ of a rigid body can be associated as $\boldsymbol{\omega} = \mathring{\boldsymbol{a}}$.

Now we prepare some operators to relate the discrete-time representations of the attitude quaternions and the angular velocity vectors. First, the quaternion multiplication $\otimes : \mathbb{H} \times \mathbb{H} \to \mathbb{H}$ is defined as follows:

$$\begin{bmatrix} a_w \\ a_x \\ a_y \\ a_z \end{bmatrix} \otimes \begin{bmatrix} b_w \\ b_x \\ b_y \\ b_z \end{bmatrix} \triangleq \begin{bmatrix} a_w b_w - a_x b_x - a_y b_y - a_z b_z \\ a_x b_w + a_w b_x - a_z b_y + a_y b_z \\ a_y b_w + a_z b_x + a_w b_y - a_x b_z \\ a_z b_w - a_y b_x + a_x b_y + a_w b_z \end{bmatrix}.$$

Based on this, let the operators $\ominus : \mathbb{H} \times \mathbb{H} \to \mathbb{R}^3$ and $\oplus : \mathbb{H} \times \mathbb{R}^3 \to \mathbb{H}$ be defined as follows:[3]

$$\boldsymbol{a} \ominus \boldsymbol{b} \triangleq \mathrm{q2v}\left(\boldsymbol{a} \otimes \mathrm{inv}(\boldsymbol{b})\right) \tag{83}$$

$$\boldsymbol{a} \oplus \boldsymbol{r} \triangleq \mathrm{v2q}(\boldsymbol{r}) \otimes \boldsymbol{a} \tag{84}$$

where $\mathrm{inv} : \mathbb{H} \to \mathbb{H}$, $\mathrm{q2v} : \mathbb{H} \to \mathbb{R}^3$, and $\mathrm{v2q} : \mathbb{R}^3 \to \mathbb{H}$ are defined as follows:

$$\mathrm{inv}\left(\begin{bmatrix} a_w \\ \boldsymbol{a}_{xyz} \end{bmatrix}\right) \triangleq \begin{bmatrix} a_w \\ -\boldsymbol{a}_{xyz} \end{bmatrix} \tag{85}$$

$$\mathrm{q2v}\left(\begin{bmatrix} a_w \\ \boldsymbol{a}_{xyz} \end{bmatrix}\right) \triangleq \frac{2\,\mathrm{sgn}(a_w)}{\mathrm{sinc}\left(\mathrm{asin}(\|\boldsymbol{a}_{xyz}\|)\right)} \boldsymbol{a}_{xyz} \tag{86}$$

$$\mathrm{v2q}\left(\boldsymbol{r}\right) \triangleq \begin{bmatrix} \cos(\|\boldsymbol{r}\|/2) \\ \boldsymbol{r}\,\mathrm{sinc}(\|\boldsymbol{r}\|/2)/2 \end{bmatrix}. \tag{87}$$

The functions q2v and v2q are to associate a unit quaternion to its correspondent rotation vector. With $\boldsymbol{a} \in \mathbb{H}$, $\boldsymbol{b} \in \mathbb{H}$, and $\boldsymbol{r} \in \mathbb{R}^3$, the operators $\ominus$ and $\oplus$ satisfy the following relation:

$$\boldsymbol{a} \ominus \boldsymbol{b} = \boldsymbol{r} \iff \boldsymbol{a} = \boldsymbol{b} \oplus \boldsymbol{r} \quad \text{if} \quad \|\boldsymbol{c}\| < \pi, \tag{88}$$

which can be shown through tedious but straightforward derivations.

With the operator $\ominus$, the following relation holds true:

$$\lim_{T \to 0} \frac{(\boldsymbol{a} + T\dot{\boldsymbol{a}}) \ominus \boldsymbol{a}}{T} = \mathring{\boldsymbol{a}}. \tag{89}$$

From (89), the backward Euler discretization of the relation $\boldsymbol{\omega} = \mathring{\boldsymbol{a}}$ between the attitude $\boldsymbol{a} \in \mathbb{H}$ and the angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ can be given as follows:

$$\boldsymbol{\omega} = (\boldsymbol{a} \ominus \boldsymbol{a}_{\mathrm{prv}})/T, \quad \boldsymbol{a} = \boldsymbol{a}_{\mathrm{prv}} \oplus T\boldsymbol{\omega} \tag{90}$$

where T is the timestep size and $\boldsymbol{a}_{\mathrm{prv}}$ is the value of $\boldsymbol{a}$ in the previous timestep. The expression (90) is particularly convenient due to its representational similarity to the backward-Euler discretization of vector quantities in ordinary vector spaces, such as translational positions or velocities. Moreover, these operators can be implemented by overloading the operators '$-$' and '$+$' in the C++ language. It should be noted, however, that the operator '$\oplus$' is not commutative, unlike its counterpart '$+$' in ordinary vector spaces.

In order to deal with the position and the attitude of a rigid body in brief expressions, we define the set $\mathbb{P} \triangleq \mathbb{R}^3 \times \mathbb{H}$ and redefine the operators $\ominus : \mathbb{P} \times \mathbb{P} \to \mathbb{R}^6$ and $\oplus : \mathbb{P} \times \mathbb{R}^6 \to \mathbb{P}$ as follows:

$$\begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{a}_1 \end{bmatrix} \ominus \begin{bmatrix} \boldsymbol{r}_2 \\ \boldsymbol{a}_2 \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{r}_1 - \boldsymbol{r}_2 \\ \boldsymbol{a}_1 \ominus \boldsymbol{a}_2 \end{bmatrix} \tag{91}$$

$$\begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{a}_1 \end{bmatrix} \oplus \begin{bmatrix} \boldsymbol{r}_2 \\ \boldsymbol{r}_3 \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{r}_1 + \boldsymbol{r}_2 \\ \boldsymbol{a}_1 \oplus \boldsymbol{r}_3 \end{bmatrix} \tag{92}$$

where $\boldsymbol{r}_* \in \mathbb{R}^3$ and $\boldsymbol{a}_* \in \mathbb{H}$. Note that $\mathbb{P}$ can be seen as an alternative representation of $\mathrm{SE}(3)$. The operators '$\ominus$' and '$\oplus$' defined with $\mathbb{P}$ and $\mathbb{R}^6$ allow concise representations of the Euler discretizations of translational and rotational equations of motion. In addition, we define the time derivative operator '$\circ$' for $\boldsymbol{p} = [\boldsymbol{r}^T, \boldsymbol{a}^T]^T \in \mathbb{P}$ as follows:

$$\mathring{\boldsymbol{p}} \triangleq [\dot{\boldsymbol{r}}^T, \mathring{\boldsymbol{a}}^T]^T \in \mathbb{R}^6. \tag{93}$$

It should be emphasized that $\boldsymbol{p} \in \mathbb{P}$ and $\mathring{\boldsymbol{p}} \in \mathbb{R}^6$.

Now let us consider an n-DOF rigid-link manipulator with a single end-effector. Let $\boldsymbol{q} \in \mathbb{R}^n$ be the joint variable vector of the manipulator, and $\boldsymbol{p} \in \mathbb{P}$ be the position and the attitude of the end-effector. Let $\boldsymbol{u} \in \mathbb{R}^n$ be the joint velocity vector defined as $\boldsymbol{u} = \dot{\boldsymbol{q}}$ and $\boldsymbol{v} \in \mathbb{R}^6$ be the vector of the translational velocity and the angular velocity of the end-effector, which satisfies $\boldsymbol{v} = \mathring{\boldsymbol{p}}$. With the forward kinematics $\boldsymbol{\Phi} : \mathbb{R}^n \to \mathbb{P}$ of the manipulator, we have $\boldsymbol{p} = \boldsymbol{\Phi}(\boldsymbol{q})$. We can define the Jacobian of $\boldsymbol{\Phi}$ based on the operator $\ominus$, which is defined as the function $\boldsymbol{J} : \mathbb{R}^n \to \mathbb{R}^{6\times n}$ satisfying the following:

$$\boldsymbol{J}(\boldsymbol{q})\boldsymbol{u} = \lim_{T \to 0} \frac{\boldsymbol{\Phi}(\boldsymbol{q} + T\boldsymbol{u}) \ominus \boldsymbol{\Phi}(\boldsymbol{q})}{T} \quad \forall \boldsymbol{u} \in \mathbb{R}^n. \tag{94}$$

With such a function $\boldsymbol{J}$, we can relate the velocities $\boldsymbol{v} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^6$ by $\boldsymbol{v} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{u}$. Moreover, with the backward Euler discretization $\boldsymbol{u} = (\boldsymbol{q} - \boldsymbol{q}_{\mathrm{prv}})/T$, we have

$$(\boldsymbol{\Phi}(\boldsymbol{q}) \ominus \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{prv}}))/T \approx \boldsymbol{J}(\boldsymbol{q}_{\mathrm{prv}})\boldsymbol{u} \tag{95}$$

$$\boldsymbol{\Phi}(\boldsymbol{q}) \approx \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{prv}}) \oplus T\boldsymbol{J}(\boldsymbol{q}_{\mathrm{prv}})\boldsymbol{u} \tag{96}$$

if $T > 0$ is small enough.

[3] The definitions of these notations are equivalent to those in [23], but different from those in [34]. In our definitions, the attitude $\boldsymbol{a} \oplus \boldsymbol{r}$ is the attitude $\boldsymbol{a}$ rotated by $\boldsymbol{r}$ seen from the world frame, but in [34], it is by $\boldsymbol{r}$ seen from the body frame. Similar notations are also found in [35], [36].

Appendix B
Proofs of Propositions in Section II-B

In order to provide a proof of Theorem 1, we here introduce the following lemma:

Lemma 1. *The following statement holds true:*

$$\operatorname*{arglexmin}_{\boldsymbol{\xi}\in\mathbb{R}^n}\{\|\boldsymbol{\Sigma}\boldsymbol{\xi}-\boldsymbol{\beta}\|^2, \|\boldsymbol{\xi}\|^2\} = \boldsymbol{\Sigma}^{+}\boldsymbol{\beta} \tag{97}$$

where $\boldsymbol{\Sigma}\in\mathbb{R}^{m\times n}$ is a rectangular diagonal matrix with non-negative diagonal elements and $\boldsymbol{\beta}\in\mathbb{R}^m$ is a vector.

Proof: Because of the property of $\boldsymbol{\Sigma}$, we have the following:

$$\|\boldsymbol{\Sigma}\boldsymbol{\xi}-\boldsymbol{\beta}\|^2 = \sum_{i=1}^{\min(m,n)} (\sigma_i\xi_i-\beta_i)^2 \tag{98}$$

where σ_i is the i-th diagonal element of $\boldsymbol{\Sigma}$ and ξ_i and β_i are the i-th elements of $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$, respectively. Let $\mathcal{I} \triangleq \{i \in \{1,\cdots,\min(m,n)\}\,|\,\sigma_i > 0\}$. Then, $\|\boldsymbol{\Sigma}\boldsymbol{\xi}-\boldsymbol{\beta}\|^2$ takes its minimum $\sum_{i\notin\mathcal{I}}\beta_i^2$ if $\xi_i=\beta_i/\sigma_i$ for all $i\in\mathcal{I}$. The other elements, ξ_i with $i\notin\mathcal{I}$, should be zero to minimize $\|\boldsymbol{\xi}\|$. Therefore, the left-hand side of (97) is $\boldsymbol{\xi}$ whose elements are:

$$\xi_i = \begin{cases} \beta_i/\sigma_i & \text{if } i\in\mathcal{I} \\ 0 & \text{otherwise,} \end{cases} \tag{99}$$

which is equivalent to $\boldsymbol{\Sigma}^{+}\boldsymbol{\beta}$. ■

Based on this Lemma 1, we can prove Theorem 1 as follows:

Proof of Theorem 1: It is known that $\boldsymbol{A}$ can be decomposed into the form of $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ where $\boldsymbol{U}\in\mathbb{R}^{m\times m}$ and $\boldsymbol{V}\in\mathbb{R}^{n\times n}$ are orthogonal matrices and $\boldsymbol{\Sigma}\in\mathbb{R}^{m\times n}$ is a rectangular diagonal matrix with non-negative diagonal elements. Therefore, the left-hand side (lhs) and the right-hand side (rhs) of (10) can be connected as follows:

$$\begin{aligned}\text{lhs of (10)} &= \operatorname*{arglexmin}_{\boldsymbol{x}\in\mathbb{R}^n}\{\|\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}-\boldsymbol{b}\|^2, \|\boldsymbol{x}\|^2\} \\ &= \boldsymbol{V}\operatorname*{arglexmin}_{\boldsymbol{V}^T\boldsymbol{x}\in\mathbb{R}^n}\{\|\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}-\boldsymbol{U}^T\boldsymbol{b}\|^2, \|\boldsymbol{V}^T\boldsymbol{x}\|^2\} \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^{+}(\boldsymbol{U}^T\boldsymbol{b}) = \boldsymbol{A}^{+}\boldsymbol{b} = \text{rhs of (10).}\end{aligned} \tag{100}$$

Here we used Lemma 1 by substituting $\boldsymbol{\beta}=\boldsymbol{U}^T\boldsymbol{b}$ and $\boldsymbol{\xi}=\boldsymbol{V}^T\boldsymbol{A}$, and also used the fact that $\|\boldsymbol{U}\boldsymbol{x}\|=\|\boldsymbol{x}\|$ is satisfied for an arbitrary orthogonal matrix $\boldsymbol{U}$. ■

Likewise, a proof of Corollary 1 can be given as follows:

Proof of Corollary 1: The left- and right-hand sides of (11) can be connected as follows:

$$\begin{aligned}&\text{lhs of (11)} \\ &= \operatorname*{arglexmin}_{\boldsymbol{C}^{-1}(\boldsymbol{\xi}+\boldsymbol{d})\in\mathbb{R}^n}\{\|\boldsymbol{A}\boldsymbol{C}^{-1}(\boldsymbol{\xi}+\boldsymbol{d})-\boldsymbol{b}\|^2, \|\boldsymbol{\xi}\|^2\} \\ &= \boldsymbol{C}^{-1}\operatorname*{arglexmin}_{\boldsymbol{\xi}\in\mathbb{R}^n}\{\|\boldsymbol{A}\boldsymbol{C}^{-1}\boldsymbol{\xi}-(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{C}^{-1}\boldsymbol{d})\|^2, \|\boldsymbol{\xi}\|^2\} \\ &\qquad +\boldsymbol{C}^{-1}\boldsymbol{d} \\ &= \boldsymbol{C}^{-1}(\boldsymbol{A}\boldsymbol{C}^{-1})^{+}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{C}^{-1}\boldsymbol{d})+\boldsymbol{C}^{-1}\boldsymbol{d} \\ &= \text{rhs of (11).}\end{aligned} \tag{101}$$

In the first line, we used $\boldsymbol{x}=\boldsymbol{C}^{-1}(\boldsymbol{\xi}+\boldsymbol{d})$ where $\boldsymbol{\xi}=\boldsymbol{C}\boldsymbol{x}-\boldsymbol{d}$. The equivalence between the second line and the third line is from Theorem 1. ■

References

[1] A. Peer and M. Buss, "A new admittance-type haptic interface for bimanual manipulations," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 4, pp. 416–428, 2008.

[2] R. Q. Van Der Linde and P. Lammertse, "HapticMaster – a generic force controlled robot for human interaction," *Industrial Robot*, vol. 30, no. 6, pp. 515–524, 2003.

[3] E. G. Kaigom and J. Roßmann, "Physics-based simulation for manual robot guidance – an eRobotics approach," *Robotics and Computer-Integrated Manufacturing*, vol. 43, pp. 155–163, 2017.

[4] H. Chen, W. Xu, W. Guo, and X. Sheng, "Variable admittance control using velocity-curvature patterns to enhance physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5054–5061, 2024.

[5] A. Campeau-Lecours, M. J.-D. Otis, and C. Gosselin, "Modeling of physical human-robot interaction: Admittance controllers applied to intelligent assist devices with large payload," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, pp. 1–12, 2016.

[6] I. Yoon, M. Na, and J.-B. Song, "Assembly of low-stiffness parts through admittance control with adaptive stiffness," *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102678, 2024.

[7] A. Morbi, M. Ahmadi, A. D. C. Chan, and R. Langlois, "Stability-guaranteed assist-as-needed controller for powered orthoses," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 745–752, 2014.

[8] A. Morbi and M. Ahmadi, "Safely rendering small impedance in admittance-controlled haptic devices," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1272–1280, 2016.

[9] T. Osa, S. Uchida, N. Sugita, and M. Mitsuishi, "Hybrid rate-admittance control with force reflection for safe teleoperated surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2379–2390, 2015.

[10] R. Kikuuwe, "Torque-bounded admittance control realized by a set-valued algebraic feedback," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1136–1149, 2019.

[11] V. Kurtz, P. M. Wensing, and H. Lin, "Control barrier functions for singularity avoidance in passivity-based manipulator control," in *Proceedings of IEEE Conferenceon Decision and Control*, 2021, pp. 6125–6130.

[12] F. Dimeas, V. C. Moulianitis, and N. Aspragathos, "Manipulator performance constraints in human-robot cooperation," *Robotics and Computer-Integrated Manufacturing*, vol. 50, pp. 222–233, 2018.

[13] J. Šifrer and T. Petrič, "Leveraging environmental contact and sensor feedback for precision in robotic manipulation," *Sensors*, vol. 24, p. 7006, 2024.

[14] M. G. Carmichael, D. Liu, and K. J. Waldron, "A framework for singularity-robust manipulator control during physical human-robot interaction," *International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 861–876, 2017.

[15] A. Colomé and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 944–955, 2015.

[16] H.-J. Kang and R. A. Freeman, "Null space damping method for local joint torque optimization of redundant manipulators," *Journal of Robotic Systems*, vol. 10, no. 2, pp. 249–270, 1992.

[17] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2014.

[18] A. Wahrburg, J. Boes, B. Matthias, F. Dai, and H. Ding, "Sensorless null-space admittance control for redundant manipulators," in *Proceedings of International Symposium on Robotics*, 2016, pp. 1–7.

[19] L. Zhang, H. Yu, and X. Cui, "Impedance-based null-space control of redundant torque-controlled robot," *Industrial Robots*, vol. 51, no. 5, pp. 799–808, 2024.

[20] M. Benallegue, R. Cisneros, A. Benallegue, A. Tanguy, A. Escande, M. Morisawa, and F. Kanehiro, "On compliance and safety with torque-control for robots with high reduction gears and no joint-torque feedback," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 6262–6269.

[21] J. Hermus, J. Lachner, D. Verdi, and N. Hogan, "Exploiting redundancy to facilitate physical interaction," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 599–615, 2022.

[22] F. Benzi and C. Secchi, "A null-space based approach for a safe and effective human-robot collaboration," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 3694–3700.

[23] Y. Michel, Y. Abdelhalem, and G. Cheng, “Passivity-based teleoperation with variable rotational impedance control,” *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11 658–11 665, 2024.

[24] M. P. A. Fonseca, B. V. Adorno, and P. Fraisse, “Coupled task-space admittance controller using dual quaternion logarithmic mapping,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6057–6064, 2020.

[25] V. Acary and B. Brogliato, *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics*, ser. Lecture Notes in Applied and Computational Mechanics. Springer, 2008, vol. 35.

[26] Y. Wang, R. Wang, H. Meng, and B. Zhang, “An investigation of the dynamic performance of lateral inerter-based vibration isolator with geometrical nonlinearity,” *Archive of Applied Mechanics*, vol. 89, pp. 1953–1972, 2019.

[27] J. Angeles, *Fundamentals of Robotic Mechanical Systems*. Springer, 2014.

[28] J.-Y. Rhee and B. Lee, “Analytical method for differentiation of robot Jacobian,” *Electronics Letters*, vol. 53, no. 6, pp. 386–387, 2017.

[29] D. Di Vito, C. Natale, and G. Antonelli, “A comparison of damped least squares algorithms for inverse kinematics of robot manipulators,” in *IFAC PapersOnLine*, vol. 50, 2017, pp. 6869–6874.

[30] A. Dietrich, C. Ott, and A. Albu-Schäffer, “An overview of null space projections for redundant, torque-controlled robots,” *International Journal of Robotics Research*, vol. 34, no. 11, pp. 1385–1400, 2015.

[31] O. Khatib, “Inertial properties in robotic manipulation: An object-level framework,” *International Journal of Robotics Research*, vol. 13, no. 1, pp. 19–36, 1995.

[32] R. Featherstone and O. Khatib, “Load independence of the dynamically consistent inverse of the Jacobian matrix,” *International Journal of Robotics Research*, vol. 16, no. 2, pp. 168–170, 1997.

[33] Kinova inc., “User guide: KINOVA Gen3 Ultra lightweight robot,” 2022, available online at: https://www.kinovarobotics.com/uploads/User-Guide-Gen3-R07.pdf.

[34] J. Solà, “Quaternion kinematics for the error-state kalman filter,” arXiv:1711.02508v1, 2017.

[35] N. Ramuzat, O. Stasse, and S. Boria, “Benchmarking whole-body controllers on the TALOS humanoid robot,” *Frontiers in Robotics and AI*, vol. 9, p. 826491, 2022.

[36] B. Wingo, A. S. Sathya, S. Caron, S. Hutchinson, and J. Carpentier, “Linear-time differential inverse kinematics: an augmented Lagrangian perspective,” in *Proceedings of Robotics: Science and Systems*, 2024, p. 110.