Task Space Control of Articulated Robot Near Kinematic Singularity: Forward Dynamics Approach

Donghyeon Lee, Woongyong Lee, Jonghoon Park, *Member, IEEE*, and Wan Kyun Chung, *Fellow, IEEE*

***Abstract*—In this study, a forward dynamics-based control (FDC) framework is proposed for task space control of a non-redundant robot manipulator. The FDC framework utilizes forward dynamic robot simulation and an impedance controller to solve the inverse kinematics problem. For the practical use of the proposed control framework, the accuracy, robustness, and stability of robot motion are considered. Taking advantage of the stability of the implicit Euler method, a high-gain PD controller enables accurate end-effector pose tracking in the task space without losing stability even near the kinematic singularities. Also, the robustness of the controller is enhanced by borrowing the structure of the nonlinear robust internal-loop compensator. Lastly, the selective joint damping injection and spring force saturation are applied to the impedance controller so that the robot motion can always stay within the given dynamic constraints. This study suggests a new, effective solution for the kinematic singularity problem of non-redundant robot manipulators.**

***Index Terms*—Motion control, kinematics, compliance and impedance control.**

I. Introduction

RECENT advances in robotic technology have expanded the use of the articulated robot and enlarged the market size of robot manipulators. Especially, a new concept of robots, namely, the collaborative robot, cobot, is an emerging topic in robotics research and industry [1], [2]. By enhancing the safety and simplifying the user interface, the cobot has become more accessible to users, and so, the applications of cobot are increasing in many different industries such as manufacturing, logistics, and service. For articulated robotic arms, tracking a given trajectory with the robot end-effector in the Cartesian task space is the most basic and essential function for practical application.

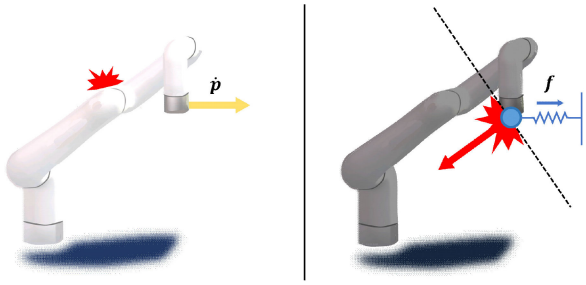


Fig. 1. Kinematic singularity of an articulated robot manipulator. Motion constraint due to robot joint kinematics (left) can be reinterpreted as an interaction with a virtual surface in the view of the forward dynamics approach (right).

However, there is a critical issue regarding the task space control of robot manipulators, the kinematic singularity problem. A kinematic singularity (or simply, the singularity) refers to a configuration of a robot where the robot cannot move towards a certain direction in task space. The immobility at the singularity is connected to the rank deficiency of the Jacobian matrix $\boldsymbol{J}$, and the rank deficiency makes the inverse kinematic solution (which utilizes $\boldsymbol{J}^{-1}$) diverge. Such divergent motion occurs near the singularities, even if the robot is not at exact singular configuration. So the singularity problem substantially limits the reachable region of the robots.

To deal with the singularity problem, various singularity avoidance algorithms have been suggested. An effective solution is to avoid singularities in advance by utilizing self-motion (null motion) [3], [4], but it cannot be applied to many commercial non-redundant (6-DOF) robot manipulators. Another well-known singularity avoidance algorithm, the damped pseudo-inverse (damped least square) method, modifies the ill-conditioned Jacobian matrix so that $\boldsymbol{J}^{-1}$ does not diverge near the singularity configuration [5], [6]. On the other hand, there are methods to reshape the given input (trajectory or force) to prevent a robot from reaching the singularity configurations [7], [8]. The damped pseudo-inverse and trajectory reshaping methods above can avoid the divergence of the robot by preventing the robot from approaching singularities, but the parameters for stable robot motion depend on the target trajectory. Thus, the parameters should be tuned manually according to the target trajectory [9]. Also, the modified path based on the kinematic criteria can affect the stability of the robot when the robot is performing interaction tasks.

Manuscript received September 10, 2019; accepted December 15, 2019. Date of publication January 9, 2020; date of current version January 22, 2020. This letter was recommended for publication by Associate Editor S. Briot and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported in part by the Bio & Medical Technology Development Program of the National Research Foundation (NRF), funded by the Ministry of Science and ICT under Grant NRF-2017M3A9F6029736 and in part by the ITECH R&D program of MOTIE/KEIT (Project 10078323. Commercialization of impedance control based low-cost cobot platform and development of application domain specific SW for easy automation application). *(Corresponding author: Wan Kyun Chung).*

D. Lee and W. K. Chung are with the Robotics Laboratory, School of Mechanical Engineering, Pohang University of Science and Technology (POSTECH) 37673, Gyeongbuk, South Korea (e-mail: sansoveria@postech.ac.kr; wkchung@postech.ac.kr).

W. Lee is with the Robotics Laboratory, School of Mechanical Engineering, Pohang University of Science and Technology (POSTECH) 37673, Gyeongbuk, South Korea, and also with the Neuromeka 06023, Seoul, South Korea (e-mail: leewoongyong@postech.ac.kr).

J. Park is with Neuromeka 06023, Seoul, South Korea (e-mail: coolcat@neuromeka.com).

Digital Object Identifier 10.1109/LRA.2020.2965071

Alternatively, the Jacobian transpose method utilizes the dynamics of a robot to solve the inverse kinematic problem. The pose of the robot end-effector can be controlled directly using impedance controllers [10], [11]. For example, the impedance controller can be a simple PD controller with gravity compensation:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau_c, \tag{1}$$

$$\tau_c = J^{\top} f_c + g(q) \text{ and } f_c = Ke + D\dot{e} \tag{2}$$

where the task space pose error e. The Jacobian transpose method utilizes $J^{\top}$ instead of J^{-1} so that the robot can stay stable even near singularities [10]. Taking advantage of the Jacobian transpose method, the forward dynamics based control (FDC) was proposed in [12]. FDC framework utilizes the forward dynamic simulation to generate the joint (reference) trajectory from the given task space trajectory. Then, the joint trajectory is followed by a robot using joint space position controller. However, the impedance controllers of the Jacobian transpose method can hardly compensate for the nonlinearity between the task space and joint space, leading to a large pose error of the resultant motion even at the non-singular configuration. Such low accuracy is critical to the practical use of the robot manipulator.

In this study, a practical task space control framework is proposed to maximize the manipulable region of the non-redundant articulated robot by enabling stable robot manipulation near the kinematic singularity. Starting from the work of [12], we improved the pose tracking accuracy of the controller and enhanced the stability near the singularities. By reinterpreting the kinematic singularity from the perspective of impedance (compliance) control, the nonlinearity between the joint and the task space is considered as a perturbation (Fig. 1). At the singularity, the system energy rapidly increases, so the impedance of the controller is reshaped to limit the system energy selectively.

In summary, the contributions of this work are:

1) Reinterpretation of kinematic singularity from the perspective of interaction task of a robot
2) Pose tracking accuracy improvement by applying a stable numerical method in the forward dynamics simulation and borrowing a robust control structure
3) Design of impedance reshaping rules for smooth motion near singularities

The rest of this paper is organized as follows. In Section II, the problems and objectives addressed by this work are defined. In Section III, the details of the proposed approach are presented component by component. In Sections IV and V, the simulation and experiment results are provided to verify the proposed controller. Lastly, the discussion is presented in Section V.

II. Problem Statement

A robot end-effector pose trajectory for an arbitrary task $p_d(t)$ is defined in the Cartesian space. The goal of the task space control of the robot is to minimize $e(t) = p_d(t) - p(t)$ using the limited bandwidth of the robot. When it comes to the task of the cobot, more restrict constraints are assigned for safety. For example, the dynamic constraint can be given as:

$$\dot{q}_{min,i} \le \dot{q}_i(t) \le \dot{q}_{max,i},$$
$$\ddot{q}_{min,i} \le \ddot{q}_i(t) \le \ddot{q}_{max,i}, \tag{3}$$

throughout the robot motion. The joint angle limits are not considered in this work because avoiding the joint angle limit while following the given task space trajectory is closer to the path planning problem, which requires a global configuration search rather than dynamic control problem.

When the inverse kinematics problem is dealt with in the forward dynamics approach, the problem becomes to design an impedance controller to let the nominal robot model generate adequate motion. The impedance controller exerts a force and torque on the robot end-effector so that the resultant robot motion satisfies the objectives given above. In the task space, the admittance of the robot end-effector can be expressed as:

$$\begin{aligned} f_c &= \Lambda\ddot{p} + \Gamma\dot{p} + \xi, \\ \Lambda &= J^{-\top} M J^{-1}, \\ \Gamma &= J^{-\top}(C - MJ^{-1}\dot{J})J^{-1}, \\ \xi &= J^{-\top} g, \end{aligned} \tag{4}$$

where Λ, Γ, ξ are the effective task space inertia, Coriolis matrix, and robot body gravity each (notation for the dependency on q, $\dot{q}$ are omitted for the convenience). Λ and Γ go to infinity as the robot reaches the singularity because $|J(q)|$ becomes zero. Such an increase in inertia and damping obstructs the robot motion as if the robot is perturbed by an external force (Fig. 1).

The impedance controller inherently generates compliant motion under perturbations, but it can store excessive energy with the large pose error and generate motion, which violates the dynamic constraints. Thus, a more conservative control scheme should be considered to generate moderate robot motion, which stays within the given dynamic constraints.

Therefore, the impedance controller for the task space control with the forward dynamics approach should be designed to have the properties below:

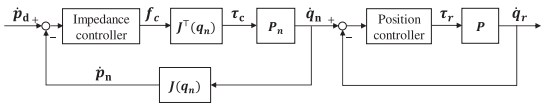
- The tracking error should be minimized when the robot travels in a non-singular configuration.
- When approaching a singularity, the joint velocity should be reduced to minimize the impact (abrupt acceleration) due to singularity.
- When leaving a singularity, the control input τ_c should be limited to prevent excessive velocity and acceleration.

III. Proposed Approach

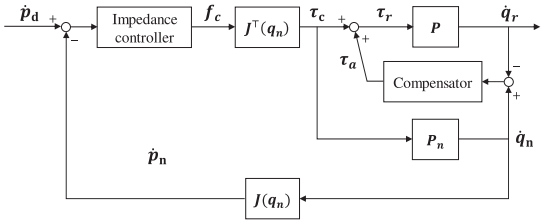
A. Forward Dynamics Approach for Task Space Control

In [12], the forward dynamics approach for task space control consists of two parts: forward dynamics simulation and joint space trajectory tracking (Fig. 2(a)). The nominal plant P_n is the virtual (ideal) robot model of the forward dynamics simulation, and it generates nominal joint trajectory q_n with the control input τ_c from the impedance controller.

The use of a virtual robot separates the inverse kinematics problem from the robust robot control problem, which should



(a) Model reference control (reference trajectory tracking)



(b) Reconstruction of controller in the view of robust control

Fig. 2. Task space control structure using the forward dynamics approach. $\boldsymbol{P}_n$ is a nominal plant (virtual model), $\boldsymbol{P}$ is a real plant and $\boldsymbol{C}$ is a inner-loop compensator.

reject the disturbances (e.g., friction) and model uncertainty (the difference between $\boldsymbol{P}_n$ and $\boldsymbol{P}$). The separation of two coexisting problems in task space control makes the controller design simpler. In the FDC framework, the role of the impedance controller is to minimize the pose error in the task space, and the position controller should reject the external disturbances and compensate for the model uncertainty.

In general, it is a widely used control scheme to calculate the model response and let the real plant to follow it. For example, the nonlinear robust inner-loop compensator (NRIC) guarantees the $\mathcal{H}^\infty$ optimality by using the simulation of a nominal plant, and it can be combined with an arbitrary controller and enhance the robustness of the system [13]. The structure of the FDC framework is similar to the structure of NRIC, so the FDC framework can be reconstructed into a double-loop control structure by borrowing the NRIC structure (Fig. 2(b)). In the NRIC framework, the behavior of a discretized nominal plant is simulated as:

$$\begin{aligned} \ddot{\boldsymbol{q}}_{n,t} &= \boldsymbol{M}_{n,t}^{-1}\{\boldsymbol{\tau}_{c,t} - \boldsymbol{C}_{n,t}\dot{\boldsymbol{q}}_{n,t} - \boldsymbol{g}_{n,t}\}, \\ \boldsymbol{q}_{n,t+1} &\leftarrow \boldsymbol{q}_{n,t} + \Delta t\dot{\boldsymbol{q}}_{n,t}, \\ \dot{\boldsymbol{q}}_{n,t+1} &\leftarrow \dot{\boldsymbol{q}}_{n,t} + \Delta t\ddot{\boldsymbol{q}}_{n,t}, \end{aligned} \tag{5}$$

where $\boldsymbol{\tau}_c$ is the output of the impedance controller (control input of the nominal plant). $\boldsymbol{q}_n$ is the joint angle of the nominal plant and $\boldsymbol{M}_{n,t} = \boldsymbol{M}_n(\boldsymbol{q}_t)$, $\boldsymbol{C}_{n,t} = \boldsymbol{C}_n(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t)$, $\boldsymbol{g}_{n,t} = \boldsymbol{g}_n(\boldsymbol{q}_t)$ are the model parameters of the nominal plant.

To make the real plant follow the nominal plant, the auxiliary input $\boldsymbol{\tau}_a$ is added to $\boldsymbol{\tau}_c$, and the control input to the real robot $\boldsymbol{\tau}_r$ is defined as:

$$\begin{aligned} \boldsymbol{\tau}_r &= \boldsymbol{\tau}_c + \boldsymbol{\tau}_a, \\ \boldsymbol{\tau}_a &= \boldsymbol{K}_a\left(\dot{\boldsymbol{e}}_{nr} + \boldsymbol{K}_p\boldsymbol{e}_{nr} + \boldsymbol{K}_i\int \boldsymbol{e}_{nr}\right), \end{aligned} \tag{6}$$

where $\boldsymbol{e}_{nr} = \boldsymbol{q}_n - \boldsymbol{q}_r$ is the joint angle error between the real plant $\boldsymbol{q}_r$ and the nominal plant $\boldsymbol{q}_n$.

The use of the nominal plant dynamics enhances the pose tracking accuracy of the real plant in the NRIC structure. With a more accurate dynamic model, the accuracy increases, but in the practical robotic applications, the dynamic model of a robot is rarely provided. If the dynamic model of the nominal plant is very far from the real plant, the feed-forward input $\boldsymbol{\tau}_c$ becomes less effective. However, the auxiliary input $\boldsymbol{\tau}_a$ (PID controller) suppresses the error, so the controller still works, provided the sufficiently large control gain $\boldsymbol{K}_a$.

Assuming the real plant follows the nominal trajectory in the joint space accurately, the remaining problem is to reduce the nominal pose error $\boldsymbol{e}_n = \boldsymbol{p}_d - \boldsymbol{p}_n$ in the task space. The following subsections are about the methodology to design and implement the impedance controller, which controls the $\boldsymbol{P}_n$. Thus, in the following subsections, the nominal plant is assumed to be identical to the real plant $\boldsymbol{P}$, and the subscripts n in the variables are omitted for simplicity.

B. Stable PD Controller for Accurate Pose Tracking

The pose error $\boldsymbol{e}$ can be reduced by increasing the control gain (stiffness) of the impedance controller in the forward dynamics simulation, but there is an issue regarding the implementation of such a high-gain controller: the numerical stability problem of a discretized system. Even if an impedance controller is designed to be passive in the continuous-time domain, in the discrete system, the discretization of the system can generate additional energy so that the controller becomes non-passive [14]. As a result, the control gain of the impedance controller is limited by the numerical stability in the forward dynamic simulation.

Thus, in this work, a variation of the PD controller, called a stable PD controller (SPD), is utilized. The concept of SPD is originally proposed as a computer simulation method [15]. The SPD method tackles the numerical instability of the explicit Euler method, which uses the current system variables to calculate the target function [16]:

$$y_t = f(x_t, t). \tag{7}$$

The conditionally stable property of the explicit Euler method bounds the control gain of the PD controller even though the PD controllers are always passive in the continuous domain. On the other hand, the implicit Euler method utilizes the future value of variables to calculate a function value:

$$y_t = f(x_{t+1}, t+1), \tag{8}$$

and it is known to be unconditionally stable, provided that the system is stable in the continuous-time domain [16]. Thus, we have adopted the implicit Euler method into the impedance controller, so that the control gain can be extremely high for better control performance.

In the discrete-time domain, the robot system (1) becomes

$$\boldsymbol{\tau}_{c,t} = \boldsymbol{M}_t\ddot{\boldsymbol{q}}_t + \boldsymbol{C}_t\dot{\boldsymbol{q}}_t + \boldsymbol{g}_t, \tag{9}$$

where $\boldsymbol{q}_t = \boldsymbol{q}(t)$, $\boldsymbol{M}_t = \boldsymbol{M}(\boldsymbol{q}_t)$, $\boldsymbol{C}_t = \boldsymbol{C}(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t)$ and $\boldsymbol{g}_t = \boldsymbol{g}(\boldsymbol{q}_t)$. Also, the output of a PD controller (2) can be written in the discrete-time domain by using the implicit Euler method as follows:

$$\begin{aligned} \boldsymbol{f}_{ie,t} &= \boldsymbol{K}\boldsymbol{e}_{t+1} + \boldsymbol{D}\dot{\boldsymbol{e}}_{t+1} \\ &= \boldsymbol{K}(\boldsymbol{p}_{des} - \boldsymbol{p}_{t+1}) + \boldsymbol{D}(\dot{\boldsymbol{p}}_{des} - \dot{\boldsymbol{p}}_{t+1}) \end{aligned}$$

$$= \boldsymbol{K}\boldsymbol{e}_t + \boldsymbol{D}\dot{\boldsymbol{e}}_t - \Delta t\boldsymbol{K}\dot{\boldsymbol{p}}_t - \Delta t\boldsymbol{D}\ddot{\boldsymbol{p}}_t)$$

$$= \boldsymbol{K}\boldsymbol{e}_t + \boldsymbol{D}\dot{\boldsymbol{e}}_t - \Delta t\boldsymbol{K}\dot{\boldsymbol{p}}_t - \Delta t\boldsymbol{D}(\boldsymbol{J}_t\ddot{\boldsymbol{q}}_t + \dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}}), \quad (10)$$

where $\boldsymbol{J}_t = \boldsymbol{J}(q(t)$, and $\boldsymbol{K}, \boldsymbol{D}$ are stiffness and damping matrices of the PD controller. With gravity compensation, the control input $\boldsymbol{\tau}_{ie,c}$ can be obtained as:

$$\boldsymbol{\tau}_{c,ie} = \boldsymbol{J}_t^\top \boldsymbol{f}_{ie,t} + \boldsymbol{g}_t. \quad (11)$$

Then, $\ddot{\boldsymbol{q}}_t$ is obtained in a closed form by substituting (11) into (9):

$$\ddot{\boldsymbol{q}}_t = \bar{\boldsymbol{M}}_t^{-1}[\boldsymbol{J}_t^\top \boldsymbol{K}(\boldsymbol{e}_t - \Delta t\dot{\boldsymbol{p}}_t) + \boldsymbol{J}_t^\top \boldsymbol{D}(\dot{\boldsymbol{e}}_t - \Delta t\dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}}_t) - \boldsymbol{C}_t\dot{\boldsymbol{q}}_t], \quad (12)$$

where $\bar{\boldsymbol{M}}_t = \boldsymbol{M}_t + \Delta t\boldsymbol{J}_t^\top \boldsymbol{D}\boldsymbol{J}_t$, and the resultant force $\boldsymbol{f}_{ie,t}$ is obtained by substituting (12) into (10).

With the implicit Euler method, the PD controller maintains its stability regardless of the scale of $\boldsymbol{K}$ with damping ($\boldsymbol{D} > \Delta t\boldsymbol{K}$).[1] The effectiveness of using SPD controller (implicit Euler method) is shown by the simulation in Section IV-B and the experiment in Section V-A.

C. Energy-Based Time-Varying Impedance Reshaping

The stable PD controller introduced in Section III-B guarantees the input-to-output stability, but it is not sufficient condition for the robot to stay within the dynamic constraints (3). Thus, the robot motion should be restricted more strictly to satisfy the safety criterion of the cobot, and it can be achieved by limiting the stored energy in the system.[2] Limiting the system energy is widely used in designing the controller for the interaction motion of a robot [18]–[20].

When the robot system (1) is controlled with the classic PD controller (2), the energy storage function S can be defined as

$$S = \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\boldsymbol{e}^\top \boldsymbol{K}\boldsymbol{e}. \quad (13)$$

Then, the rate of change of S is

$$\dot{S} = \frac{1}{2}\dot{\boldsymbol{q}}^\top \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^\top \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{e}}^\top \boldsymbol{K}\boldsymbol{e}. \quad (14)$$

By substituting (1) and (2) into (14), and utilizing the skew-symmetry of $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$, $\dot{S}$ can be simplified as:

$$\dot{S} = \frac{1}{2}\dot{\boldsymbol{q}}^\top(\dot{\boldsymbol{M}} - 2\boldsymbol{C})\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^\top \boldsymbol{J}^\top \boldsymbol{f}_c + \dot{\boldsymbol{e}}^\top \boldsymbol{K}\boldsymbol{e}$$

$$= \dot{\boldsymbol{p}}^\top \boldsymbol{f}_c + \dot{\boldsymbol{e}}^\top(\boldsymbol{f}_c - \boldsymbol{D}\dot{\boldsymbol{e}})$$

$$= \dot{\boldsymbol{p}}_d^\top \boldsymbol{K}\boldsymbol{e} + \dot{\boldsymbol{p}}^\top \boldsymbol{D}\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}^\top \boldsymbol{D}\dot{\boldsymbol{p}}. \quad (15)$$

The terms associated with $\dot{\boldsymbol{p}}_d$ are indefinite, so system energy can increase while tracking the input task velocity $\dot{\boldsymbol{p}}_d$. Especially, non-zero $\dot{\boldsymbol{p}}_d$ continuously raises $\boldsymbol{e}$ because the robot can hardly move near the singularity. Thus, the stored energy S keeps increasing with time near the singularity.

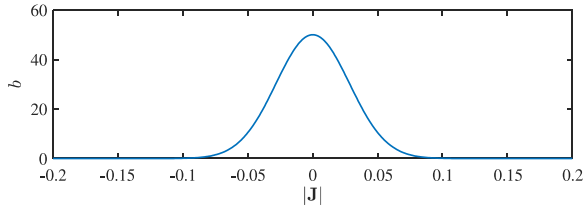


Fig. 3. Dirac delta function to determine damping constant. $b_0 = 50$ and $a = 1/25$.

Once $\boldsymbol{e}$ increased, the stored energy is converted into the kinetic energy until the energy is dissipated. While dissipating the stored energy, excessive acceleration or velocity can occur, and the robot can violate the dynamic constraints. Such a situation can be prevented by adding enough dissipative components (damping) to the controller, but the large damping degrades the overall tracking performance. Therefore, the impedance of the system needs to be reshaped properly, so that the energy storage is restricted to satisfy the given dynamic constraints while maintaining the tracking performance.

Here, two impedance modifications are proposed to prevent excessive energy elevation: the selective damping injection and the spring force saturation. First, the selective damping injection is to add joint damping torque to the control input $\boldsymbol{\tau}_c$:

$$\boldsymbol{\tau}_c = \boldsymbol{J}^\top \boldsymbol{f}_c + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \boldsymbol{B}\dot{\boldsymbol{q}}. \quad (16)$$

where $\boldsymbol{B} = b\boldsymbol{I}$ is a positive definite joint damping matrix. The added joint damping reduces the joint velocity before a robot reaches the singularity. As described in Section II, the effect of the task space inertia and Coriolis rapidly increase near singularities, so that certain joint motion can be rapidly changed. The faster the robot approaches the singularity, the larger $\ddot{\boldsymbol{q}}$ peaks it generates. Therefore, the joint velocity should be regulated depending on the distance metric between the current joint configuration and the singularity.

For the metric of the distance from the singularity, $|\boldsymbol{J}|$ can be used. $|\boldsymbol{J}|$ value becomes zero when the robot is at the singularity. Thus, it is reasonable to raise b as $|\boldsymbol{J}|$ tends to zero. However, unnecessarily large joint damping at the non-singular configuration degrades the tracking accuracy, so the Dirac delta function-shaped nonlinear function is used to determine b:

$$b = b_0 \exp\left(-\frac{|\boldsymbol{J}|^2}{a^2}\right), \quad (17)$$

where b_0 is the maximum damping constant, a is a constant which determines the width of the damping peak (Fig. 3).

Next, the spring force saturation modifies the output force of the impedance controller $\boldsymbol{f}_c$ as:

$$\boldsymbol{f}_c = \begin{cases} \boldsymbol{K}\boldsymbol{e} + \boldsymbol{D}\dot{\boldsymbol{e}}, & \|\boldsymbol{K}\boldsymbol{e}\| < f_{cut}, \\ f_{cut}\frac{\boldsymbol{e}}{\|\boldsymbol{e}\|} + \boldsymbol{D}\dot{\boldsymbol{e}}, & \|\boldsymbol{K}\boldsymbol{e}\| \geq f_{cut}. \end{cases} \quad (18)$$

The f_{cut} is a positive scalar value which limits the virtual spring force. If the pose error increases ($\|\boldsymbol{K}\boldsymbol{e}\| \geq f_{cut}$), the spring

[1]This condition does not guarantee the stability analytically for the nonlinear systems but is recommended in [15].

[2]There is another approach to satisfy the dynamic constraints, mathematical optimization. The optimization scheme is to formulate the dynamic constraints as the constraints of the optimization problem and solve it [17] However, it requires heavy computational cost, and the kinematic singularity makes the problem formulation very difficult.

force does not increase anymore, and only the direction of the force changes.

With the reshaped impedance controller, the storage function S is changed into S':

$$S' = \begin{cases} \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\boldsymbol{e}^\top \boldsymbol{K}\boldsymbol{e}, & \|\boldsymbol{K}\boldsymbol{e}\| < f_{cut}, \\ \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + f_{cut}\|\boldsymbol{e}\|, & \|\boldsymbol{K}\boldsymbol{e}\| \geq f_{cut}. \end{cases} \tag{19}$$

Then, $\dot{S}'$ becomes

$$\dot{S}' = \dot{\boldsymbol{p}}_d^\top \boldsymbol{K}\boldsymbol{e} + \dot{\boldsymbol{p}}_d^\top \boldsymbol{D}\boldsymbol{dotp} - \dot{\boldsymbol{p}}^\top \boldsymbol{D}\dot{\boldsymbol{p}} - \dot{\boldsymbol{q}}^\top \boldsymbol{B}\dot{\boldsymbol{q}} \tag{20}$$

when $\|\boldsymbol{K}\boldsymbol{e}\| < f_{cut}$ and

$$\dot{S}' = \frac{f_{cut}}{\|\boldsymbol{e}\|}\dot{\boldsymbol{p}}_d^\top \boldsymbol{e} + \dot{\boldsymbol{p}}_d^\top \boldsymbol{D}\dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}^\top \boldsymbol{D}\dot{\boldsymbol{p}} - \dot{\boldsymbol{q}}^\top \boldsymbol{B}\dot{\boldsymbol{q}} \tag{21}$$

when $\|\boldsymbol{K}\boldsymbol{e}\| \geq f_{cut}$.

The selective joint damping adds the term $-\dot{\boldsymbol{q}}^\top \boldsymbol{B}\dot{\boldsymbol{q}}$, which rapidly dissipates the stored energy when the robot reaches the singularity. Also, the spring force saturation limits the scale of the energy stored in the virtual spring and removes the effect of the integral behavior of the pose error $\boldsymbol{e} = \int(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}})$, so the resultant motion becomes relaxed even if a large error occurs due to the singularity.

IV. Simulations

In this section, the simulation is conducted to show the effectiveness of each proposed component before the empirical verification. In the simulations, the dynamic model of a commercial 6-DOF collaborative robot (Indy7, Neuromeka, Korea) is utilized, and the control frequency is 2000 Hz. Also, in the FDC framework, unstable (excessive) motions are mainly caused by the motion of the virtual robot, so only the motion of the virtual robot is investigated in the simulations without joint trajectory tracking of the real robot.

A. Stability of a PD Controller With High Control Gains

A simple pose regulation is simulated to show how the SPD controller enhances the stability of the impedance controller. The SPD controller (with the implicit Euler method) and the normal PD controller (with the explicit Euler method) method are compared, and the stability and regulation performance are investigated with various control gains k ($\boldsymbol{K} = k\boldsymbol{I}$). For both methods, the damping constant is set as $\boldsymbol{D} = 10\Delta t\boldsymbol{K}$.

In the simulation, (11) is used as the control input of the SPD controller $\boldsymbol{\tau}_{c,ie}$, and the control input of the normal PD controller $\boldsymbol{\tau}_{c,ee}$ is obtained as:

$$\boldsymbol{\tau}_{c,ee} = \boldsymbol{J}_t^\top \boldsymbol{f}_{ee,t} + \boldsymbol{g}_t \text{ where } \boldsymbol{f}_{ee,t} = \boldsymbol{K}\boldsymbol{e}_t + \boldsymbol{D}\dot{\boldsymbol{e}}_t. \tag{22}$$

In Fig. 4, the normal PD controller diverges when k exceeds a certain value (dotted green line). With the lower control gain ($k = 500$), the system does not diverge (dashed blue line), but the convergence rate is too slow. On the other hand, the SPD controller maintains its stability even when k is 10,000 (black dash-dot line). With extremely large $k = 1,000,000$ (solid red line), the system converges to the target point very fast and smoothly.

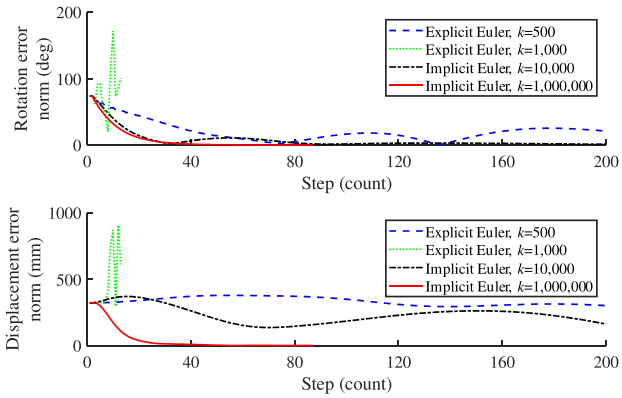


Fig. 4. Simulation results of the robot pose regulation.

B. Stability Analysis of the Nominal Plant Near the Singularity

In the second simulation, the effect of the impedance reshaping in Section III-C is verified. The impedance modifications (damping injection and spring force saturation) are applied one by one, and the pose tracking simulation is performed to evaluate the stability.

In the simulation, the control gains of the stable PD controller are

$$\boldsymbol{K} = 1,000,000 \text{ diag}\left\{1, 1, 1, \frac{1}{30}, \frac{1}{30}, \frac{1}{30}\right\},$$
$$\boldsymbol{D} = 30\Delta t\boldsymbol{K}. \tag{23}$$

The parameters for the selective damping are $b_0 = 50$ and $a = 1/25$. Also, the dynamic constraints are given as

$$\begin{aligned} \dot{\boldsymbol{q}}_{max,i} &= -\dot{\boldsymbol{q}}_{min,i} = 159 \text{ deg}/s && \text{for } i = 1,2,3,4, \\ \dot{\boldsymbol{q}}_{max,i} &= -\dot{\boldsymbol{q}}_{min,i} = 179 \text{ deg}/s && \text{for } i = 5,6, \\ \ddot{\boldsymbol{q}}_{max,i} &= -\ddot{\boldsymbol{q}}_{min,i} = 2000 \text{ deg}/s^2 && \text{for all } i. \end{aligned} \tag{24}$$

The target motion trajectory of the simulation passes through the wrist singularity in which the 5th joint is fully stretched so that the 4th and 6th joint axes become aligned (Fig. 5). Here, the simulations are conducted with the three impedance controllers: vanilla FDC (*Simple*), FDC with selective joint damping (*JD*), FDC with selective joint damping and spring force saturation (*JD+CFS*).

In Fig. 6, the robot easily exceeds the dynamic constraints (thick red lines) in the *Simple* case. When the selective joint damping is applied (*JD* case), the robot reaches the singularity more slowly ($|\boldsymbol{J}|$ decrease slowly), and the velocity and acceleration fluctuation are alleviated. However, after the robot passes by the singularity, the joint damping b decreases, and the robot is accelerated by the stored potential energy of virtual spring, and it exceeds the dynamic limits. In the case of *JD+CFS*, spring force saturation limits the energy storage, and it suppresses the velocity peak even when the large error occurs due to the immobility near the singularity. In conclusion, the selective damping injection and limiting the spring force degrades the pose tracking accuracy near the singularity, but a smooth dynamic response

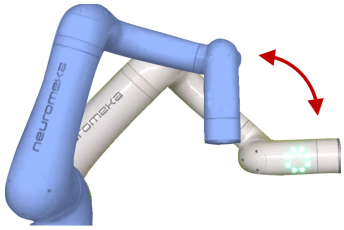


Fig. 5. Target trajectory for the experiment in Section IV-B (includes wrist singularity).

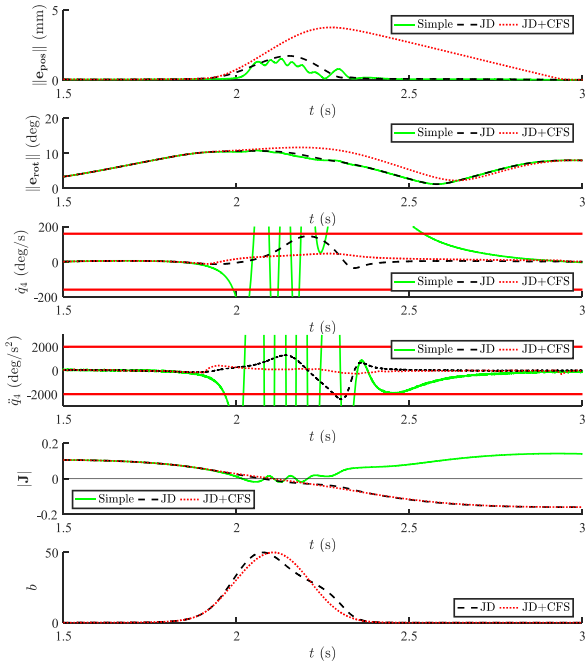


Fig. 6. End-effector pose tracking experiment result of FDC with various impedance design. The position and orientation error are plotted. The 4th joint showed the highest frequency motion, so $\dot{\boldsymbol{q}}_4, \ddot{\boldsymbol{q}}_4$ are plotted. The thick red lines are dynamic constraints of robot.

that satisfies the dynamic constraints could be accomplished with the selective joint damping and the spring force saturation.

V. Experiments

In this section, the proposed control framework is verified by real robot experiments. The proposed controller is implemented on the real Indy 7 robot of Neuromeka. Throughout the experiments, all the parameters and the control gains of the forward dynamic simulation are same as the simulation (23), (24) in Section IV-B, and the control gains of the NRIC structure are

$$\begin{aligned}\boldsymbol{K}_a &= \mathrm{diag}\{400, 400, 400, 400, 600, 600\},\\ \boldsymbol{K}_p &= \mathrm{diag}\{10, 10, 10, 10, 20, 20\},\\ \boldsymbol{K}_i &= 8\boldsymbol{I}.\end{aligned} \tag{25}$$

The experiment results show that the pose tracking accuracy of the proposed controller is comparable to that of the controllers that use the inverse Jacobian method. Also, the following experiment results show the stability and the safety of the robot motion near the singular configuration.

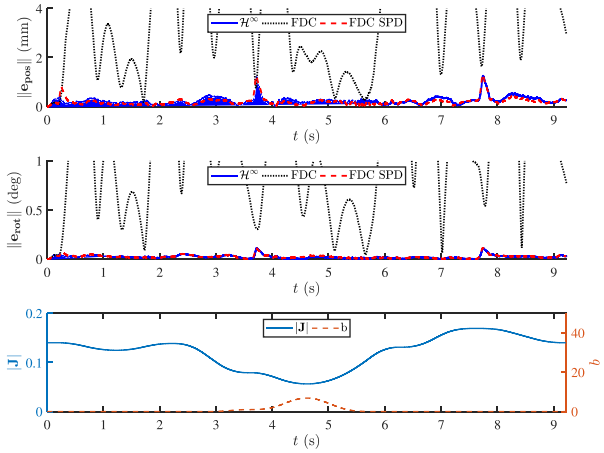


Fig. 7. End-effector pose tracking result of robot manipulator at non-singular configuration.

TABLE I
End-Effector Pose Tracking Error of the Controllers

Controller	e_{pos} (mm)		e_{rot} (deg)	
	max	**mean**	**max**	**mean**
FDC	11.62	3.19	4.12	0.94
FDC+SPD	1.31	0.21	0.11	0.02
$\mathcal{H}^\infty$	1.26	0.26	0.11	0.02

A. Pose Tracking Performance on the Non-Singular Path

First of all, the pose tracking performance of the proposed control scheme is investigated. The robot traveled following a trajectory defined in the task space, which did not include the singularity within its path. To evaluate the tracking accuracy of the proposed task space controller, the result (*FDC+SPD*) is compared with the tracking results of two existing controllers: FDC with normal PD controller (*FDC*) in [12] and $\mathcal{H}_\infty$ optimal PID controller ($\mathcal{H}_\infty$) [21].

The $\mathcal{H}_\infty$ optimal PID controller utilizes $\boldsymbol{J}^{-1}$ to map the task space trajectory into joint space and has $\mathcal{H}_\infty$ optimality with respect to a disturbance input, so it can be a reasonable reference to evaluate the pose tracking accuracy of the proposed controller. The $\mathcal{H}_\infty$ optimal PID controller input is obtained as [21], [22]:

$$\begin{aligned}\boldsymbol{\tau}_{c,\infty} = \boldsymbol{M}_n\ddot{\boldsymbol{q}}_{ref} + \boldsymbol{C}_n\dot{\boldsymbol{q}}_{ref} + \boldsymbol{g}_n\\ + \boldsymbol{J}^\top\boldsymbol{K}_a(\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i\int\boldsymbol{e}),\end{aligned} \tag{26}$$

where $\boldsymbol{M}_n, \boldsymbol{C}_n, \boldsymbol{g}_n$ are model dynamics matrices, and $\boldsymbol{q}_{ref}$ is the reference trajectory. The reference trajectory $\boldsymbol{q}_{ref}$ is defined as:

$$\begin{aligned}\ddot{\boldsymbol{p}}_{ref} &= \ddot{\boldsymbol{p}}_{des} + \boldsymbol{K}_p\dot{\boldsymbol{e}} + \boldsymbol{K}_i\boldsymbol{e}\\ \dot{\boldsymbol{p}}_{ref} &= \dot{\boldsymbol{p}}_{des} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i\int\boldsymbol{e}\\ \ddot{\boldsymbol{q}}_{ref} &= \boldsymbol{J}^{-1}\ddot{\boldsymbol{p}}_{ref} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}_{ref}\\ \dot{\boldsymbol{q}}_{ref} &= \boldsymbol{J}^{-1}\dot{\boldsymbol{p}}_{ref},\end{aligned} \tag{27}$$

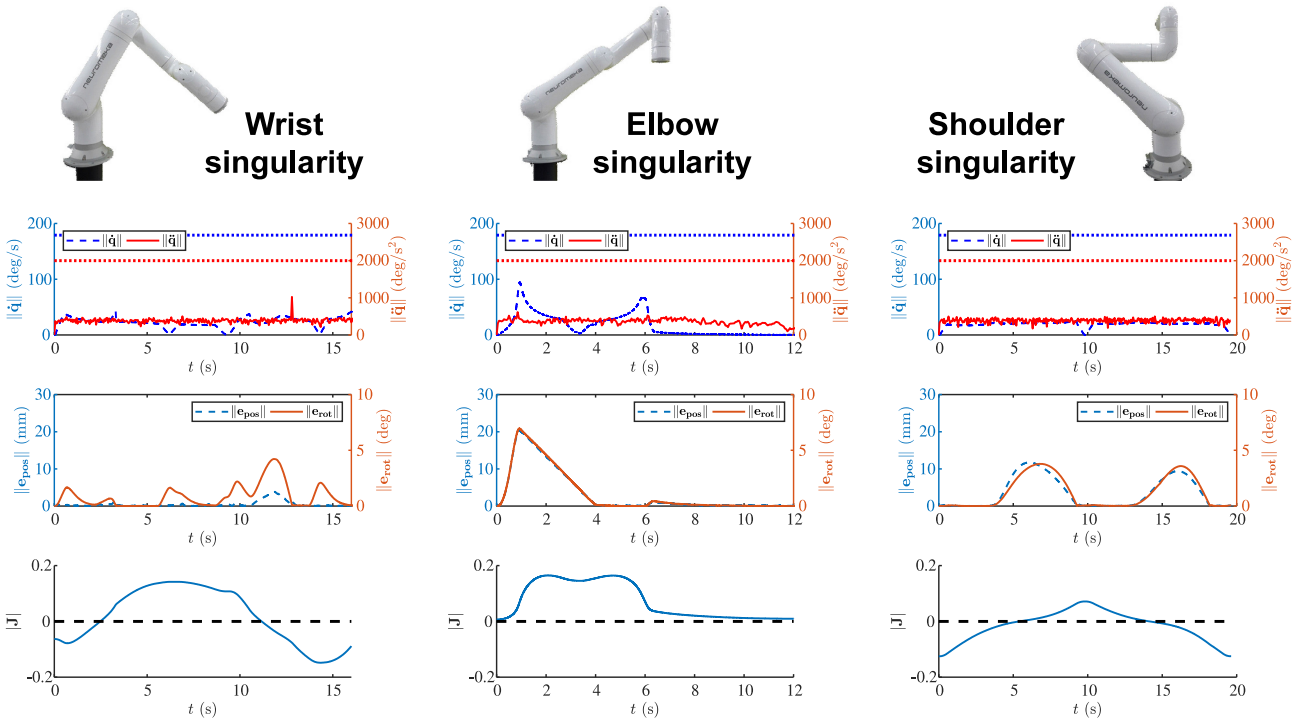


Fig. 8. Pose tracking result with various trajectories. Three trajectories includes wrist, elbow and shoulder singularity each. The blue dotted line is $\dot{q}_{max,i}$ and the red dotted line is $\ddot{q}_{max,i}$. All trajectories meet singularity twice during the motion.

where $\boldsymbol{p}_{des}$ is the given desired end-effector pose and $\boldsymbol{e} = \boldsymbol{p}_{des} - \boldsymbol{p}$.

For the result of the *FDC*, the force is calculated with the explicit Euler method (22). Due to the instability with high control gain, the stiffness of the PD controller is limited to $\boldsymbol{K} = 1,000\ \mathrm{diag}\{1, 1, 1, \frac{1}{30}, \frac{1}{30}, \frac{1}{30}\}$ even with the high damping constant $\boldsymbol{D} = 100\Delta t\boldsymbol{K}$. For the proposed controller, the control input is calculated using the SPD controller in (11), and control gains in (23) are used. The joint trajectories generated by the both forward dynamics simulation are followed by the robot using NRIC structure in (6).

The result is shown in Fig. 7 and Table I. The tracking accuracy of *FDC* was far low comparing the other controllers due to its limited control gain. On the other hand, *FDC+SPD* showed an accurate tracking result, which is comparable to or sometimes even better than the result of the $\mathcal{H}_\infty$ optimal PID controller.

B. Robot Manipulation Near Various Singularities

From the practical point of view, the proposed controller needs to maintain the stability shown in Section IV-B regardless of the geometry of the given target trajectory. Thus, the proposed controller is tested by tracking the trajectories which includes the three different singular configurations (Fig. 8).

The result shows that the proposed forward dynamics based controller generates a smooth joint space trajectory without violating dynamic constraints regardless of the given trajectory (Fig. 8). The controller induces position and orientation errors when the robot moves near the singularity ($|\boldsymbol{J}| = 0$), but it converges to the target trajectory asymptotically as it leaves the singularity.

For the more strict verification of the stability, the admittance control is applied with a 6 DoF FT sensor. The FT sensor is attached to the end-effector of the robot, and an arbitrary force is exerted by the user so that the irregular trajectory is generated. The resultant trajectory passes through the singularities, and the smooth robot motion shows the stability of the system (supplementary video).

VI. Discussion

In this work, we have proposed a controller that allows the robot to travel in the task space even in the vicinity of singularities without the need of kinematic redundancy. By reinterpreting the singularity problem as a strong perturbation in the view of the interaction control, the impedance controller is designed to generate a smooth joint space trajectory. In the result of experiments, the robot motion shows accurate pose tracking and does not generate any unsafe motion, which violates the dynamic constraints of the robot.

To deal with the singularity problem, many different control algorithms are combined in this work: the FDC framework, NRIC structure, SPD controller, spring force saturation and selective joint damping injection. Although each control algorithm may not be novel in the robotics and control research, the control schemes for different purposes are combined together and provides a new, practical solution for the singularity problem.

In the simulations and experiments, the dynamic model of the nominal plant is utilized to generate the nominal joint trajectory, but the accuracy of the nominal model is not critical to the control

performance if the control gains of the stable PD controller and $\boldsymbol{K}_a$ are large enough. Thus, the nominal model can be freely designed, and the accurate tracking can be achieved by reducing the scale of the inertia and Coriolis matrix of the nominal plant as well as increasing the control gains of the stable PD controller.

The parameters in the proposed control framework can be easily tuned because the stable PD controller guarantees the input-to-output stability of the nominal plant regardless of the control gain $\boldsymbol{K}$. For the parameters of the NRIC structure, the straightforward gain tuning rule is already suggested [13]. In the case of the impedance reshaping, tuning the parameters b_0, a, f_{cut} would be the trade-off between pose tracking accuracy and conservative dynamic constraint. Especially, the f_{cut} value of the constant force spring affects the convergence rate of the robot to return to its target trajectory after passing a singularity, so it would be a key parameter to adjust the tracking performance and the smoothness of the robot motion.

The spring force saturation and selective joint damping greatly enhance the stability of the robot motion, but it cannot guarantee that the robot motion always satisfies the dynamic constraints. The input-to-output stability of the proposed system can store larger energy as the larger input is given. Thus, the trade-off between the tracking performance and the safety of the motion is an inevitable problem, and it is required to provide analytic criteria which can be calculated from the dynamic constraints explicitly. The parameters b_0, a, f_{cut} in the impedance controller are determined empirically in this work, so designing a tuning rule for the parameters would be future research for this work.

REFERENCES

[1] T.-M. Wang, Y. Tao, and H. Liu, "Current researches and future development trend of intelligent robot: A review," *Int. J. Autom. Comput.*, vol. 15, no. 5, pp. 525–546, 2018.

[2] E. Estolatan, A. Geuna, M. Guerzoni, and M. Nuccio, "Mapping the evolution of the robotics industry: A cross country comparison," University of Turin, Turin, Italy, Working Papers, 2018.

[3] L. Huo and L. Baron, "The jointlimits and singularity avoidance in robotic welding," *Ind. Robot: The Int. J. Robot. Res. Appl.*, vol. 35, no. 5, pp. 456–464, 2008.

[4] A. Dietrich, X. Wu, K. Bussmann, C. Ott, A. Albu-Schffer, and S. Stramigioli, "Passive hierarchical impedance control via energy tanks," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 522–529, Apr. 2017.

[5] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[6] F. Caccavale, S. Chiaverini, and B. Siciliano, "Second-order kinematic control of robot manipulators with Jacobian damped least-squares inverse: Theory and experiments," *IEEE/ASME Trans. Mechatron.*, vol. 2, no. 3, pp. 188–194, 1997.

[7] G. Marani, J. Kim, J. Yuh, and W. K. Chung, "A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators," in *Proc. IEEE Int. Conf. Robot. Autom. (Cat. No.02CH37292)*, 2002, vol. 2, pp. 1973–1978.

[8] F. Dimeas, V. C. Moulianitis, C. Papakonstantinou, and N. Aspragathos, "Manipulator performance constraints in cartesian admittance control for human-robot cooperation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3049–3054.

[9] D. Di Vito, C. Natale, and G. Antonelli, "A comparison of damped least squares algorithms for inverse kinematics of robot manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6869–6874, 2017.

[10] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *J. Dyn. Syst., Meas. Control*, vol. 103, no. 2, pp. 119–125, 1981.

[11] C.-C. Cheah, "Task-space pd control of robot manipulators: Unified analysis and duality property," *Int. J. Robot. Res.*, vol. 27, no. 10, pp. 1152–1170, 2008.

[12] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (FDCC): A new approach to cartesian compliance for robotic manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4568–4575.

[13] M. J. Kim, Y. Choi, and W. K. Chung, "Bringing nonlinear $\mathcal{H}_\infty$ optimality to robot controllers," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 682–698, Jun. 2015.

[14] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 1–10, Feb. 2002.

[15] J. Tan, K. Liu, and G. Turk, "Stable proportional-derivative controllers," *IEEE Comput. Graph. Appl.*, vol. 31, no. 4, pp. 34–44, Jul.-Aug. 2011.

[16] P. Moin, *Fundamentals of Engineering Numerical Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[17] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, "Predictive inverse kinematics for redundant manipulators with task scaling and kinematic constraints," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 278–285, Feb. 2019.

[18] K. Kronander and A. Billard, "Passive interaction control with dynamical systems," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 106–113, Jan. 2016.

[19] S. S. M. Salehian and A. Billard, "A dynamical-system-based approach for controlling robotic manipulators during noncontact/contact transitions," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2738–2745, Oct. 2018.

[20] G. Raiola, C. A. Cardenas, T. S. Tadele, T. de Vries, and S. Stramigioli, "Development of a safety-and energy-aware impedance controller for collaborative robots," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1237–1244, Apr. 2018.

[21] J. Park and W. Chung, "Design of a robust $\mathcal{H}^\infty$ PID control for industrial manipulators," *J. Dyn. Syst., Meas. Control*, vol. 122, no. 4, pp. 803–812, 2000.

[22] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.