

RESEARCH ARTICLE

Open Access



# Stiff and safe task-space position and attitude controller for robotic manipulators

Gyuho Byun<sup>1\*</sup> and Ryo Kikuuwe<sup>2</sup>

## Abstract

This paper proposes a stiff and safe task-space position and attitude control scheme for robotic manipulators. This study extends the work of Kikuuwe et al.'s. (2006) velocity-bounding proxy-based sliding mode control by explicitly addressing the attitude part. The proposed controller has a Jacobian-based structure, which realizes smooth trajectories when the desired attitude is far rotated from the actual attitude. It also imposes arbitrary magnitude limits on the end-effector velocity, angular velocity, and each actuator force without sacrificing a stiffness, which is the same level as a high-gain PID position control below the limits. The benefit of the proposed controller becomes apparent after the robot yields to external forces due to force saturations, when the robot makes contact with obstacles. In such a situation, if the external forces disappear, the controller generates overdamped resuming motion from large tracking errors. The proposed controller can be expected to enhance the safety of robotic applications for the human–robot interaction. The proposed method is validated by experiments employing a six-degree of freedom industrial manipulator.

**Keywords:** Human–robot interaction, Safety regulation, Multi-dimensional task space, Quaternion representation, Attitude control, Proportional-integral-derivative (PID) position control, Proxy-based sliding mode control (PSMC)

## Introduction

When the robotic manipulators work in three dimensional task space, the robot's trajectory is usually described by task-space coordinate vectors that contain a position vector and attitude of the robot end-effector. The control of the robot end-effector is usually performed by torque commands to each actuator. In order to determine the actuator torque command, one method is to use the inverse kinematics and the joint angle controllers and another one is to use the transpose Jacobian and the forward kinematics [1]. A weak point of the control schemes using the inverse kinematics is that it can produce excessive large torques in the vicinity of singular configurations. In such situation, a control scheme based on the transpose Jacobian avoids the problem numerically [2].

Robotic manipulators are usually affected by nonlinear factors such as inertia and joint friction. To suppress such nonlinear factors, a stiff position controller is necessary for tracking accuracy. The proportional-derivative (PD) and proportional-integral-derivative (PID) position control are widely used for the robotic manipulator as a stiff position controller [3]. It can be extended to a Jacobian-based control scheme using the errors between the actual position/attitude and the desired position/attitude [4, 5], in which the proportional gain sets the robot stiffness.

While the position error between the desired position and actual position of robot end-effector is simply computed as a vector difference, the attitude error is not straightforward. Some previous researchers have studied the PD position controller combined with a Jacobian-based control scheme for tracking the desired position/attitude in three-dimensional task space. One example is Luh et al.'s [6] controller. Their control scheme can be seen as a PD-type controller combined with dynamics compensations, in which the attitude error is represented by a rotation matrix. Khatib et al.'s [7] used the attitude

\*Correspondence: byun@ctrl.mech.kyushu-u.ac.jp

<sup>1</sup> Department of Mechanical Engineering, Kyushu University, 744 Motoooka, Nishi-ku, Fukuoka 819-0395, Japan  
Full list of author information is available at the end of the article

error in a PD-type controller based on three Euler angles. The controllers employing Euler angles are not free from the representational singularity [8]. Caccavale et al. [9] proposed a PD-type controller using the attitude error in terms of an unit quaternion, which has no representational singularities. This control framework has been also applied to force control schemes [10] as a low-level controller.

One drawback of the PD-type controllers is that it can cause undesirable trajectories when a desired attitude is far rotated from an actual attitude. It may lead to a trouble with the robot configuration. Moreover, the controller with high gains produces excessively high speed from the large position and attitude errors. Such behaviors usually result in large oscillations and overshoots, which are not suitable for the robot that performs tasks with humans in the same workspace.

For improving the safety of the stiff position controller, Kikuuwe and his colleagues proposed a proxy-based sliding mode control (PSMC) [11–13], which is an extension of PID position controller and also can be seen as an approximation of a sliding mode controller (SMC). The advantage of PSMC is that it produces overdamped motion from large position error after the actuator force saturation, but its tracking accuracy is the same level as a high-gain PID control as long as the actuator force is not saturated. It has been applied to many applications, such as rehabilitation robots [14–16], passive actuators [17], pneumatic actuators [18], motion platforms [19–21] and bilateral master-slave system [22]. After initial disclosure of PSMC, they proposed a velocity bounding PSMC (VB-PSMC) for complementing the original PSMC [23]. It imposes an arbitrary magnitude limit on velocity without sacrificing tracking accuracy below the limit. An extension of VB-PSMC into the three-dimensional task space has not been considered in the previous work where an attitude control is not trivial especially.

This paper proposes a stiff and safe task-space controller for robotic manipulators, which is an extension of VB-PSMC with addressing the attitude part using the unit quaternion. The proposed controller has a Jacobian-based structure, which does not produce excessively large speed in the vicinity of singular configurations. It imposes arbitrary magnitude limits on the end-effector velocity, angular velocity and each actuator force without sacrificing its stiffness, which is the same level as a high-gain PID position control below the limits. The additional benefit of the controller appears after the robot yields to external forces due to the actuator force saturation, when the robot makes contact with obstacles. In such situation, if the external forces disappear, the controller generates overdamped resuming motion from the large tracking errors. This behavior looks similar to the compliant behavior that is realized by an

impedance control for the robotic manipulators [24–26], but its tracking accuracy is the same level as the high-gain PID position control during normal operation. The proposed controller can be expected to be useful in a robotic application, when the robot works in the workspace shared with the human.

The rest of this paper is organized as follows. “Background” section prepares mathematical preliminaries and an overview about VB-PSMC. “Robot kinematics” section discusses robot kinematics and a Jacobian-based structure. “Task-space controller” section presents a new task-space controller and its implementation by implicit Euler discretization, which avoids chattering problem that is the inherent problem of SMC. “Experiment” section presents experimental results. “Conclusion” section provides the concluding remarks.

## Background

### Mathematical preliminaries

In this paper,  $\mathbb{R}$  denotes the set of all real numbers and  $\mathbb{D}_n$  denotes the set of all  $n \times n$  diagonal matrices of which diagonal elements are all strictly positive. Symbols in boldfaces denote a vector or a matrix and the symbol  $\mathbf{o}$  and  $\mathbf{I}$  denote the zero vector and identity matrix of appropriate dimensions, respectively.

Let us define scalar functions  $\text{sgn}(\cdot)$  and  $\text{sat}(\cdot)$  as follows:

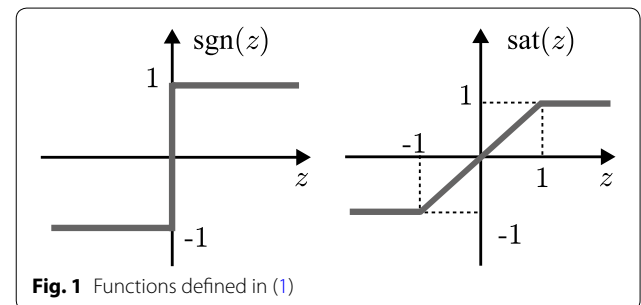
$$\text{sgn}(z) \triangleq \begin{cases} z/|z|, & \text{if } z \neq 0 \\ [-1, 1], & \text{if } z = 0 \end{cases} \quad (1a)$$

$$\text{sat}(z) \triangleq \frac{z}{\max(1, |z|)}. \quad (1b)$$

where  $z \in \mathbb{R}$ . These functions are illustrated in Fig. 1 and they are related to each other as follows:

$$y \in \text{sgn}(z - y) \iff y = \text{sat}(z) \quad \forall y, z \in \mathbb{R}, \quad (2)$$

which is proved by Kikuuwe et al.'s [12]. By using the functions  $\text{sgn}(\cdot)$  and  $\text{sgn}(\cdot)$ , we define  $\text{Sgn}(\cdot)$  and  $\text{Sat}(\cdot)$ , which are the element-wise vector versions of  $\text{sgn}(\cdot)$  and  $\text{sat}(\cdot)$  as follows:



**Fig. 1** Functions defined in (1)

$$\text{Sgn}(\mathbf{z}) \triangleq [\text{sgn}(z_1) \dots \text{sgn}(z_n)]^T \in \mathbb{R}^n \quad (3a)$$

$$\text{Sat}(\mathbf{z}) \triangleq [\text{sat}(z_1) \dots \text{sat}(z_n)]^T \in \mathbb{R}^n \quad (3b)$$

where  $\mathbf{z} = [z_1, z_2, \dots, z_n]^T \in \mathbb{R}^n$ . Because the function  $\text{sgn}(\cdot)$  is a set-valued function, the function  $\text{Sgn}(\cdot)$  is a set-valued vector function. Based on the relation (2),  $\text{Sgn}(\cdot)$  and  $\text{Sat}(\cdot)$  also have a relationship as follows:

$$\mathbf{y} \in \text{Sgn}(\mathbf{z} - \mathbf{y}) \iff \mathbf{y} = \text{Sat}(\mathbf{z}) \quad \forall \mathbf{y}, \mathbf{z} \in \mathbb{R}^n, \quad (4)$$

which can be used in the following relation:

$$\mathbf{y} \in \mathbf{Z} \text{Sgn}(\mathbf{Y}(\mathbf{z} - \mathbf{y})) \iff \mathbf{y} = \mathbf{Z} \text{Sat}(\mathbf{Z}^{-1}\mathbf{z}) \quad (5)$$

where  $\{\mathbf{Y}, \mathbf{Z}\} \in \mathbb{D}_n$  and  $\{\mathbf{y}, \mathbf{z}\} \in \mathbb{R}^n$ . Since  $\text{Sgn}(\mathbf{Y}\mathbf{z})$  is equivalent to  $\text{Sgn}(\mathbf{z})$ , this relation holds true. It will be used in the derivation in “Task-space controller” section.

#### Velocity bounding proxy-based sliding mode control

The concept of VB-PSMC was developed from PSMC. Both PSMC and VB-PSMC can be explained by an imaginary dynamical system that includes a massless virtual object (proxy) illustrated in Fig. 2. In this system, the proxy is connected to a controlled object through a PID-type virtual coupling (PID controller) and the other side is connected to the given desired position through a sliding mode-like set-valued controller. Based on the imaginary dynamical system in Fig. 2, the control algorithm of VB-PSMC can be derived from the analytical solution of a differential algebraic inclusion.

Let  $p_s \in \mathbb{R}$  and  $p_d \in \mathbb{R}$  denote the actual position and the desired position of the controlled object, respectively.  $p_x \in \mathbb{R}$  denotes the proxy's position. Then, the PID-type

virtual coupling produces the force  $f_{\text{PID}} \in \mathbb{R}$ , which can be determined as follows:

$$f_{\text{PID}} = La + K\dot{a} + B\ddot{a} \quad (6)$$

where

$$a \triangleq \int (p_x - p_s) dt \quad (7)$$

and  $L$ ,  $K$ , and  $B$  are positive real numbers, which represent the integral, proportional, and derivative gains, respectively. These parameters should be chosen so that  $p_s$  is controlled to follow  $p_x$ . On the other side of the proxy, there is a difference between PSMC and VB-PSMC. The original PSMC involves a force  $f_{\text{SMC}} \in \mathbb{R}$  produced by a sliding mode-like set-valued controller, which is a simple version of sliding mode controller (SMC) presented in Slotine et al.'s work [27]. The force  $f_{\text{SMC}} \in \mathbb{R}$  can be determined as follows:

$$\sigma_x = p_d - p_x + H(\dot{p}_d - \dot{p}_x) \quad (8a)$$

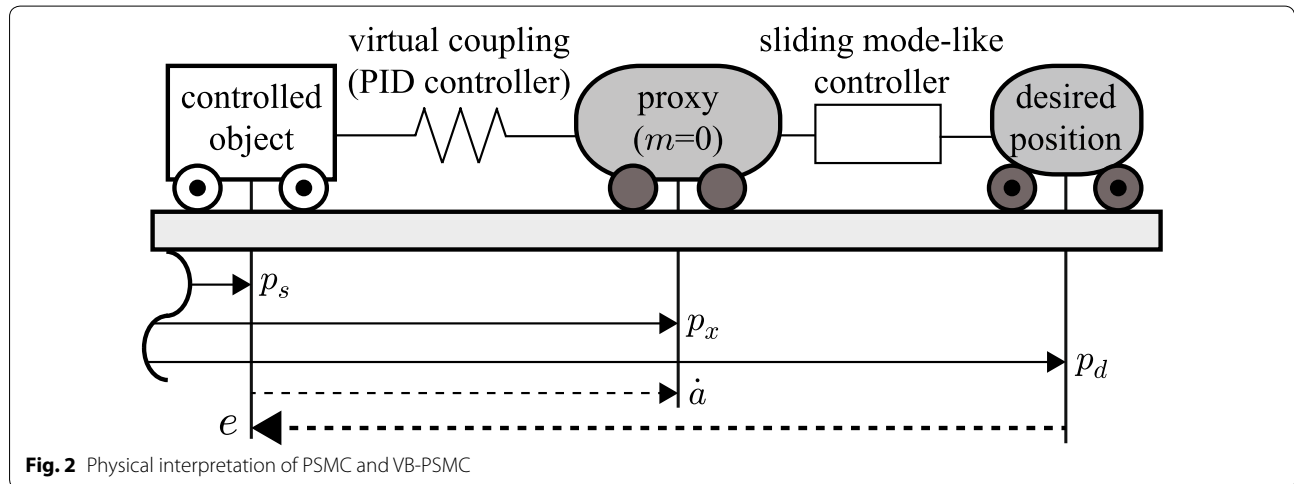
$$f_{\text{SMC}} \in F \text{sgn}(\sigma_x) \quad (8b)$$

where  $F$  and  $H$  are positive real numbers for control parameters. The set  $\sigma_x = 0$  is a sliding manifold in the proxy's state space  $\{p_x, \dot{p}_x\}$  and the control law (8) attracts the proxy's states toward the sliding manifold.

One potential risk of the control law (8) in the PSMC is that it can produce large velocity before the proxy's states reach to the sliding manifold. In order to complement the drawback, VB-PSMC involves the following controller:

$$\sigma_x = p_d - p_x + H(\dot{p}_d - \dot{p}_x) \quad (9a)$$

$$s_x = V \text{sat}((A\sigma + \dot{p}_x)/V) - \dot{p}_x \quad (9b)$$



**Fig. 2** Physical interpretation of PSMC and VB-PSMC

$$f_{\text{SMC}} \in F \text{sgn}(s_x) \quad (9c)$$

where  $V$  and  $A$  are positive real numbers. The control law (9) can be seen as a set-point controller that employs a sliding manifold as  $s_x = 0$ . It implies that  $\dot{p}_x$  is equal to or smaller than the parameter  $V$  because the right-hand side of (9b, c) is always smaller than  $V$ . Therefore, a magnitude of  $\dot{p}_x$  is bounded in the parameter  $V$  as long as  $s_x$  stays on the sliding manifold. As a result, when the set-point controller satisfies  $|\dot{p}_x| \leq V$ , it is equivalent to SMC.

The imaginary dynamical system shown in Fig. 2 satisfies  $m\ddot{p}_x = f_{\text{SMC}} - f_{\text{PID}}$  where  $m$  is the mass of the proxy. By setting  $m = 0$ , (6) and (9) can be written in the following equation:

$$\sigma_x = p_d - p_s + H(\dot{p}_d - \dot{p}_s) - \dot{a} - H\ddot{a} \quad (10a)$$

$$s_x = V \text{sat}((A\sigma_x + \dot{p}_s + \ddot{a})/V) - \dot{p}_s - \ddot{a} \quad (10b)$$

$$0 \in La + K\dot{a} + B\ddot{a} + F \text{sgn}(s_x) \quad (10c)$$

$$f = La + K\dot{a} + B\ddot{a}, \quad (10d)$$

which is a continuous-time representation of the one-dimensional VB-PSMC. As has been detailed in Kikuuwe et al.'s work [23], taking the implicit Euler discretization of the continuous-time expression (10) and the application of (5) result in the following discrete-time representation:

$$u^*(k) = \frac{p_d(k) - p_s(k) + H(\nabla p_d(k) - \nabla p_s(k))/T}{T + H} - \frac{\nabla a(k-1)/T}{T + H} + \frac{\nabla p_s(k)}{T} \quad (11a)$$

$$u(k) = V \text{sat}\left(\frac{u^*(k)}{V}\right) - \frac{\nabla p_s(k)}{T} \quad (11b)$$

$$f^*(k) = La(k-1) + (LT + K)\nabla a(k-1)/T + (LT^2 + KT + B)u(k) \quad (11c)$$

$$f(k) = F \text{sat}\left(\frac{f^*(k)}{F}\right) \quad (11d)$$

$$a(k) = \frac{(2B + KT)a(k-1) - Ba(k-2) + T^2 f(k)}{LT^2 + KT + B} \quad (11e)$$

where  $k$  is the integer representing a discrete-time index,  $T$  is the timestep size and  $\nabla$  denotes the backward-difference operator, which is defined as  $\nabla z(k) = z(k) - z(k-1)$  and  $\nabla^2 z(k) = z(k) - 2z(k-1) + z(k-2)$ .

## Robot kinematics

This section presents a task-space coordinate using an unit quaternion and a Jacobian-based structure based on a transpose Jacobian matrix. The structure will be combined with a proposed controller to produce smooth trajectories when the desired attitude is far rotated from the actual attitude.

### Task-space coordinate

The kinematics of robotic manipulators is the relation among joint variables  $\mathbf{q}_s \in \mathbb{R}^6$ , a position vector  $\mathbf{p}_s = [p_x, p_y, p_z]^T \in \mathbb{R}^3$  and attitude. While the end-effector position is easily obtained by the position vector, several representations exist to represent the attitude.

Let us consider an unit quaternion, which is composed of four elements for describing the attitude without a representational singularity resulted in an Euler angle representation. The unit quaternion representation of the attitude can be expressed as follows:

$$\boldsymbol{\alpha}_s = [\eta_s, \tilde{\boldsymbol{\alpha}}_s^T]^T \quad (12a)$$

$$= \left[ \cos\left(\frac{\theta_s}{2}\right), \sin\left(\frac{\theta_s}{2}\right)\boldsymbol{\mu}_s^T \right]^T \quad (12b)$$

where  $\eta_s \in \mathbb{R}$  denotes the scalar part,  $\tilde{\boldsymbol{\alpha}}_s = [\alpha_x, \alpha_y, \alpha_z]^T \in \mathbb{R}^3$  denotes the vector part, in which  $\theta_s$  is a rotation angle around the unit vector of axis  $\boldsymbol{\mu}_s \in \mathbb{R}^3$ . Note that the scalar part and vector part are constrained by the following equation:

$$\eta_s^2 + \tilde{\boldsymbol{\alpha}}_s^T \tilde{\boldsymbol{\alpha}}_s = 1, \quad (13)$$

which implies that both  $[\eta_s, \tilde{\boldsymbol{\alpha}}_s^T]^T$  and  $[-\eta_s, -\tilde{\boldsymbol{\alpha}}_s^T]^T$  describe the same attitude. If the rotation angle is set to the range  $\{-180^\circ \leq \theta_s \leq 180^\circ\}$ , then the scalar part is always positive and its representation of the attitude is unique. Based on the unit quaternion, the task-space coordinate can be expressed as the actual vector  $\mathbf{P}_s = [\mathbf{p}_s^T, \tilde{\boldsymbol{\alpha}}_s^T]^T \in \mathbb{R}^6$  and the desired vector  $\mathbf{P}_d = [\mathbf{p}_d^T, \tilde{\boldsymbol{\alpha}}_d^T]^T \in \mathbb{R}^6$  where  $\mathbf{p}_d \in \mathbb{R}^3$  denotes the desired position vector and  $\tilde{\boldsymbol{\alpha}}_d \in \mathbb{R}^3$  denotes the vector part of the unit quaternion representing the desired attitude  $\boldsymbol{\alpha}_d = [\eta_d, \tilde{\boldsymbol{\alpha}}_d^T]^T \in \mathbb{R}^4$ .

While the end-effector position error is simply computed as the vector difference  $\mathbf{p}_e = \mathbf{p}_s - \mathbf{p}_d$ , the attitude error is not straightforward. The unit quaternion representation of the attitude error can be expressed as follows:

$$\boldsymbol{\alpha}_e = \boldsymbol{\alpha}_s \circ \bar{\boldsymbol{\alpha}}_d \quad (14)$$

where the symbol  $\circ$  denotes the quaternion product and  $\bar{\boldsymbol{\alpha}}_d$  is the conjugate of the quaternion representing the desired attitude. It can be written as follows:



$$\alpha_e = [\eta_e, \tilde{\alpha}_e^T]^T \quad (15a)$$

$$= \left[ \cos\left(\frac{\theta_e}{2}\right), \sin\left(\frac{\theta_e}{2}\right) \mu_e^T \right]^T \quad (15b)$$

where  $\theta_e$  denotes the rotation angle between the desired attitude and actual attitude and  $\mu_e \in \mathbb{R}^3$  denotes the unit vector of axis for  $\theta_e$ . If  $\tilde{\alpha}_e$  goes to zero, the quaternion (15) goes closer to the identity quaternion  $[1, 0, 0, 0]^T$ . Thus, the position and attitude errors are represented by the following vector:

$$P_e = \Psi(P_d, q_s) \quad (16a)$$

$$= [p_e^T, \tilde{\alpha}_e^T]^T, \quad (16b)$$

which is referred to as an error vector  $P_e \in \mathbb{R}^6$  where  $\tilde{\alpha}_e \in \mathbb{R}^3$  denotes the attitude error,  $\Psi(\cdot)$  denotes the kinematics function for formulating the error vector with respect to the desired vector  $P_d$  and the joint variables  $q_s$ .

#### Jacobian-based structure

By using the kinematics function  $\Psi(\cdot)$ , the relation between the error vector  $P_e$  and joint variables can be written as follows:

$$\dot{P}_e = J(P_d, q_s) \dot{q}_s \quad (17)$$

where  $J(P_d, q_s) \in \mathbb{R}^{6 \times 6}$  denotes the Jacobian matrix defined as follows:

$$J(P_d, q_s) = \frac{\partial \Psi(P_d, q_s)}{\partial q_s}. \quad (18)$$

In the use of the relation (17), the force relationship between task-space and joint space can be written in the following equation:

$$\delta W = (\delta q_s)^T \tau_c - (\delta P_e)^T f_c \quad (19a)$$

$$= 0 \quad (19b)$$

where  $\delta W$  denotes the virtual work and  $f_c \in \mathbb{R}^6$  denotes a force vector and  $\tau_c \in \mathbb{R}^6$  denotes actuator force. The force vector and the actuator forces should be statically equated with each other.

Based on the relationship (19), a Jacobian-based structure can be written as follows:

$$\tau_c = J(P_d, q_s)^T f_c \quad (20)$$

where  $f_c$  is determined by the proposed controller using the error vector  $P_e$  as the input of the controller, which will be discussed in “Task-space controller” section specifically.

#### Task-space controller

This paper considers a torque-commanded manipulator having six-degrees-of-freedom, which can be described in the following form:

$$M(q_s) \ddot{q}_s = \tau_c + \tau_g + h + d \quad (21)$$

where  $M(q_s) \in \mathbb{R}^{6 \times 6}$  denotes a symmetric positive-definite matrix that represents the inertia,  $\tau_g \in \mathbb{R}^6$  denotes gravity compensation torques,  $\tau_c \in \mathbb{R}^6$  represents actuator force generated from the output of the controller,  $h \in \mathbb{R}^6$  denotes forces applied to the object from external forces and  $d \in \mathbb{R}^6$  denotes the sum of forces resulted from all unmodeled factors. In order to suppress the influence of  $h$  and  $d$ , the high-gain PID position/attitude controller is necessary mentioned in “Introduction” section.

In the use of the torque-commanded manipulator, one problem of the high-gain PID position/attitude controller is that it produces excessively high speed and overshoot when the desired position/attitude are far separated from the actual position/attitude. The cascade control scheme is widely used in the industrial robot [28, 29]. Simple torque and velocity limiters in the cascade control scheme does not provide direct solutions of the problem. This is because the cascade controller is joint based controller, of which the magnitude limits on the velocity/angular velocity cannot be achieved in task space directly. In addition, it does not guarantee overdamped and smooth motion below the limits.

Another imaginable solution to produce a damped response is to use a very high velocity feedback gain. It can magnify the noise in the velocity measurements and the magnified noise will deteriorate the tracking accuracy of the controller during normal operation.

#### Extension of VB-PSMC

For this reason, let us consider an extension of VB-PSMC into the three-dimensional task space. We consider the following controller as a direct extension of (10):

$$P_e = \Psi(P_d, q_s) \quad (22a)$$

$$\sigma_c = -P_e - H\dot{P}_e - \dot{a} - H\ddot{a} \quad (22b)$$

$$s_c = V \text{Sat}(V^{-1}(A\sigma_c + \dot{P}_e + \ddot{a})) - \dot{P}_e - \ddot{a} \quad (22c)$$

$$o \in La + K\dot{a} + B\ddot{a} - F \text{Sgn}(s_c) \quad (22d)$$

$$f_c = La + K\dot{a} + B\ddot{a} \quad (22e)$$

where

$$a \triangleq \int (P_x - P_e) dt, \quad (23)$$

$\mathbf{P}_x \in \mathbb{R}^6$  denotes a proxy's vector,  $\{\mathbf{K}, \mathbf{L}, \mathbf{B}\} \in \mathbb{D}_6$  denote control gain matrices,  $\{\mathbf{V}, \mathbf{F}\} \in \mathbb{D}_6$ , and  $A$  and  $H$  are positive real numbers, respectively.

This is the continuous-time representation of an extension of VB-PSMC into the three-dimensional task space, in which  $\mathbf{f}_c$  is a force vector that should be statically equivalent to actuator forces. Note that (22) is an algebraic inclusion including discontinuity, which should be solved in the implementation.

### Discrete-time implementation

For the implementation of the proposed algorithm, the continuous-time representation (22) must be approximated by a discrete-time representation with preserving its feature. Based on the implicit Euler discretization, the discrete-time representation of (22) can be rewritten as follows:

$$\mathbf{P}_e(k) = \Psi(\mathbf{P}_d, \mathbf{q}_s(k)) \quad (24a)$$

$$\sigma_c(k) = -\mathbf{P}_e(k) - \frac{H\nabla\mathbf{P}_e(k)}{T} - \frac{\nabla\mathbf{a}(k)}{T} - \frac{H\nabla^2\mathbf{a}(k)}{T^2} \quad (24b)$$

$$s_c(k) = \mathbf{V} \text{Sat} \left( \mathbf{V}^{-1} \left( \mathbf{A}\sigma_c(k) + \frac{\nabla\mathbf{P}_e(k)}{T} + \frac{\nabla^2\mathbf{a}(k)}{T^2} \right) - \frac{\nabla\mathbf{P}_e(k)}{T} - \frac{\nabla^2\mathbf{a}(k)}{T^2} \right) \quad (24c)$$

$$\mathbf{o} \in \mathbf{L}\mathbf{a}(k) + \frac{\mathbf{K}\nabla\mathbf{a}(k)}{T} + \frac{\mathbf{B}\nabla^2\mathbf{a}(k)}{T^2} - \mathbf{F} \text{Sgn}(s_c(k)) \quad (24d)$$

$$\mathbf{f}_c(k) = \mathbf{L}\mathbf{a}(k) + \frac{\mathbf{K}\nabla\mathbf{a}(k)}{T} + \frac{\mathbf{B}\nabla^2\mathbf{a}(k)}{T^2} \quad (24e)$$

Here, one can see that  $\mathbf{a}(k)$  in (24b) and (24c) are an unknown vector. By setting the parameter  $A = 1/(T + H)$ , (24c) is written as follows:

$$\mathbf{u}^*(k) = \frac{-\mathbf{P}_e(k) - H\nabla\mathbf{P}_e(k) - \nabla\mathbf{a}(k-1)/T}{T + H} \quad (25a)$$

$$\mathbf{u}(k) = \mathbf{V} \text{Sat} \left( \mathbf{V}^{-1} \left( \mathbf{u}^*(k) + \frac{\nabla\mathbf{P}_e(k)}{T} \right) - \frac{\nabla\mathbf{P}_e(k)}{T} \right) \quad (25b)$$

$$s_c(k) = \mathbf{u}(k) - \frac{\nabla^2\mathbf{a}(k)}{T^2}. \quad (25c)$$

Note that (24d) is an algebraic inclusion with respect to the unknown  $\mathbf{a}(k)$ . In order to solve the algebraic inclusion, we now express (24e) as follows:

$$\nabla\mathbf{a}(k) = (\mathbf{L}T^2 + \mathbf{K}T + \mathbf{B})^{-1}((\mathbf{B}\nabla\mathbf{a}(k-1)) - \mathbf{L}T^2\mathbf{a}(k-1) + T^2\mathbf{f}_c(k)) \quad (26a)$$

$$\mathbf{a}(k) = (\mathbf{L}T^2 + \mathbf{K}T + \mathbf{B})^{-1}((2\mathbf{B} + \mathbf{K}T)\mathbf{a}(k-1) - \mathbf{B}\mathbf{a}(k-2) + T^2\mathbf{f}_c(k)). \quad (26b)$$

Substituting (26b) into (24d) yields:

$$\mathbf{o} \in \mathbf{f}_c(k) - \mathbf{F} \text{Sgn} \left( \mathbf{u}(k) - \frac{\nabla\mathbf{a}(k) - \nabla\mathbf{a}(k-1)}{T^2} \right), \quad (27)$$

and substituting (26a) into (27) obtains as follows:

$$\mathbf{f}_c(k) \in \mathbf{F} \text{Sgn}((\mathbf{L}T^2 + \mathbf{K}T + \mathbf{B})^{-1}(\mathbf{f}_\alpha^*(k) - \mathbf{f}_c(k))) \quad (28)$$

where

$$\mathbf{f}_\alpha^*(k) = \mathbf{L}\mathbf{a}(k-1) + (\mathbf{L}T + \mathbf{K})\nabla\mathbf{a}(k-1)/T + (\mathbf{L}T^2 + \mathbf{K}T + \mathbf{B})\mathbf{u}(k). \quad (29)$$

Based on (5), (28) can be expressed as follows:

$$\mathbf{f}_c(k) = \mathbf{F} \text{Sat}(\mathbf{F}^{-1}\mathbf{f}_\alpha^*(k)), \quad (30)$$

of which the discontinuities are eliminated by using the application of (5) and  $\mathbf{f}_c(k)$  is a force vector that should be applied to the robot end-effector.

### Force limiter

For situations where actuator forces should be limited before the force vector is saturated in  $\mathbf{F}$ ,  $\mathbf{f}_\alpha^*(k)$  is used as a force vector, which can be bounded by using a transposed Jacobian matrix (18) as follows:

$$\boldsymbol{\tau}_c^*(k) = \mathbf{J}(\mathbf{P}_d, \mathbf{q}_s(k))^T \mathbf{f}_\alpha^*(k) \quad (31a)$$

$$\zeta(k) = \xi(\mathbf{C}, \boldsymbol{\tau}_c^*(k)) \quad (31b)$$

$$\boldsymbol{\tau}_c(k) = \zeta(k)\boldsymbol{\tau}_c^*(k) \quad (31c)$$

$$\mathbf{f}_\alpha(k) = \zeta(k)\mathbf{f}_\alpha^*(k), \quad (31d)$$

where  $\zeta(k) \in \mathbb{R}$  denotes a scaling factor and

$$\xi(\mathbf{C}, \boldsymbol{\tau}_c^*(k)) = \min\left(1, \frac{C_1}{|\tau_{c1}^*(k)|}, \frac{C_2}{|\tau_{c2}^*(k)|}, \dots, \frac{C_n}{|\tau_{cn}^*(k)|}\right), \quad (32)$$

in which  $\mathbf{C} \in \mathbb{D}_6$  denotes a matrix for the magnitude limits on actuator forces,  $C_n$  and  $\tau_{cn}^*(k)$  ( $n \in \{1, 2, \dots, 6\}$ ) denote the  $n$ -th diagonal elements of  $\mathbf{C}$  and  $n$ -th elements of  $\boldsymbol{\tau}_c^*(k)$ , respectively. It realizes the statical equivalence of the pairs  $\{\boldsymbol{\tau}_c^*(k), \mathbf{f}_\alpha^*(k)\}$  and  $\{\boldsymbol{\tau}_c(k), \mathbf{f}_\alpha(k)\}$ . By using the scaling factor, the force vector  $\mathbf{f}_c(k)$  can be written as follows:

$$\mathbf{f}_c(k) = \mathbf{F} \text{Sat}(\mathbf{F}^{-1} \mathbf{f}_\alpha(k)), \quad (33)$$

which is saturated when  $\boldsymbol{\tau}_c(k)$  is saturated in  $\mathbf{C}$ . It results that the magnitude of  $\mathbf{f}_\alpha^*(k)$  is decreased by reducing the magnitude of  $\mathbf{a}(k)$  according to (26b). This is especially useful to improve safety when the robot yields to external forces by the actuator force saturations. In addition,

$\mathbf{f}_c(k)$  is also limited by  $\mathbf{F}$  itself when actuator forces are not saturated in  $\mathbf{C}$  from the large tracking errors.

#### Discrete-time control algorithm

In conclusion, based on (24a), (25a), (25b), (26b), (29), (31) and (33), the numerical solution of (24) is obtained as follows:

$$\mathbf{P}_e(k) = \Psi(\mathbf{P}_d, \mathbf{q}_s(k)) \quad (34a)$$

$$\mathbf{u}^*(k) = \frac{-\mathbf{P}_e(k) - H \nabla \mathbf{P}_e(k) - \nabla \mathbf{a}(k-1)/T}{T + H} \quad (34b)$$

$$\mathbf{u}(k) = \mathbf{V} \text{Sat}\left(\mathbf{V}^{-1}\left(\mathbf{u}^*(k) + \frac{\nabla \mathbf{P}_e(k)}{T}\right)\right) - \frac{\nabla \mathbf{P}_e(k)}{T} \quad (34c)$$

$$\mathbf{f}_\alpha^*(k) = \mathbf{L} \mathbf{a}(k-1) + (\mathbf{L}T + \mathbf{K}) \nabla \mathbf{a}(k-1)/T + (\mathbf{L}T^2 + \mathbf{K}T + \mathbf{B}) \mathbf{u}(k) \quad (34d)$$

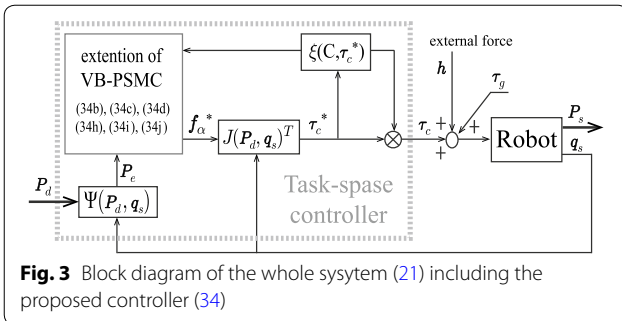
$$\boldsymbol{\tau}_c^*(k) = \mathbf{J}(\mathbf{P}_d, \mathbf{q}_s(k))^T \mathbf{f}_\alpha^*(k) \quad (34e)$$

$$\zeta(k) = \xi(\mathbf{C}, \boldsymbol{\tau}_c^*(k)) \quad (34f)$$

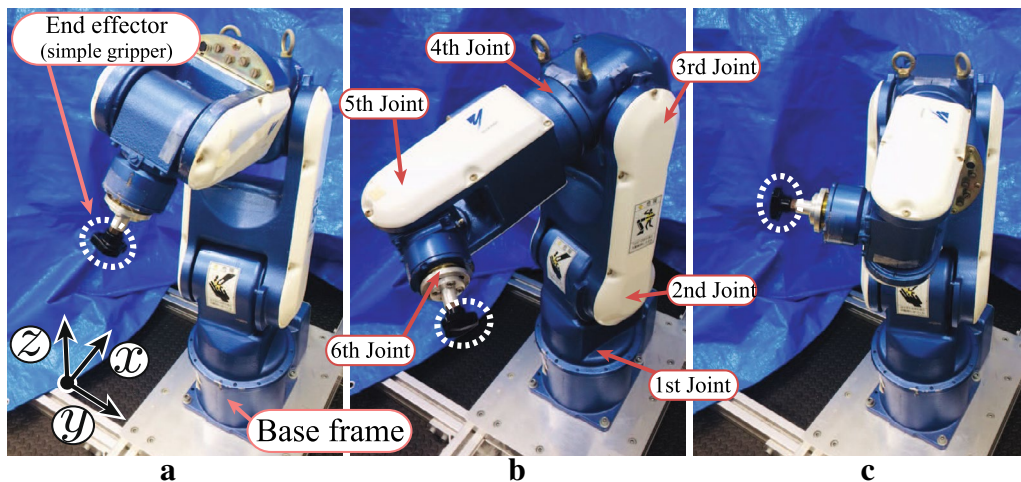
$$\boldsymbol{\tau}_c(k) = \zeta(k) \boldsymbol{\tau}_c^*(k) \quad (34g)$$

$$\mathbf{f}_\alpha(k) = \zeta(k) \mathbf{f}_\alpha^*(k) \quad (34h)$$

$$\mathbf{f}_c(k) = \mathbf{F} \text{Sat}(\mathbf{F}^{-1} \mathbf{f}_\alpha(k)) \quad (34i)$$



**Fig. 3** Block diagram of the whole system (21) including the proposed controller (34)



**Fig. 4** Six degree-of-freedom manipulator used in the experiments. Situations of Experiment I. **a** Configuration A ( $\mathbf{p}_d = [0.26, -0.01, 0.52]^T \text{ m}$ ,  $\tilde{\mathbf{a}}_d = [-0.152, 0.507, -0.07]^T$ ). **b** Configuration B ( $\mathbf{p}_d = [0.31, 0.02, 0.45]^T \text{ m}$ ,  $\tilde{\mathbf{a}}_d = [0.455, 0.377, 0.309]^T$ ). **c** Configuration C ( $\mathbf{p}_d = [0.23, -0.06, 0.54]^T \text{ m}$ ,  $\tilde{\mathbf{a}}_d = [-0.653, 0.334, -0.405]^T$ )

$$\begin{aligned} \mathbf{a}(k) = & (LT^2 + KT + \mathbf{B})^{-1}((2\mathbf{B} + KT)\mathbf{a}(k-1) \\ & - \mathbf{B}\mathbf{a}(k-2) + T^2\mathbf{f}_c(k)) \end{aligned} \quad (34j)$$

where  $\tau_c(k)$  is the output of the controller that must be sent to the robot actuators. The whole system is illustrated as the block diagram shown in Fig. 3. This is the discrete-time control law of an extension of VB-PSMC combined with the Jacobian-based structure.

#### Behavior of controller

When removing the velocity limit from the proposed controller (34) (i.e., setting  $V \rightarrow \infty I$ ), it can be expressed as follows:

$$\mathbf{P}_e(k) = \Psi(\mathbf{P}_d, \mathbf{q}_s(k)) \quad (35a)$$

$$\sigma_c(k) = -\mathbf{P}_e(k) - H \frac{\nabla \mathbf{P}_e(k)}{T} \quad (35b)$$

$$\begin{aligned} \mathbf{f}_\alpha^*(k) = & \frac{\mathbf{B} + KT + LT^2}{H + T} \sigma_c(k) \\ & + \frac{KH - \mathbf{B} + LT(2H + T)}{(H + T)T} \mathbf{a}(k-1) \\ & - \frac{KH - \mathbf{B} + LTH}{(H + T)T} \mathbf{a}(k-2) \end{aligned} \quad (35c)$$

$$\tau_c^*(k) = \mathbf{J}(\mathbf{P}_d, \mathbf{q}_s(k))^T \mathbf{f}_\alpha^*(k) \quad (35d)$$

$$\zeta(k) = \xi(\mathbf{C}, \tau_c^*(k)) \quad (35e)$$

$$\tau_c(k) = \zeta(k) \tau_c^*(k) \quad (35f)$$

$$\mathbf{f}_\alpha(k) = \zeta(k) \mathbf{f}_\alpha^*(k) \quad (35g)$$

$$\mathbf{f}_c(k) = \mathbf{F} \text{Sat}(\mathbf{F}^{-1} \mathbf{f}_\alpha(k)) \quad (35h)$$

$$\begin{aligned} \mathbf{a}(k) = & (LT^2 + KT + \mathbf{B})^{-1}((2\mathbf{B} + KT)\mathbf{a}(k-1) \\ & - \mathbf{B}\mathbf{a}(k-2) + T^2\mathbf{f}_c(k)), \end{aligned} \quad (35i)$$

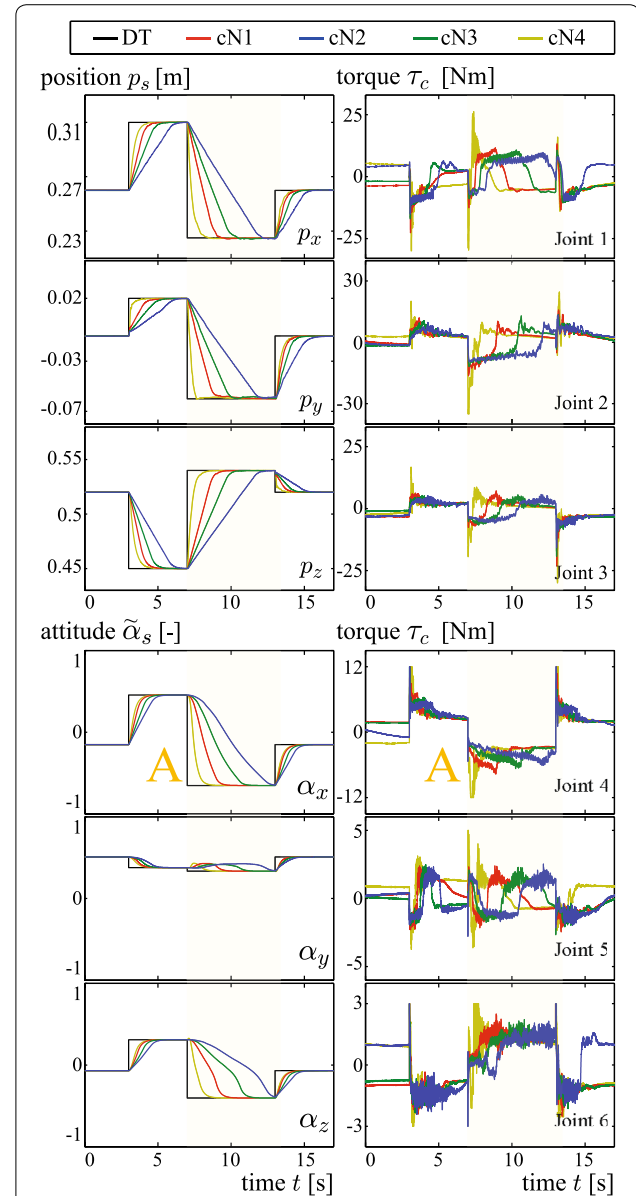
which can be seen as the controller based on PSMC. In such a framework, smooth and overdamped motions from large tracking errors are realized by setting  $H$  to be appropriately large, which is mentioned in the previous paper [12] in detail. Note that (34) is equivalent to (35) as long as all elements of  $\mathbf{u}^*(k)$  is smaller than their correspondent diagonal elements of  $\mathbf{V}$ .

PSMC is an extension of the conventional PID control. By setting  $\mathbf{F} \rightarrow \infty I$  and  $H = 0$ , the controller (35) behaves as follows:

$$\mathbf{P}_e(k) = \Psi(\mathbf{P}_d, \mathbf{q}_s(k)) \quad (36a)$$

$$\mathbf{a}(k) = \mathbf{a}(k-1) - T\mathbf{P}_e(k) \quad (36b)$$

$$\mathbf{f}_c(k) = \mathbf{L}\mathbf{a}(k) + \frac{\mathbf{K}\nabla \mathbf{a}(k)}{T} + \frac{\mathbf{B}\nabla^2 \mathbf{a}(k)}{T^2} \quad (36c)$$



**Fig. 5** Results of Experiment I. The end-effector position  $\mathbf{p}_s$  and attitude  $\tilde{\alpha}_s$ , and each joint torque during set-point controllers under **cN1**, **cN2**, **cN3** and **cN4** where **DT** is desired trajectories, which are pointed in Fig. 4

$$\tau_c^*(k) = J(P_d, q_s(k))^T f_c(k) \quad (36d)$$

$$\zeta(k) = \xi(C, \tau_c^*(k)) \quad (36e)$$

$$\tau_c(k) = \zeta(k) \tau_c^*(k), \quad (36f)$$

which can be seen as the PID-type controller with the torque limiter using the scaling factor. The above derivation shows that the proposed controller (34) is advantageous over the controller (36) in the choices of  $V$ ,  $F$  and  $H$ , which can be set according to safety regulations.

### Experiment

The proposed controller (34) was experimentally tested by using a 6-DOF industrial manipulator MOTOMAN-UPJ (Yaskawa Electric Corporation), which is shown in Fig. 4. It was controlled through a PC running the ART-Linux operating system. This manipulator has six joints, which consist of AC servo motors integrated with harmonic drive gears. In the experiment, the sampling interval was set to be  $T = 0.001$ s and six joints were used, in which each joint angles were measured by optical encoders attached to the actuators.

For the comparison, the following controllers were used:

- **cn1**: The proposed controller (34) with the following settings:

$$K = \text{diag}(K_1, K_1, K_1, K_2, K_2, K_2) \quad (37a)$$

$$L = \text{diag}(L_1, L_1, L_1, L_2, L_2, L_2) \quad (37b)$$

$$B = \text{diag}(B_1, B_1, B_1, B_2, B_2, B_2) \quad (37c)$$

$$V = \text{diag}(V_1, V_1, V_1, V_2, V_2, V_2) \quad (37d)$$

$$F = \text{diag}(F_1, F_1, F_1, F_2, F_2, F_2) \quad (37e)$$

$$C = \text{diag}(30, 35, 30, 12, 5, 3)(\text{Nm}) \quad (37f)$$

$$\{K_1, K_2\} = \{30000(\text{N/m}), 12000(\text{Nm})\} \quad (37g)$$

$$\{L_1, L_2\} = \{40000(\text{N/ms}), 4000(\text{Nm/s})\} \quad (37h)$$

$$\{B_1, B_2\} = \{200(\text{Ns/m}), 30(\text{Nms})\} \quad (37i)$$

$$\{V_1, V_2\} = \{0.08(\text{m/s}), 1(\text{s}^{-1})\} \quad (37j)$$

$$\{F_1, F_2\} = \{200(\text{N}), 30(\text{Nm})\} \quad (37k)$$

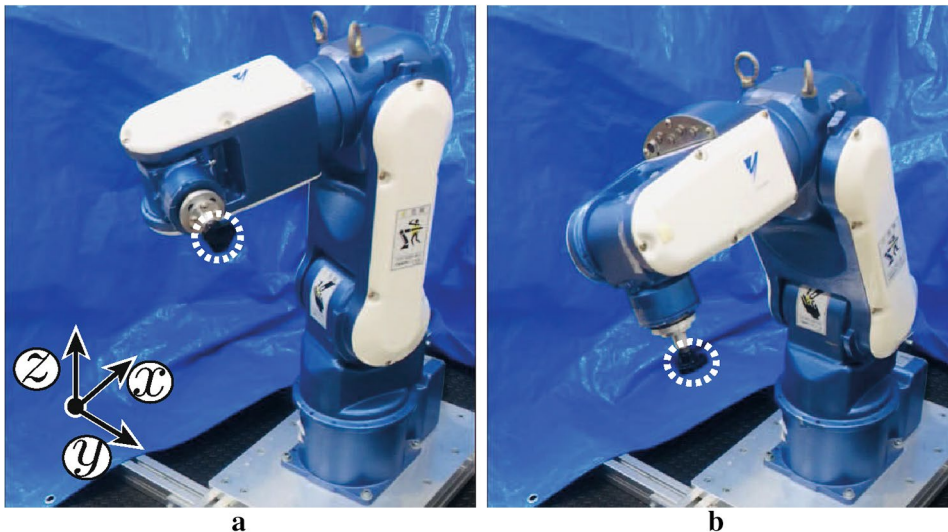
$$H = 0.3\text{s}, \quad (37l)$$

which were chosen through trial and error. Its optimal design of these parameter values are subject to further research.

- **cn2**: **cn1** with setting matrices as follows:

$$\{V_1, V_2\} = \{0.05(\text{m/s}), 0.6(\text{s}^{-1})\}. \quad (38)$$

- **cn3**: **cn1** with setting matrices as follows:



**Fig. 6** Situations of Experiment II. **a** Configuration D ( $p_d = [0.3, 0.01, 0.5]^T \text{m}$ ,  $\tilde{\alpha}_d = [0.592, 0.449, 0.296]^T$ ). **b** Configuration E ( $p_d = [0.36, 0.06, 0.4]^T \text{m}$ ,  $\tilde{\alpha}_d = [0.178, 0.711, 0.088]^T$ )



$$\{V_1, V_2\} = \{0.03(\text{m/s}), 0.4(\text{s}^{-1})\}. \quad (39)$$

- **cN4: cN1** with setting matrices as follows:

$$\{V_1, V_2\} = \{\infty(\text{m/s}), \infty(\text{s}^{-1})\}, \quad (40)$$

which can be seen as the controller based on PSMC (35).

- **cSPID**: a PID-type controller combined with the Jacobian-based structure (20) and a simple torque limiter:

$$P_e(k) = \Psi(P_d, q_s(k)) \quad (41a)$$

$$a(k) = a(k-1) - TP_e(k) \quad (41b)$$

$$f_c(k) = La(k) + \frac{K\nabla a(k)}{T} + \frac{B\nabla^2 a(k)}{T^2} \quad (41c)$$

$$\tau_c^*(k) = J(P_d, q_s(k))^T f_c(k) \quad (41d)$$

$$\tau_c(k) = C \text{Sat}(C^{-1} \tau_c^*(k)) \quad (41e)$$

where parametric matrices were set as the same as **cN4**.

#### Experiment I: large position and attitude errors

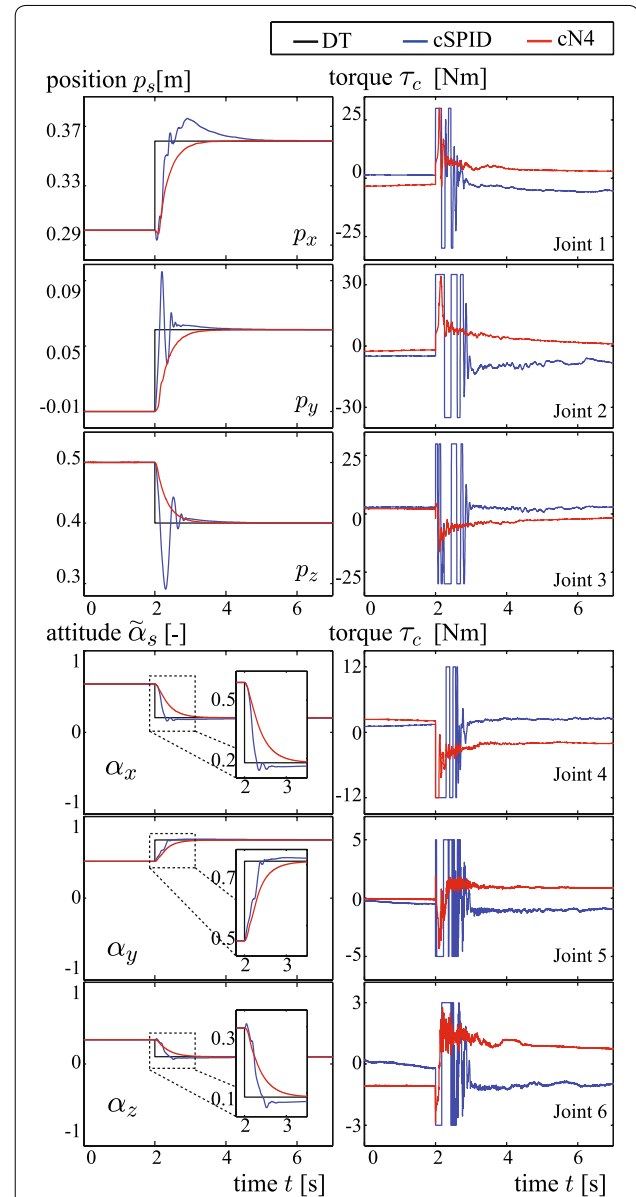
In order to show the effectiveness of the proposed controller (34), a comparison among **cN1**, **cN2**, **cN3** and **cN4** was performed to show their trajectories from large attitude error. The desired vector  $P_d$  was changed discontinuously among the three configurations shown in Fig. 4. To be more specific, the initial vector  $P_s$  was chosen as configuration A in Fig. 4. The desired vector  $P_d$  was discontinuously changed to configuration B at  $t = 3\text{s}$ , then to configuration C at  $t = 7\text{s}$ , and finally, again to configuration A at  $t = 13\text{s}$ .

The results show that **cN1**, **cN2**, **cN3** and **cN4** produce overdamped motion without overshoot from large position/attitude errors shown in Fig. 5. In Period A in Fig. 5, one can see that the controllers produce smooth trajectories from large attitude error, which is visible as errors in  $\{\alpha_x, \alpha_z\}$ . In addition, their velocity and angular velocity are bounded at the values specified by the matrix  $V$ . Such behavior is correspond to the feature of VB-PSMC mentioned in “Background” section.

#### Experiment II: comparison between cN4 and cSPID

Second set of experiments were performed to compare **cN4** with **cSPID**. The desired vector  $P_d$  was changed between the two configurations shown in Fig. 6. The results are shown in Fig. 7. At  $t = 2\text{s}$ , the robot end-effector started to accelerate due to the large tracking errors.

**cSPID** produced high speed as expected, which leads to large oscillations and overshoots. The results indicate that **cN4** and **cSPID** have distinctly different transient responses after acceleration. Specifically, the result from **cN4** shows that the actual position/attitude exponentially move toward the desired position/attitude. It leads to smooth and overdamped motion, which can be adjusted by the parameter  $H$ . This property can be considered beneficial for safety when the desired position/attitude are far separated from the actual position/attitude.



**Fig. 7** Results of Experiment II. The end-effector position  $p_s$  and attitude  $\tilde{\alpha}_s$  and each joint torque during set-point controllers under **cN4** and **cSPID** where the desired trajectories **DT** are pointed in Fig. 6

### Experiment III: human–robot interaction

In the final set of experiments, the robot was moved by the experimenter's hand through the robot's link that is connected to the 5th joint. In the experiment, a comparison between **cN1** and **cSPID** was performed to show the effectiveness of **cN1**. It was started with the initial configuration F shown in Fig. 8. The experiment results are shown in Fig. 9. In Period B, the experimenter applied a torque smaller than 5 Nm to the robot's link, which appears as a reaction torque generated by the 5th joint. In Period C, the experimenter applied a torque larger than 5 Nm to the robot's link. After Period C, the experimenter released his hand from the robot.

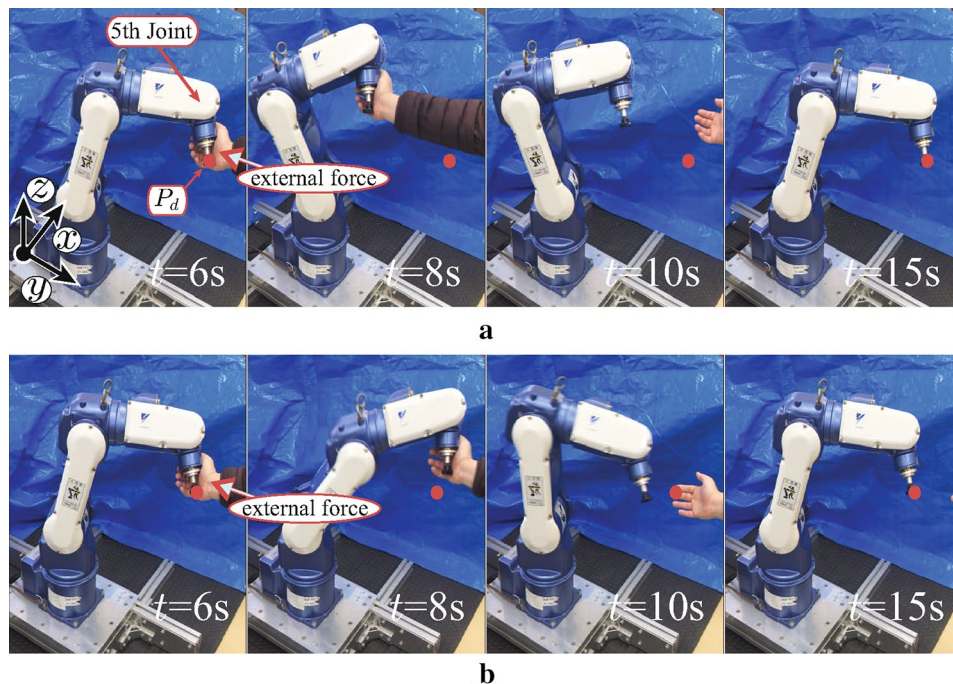
Fig. 9 shows that, in Period B, the robot did not move to any direction while the external force is applied to the robot link. It can be seen that the controller **cN1** and **cSPID** resist the external force under the actuator force limiter. In Period C, the robot yields to external force due to the actuator force saturation where **cN1** and **cSPID** produce distinctly different motion. The result produced by **cSPID** shows that the robot does not move toward the direction of the external force, while the 5th joint yields to external force shown in Figs. 8b and 9. This motion leads to the undesirable robot configuration. In addition, it produces excessively high speed after Period C due to large tracking errors. With the controller **cN1**, in contrast, the robot moves toward the direction of the

external force shown in Figs. 8a and 9. In this Period, the force vector is saturated when 5th joint torque is saturated, which is consistent with the point discussed in “Force limiter” section. It is a preferable response when the robot should be moved by external environments. After Period C, the controller **cN1** produces overdamped resuming motion from the large tracking errors. These behaviors can be seen as a main advantage of the proposed controller.

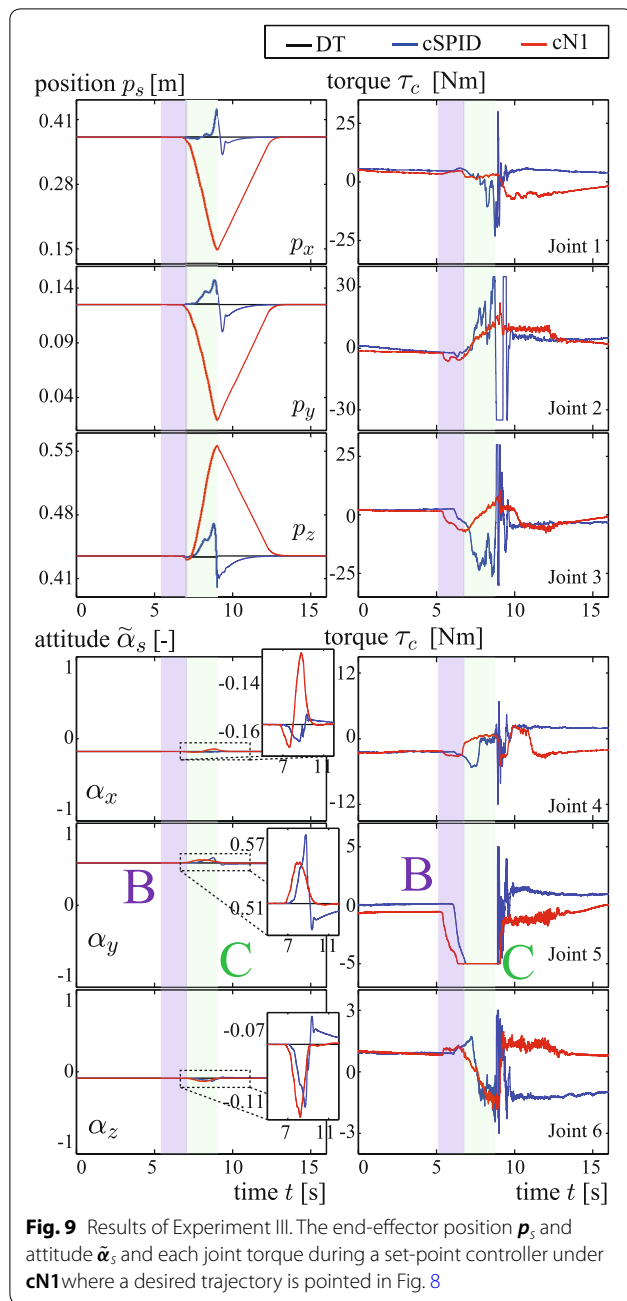
### Conclusion

This paper has proposed a new task-space controller for robotic manipulators. It can be seen as an extension of VB-PSMC to deal with position and attitude in the three-dimensional task space. The proposed controller has a Jacobian-based structure, which realizes smooth trajectories when the desired attitude is far rotated from the actual attitude. It also imposes arbitrary magnitude limits on the end-effector velocity, angular velocity and each actuator force without sacrificing the stiffness, which is the same level as a PID-type controller below the limits.

This controller was tested through experiments employing a 6-DOF industrial manipulator equipped with harmonic drive gearings. The results show that it produces high tracking accuracy and safe behavior as the advantage inherited from VB-PSMC. In addition, the force vector is saturated when actuator forces are saturated by external



**Fig. 8** Situations of Experiment III. **a** The robot is moved by the experimenter's hand under **cN1**. **b** The robot is moved by the experimenter's hand under **cSPID**. Both (a, b) are with the initial configuration F, which is  $(p_d = [0.38, 0.13, 0.44]^T \text{ m}, \hat{q}_d = [-0.157, 0.515, -0.078]^T)$



forces. It realizes that the robot is moved by external forces smoothly after actuator forces are saturated. The benefit of the proposed controller becomes apparent after the robot yields to external forces. In situations where the external forces disappear after the robot yields to external forces, the controller generates overdamped resuming motion from large tracking errors.

The proposed controller can be expected to enhance the safety of robots that work in the workspace shared with humans. Future study should address better guidelines for

the choice of the matrices  $\{V, F\}$  and theoretical details on the Jacobian-based structure (20).

#### Acknowledgements

Not applicable.

#### Authors' contributions

All authors read and approved the final manuscript.

#### Funding

Not applicable.

#### Data availability statement

The data used to support the findings of this study are available from the corresponding author upon request

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup> Department of Mechanical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan. <sup>2</sup> Department of Mechanical Systems Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan.

Received: 28 November 2019 Accepted: 13 March 2020

Published online: 30 March 2020

#### References

1. Siciliano B, Sciavicco L, Villani L, Oriolo G (2010) Robotics: modelling, planning and control. Springer, Berlin
2. Buss SR (2004) Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE J Robot Autom* 17(1–19):16
3. Arimoto S, Miyazaki F (1984) Stability and robustness of pid feedback control for robot manipulators of sensory capability. In: Robotics research, first international symposium. MIT Press, pp. 783–799
4. Chiaverini S, Siciliano B, Villani L (1999) A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Trans Mechatron* 4(3):273–285
5. Caccavale F, Natale C, Siciliano B, Villani L (2005) Integration for the next generation: embedding force control into industrial robots. *IEEE Robot Autom Mag* 12(3):53–64
6. Luh J, Walker M, Paul R (1980) Resolved-acceleration control of mechanical manipulators. *IEEE Trans Autom Control* 25(3):468–474
7. Khatib O (1987) A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE J Robot Autom* 3(1):43–53
8. Diebel J (2006) Representing attitude: euler angles, unit quaternions, and rotation vectors. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.5134>
9. Caccavale F, Chiacchio P, Chiaverini S (2000) Task-space regulation of cooperative manipulators. *Automatica* 36(6):879–887
10. Caccavale F, Chiacchio P, Marino A, Villani L (2008) Six-dof impedance control of dual-arm cooperative manipulators. *IEEE/ASME Trans Mechatron* 13(5):576–586
11. Kikuuwe R, Fujimoto H (2006) Proxy-based sliding mode control for accurate and safe position control. In: Proceedings of the 2006 IEEE international conference on robotics and automation, pp. 25–30
12. Kikuuwe R, Yasukouchi S, Fujimoto H, Yamamoto M (2010) Proxy-based sliding mode control: a safer extension of PID position control. *IEEE Trans Robot* 26(4):670–683
13. Kikuuwe R (2018) Some stability proofs on proxy-based sliding mode control. *IMA J Math Control Inform* 35(4):1319–1341
14. Watanabe T, Yano K, Aoki T, Nishimoto Y (2011) Extension motion assistance for upper limb using proxy-based sliding mode control. In: IEEE international conference on systems, man, and cybernetics (SMC), pp. 2885–2890

15. Liao Y, Zhou Z, Wang Q (2015) BioKEX: A bionic knee exoskeleton with proxy-based sliding mode control. In: Proceedings of 2015 IEEE international conference on industrial technology, pp. 125–130
16. Chen G, Zhou Z, Vanderborght B, Wang N, Wang Q (2016) Proxy-based sliding mode control of a robotic ankle-foot system for post-stroke rehabilitation. *Adv Robot* 30(15):992–1003
17. Kashiri N, Lee J, Tsagarakis NG, Van Damme M, Vanderborght B, Caldwell DG (2016) Proxy-based position control of manipulators with passive compliant actuators: stability analysis and experiments. *Robot Auton Syst* 75:398–408
18. Van Damme M, Beyl P, Vanderborght B, Versluys R, Van Ham R, Vanderniepen I, Daerden F, Lefeber D (2010) The safety of a robot actuated by pneumatic muscles—a case study. *Int J Soc Robot* 2(3):289–303
19. Hastürk Ö, Erkmen AM, Erkmen İ (2011) Proxy-based sliding mode stabilization of a two-axis gimbaled platform. In: Proceedings of the world congress on engineering and computer science, pp. 370–376
20. Prieto PJ, Rubio E, Hernández L, Urquijo O (2013) Proxy-based sliding mode control on platform of 3 degree of freedom (3-DOF). *Adv Robot* 27(10):773–784
21. Gu G-Y, Zhu L-M, Su C-Y, Ding H, Fatikow S (2015) Proxy-based sliding-mode tracking control of piezoelectric-actuated nanopositioning stages. *IEEE/ASME Trans Mechatron* 20(4):1956–1965
22. Nishi F, Katsura S (2015) Ultrafine manipulation considering input saturation using proxy-based sliding mode control. In: Proceedings of 2015 IEEE international conference on mechatronics, pp. 547–552
23. Kikuuwe R, Yamamoto T, Fujimoto H (2006) Velocity-bounding stiff position controller. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems, pp. 3050–3055
24. Hogan N (1985) Impedance control: an approach to manipulation: part i-theory. *Trans ASME J Dyn Syst Meas Control* 107(1):1–7
25. Hogan N (1985) Impedance control: an approach to manipulation: part ii-implementation. *Trans ASME J Dyn Syst Meas Control* 107(1):8–16
26. Hogan N (1985) Impedance control: an approach to manipulation: part iii-applications. *Trans ASME J Dyn Syst Meas Control* 107(1):17–24
27. Slotine J-JE, Sastry SS (1983) Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators. *Int J Control* 38(2):465–492
28. Lange F, Hirzinger G (1999) Adaptive minimization of the maximal path deviations of industrial robots. In: 1999 European control conference (ECC). IEEE, pp. 1914–1919
29. Xie Y, Jin J, Tang X, Ye B, Tao J (2018) Robust cascade path-tracking control of networked industrial robot using constrained iterative feedback tuning. *IEEE Access* 7:8470–8482

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)