Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

# Velocity-Bounding Stiff Position Controller

Ryo Kikuuwe, Takahiro Yamamoto, and Hideo Fujimoto

*Nagoya Institute of Technology, Nagoya, Aichi 466-8555, Japan*
*kikuuwe@ieee.org*

***Abstract*— High-gain PID position control, which is widely used with robots having complex dynamics, involves some risks in cases of abnormal events, such as unexpected environment contacts, temporal power failures, and wrong position commands from a higher-level controller. We previously proposed a proxy-based sliding mode control (PSMC) scheme, which is an extension of PID control that ensures slow, moderate, overdamped recovery from a large positional error at abnormal events without sacrificing tracking accuracy during normal operation. Its weakness is that the velocity is not bounded. This paper proposes a velocity-bounding PSMC (VB-PSMC) scheme, which is an extension of PSMC to impose an arbitrary magnitude limit on the velocity. The VB-PSMC method can be used as a lowest-level position servo that is safer than PSMC and much safer than conventional PID control. The advantage of VB-PSMC was demonstrated through implementation experiments.**

## I. Introduction

The dynamics of industrial robots are usually difficult to be modeled due to the existence of many nonlinear elements, such as joint frictions. To suppress the influence of unmodeled dynamics, an industrial robot usually requires a stiff position controller as the lowest level controller. Low-level position controllers are necessary even for implicit force control [1] and admittance control [2]. The PD (proportional-derivative) and PID (proportional-integral-derivative) control schemes are among the most widely used methods for position control.

One drawback of conventional stiff position control is difficulty in ensuring safety. It is not safe when the desired position determined by a higher-level controller is far separated from the actual end-effector position. This unsafe situation occurs in cases of unexpected environment contacts, temporal power failures to the actuators, wrong position commands from the higher-level controller, and so on. In such cases, the actuator force, determined by the position controller, increases as the error between the desired and actual positions increases. This can cause damage to the environment and the robot, and can cause excessive speed and overshoots during recovery to the desired position. A high velocity-feedback gain (D-gain) will prevent excessive speed and overshoots, but it will deteriorate the tracking accuracy during normal operations and will magnify the noise in the velocity signal.

In a previous paper [3], we proposed an extension of PID control, which we call proxy-based sliding mode control (PSMC). This method can also be interpreted as a modified version of sliding mode control (SMC). Tracking accuracy of PSMC during normal operation is the same as the conventional PID control. However, it is capable of exhibiting slow, overdamped motions during recovering from large positional errors. This property prevents overshoots after recovery from abnormal events. The time constant of this recovering motion can be set as an arbitrary user parameter.

In the PSMC scheme, the system state (that consists of position and velocity) is attracted to a linear sliding manifold in the state space, which is illustrated in Fig. 1(a). This linear manifold corresponds to a first order differential equation with respect to position, which represents overdamped convergent dynamics toward the desired position. One imaginable risk in using PSMC is that there is no limit on the velocity during recovering motions. When the distance between the actual and desired positions are large, the robot produces a constant force until the system state reaches the sliding manifold. Therefore, when the error is too large, the velocity can become excessively large before it reaches the sliding manifold.

This paper proposes a modification of PSMC to enable users to set an arbitrary magnitude limit on the velocity. We term the proposed method as the velocity-bounding proxy-based sliding mode control (VB-PSMC). Basic idea is simple; we use a sliding manifold illustrated in Fig. 1(b), on which the velocity is saturated. We present a multidimensional expression of this sliding manifold and derive a discrete-time control law based on this sliding manifold. The new method allows users to impose an arbitrary magnitude limit on velocity without sacrificing tracking accuracy below this limit. As an advantage inherited from the previous PSMC, the VB-PSMC scheme also has the capability of overdamped recovering motion from large positional errors. Thus, VB-PSMC can be used as a low-level position controller safer than PSMC and much safer than conventional PID control. The literature includes some control schemes for imposing state (position and/or velocity) constraints [4], [5] based on passivity-based energy shaping scheme [6], but control accuracy during normal operation

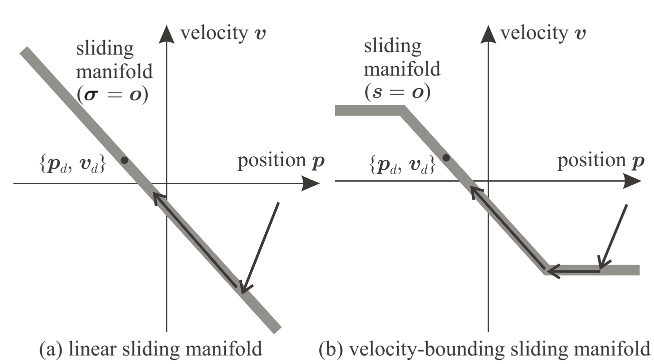



Fig. 1. Sliding manifolds.

has not been investigated. Rate constraints using time-scale transformations, such as used in [7], are not suitable for low-level position controllers that accept unpredictable position commands from upper level controllers.

The rest of this paper is organized as follows. Section II presents a new VB-PSMC scheme that incorporates the sliding manifold illustrated in Fig. 1(b). Section III presents the results of implementation experiments. Section IV provides the concluding remarks.

## II. Velocity-bounding Proxy-based Sliding Mode Control

This section provides the new control law, VB-PSMC, which is an extension of the previously-proposed PSMC method [3]. In the following discussions, lower- and upper-case symbols in boldface denote $n$-dimensional vectors and matrices, respectively. Plain symbols denote scalars. The symbol $\boldsymbol{o}$ denotes the $n$-dimensional zero vector.

### A. Mathematical Preparation

We prepare two $n$-dimensional functions: $\mathbf{sgn}(\cdot)$ and $\mathbf{sat}(\cdot)$. The function $\mathbf{sgn}(\cdot)$ denotes the normalization function, which is the map from $\mathcal{R}^n$ into $\mathcal{R}^n$ that is defined by

$$\mathbf{sgn}(\boldsymbol{x}) \begin{cases} = \boldsymbol{x}/\|\boldsymbol{x}\| & \text{if } \boldsymbol{x} \neq \boldsymbol{o} \\ \in \{\boldsymbol{e} \in \mathcal{R}^n \mid \|\boldsymbol{e}\| \leq 1\} & \text{if } \boldsymbol{x} = \boldsymbol{o}, \end{cases} \tag{1}$$

where $\boldsymbol{x}$ is an arbitrary $n$-dimensional vector. This function can be considered as the $n$-dimensional version of signum function. Under the definition (1), the statement $\boldsymbol{y} = \mathbf{sgn}(\boldsymbol{x})$ is equivalent to

$$(\boldsymbol{y} = \boldsymbol{x}/\|\boldsymbol{x}\| \wedge \boldsymbol{x} \neq \boldsymbol{o}) \vee (\|\boldsymbol{y}\| \leq 1 \wedge \boldsymbol{x} = \boldsymbol{o}). \tag{2}$$

The function $\mathbf{sat}(\cdot)$, on the other hand, denotes the unit saturation function, which is the map from $\mathcal{R}^n$ into $\mathcal{R}^n$ that is defined by

$$\mathbf{sat}(\boldsymbol{x}) = \begin{cases} \boldsymbol{x}/\|\boldsymbol{x}\| & \text{if } \|\boldsymbol{x}\| > 1 \\ \boldsymbol{x} & \text{if } \|\boldsymbol{x}\| \leq 1. \end{cases} \tag{3}$$

The functions $\mathbf{sgn}(\cdot)$ and $\mathbf{sat}(\cdot)$ are related by the following theorem and corollary:

**Theorem 1.** *With two $n$-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, the following statement holds true:*

$$\boldsymbol{y} = \mathbf{sgn}(\boldsymbol{x} - \boldsymbol{y}) \iff \boldsymbol{y} = \mathbf{sat}(\boldsymbol{x}). \tag{4}$$

*Proof.* From the definition of $\mathbf{sgn}(\cdot)$ function, we have

$$\begin{aligned} &\boldsymbol{y} = \mathbf{sgn}(\boldsymbol{x} - \boldsymbol{y}) \\ \iff &\left(\boldsymbol{y} = \frac{\boldsymbol{x} - \boldsymbol{y}}{\|\boldsymbol{x} - \boldsymbol{y}\|} \wedge \boldsymbol{x} \neq \boldsymbol{y}\right) \vee (\boldsymbol{y} = \boldsymbol{x} \wedge \|\boldsymbol{y}\| \leq 1). \end{aligned} \tag{5}$$

Assume that $\boldsymbol{y} = (\boldsymbol{x} - \boldsymbol{y})/\|\boldsymbol{x} - \boldsymbol{y}\|$ and $\boldsymbol{x} \neq \boldsymbol{y}$ are satisfied. Letting $r = \|\boldsymbol{x} - \boldsymbol{y}\| > 0$, we have $(1+r)\boldsymbol{y} = \boldsymbol{x}$. This implies $\|\boldsymbol{x}\| = 1+r$ because of $\|\boldsymbol{y}\| = 1$. This indicates $\|\boldsymbol{x}\| > 1$, and substituting $\|\boldsymbol{x}\| = 1+r$ into $(1+r)\boldsymbol{y} = \boldsymbol{x}$ yields $\boldsymbol{y} = \boldsymbol{x}/\|\boldsymbol{x}\|$. Therefore, we have

$$\left(\boldsymbol{y} = \frac{\boldsymbol{x} - \boldsymbol{y}}{\|\boldsymbol{x} - \boldsymbol{y}\|} \wedge \boldsymbol{x} \neq \boldsymbol{y}\right) \Rightarrow (\boldsymbol{y} = \boldsymbol{x}/\|\boldsymbol{x}\| \wedge \|\boldsymbol{x}\| > 1). \tag{6}$$

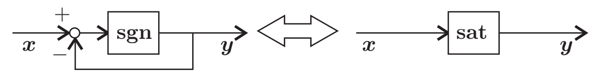



Fig. 2. Block-diagram representation of Theorem 1.

The converse of the above statement is trivial. It is also trivial that $(\boldsymbol{y} = \boldsymbol{x} \wedge \|\boldsymbol{y}\| \leq 1)$ is equivalent to $(\boldsymbol{y} = \boldsymbol{x} \wedge \|\boldsymbol{x}\| \leq 1)$. Therefore, we have

$$\begin{aligned} &\boldsymbol{y} = \mathbf{sgn}(\boldsymbol{x} - \boldsymbol{y}) \\ \iff &(\boldsymbol{y} = \boldsymbol{x}/\|\boldsymbol{x}\| \wedge \|\boldsymbol{x}\| > 1) \vee (\boldsymbol{y} = \boldsymbol{x} \wedge \|\boldsymbol{x}\| \leq 1) \\ \iff &\boldsymbol{y} = \mathbf{sat}(\boldsymbol{x}). \end{aligned} \tag{7}$$

Thus, the proof is complete. □

**Corollary 1.** *With two $n$-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ and two positive real numbers $X$ and $Y$, the following statement holds true:*

$$\boldsymbol{y} = X\mathbf{sgn}(Y(\boldsymbol{x} - \boldsymbol{y})) \iff \boldsymbol{y} = X\mathbf{sat}(\boldsymbol{x}/X). \tag{8}$$

*Proof.* Because $\mathbf{sgn}(Z\boldsymbol{z}) = \mathbf{sgn}(\boldsymbol{z})$ for all $Z > 0$ and $\boldsymbol{z} \in \mathcal{R}^n$, (8) can be rewritten as

$$\boldsymbol{y}/X = \mathbf{sgn}(\boldsymbol{x}/X - \boldsymbol{y}/X) \iff \boldsymbol{y}/X = \mathbf{sat}(\boldsymbol{x}/X). \tag{9}$$

The above holds true because of Theorem 1. □

Fig. 2 shows a block-diagram representation of Theorem 1. Theorem 1 is a multidimensional extension of the theorem introduced in the previous paper [3]. Theorem 1 and Corollary 1 imply that if the discontinuous function $\mathbf{sgn}(\cdot)$ is enclosed within a closed loop without time delay, it can be removed by using the continuous function $\mathbf{sat}(\cdot)$.

### B. Continuous-Time Representation

Let us consider a position control system in $n$-dimensional Cartesian space. Let $\boldsymbol{p}_d$ and $\boldsymbol{p}$ denote the desired and actual positions of a controlled object, respectively. Let $\boldsymbol{v}_d$ and $\boldsymbol{v}$ denote the time derivatives of $\boldsymbol{p}_d$ and $\boldsymbol{p}$, respectively. The controller accepts $\boldsymbol{p}_d$ and $\boldsymbol{p}$ as inputs and determines an actuator force $\boldsymbol{f}$ as an output. The force $\boldsymbol{f}$ is applied to the controlled object. We use multidimensional representations to allow the extensibility of discussion. When $n = 1$, the following discussion can be applied to angle control in each of the joints of general manipulators.

The concept of PSMC [3] can be schematically illustrated as Fig. 3. The block diagram of this system is shown in Fig. 4. This system includes a massless virtual object, called a *proxy*, whose position is denoted by $\boldsymbol{p}_s$. The actual controlled object and proxy are connected with a stiff virtual spring-like element, which is often called a virtual coupling [8]. As apparent from Fig. 3, the proxy receives forces from the virtual coupling and a (virtual) sliding mode controller. Because the proxy is massless, these two forces balance each other. Both of them are denoted by $\boldsymbol{f}$ in Fig. 3 and Fig. 4. The same force $\boldsymbol{f}$ is produced by the real actuator and applied to the real controlled object.

The virtual coupling is assumed to produce a force based on a PID control action to maintain its length to be zero. It is represented in Fig. 4 as the block $L/s + K + Bs$, where $L$, $K$, and $B$ are the integral, proportional, and derivative gains, respectively. This means that the output of the virtual coupling, $\boldsymbol{f}$, satisfies the following equation:

$$\boldsymbol{f} = L\int(\boldsymbol{p}_s - \boldsymbol{p})\,dt + K(\boldsymbol{p}_s - \boldsymbol{p}) + B(\boldsymbol{v}_s - \boldsymbol{v}). \quad (10)$$

Meanwhile, the sliding mode controller in Fig. 3 and Fig. 4 is assumed to produce the force $\boldsymbol{f}$ based on a SMC law. The previous method [3] employed the following control law:

$$\boldsymbol{f} = F\mathbf{sgn}(\boldsymbol{\sigma}_s) \quad (11a)$$
$$\boldsymbol{\sigma}_s = \boldsymbol{p}_d - \boldsymbol{p}_s + H(\boldsymbol{v}_d - \boldsymbol{v}_s). \quad (11b)$$

Here, $F$ and $H$ are positive real numbers. This control law can be illustrated as the block shown in Fig. 5. In the $2n$-dimensional state space of the proxy ($\{\boldsymbol{p}_s, \boldsymbol{v}_s\}$), the manifold $\boldsymbol{\sigma}_s = \boldsymbol{o}$ is an $n$-dimensional subspace, as illustrated in Fig. 1(a). This manifold is called the sliding manifold, and the control law (11) acts to attract the proxy's state $\{\boldsymbol{p}_s, \boldsymbol{v}_s\}$ toward the sliding manifold.

One potential risk of using the control law (11) is that it can produce excessive speed before the proxy's state reaches the sliding manifold. In stead of the control law (11), we here propose to use the following SMC law:

$$\boldsymbol{f} = F\mathbf{sgn}(\boldsymbol{s}_s) \quad (12a)$$
$$\boldsymbol{s}_s = V\mathbf{sat}\left((A\boldsymbol{\sigma}_s + \boldsymbol{v}_s)/V\right) - \boldsymbol{v}_s \quad (12b)$$
$$\boldsymbol{\sigma}_s = \boldsymbol{p}_d - \boldsymbol{p}_s + H(\boldsymbol{v}_d - \boldsymbol{v}_s). \quad (12c)$$

Here, $V$ and $A$ are positive real numbers.

The set $\boldsymbol{s}_s = \boldsymbol{o}$ is an $n$-dimensional manifold in the $2n$-dimensional state space $\{\boldsymbol{p}_s, \boldsymbol{v}_s\}$, which is schematically illustrated in Fig. 6. Equation (12b) shows that $\boldsymbol{s}_s = \boldsymbol{o}$ implies

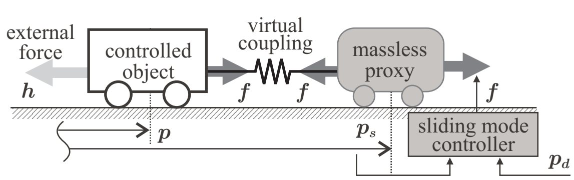



Fig. 3. A proxy-based implementation of a sliding mode control law.

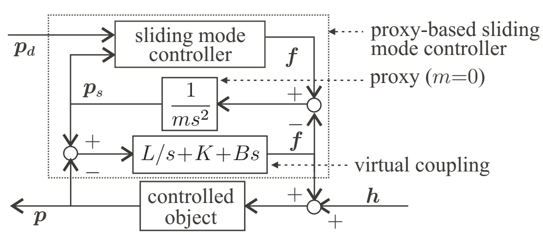



Fig. 4. Block-diagram representation of Fig. 3.

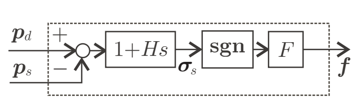



Fig. 5. Sliding mode controller with a liner sliding manifold in [3].

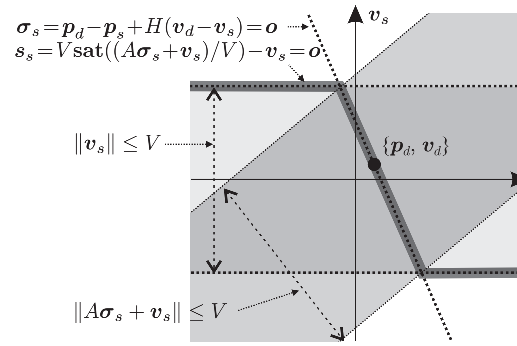



Fig. 6. Velocity-bounding sliding manifold.

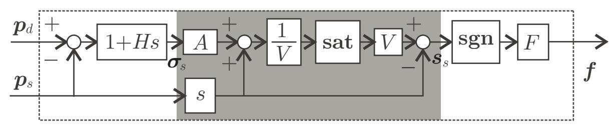



Fig. 7. Velocity-bounding sliding mode controller. The gray portion highlights the difference from Fig. 5.

$\|\boldsymbol{v}_s\| \le V$ because the norm of the first term in the right-hand side of (12b) is always smaller than $V$. This means that the magnitude of the proxy's velocity is equal to or smaller than $V$ as long as the proxy's state is on the manifold $\boldsymbol{s}_s = \boldsymbol{o}$. When $\boldsymbol{s}_s = \boldsymbol{o}$ and $\|\boldsymbol{v}_s\| < V$, $\|(A\boldsymbol{\sigma}_s + \boldsymbol{v}_s)/V\|$ is smaller than 1 and as a result, $A\boldsymbol{\sigma}_s + \boldsymbol{v}_s - \boldsymbol{v}_s = \boldsymbol{o}$ is satisfied. This means that $\boldsymbol{\sigma}_s = \boldsymbol{o}$ is satisfied if $\boldsymbol{s}_s = \boldsymbol{o}$ and $\|\boldsymbol{v}_s\| < V$. Therefore, as long as $\|A\boldsymbol{\sigma}_s + \boldsymbol{v}_s\| \le V$ is satisfied, (12) is equivalent to (11). Notice that the parameter $A$ does not influence the manifold $\boldsymbol{s}_s = \boldsymbol{o}$. Fig. 7 shows the block diagram of the control law (12).

As a whole, the controller represented by Fig. 4 and Fig. 7 satisfies the following equations:

$$\boldsymbol{f} = L\int(\boldsymbol{p}_s - \boldsymbol{p})dt + K(\boldsymbol{p}_s - \boldsymbol{p}) + B(\boldsymbol{v}_s - \boldsymbol{v}) \quad (13a)$$
$$\boldsymbol{f} = F\mathbf{sgn}(\boldsymbol{s}_s) \quad (13b)$$
$$\boldsymbol{s}_s = V\mathbf{sgn}\left((A\boldsymbol{\sigma}_s + \boldsymbol{v}_s)/V\right) - \boldsymbol{v}_s \quad (13c)$$
$$\boldsymbol{\sigma}_s = \boldsymbol{p}_d - \boldsymbol{p}_s + H(\boldsymbol{v}_d - \boldsymbol{v}_s). \quad (13d)$$

Note that $\boldsymbol{f}$ satisfies both of (13a) and (13b).

### C. Discrete-Time Representation

Based on the Euler approximation, we have the discrete-time representation of (13) as follows:

$$\boldsymbol{f}(k) = L\boldsymbol{a}(k) + K\nabla\boldsymbol{a}(k)/T + B\nabla^2\boldsymbol{a}(k)/T^2 \quad (14a)$$
$$\boldsymbol{f}(k) = F\mathbf{sgn}(\boldsymbol{s}_s(k)) \quad (14b)$$
$$\boldsymbol{s}_s(k) = V\mathbf{sat}\left((A\boldsymbol{\sigma}_s(k) + \boldsymbol{v}_s(k))/V\right) - \boldsymbol{v}_s(k) \quad (14c)$$
$$\boldsymbol{\sigma}_s(k) = \boldsymbol{p}_d(k) - \boldsymbol{p}_s(k) + H(\boldsymbol{v}_d(k) - \boldsymbol{v}_s(k)) \quad (14d)$$
$$\nabla\boldsymbol{a}(k) = T(\boldsymbol{p}_s(k) - \boldsymbol{p}(k)). \quad (14e)$$

Here, symbols in the parentheses, such as $k$, denote discrete-time indices. The operator $\nabla$ denotes the backward difference operator, which is defined by $\nabla\boldsymbol{x}(k) = \boldsymbol{x}(k) - \boldsymbol{x}(k-1)$ and satisfies $\nabla^2\boldsymbol{x}(k) = \nabla\boldsymbol{x}(k) - \nabla\boldsymbol{x}(k-1) = \boldsymbol{x}(k) - 2\boldsymbol{x}(k) + \boldsymbol{x}(k-2)$. We assume $\boldsymbol{v}(k) = \nabla\boldsymbol{p}(k)/T$, $\boldsymbol{v}_s(k) = \nabla\boldsymbol{p}_s(k)/T$, and $\boldsymbol{v}_d(k) = \nabla\boldsymbol{p}_d(k)/T$. Notice that (14) cannot be considered as a computational procedure, but as a set of algebraic constraints that must be satisfied. The inputs to the system are $\boldsymbol{p}_d(k)$ and $\boldsymbol{p}(k)$ and their derivatives, and thus they can be treated as known variables. Once they are provided, the actuator force $\boldsymbol{f}(k)$ and the state variable $\boldsymbol{a}(k)$ (or its time difference $\nabla\boldsymbol{a}(k)$) must be determined so that (14) is satisfied.

The block diagrams of Fig. 4 and Fig. 7 show that the discontinuous $\mathbf{sgn}(\cdot)$ element is surrounded by a feedback loop without time delay within the controller software. Therefore, due to Corollary 1, the function $\mathbf{sgn}(\cdot)$ in (14b) can be removed and an analytical solution for (14) can be obtained at least if the first term of the right-hand side of (14c) is known. Because $A$ is an arbitrary positive value, we can choose $A$ for the convenience of analytical derivations. Specifically, we choose $A$ so that

$$\boldsymbol{u}^*(k) = A\boldsymbol{\sigma}_s(k) + \boldsymbol{v}_s(k) \tag{15}$$

(which appears in (14c)) is independent from unknown variables. We can write $\boldsymbol{v}_s(k)$ and $\boldsymbol{\sigma}_s(k)$ as follows:

$$\boldsymbol{v}_s(k) = \boldsymbol{v}(k) + \nabla\boldsymbol{a}(k)/T^2 - \nabla\boldsymbol{a}(k-1)/T^2 \tag{16}$$

$$\begin{aligned}\boldsymbol{\sigma}_s(k) &= \boldsymbol{p}_d(k) - \boldsymbol{p}(k) - \nabla\boldsymbol{a}(k)/T + H(\boldsymbol{v}_d(k) - \boldsymbol{v}(k) \\ &\qquad -(\nabla\boldsymbol{a}(k) - \nabla\boldsymbol{a}(k-1))/T^2) \\ &= \boldsymbol{p}_d(k) - \boldsymbol{p}(k) + H(\boldsymbol{v}_d(k) - \boldsymbol{v}(k)) \\ &\qquad -(H+T)\nabla\boldsymbol{a}(k)/T^2 + H\nabla\boldsymbol{a}(k-1)/T^2. \end{aligned}\tag{17}$$

Here, note that $\nabla\boldsymbol{a}(k)$ is unknown. By setting

$$A = 1/(T+H), \tag{18}$$

we can make $\boldsymbol{u}^*(k)$ independent from $\nabla\boldsymbol{a}(k)$, as follows:

$$\boldsymbol{u}^*(k) = \frac{\boldsymbol{p}_d(k) - \boldsymbol{p}(k) + H\boldsymbol{v}_d(k) + T\boldsymbol{v}(k) - \nabla\boldsymbol{a}(k-1)/T}{H+T}. \tag{19}$$

This means that setting $A = 1/(T+H)$ allows us to treat $\boldsymbol{u}^*(k)$ as a known variable.

Letting

$$\boldsymbol{u}(k) = V\mathbf{sat}(\boldsymbol{u}^*(k)/V), \tag{20}$$

we can simply rewrite (14b) as follows:

$$\boldsymbol{f}(k) = F\mathbf{sgn}(\boldsymbol{u}(k) - \boldsymbol{v}_s(k)). \tag{21}$$

Here, $\boldsymbol{v}_s(k)$ depends on the unknown variable $\boldsymbol{f}(k)$. From (14a), we have

$$\nabla\boldsymbol{a}(k) = \frac{T^2\boldsymbol{f}(k) - LT^2\boldsymbol{a}(k-1) + B\nabla\boldsymbol{a}(k-1)}{LT^2 + KT + B}. \tag{22}$$

Substituting the above into (16) yields

$$\begin{aligned}\boldsymbol{v}_s(k) = \boldsymbol{v}(k) &- \frac{LT\boldsymbol{a}(k-1) + (LT+K)\nabla\boldsymbol{a}(k-1)}{T(LT^2+KT+B)} \\ &+ \frac{\boldsymbol{f}(k)}{LT^2+KT+B}.\end{aligned}\tag{23}$$

Therefore, by using

$$\begin{aligned}\boldsymbol{f}^*(k) :=\, & L\boldsymbol{a}(k-1) + (LT+K)\nabla\boldsymbol{a}(k-1)/T \\ & +(LT^2+KT+B)(\boldsymbol{u}(k) - \boldsymbol{v}(k)),\end{aligned}\tag{24}$$

we obtain

$$\boldsymbol{u}(k) - \boldsymbol{v}_s(k) = (\boldsymbol{f}^*(k) - \boldsymbol{f}(k))/(LT^2+KT+B). \tag{25}$$

Substituting the above into (21) yields

$$\boldsymbol{f}(k) = F\mathbf{sgn}\left(\frac{\boldsymbol{f}^*(k) - \boldsymbol{f}(k)}{LT^2+KT+B}\right). \tag{26}$$

By using Corollary 1, we have

$$\boldsymbol{f}(k) = F\mathbf{sgn}\left(\boldsymbol{f}^*(k)/F\right), \tag{27}$$

which provides the solution for $\boldsymbol{f}(k)$. Once $\boldsymbol{f}(k)$ is obtained, the other unknown $\boldsymbol{a}(k)$ can be calculated by using (14a).

In conclusion, the control law of VB-PSMC, i.e., the computational procedure to calculate $\boldsymbol{f}(k)$ and $\boldsymbol{a}(k)$ that satisfy (14), is obtained as follows:

$$\boldsymbol{u}^*(k) := \frac{\boldsymbol{p}_d(k) - \boldsymbol{p}(k) + H\boldsymbol{v}_d(k) + T\boldsymbol{v}(k) - \nabla\boldsymbol{a}(k-1)/T}{H+T} \tag{28a}$$

$$\boldsymbol{u}(k) := V\mathbf{sat}(\boldsymbol{u}^*(k)/V) \tag{28b}$$

$$\begin{aligned}\boldsymbol{f}^*(k) :=\, & L\boldsymbol{a}(k-1) + (LT+K)\nabla\boldsymbol{a}(k-1)/T \\ & +(LT^2+KT+B)(\boldsymbol{u}(k) - \boldsymbol{v}(k))\end{aligned}\tag{28c}$$

$$\boldsymbol{f}(k) := F\mathbf{sat}(\boldsymbol{f}^*(k)/F) \tag{28d}$$

$$\boldsymbol{a}(k) := \frac{(KT+B)\boldsymbol{a}(k-1) + B\nabla\boldsymbol{a}(k-1) + T^2\boldsymbol{f}(k)}{LT^2+KT+B}. \tag{28e}$$

### D. Relation to Previous Methods

The control law (28) is an extension of the previously-proposed PSMC. Removing the velocity limit from the control law (28) (i.e., setting $V \to \infty$ with (28)) yields the following:

$$\begin{aligned}\boldsymbol{f}^*(k) :=\, & \frac{LT^2+KT+B}{H+T}(\boldsymbol{p}_d(k) - \boldsymbol{p}(k) + H(\boldsymbol{v}_d(k) - \boldsymbol{v}(k))) \\ & + L\boldsymbol{a}(k-1) + \frac{(K+LT)H - B}{T(H+T)}\nabla\boldsymbol{a}(k-1)\end{aligned}\tag{29a}$$

$$\boldsymbol{f}(k) := F\mathbf{sat}(\boldsymbol{f}^*(k)/F) \tag{29b}$$

$$\boldsymbol{a}(k) := \frac{(KT+B)\boldsymbol{a}(k-1) + B\nabla\boldsymbol{a}(k-1) + T^2\boldsymbol{f}(k)}{LT^2+KT+B}, \tag{29c}$$

which is the control law of PSMC based on the linear sliding manifold (11). It is also important to notice that the VB-PSMC (28) is equivalent to PSMC (29) as long as $\|\boldsymbol{u}^*(k)\| \le V$.

As is stated in the previous paper [3], the PSMC is an extension of the conventional PID control. Thus, the VB-PSMC is also an extension of the PID control. Setting $F \to \infty$ makes (29) to be equivalent to the following:

$$\begin{aligned}\boldsymbol{f}(k) :=\, & \frac{LT^2+KT+B}{H+T}(\boldsymbol{p}_d(k) - \boldsymbol{p}(k) + H(\boldsymbol{v}_d(k) - \boldsymbol{v}(k))) \\ & + L\boldsymbol{a}(k-1) + \frac{(K+LT)H - B}{T(H+T)}\nabla\boldsymbol{a}(k-1)\end{aligned}\tag{30a}$$

$$\boldsymbol{a}(k) := \frac{(KT+B)\boldsymbol{a}(k-1) + B\nabla\boldsymbol{a}(k-1) + T^2\boldsymbol{f}(k)}{LT^2+KT+B}, \tag{30b}$$

which is equivalent to

$$\begin{aligned}\boldsymbol{a}(k) :=\, & \boldsymbol{a}(k-1) + T(\boldsymbol{p}_d(k) - \boldsymbol{p}(k)) \\ & + \frac{H}{H+T}\left(\nabla\boldsymbol{a}(k-1) - T(\boldsymbol{p}_d(k-1) - \boldsymbol{p}(k-1))\right)\end{aligned}\tag{31a}$$

$$\boldsymbol{f}(k) := L\boldsymbol{a}(k) + K\nabla\boldsymbol{a}(k)/T + B\nabla^2\boldsymbol{a}(k)/T^2. \tag{31b}$$

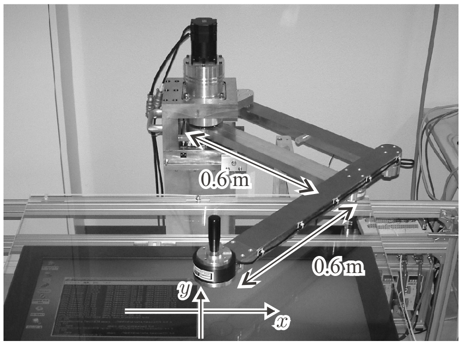



Fig. 8. Experimental setup: a parallel-link manipulator.

Letting

$$C_0 = \nabla \boldsymbol{a}(k_0) - T(\boldsymbol{p}_d(k_0) - \boldsymbol{p}(k_0)), \tag{32}$$

we can see that (31) is equivalent to

$$\boldsymbol{a}(k) := \boldsymbol{a}(k-1) + T(\boldsymbol{p}_d(k) - \boldsymbol{p}(k)) + C_0\left(\frac{H}{H+T}\right)^{k-k_0} \tag{33a}$$

$$\boldsymbol{f}(k) := L\boldsymbol{a}(k) + K\nabla\boldsymbol{a}(k)/T + B\nabla^2\boldsymbol{a}(k)/T^2, \tag{33b}$$

which is, as $k \to \infty$, equivalent to the PID control law, i.e.,

$$\boldsymbol{a}(k) := \boldsymbol{a}(k-1) + T(\boldsymbol{p}_d(k) - \boldsymbol{p}(k)) \tag{34a}$$

$$\boldsymbol{f}(k) := L\boldsymbol{a}(k) + K\nabla\boldsymbol{a}(k)/T + B\nabla^2\boldsymbol{a}(k)/T^2. \tag{34b}$$

The above derivation shows that VB-PSMC (28) is analytically equivalent to the PID control law (34) as long as $\|\boldsymbol{u}^*(k)\| \leq V$ and $\|\boldsymbol{f}^*(k)\| \leq F$ are satisfied. This means that the choices of $F$, $V$, and $H$ do not affect the tracking accuracy during normal operation as long as $\|\boldsymbol{f}^*(k)\| \leq F$ and $\|\boldsymbol{u}^*(k)\| \leq V$. The VB-PSMC can be viewed as an extension of the conventional PID control law (34) that includes

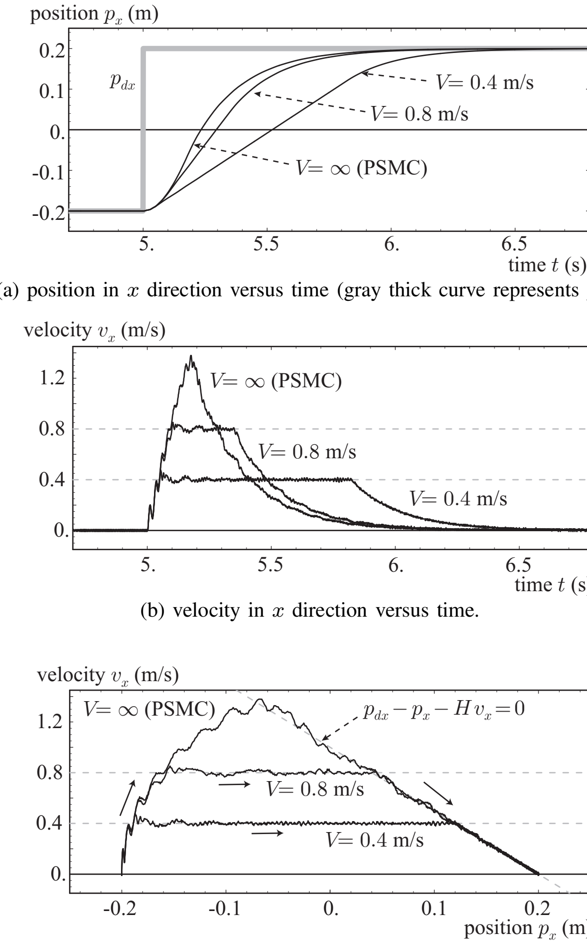
- an arbitrary magnitude limit $F$ on the actuator force,
- an arbitrary magnitude limit $V$ on the velocity, and
- overdamped recovering motions characterized by an arbitrary time constant $H$.

VB-PSMC is advantageous over PID control in that the above 3 parameters can be set independently from the PID control gains $K$, $L$, and $B$. The choices of $F$ and $V$ can be made according to safety regulations at each site where this control scheme is used.

## III. Implementation Experiments

The proposed method was implemented in the 2-DOF planar parallel link manipulator shown in Fig. 8. This manipulator had two actuators, which were AC servo motors integrated with harmonic drive gearings. The manipulator was controlled using the 2-dimensional version of the proposed control law (28). The position $\boldsymbol{p}$ of the end-effector in the Cartesian coordinate system was measured by optical encoders attached to the actuators, and the velocity $\boldsymbol{v}$ was calculated from the time difference of $\boldsymbol{p}$. The force $\boldsymbol{f}$, which should be applied to the end-effector, was determined by the computational procedure (28). The joint torque vector that is statically equivalent to $\boldsymbol{f}$ was commanded to the actuators. The sampling interval $T$ was set to be $T = 0.001$ s.

In order to exhibit the difference between VB-PSMC and PSMC, we investigated the influence of the parameter $V$. No attempts were made to test the influence of the parameters $K$, $B$, $L$, $F$, or $H$ because they are common parameters for both VB-PSMC and PSMC, and advantages of PSMC over conventional PID control have already been demonstrated in the previous paper [3]. The parameters $K$, $B$, $L$, $F$, and $H$ were fixed as $K = 80000$ N/m, $B = 600$ Ns/m, $L = 100000$ N/(ms), $F = 80$ N, and $H = 0.2$ s.




(a) position in $x$ direction versus time (gray thick curve represents $\boldsymbol{p}_d$).



(b) velocity in $x$ direction versus time.



(c) position versus velocity in $x$ direction.

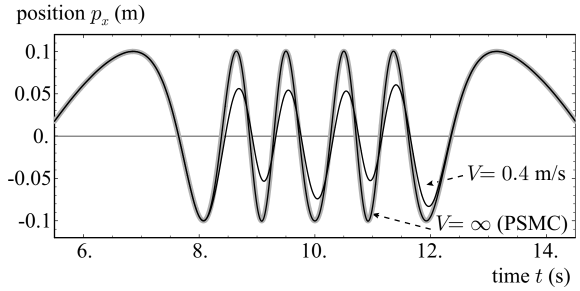
Fig. 9. Experimental results (point-to-point reaching).

In a first set of experiments, simple point-to-point reaching motions were performed. The desired position command $\boldsymbol{p}_d$ was chosen as a step-like function as follows:

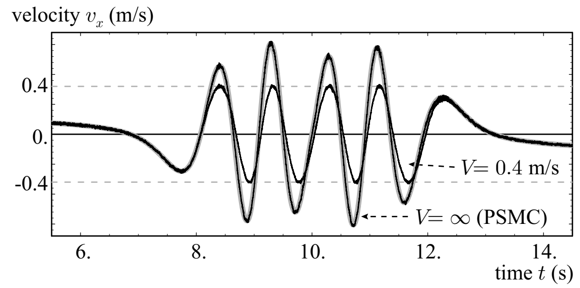
$$\boldsymbol{p}_d(t) = \begin{cases} [-0.2 \text{ m}, 0 \text{ m}]^T & \text{if } t < 5\text{s} \\ [0.2 \text{ m}, 0 \text{ m}]^T & \text{if } t > 5\text{s}. \end{cases}$$

Three trials were performed with different $V$ values: $V = 0.4$, 0.8, and $\infty$ m/s. (Setting $V = \infty$ means using PSMC (29) instead of VB-PSMC (28).)

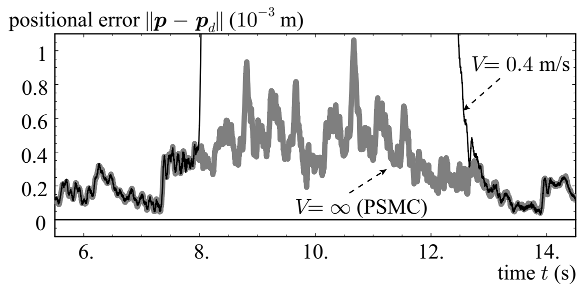
Fig. 9 shows the results of this set of experiments. At $t = 5$ s, the end-effector starts to accelerate due to the sudden




(a) position in $x$ direction versus time (gray thick curve represents $\boldsymbol{p}_d$).




(b) velocity in $x$ direction versus time (gray thick curve represents $\boldsymbol{v}_d$).




(c) magnitude of positional error versus time.

Fig. 10. Experimental results (tracking).

change in the desired position. With VB-PSMC (with $V = 0.4$ and $0.8$ m/s), the velocity becomes constant at the value of $V$, while the velocity is not limited with PSMC (with $V = \infty$). In a latter portion of each motion, the end-effector approaches to the desired position ($p_x = 0.2$ m) exhibiting an asymptotic, overdamped motion. This overdamped motion is the main advantage of PSMC (and VB-PSMC) over conventional PID control.

In a second set of experiments, the desired position command was given as follows:

$$\boldsymbol{p}_d(t) = \begin{bmatrix} 0.1\cos\Big((4e^{-\frac{(t-10)^2}{4}}+1)(|t-10|-\pi)\Big)\ \mathrm{m} \\ 0.1\sin\Big((4e^{-\frac{(t-10)^2}{4}}+1)(t-10-\mathrm{sgn}(t-10)\pi)\Big)\ \mathrm{m} \end{bmatrix}.$$

This represents a circular motion with a constant radius (0.1 m) and a time-dependent frequency; the motion becomes faster around $t = 10$ s. Two trials were performed with different $V$ values: $V = 0.4$ and $\infty$ m/s.

The results are shown in Fig. 10. As expected from the analysis presented in section II-D, Fig. 10 shows that the tracking accuracy during slow motion (before $t = 8$ s and after $t = 13$ s) is almost the same between VB-PSMC ($V = 0.4$ m/s) and PSMC ($V = \infty$), but VB-PSMC properly limits the velocity when the desired motion is fast (between $t = 8$ s and 13 s). This indicates that VB-PSMC can be safer even when erroneous position commands are provided from a higher-level controller.

## IV. Conclusions

This paper has proposed a position controller that is an extension of PID controller with improved safety. The new method, velocity-bounded proxy-based sliding mode control (VB-PSMC), allows users to arbitrarily choose

- a time constant of overdamped recovering motions from large positional errors and
- a magnitude limit on the velocity

without sacrificing accuracy and responsiveness of position control during normal operations. It is an extension of our previously-proposed proxy-based sliding mode control (PSMC) [3], in which only the time constant of recovering motions can be arbitrarily designed. The VB-PSMC method can be used as a low-level position controller for industrial robots, and it is safer than PSMC and much safer than PID control. Although the demonstrations were performed only with a 2-DOF planar manipulator, one-dimensional version of VB-PSMC can be used for independent joint angle control. Thus, VB-PSMC can also be used with general robotic manipulators with which PD or PID control has usually been used.

Future studies will be addressed to realize arbitrary bounds on actuator force and velocity vectors. This paper only considered imposing limits on L2 norms of force and velocity vectors in Cartesian space, but there will be cases where the limits should be defined in the joint space of a robotic manipulator. A complete analysis on the stability will also be an important topic.

## References

[1] P. Rocco, G. Ferretti, and G. Magnani, "Implicit force control for industrial robots in contact with stiff surfaces," *Automatica*, vol. 33, no. 11, pp. 2041–2047, 1997.

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

[3] R. Kikuuwe and H. Fujimoto, "Proxy-based sliding mode control for accurate and safe position control," in *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 26–31.

[4] K. B. Ngo and R. Mahony, "Passivity-based control of robot manipulators subject to constraints," in *Proc. of the 2005 Australian Conf. on Robotics and Automation*, 2005.

[5] ——, "Bounded torque control for robot manipulators subject to joint velocity constraints," in *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 7–12.

[6] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulator," *Trans. of the ASME: J. of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 109–125, 1981.

[7] T. Sugie, K. Fujimoto, and Y. Kito, "Obstacle avoidance of manipulators with rate constraints," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 1, pp. 168–174, 2003.

[8] J. E. Colgate, M. C. Stanley, and J. M. Brown, "Issues in the haptic display of tool use," in *Proc. of the 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, 1995, pp. 140–145.