Improving Velocity Feedback for Position Control
by Using a Discrete-Time Sliding Mode Filtering with Adaptive Windowing

Shanhai Jin, Ryo Kikuuwe* and Motoji Yamamoto

Department of Mechanical Engineering, Kyushu University,
Motooka 744, Nishi-ku, Fukuoka 819-0395, Japan;

In position control of mechatronic devices, velocity feedback is important for injecting additional damping to avoid low-frequency fluctuation around desired trajectories. In practice, velocity signal is often obtained by finite difference of position signal from an optical encoder. However, such a numerical differentiation produces high-frequency noise by magnifying quantization error contained in the position signal. As a result, the controller may produce high-frequency vibration. This paper presents a new noise-reduction discrete-time filter based on sliding mode and adaptive windowing. The presented filter is an improved version of a sliding mode filter by Jin et al. (2012), with including adaptive windowing of which the window size is determined in a similar way to that of a discrete-time adaptive windowing differentiator by Janabi-Sharifi et al. (2000). The presented filter is then applied to a position control of a mechatronic device for improving velocity feedback. Experimental results show that the presented filter provides better velocity feedback than its previous version, Janabi-Sharifi et al.’s differentiator, and combinations of these two filters.

Keywords: position control; nonlinear filter; sliding mode filter; damping

1. Introduction

Velocity feedback is important in position control of mechatronic devices. Especially for low-friction devices, velocity feedback is necessary for injecting additional damping to avoid low-frequency fluctuation around desired trajectories. In practice, velocity signal is often obtained by finite difference of position signal from an optical encoder. However, such a numerical differentiation produces high-frequency noise by magnifying quantization error contained in the position signal. As a result, the controller may produce high-frequency vibration. Thus, a filter is required for obtaining smooth and reliable velocity signal.

Linear filters are widely used for smoothing the velocity signal obtained by finite difference of the measured position signal. A linear filter, however, proportionally transfers any noise component into the output. In addition, strong noise attenuation usually produces a large phase lag, which may lead to the instability of feedback systems.

Some nonlinear filters have been used as a way to avoid drawbacks of linear filters. For example, median filters [2] are used for removing high-frequency noise, although they are computationally expensive [3]. Stochastic filters, such as Kalman filter [4–7], are also useful in some applications. They however require a dynamics model of the source of the signal, which is not always available, and their performance depend on the accuracy of the model.

*Corresponding author. Email: kikuuwe@ieee.org

The paper extends the authors’ previous conference publication [1] by including more details of the derivation process, new experimental results, and stability analysis.
In the last decade, much attention has been paid to the study of sliding mode observers based on the super-twisting algorithm [8–10]. These observers theoretically realize finite time convergence in continuous-time analysis. However, the accuracy of convergence in discrete-time implementation, typically with finite difference, depends on the sampling interval, as pointed out in [8]. In addition, they are prone to overshoot during the convergence. Moreover, a system dynamics model is also required in these observers.

The sliding mode filter employing a parabolic-shaped sliding surface has also been studied. This filter was independently proposed by Han and Wang [11] and Emaru and Tsuchiya [12, 13]. The main advantage of their filter is that it realizes finite time convergence of the output to the input value when a constant input is provided. Their filter however is prone to overshoot.

In a previous paper [14], the authors proposed a new parabolic sliding mode filter, which we referred to as PSMF, for effectively removing high-frequency noise in robotic and mechatronic control systems. It is reported in [14] that the filter produces smaller phase lag than linear filters, and it is less prone to overshoot than the sliding mode filter [11–13] that also has the parabolic-shaped sliding surface. The authors also have shown the effectiveness of PSMF in position control of a device [15]. The algorithm of PSMF is based on the backward Euler differentiation.

Meanwhile, a finite-time differentiator focusing on the window size has been presented by Janabi-Sharifi et al. [16]. Their differentiator, named as a first-order adaptive windowing (FOAW), adaptively changes its window size to find a window size that optimizes the trade-off between the output smoothness and the suppression of delay in numerical differentiation. This differentiator has been utilized for quantized signals by several researchers [17–19].

This paper presents a new noise-reduction discrete-time filter. The presented filter is an improved version of the authors’ PSMF [14], with including adaptive windowing of which the size is determined by an adaptive method inspired by FOAW. The presented filter is then applied to a position control of a mechatronic device for improving velocity feedback. Experimental results show that the presented filter rather improves velocity feedback compared to PSMF, FOAW, and combinations of these two filters in position control of a mechatronic device.

The rest of this paper is organized as follows. Section 2 gives a brief overview of previous works. Section 3 presents a new discrete-time filter and Section 4 shows its effectiveness through experimental results of position control. Section 5 concludes the paper.

2. Overview of Previous Works

2.1 Parabolic Sliding Mode Filters

This section overviews the sliding mode filter presented in a previous paper [14]. The notational conventions used in the paper are slightly different from those in the authors’ previous papers [14, 20] and other works, e.g., [21], as is detailed in Appendix A. The continuous-time representation of the filter presented in [14] can be written as follows:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &\in - \frac{F(H + 1)}{2} \text{sgn}(\sigma(F, u, x_1, x_2)) - \frac{F(H - 1)}{2} \text{sgn}(x_2)
\end{align*}
\]  

(1a)

(1b)

where

\[
\sigma(F, u, x_1, x_2) \triangleq 2F(x_1 - u) + |x_2|x_2. 
\]

(2)

Here, \( u \in \mathbb{R} \) is the input to the filter, \( x_1 \in \mathbb{R} \) and \( x_2 \in \mathbb{R} \) are the outputs of the filter, and \( F > 0 \) and \( H > 1 \) are constants. The function \( \text{sgn}(\cdot) \) is the set-valued signum function defined as
The filter (1) is an extension of a conventional parabolic sliding mode filter that is described as follows:

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &\in -F \text{sgn}(\sigma(F, x_1, x_2, u)).
\end{align*}
\]

Here, \(\sigma(F, x_1, x_2, u)\) is the same as the one defined in (2). The filter (4) was independently proposed by Han and Wang [11] and Emaru and Tsuchiya [12, 13]. This filter can be viewed as a special case of the filter (1) with \(H = 1\).

Figure 1 shows the sliding surface and trajectories of the state \((x_1, x_2)\) of the filter (1) and the filter (4). It is shown that the state of the filter (1) is attracted to the sliding surface from both sides, whereas the state of the filter (4) moves in parallel to the sliding surface in the regions of \(\sigma x_2 > 0\). Thus, the filter (1) is less prone to overshoot than the filter (4). This advantage of the filter (1) is attributed to the use of \(H > 1\) in the regions of \(\sigma x_2 > 0\), while \(H = 1\) in the filter (4).

The paper [14] also provides the filter (1)’s discrete-time algorithm (hereafter, referred to as PSMF). Specifically, by taking the backward Euler discretization of (1), (i.e., by replacing \(x_2\) by \((x_1(k) - x_1(k-1))/T\) and by careful analytical derivation detailed in [14], one can obtain the
following algorithm:

Algorithm PSMF

\[
x_2^* := FT \Phi \left( \frac{u(k) - x_1(k-1)}{FT^2} \right) \tag{5a}
\]

\[
x_2(k) := \text{clip} (\text{clip} (x_2(k-1) - HF T, x_2(k-1) - FT, 0), \text{clip} (x_2(k-1) + FT, x_2(k-1) + FTH, 0), x_2^*) \tag{5b}
\]

\[
x_1(k) := x_1(k-1) + T x_2(k) \tag{5c}
\]

RETURN \( x_1(k) \) and \( x_2(k) \). \tag{5d}

Here, \( T \) is the sampling interval, and \( \Phi() \) and \( \text{clip}() \) are functions respectively defined as follows:

\[
\Phi(x) \triangleq \text{sgn}(x) \left( \sqrt{1 + 2|x|} - 1 \right) \tag{6}
\]

\[
\text{clip}(a, b, x) \triangleq \begin{cases} 
  b & \text{if } x > b \\
  x & \text{if } x \in [a, b] \\
  a & \text{if } x < a 
\end{cases} \tag{7}
\]

where \( a \in \mathbb{R} \) and \( b \in \mathbb{R} \) are arguments that satisfy \( a \leq b \). The initial values should be set as \( x_1(-1) := u(0) \) and \( x_2(-1) := 0 \). In this algorithm, \( x_2^* \) is considered as an intermediate variable, and \( x_2(k) \) is determined so as to follow \( x_2^* \) under a constraint on the absolute value of \((x_2(k) - x_2(k-1))/T\). After the determination of \( x_2(k) \), \( x_1(k) \) is obtained by integrating \( x_2(k) \) through Euler integration.

Figure 2 shows the block diagram of PSMF. The discrete-time implementation of PSMF does not produce chattering, which has been one of major problems of conventional implementation of sliding mode techniques. This is owing to the use of the backward Euler discretization, which has been recognized to be useful to realize exact reaching to sliding modes [22–26]. This advantage would not be obtained if an explicit integration method, such as the fourth-order Runge-Kutta method, is applied to (1). In such a case, the discontinuous signum functions in the right-hand side could not be removed from the discrete-time algorithm, causing the chattering problem.

2.2 First-Order Adaptive Windowing (FOAW)

In a paper [16], Janabi-Sharifi et al. presented a discrete-time adaptive windowing differentiator that is named as a first-order adaptive windowing (FOAW). The main principle of FOAW is that the window size is set large for a slow motion to provide smooth signal, while it is set small for a fast motion to instantly reflect rapidly varying signal. Specifically, the algorithm of FOAW
Figure 3. Block diagram of FOAW.

Figure 4. Windowing method of FOAW.

is as follows:

Algorithm FOAW

FOR \( n \in \{1, \cdots, \min(k+1, N_{\text{max}})\} \) \hspace{1cm} (8a)

\( a := (u(k) - u(k-n))/(nT) \) \hspace{1cm} (8b)

FOR \( i \in \{1, \cdots, n\} \)

\( e := u(k-i) - u(k) + iTa \) \hspace{1cm} (8c)

IF \(|e| > C\) RETURN \( x_2(k) := a \) \hspace{1cm} (8d)

END FOR \hspace{1cm} (8e)

END FOR \hspace{1cm} (8f)

RETURN \( x_2(k) := a \) \hspace{1cm} (8g)

where \( u(k) \) and \( x_2(k) \) are the input and the output of the differentiator, respectively. The initial values should be set as \( u(-1) := u(0) \) and \( x_2(-1) := 0 \). The symbols \( N_{\text{max}} \in \mathbb{N} \) and \( C > 0 \) are parameters that should be chosen appropriately.

In FOAW, the current input \( u(k) \) and a previous input \( u(k-n) \) are used for windowing, as shown in Figure 3. The algorithm (8) checks whether all of the points \( \{u(k), u(k-1), \cdots, u(k-n)\} \) in the window are inside the region defined by the two end-points \( u(k) \) and \( u(k-n) \) and the constant \( C \), as shown in Figure 4. If it is the case, the window size \( n \) is further increased for obtaining smoother signal. The increase of the window size is continued until at least one point in the window lies outside the region, as also shown in Figure 4, or the window size reaches its maximum value \( N_{\text{max}} \). Then, the last \( a \) obtained by the finite-time differentiation (8b) is provided as the output \( x_2(k) \). As a whole, the output is obtained by (8b) with a window size that optimizes the trade-off between the output smoothness and the suppression of delay.

3. Discrete-Time Sliding Mode Filtering with Adaptive Windowing

This section presents a new noise-reduction discrete-time filter, which is an improved version of PSMF that is derived through a discretization scheme inspired by FOAW.
Figure 5. Block diagram of AW-PSMF.

Figure 6. Windowing in AW-PSMF

Figure 7. Outputs $x_2$ of AW-PSMF and PSMF obtained from the input $u(t) = u_0(t) + 2 \times 10^{-5}e(t)$, where $u_0(t) = \sin(t)$ and $e(t)$ is the unite white Gaussian noise. The parameters were $F = 500$ s$^{-2}$ and $H = 3$ for both filters, and $N_{max} = 50$ and $C = 0.0002$ for AW-PSMF. The sampling interval was $T = 0.001$ s.

Now, let us carefully observe the algorithm (5) of PSMF. Considering that $x_1(k)$ is an estimate of $u(k)$, one can observe that $u(k) - x_1(k-1)$, which determines $x_2^*$ in the step (5a), is approximately a numerical differentiation of $u(k)$. The latter two lines represent numerical integration to update $x_1(k)$ and $x_2(k)$. It is known that, to obtain a smooth output signal from a noisy input signal, differentiation should employ a large time-step size while integration should employ a small time-step size. In the light of this idea, we can obtain the following modified version of PSMF:

**Algorithm Modified-PSMF**

$$x_2^* := nFT\Phi \left( \frac{u(k) - x_1(k - n)}{n^2FT^2} \right)$$ (9a)

$$x_2(k) := \text{clip}(\text{clip}(x_2(k - 1) - FTH, 0, x_2(k - 1) - FT), x_2^*, \text{clip}(x_2(k - 1) + FT, 0, x_2(k - 1) + FTH))$$ (9b)

$$x_1(k) := x_1(k - 1) + Tx_2(k)$$ (9c)

**RETURN** $x_1(k)$ and $x_2(k)$. (9d)
The remaining problem is how to determine the window size $n$. Toward this problem, we apply an adaptive method inspired by FOAW to balance the trade-off between the output smoothness and the suppression of delay. Based on this, the algorithm of a new modified version of PSMF, which we name adaptive windowing parabolic sliding mode filter (AW-PSMF), is described as follows:

**Algorithm AW-PSMF**

FOR $n \in \{1, \cdots, \min(k + 1, N_{\text{max}})\}$

\[ a := (x_1(k - 1) - x_1(k - n - 1))/(nT) \quad (10\text{a}) \]

FOR $i \in \{1, \cdots, n\}$

\[ e := x_1(k - i - 1) - x_1(k - 1) + iTa \quad (10\text{d}) \]

IF $|e| > C$; EXIT LOOP

END FOR

IF $i < n$; EXIT LOOP

END FOR

\[ x_2^* := nFT\Phi \left( \frac{u(k) - x_1(k - n)}{n^2FT^2} \right) \quad (10\text{i}) \]

\[ x_2(k) := \text{clip}(\text{clip}(x_2(k - 1) - FTH, x_2(k - 1) - FT, 0), \text{clip}(x_2(k - 1) + FT, x_2(k - 1) + FTH, 0), x_2^*) \quad (10\text{j}) \]

\[ x_1(k) := x_1(k - 1) + Tx_2(k) \quad (10\text{k}) \]

RETURN $x_1(k)$ and $x_2(k)$.

Here, the initial values should be set as $x_1(-2) := x_1(-1) := u(0)$ and $x_2(-1) := 0$.

In AW-PSMF, the process from (10a) to (10h) determines the window size $n$ by using a modification of FOAW. Here, the difference of AW-PSMF from FOAW is that the previous output values are used for windowing in AW-PSMF, as shown in Figure 5 and Figure 6, whereas the current and previous input values are used in FOAW, as introduced in Section 2.2. Then, the step (10i) provides $x_2^*$ by using the determined window size. After that, in the step (10j), $x_2(k)$ is determined so as to follow $x_2^*$ under a constraint on the absolute value of $(x_2(k) - x_2(k - 1))/T$, and in step (10k), $x_1(k)$ is obtained by integrating $x_2(k)$.

Figure 7 compares the outputs $x_2$ of AW-PSMF and PSMF obtained from an input corrupted by white Gaussian noise. In this figure, the output $x_2$ of AW-PSMF is smoother than that of PSMF.

4. Experiments

4.1 Problem Demonstration

Now, the problem of position control without or with inappropriate velocity feedback is demonstrated. In the rest part of this section, the sampling interval $T = 0.002$ s was used for the experiments. First, a proportional (P) controller was used to control a device (G25 Racing Wheel, Logitech International S.A., Encoder resolution: 65536 [pulse/rev]) shown in Figure 8 for demonstrating the necessity of velocity feedback, as shown in Figure 9(a). The discrete-time expression of the P controller is as follows:

\[ \tau(k) = K_p(\theta_d(k) - \theta(k)) \quad (11) \]
Figure 8. Experimental setup: Logitech G25 Racing Wheel, Logitech International S.A., Encoder resolution: 65536 pulse/rev

\[ \theta_d(k) \rightarrow K_p \rightarrow Device \rightarrow \theta(k) \]

(a) P-controlled system

\[ \theta_d(k) \rightarrow K_p \rightarrow Device \rightarrow \theta(k) \]

\[ \frac{1-z^{-1}}{T} \omega_d(k) \]

\[ \omega(k) \frac{1-z^{-1}}{T} \]

(b) PD-controlled system

Figure 9. Block diagrams of P-controlled and PD-controlled systems.

where \( \theta_d(k) \in \mathbb{R} \) is the desired trajectory, \( \theta(k) \in \mathbb{R} \) is the current position, \( \tau(k) \in \mathbb{R} \) is the output force, and \( K_p > 0 \) is the proportional gain. Figure 10 shows the P-controlled results with \( K_p = 40.0 \) Nm/rad and the following sinusoidal desired trajectory:

\[
\theta_d(k) = \begin{cases} 
0 \text{ rad if } kT < 5 \text{ s,} \\
\sin(kT - 5) \text{ rad otherwise.} 
\end{cases} \tag{12}
\]

In Figure 10, one can observe that the P control resulted in low-frequency fluctuation of the output position around the desired trajectory. Thus, additional damping should be injected for stabilizing the device.

Next, a proportional-derivative (PD) controller was used for removing the aforementioned problem, as shown in Figure 9(b). The discrete-time expression of the PD controller is as follows:

\[
\tau(k) = K_p(\theta_d(k) - \theta(k)) + K_d(\omega_d(k) - \omega(k)) \tag{13}
\]

where \( \omega_d(k) \in \mathbb{R} \) and \( \omega(k) \in \mathbb{R} \) are the first-order derivatives of \( \theta_d(k) \) and \( \theta(k) \), respectively, and \( K_d > 0 \) is the derivative gain. Figure 11 shows the PD-controlled results with \( K_p = 40.0 \) Nm/rad, \( K_d = 1.0 \) Nm·s/rad and the desired trajectory (12). It is shown that the problem of low-frequency fluctuation was removed owing to the velocity feedback. However, undesirable high-frequency vibration was produced in the device instead. This is because the velocity signal obtained through finite difference was corrupted by high-frequency noise. Thus, a filter is required for improving velocity feedback.
4.2 Improving Velocity Feedback

The effectiveness of AW-PSMF in position control is now presented. For comparison, we used PSMF, FOAW, and their simple combinations, which are hereafter referred to as PSMF+FOAW and FOAW+PSMF. Figure 12 shows the block diagrams of the PD-controlled systems with the five filtering methods, and Table 1 shows the parameter values used in the experiment. We here do not consider other filtering methods, such as the linear low-pass filters, the conventional parabolic sliding mode filter (4), and the filter (1) with other integration methods, because they have been discussed in our previous papers [14, 20].
In the experiments, the following PD controller was used:

\[ \tau(k) = K_p(\theta_d(k) - \theta(k)) + K_d(\omega_d(k) - \omega_f(k)) \]  \hspace{1cm} (14)

where \( \omega_f(k) \in \mathbb{R} \) is the estimated velocity obtained through a filter. The gains were chosen as...
The unit and the values of $F$ for FOAW+PSMF are different from those for other filters because FOAW was applied to the angular velocity in the case of FOAW+PSMF while it was applied to the angle in other cases.

<table>
<thead>
<tr>
<th>Filter</th>
<th>$F$</th>
<th>$H$</th>
<th>$N_{\text{max}}$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AW-PSMF</td>
<td>$400 \text{ rad/s}^2$</td>
<td>3</td>
<td>10</td>
<td>0.0002 rad</td>
</tr>
<tr>
<td>PSMF</td>
<td>$300, 400, 500 \text{ rad/s}^2$</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOAW</td>
<td>$N_{\text{max}} = 10, 15$</td>
<td>3</td>
<td></td>
<td>0.0002 rad</td>
</tr>
<tr>
<td>PSMF+FOAW</td>
<td>$F = 300, 400, 500 \text{ rad/s}^2$</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOAW+PSMF</td>
<td>$F = 25000, 30000, 35000 \text{ rad/s}^2$</td>
<td>3</td>
<td></td>
<td>0.0002 rad</td>
</tr>
</tbody>
</table>

$K_p = 40.0 \text{ Nm/rad}$ and $K_d = 1.0 \text{ Nm-s/rad}$. In addition, two desired trajectories were used. One was the sinusoidal trajectory (12), and the other was the sum of two sinusoidal functions as follows:

$$\theta_d(k) = \begin{cases} 
0 \text{ rad} & \text{if } kT < 5 \text{ s} \\
\sin(kT - 5) + \sin(0.6(kT - 5)) \text{ rad} & \text{otherwise.}
\end{cases}$$

In the upcoming analysis, the knowledge of the “true” velocity is required to quantitatively evaluate the control performance. However, due to the limited encoder resolution, it is impossible to obtain the “true” velocity. Thus, we obtained approximated velocity signal in the following procedure:

**Step (1):** A velocity signal $\omega \in \mathbb{R}$ was obtained by using the finite difference, i.e.,

$$\omega(k) := (\theta(k) - \theta(k - 1))/T, \; k \in \{1, \cdots, N\}.$$

**Step (2):** Another signal $\omega_r \in \mathbb{R}$ was obtained by reversing the order of $\omega(k)$, i.e.,

$$\omega_r(k) := \omega(N - k + 1), \; k \in \{1, \cdots, N\}.$$

**Step (3):** Signals $\omega_f \in \mathbb{R}$ and $\omega_{fr} \in \mathbb{R}$ were obtained by smoothing $\omega$ and $\omega_r$, respectively, with linear low-pass filters. Here, we set the cutoff frequency of the low-pass filters as 50 times of the dominant frequency of the desired velocity $\omega_d$.

**Step (4):** The approximated velocity $\omega_{apx} \in \mathbb{R}$ was obtained by

$$\omega_{apx}(k) := (\omega_f(k) + \omega_{fr}(N - k + 1))/2, \; k \in \{1, \cdots, N\}.$$

An example of the output obtained through this procedure is shown in Figure 13. Figure 14 and Figure 15 show the position control results through the data of the average
Figure 13. Example of “true” velocity approximation.

magnitudes of $|\theta_d - \theta|$, $|\omega_d - \omega_{apx}|$ and $|\ddot{\tau}|$, which are respectively defined as follows:

\[
\text{AMP} \triangleq \frac{1}{60000 - 5000} \sum_{k=5000}^{60000} |\theta_d(k) - \theta(k)| \tag{16}
\]

\[
\text{AMV} \triangleq \frac{1}{60000 - 5000} \sum_{k=5000}^{60000} |\omega_d(k) - \omega_{apx}(k)| \tag{17}
\]

\[
\text{AMF} \triangleq \frac{1}{60000 - 5000} \sum_{k=5000}^{60000} \frac{\tau(k) - \tau(k-1)}{T} \tag{18}
\]

Here, the measures AMP and AMV provide information on position and velocity of the device, and AMF can be considered as a measure of the intensity of the device’s high-frequency vibration. One can say that, for a better position control, all AMP, AMV and AMF should be maintained small. It is shown that either PSMF or FOAW cannot simultaneously reduce AMP, AMV and AMF to the levels of those of AW-PSMF by any adjustment of their parameters. It is also shown that, compared to AW-PSMF, both PSMF+FOAW and FOAW+PSMF produced similar AMP but larger AMV and AMF. Here, it should be noted that $F$ values smaller than those used in Figure 15 resulted in undesirable low-frequency fluctuation around the desired trajectory, as shown in Figure 16. Such oscillation can be attributed to the large phase lag resulted from the small $F$ values [20]. Thus, it can be concluded that AW-PSMF improved velocity feedback compared to PSMF, FOAW, PSMF+FOAW, and FOAW+PSMF.

The advantage of AW-PSMF is also observed from typical measured position data obtained with the desired trajectories (12) and (15), which are shown in Figure 17 and Figure 18, respectively. The figures show that AW-PSMF produced weaker low-frequency fluctuation than PSMF, FOAW, PSMF+FOAW, and FOAW+PSMF did. This is attributed to the improved velocity feedback achieved by AW-PSMF.

5. Conclusions

This paper has presented a new noise-reduction discrete-time filter, named as AW-PSMF. The presented filter is an improved version of PSMF, with including adaptive windowing of which the window size is determined by an adaptive method inspired by FOAW to balance the trade-off between the output smoothness and the suppression of delay. The presented filter is then applied to a position control of a mechatronic device for obtaining reliable velocity signal. Experimental results show that AW-PSMF improved velocity feedback compared to PSMF, FOAW, and their simple combinations in position control of a mechatronic device.

One limitation of this paper is that the performance of AW-PSMF is only validated through numerical methods. Theoretical validation should be addressed in future study. Selection guidelines for the values of the parameters ($F$, $H$, $N_{max}$ and $C$) are also left to be clarified.
ACKNOWLEDGMENT

This work was supported in part by Grant-in-Aid for Scientific Research (24360098) from Japan Society for the Promotion of Science (JSPS). The authors are grateful to Dr. Vincent Acary, INRIA Grenoble Rhône-Alpes, who raised a question on the function gsgn() in the previous
Figure 16. Low-frequency fluctuation caused by inappropriate parameter values in PSMF+FOAW and FOAW+PSMF. The parameters were chosen as in Table 1 except $F = 250$ rad/s$^2$ for PSMF+FOAW and $F = 20000$ rad/s$^3$ for FOAW+PSMF, which are smaller than those in Figure 15.

papers [14, 20]. The question lead us to the expression (1b), which enabled a more detailed analysis presented in the Appendix B.

References

Figure 17. Typical position data obtained through AW-PSMF, PSMF and FOAW. The parameters were chosen as in Table 1 except $F = 400 \text{ rad/s}^2$ and $N_{\text{max}} = 10$.

Figure 18. Typical position data obtained through AW-PSMF, PSMF+FOAW and FOAW+PSMF. The parameters were chosen as in Table 1 except $F = 400 \text{ rad/s}^2$ for AW-PSMF and PSMF+FOAW and $F = 30000 \text{ rad/s}^3$ for FOAW+PSMF.


Appendix A. Notations

This appendix section provides some remarks on the notational conventions used in this paper. In the previous paper [14], the state equation of PSMF, which corresponds to (1b), was described in the following form:

\[ \dot{x}_2 \in \text{gsgn}(\text{gsgn}(-FH, x_2, -F), \sigma(F, u, x_1, x_2), \text{gsgn}(F, x_2, FH)). \] (A1)

Here, \( \text{gsgn}() \) is a set-valued function defined as follows:

\[
\text{gsgn}(A, x, B) \triangleq \begin{cases} 
    A & \text{if } x > 0 \\
    \text{Conv}(A \cup B) & \text{if } x = 0 \\
    B & \text{if } x < 0 
\end{cases}
\] (A2)

where \( A \) and \( B \) are closed intervals (a compact set in \( \mathbb{R} \)), each of which can be a single value as a special case, and \( \text{Conv}(X) \) denotes the convex hull of the set \( X \). Through a careful but straightforward derivation, one can see that the following relation holds true for all \( x, y \in \mathbb{R} \) and \( b \geq a \geq 0 \):

\[
\text{gsgn}(\text{gsgn}(-b, x, -a), y, \text{gsgn}(a, x, b)) = \frac{b + a}{2} \text{sgn}(y) + \frac{b - a}{2} \text{sgn}(x).
\] (A3)

This relation leads to the equivalence between (1b) and (A1).

In order to derive the discrete-time algorithm (5) from the expression (A1), the previous paper [14] used the following relation:

\[
\begin{align*}
    z - y \in &\ \text{gsgn}(\text{gsgn}(d, -z, c), x - z, \text{gsgn}(a, -z, b)) \\
\iff &\ z - y = \text{gsat}(\text{gsat}(d, -y, c), x - y, \text{gsat}(a, -y, b))
\end{align*}
\] (A4)

where

\[
\text{gsat}(a, x, b) \triangleq \text{clip}(a, b, x)
\] (A5)

and \( d \leq c \leq a \leq b \). The discrete-time form (5) can also be derived from (1b) without using the function \( \text{gsgn} \). To do this, it is convenient to note the following relation, which is a special case of (A4):

\[
\begin{align*}
    z - y \in &\ \frac{b + a}{2} \text{sgn}(x - z) + \frac{b - a}{2} \text{sgn}(-z) \\
\iff &\ z - y = \text{clip}(\text{clip}(-b, -a, -y), \text{clip}(a, b, -y), x - y)
\end{align*}
\] (A6)
where \( b \geq a \geq 0 \). The relation (A6) can be proven by specializing (A4) by \( d = -b \) and \( c = -a \), or by proceeding through a similar process of derivation to that in the proof of (A4) in the Appendix of [14].

It must be noted that the functions used in this paper can be expressed by using a more general notations found in the literature from the field of convex analysis. As is seen in, e.g., [21], let us denote a projection operator by \( \text{proj}_K(x) \), which is the nearest point of \( x \) in the convex set \( K \), and a normal cone operator by \( N_K(x) \), which is the normal cone at \( x \) of the convex set \( K \). Then, they are related to the functions \( \text{gsat}(), \text{clip}(), \text{sgn}() \) and \( g\text{sgn}() \) in the following manner:

\[
\text{gsat}(a, x, b) = \text{clip}(a, b, x) = \text{proj}_{[a,b]}(x)
\]

\[
\text{sgn}(x) = N_{[-1,1]}^{-1}(x)
\]

\[
\text{gsgn}(A, x, B) = \bigcup_{a \in A} \bigcup_{b \in B} N_{[a,b]}^{-1}(x).
\]

**Appendix B. Stability Analysis on Filter (1)**

This appendix section provides a stability analysis on the filter (1) under a constant input, i.e., \( \dot{u} = 0 \). Both PSMF and AW-PSMF (the algorithms (5) and (10), respectively) can be seen as discrete-time approximations of (1) with a sufficiently small \( T \).

Now, let us rewrite (1) in the following form:

\[
\dot{x}_1 = x_2 \quad \text{ (B1)}
\]

\[
\dot{x}_2 = -F(1 + \lambda)\text{sgn}(\sigma) - F\lambda\text{sgn}(x_2) \quad \text{ (B2)}
\]

where \( \lambda \equiv (H - 1)/2 \) and thus \( \lambda > 0 \). Assuming \( \dot{u} = 0 \), one can obtain \( \dot{\sigma} \) as follows:

\[
\dot{\sigma} = 2Fx_2 + 2|x_2|\dot{x}_2 \\
= -2F|x_2|\text{sgn}(\sigma) (\lambda + 1 + (\lambda - 1)\text{sgn}(\sigma x_2)).
\]

Let us define a positive definite function \( V_1 = \sigma^2 \). Then, one can see that

\[
\dot{V}_1 = -4F|\sigma x_2|(1 + \lambda + (\lambda - 1)\text{sgn}(\sigma x_2)) \\
< -8F \min(1, \lambda)|\sigma x_2| \\
= -8F \min(1, \lambda)|x_2|V_1^{1/2},
\]

which implies that the sliding mode \( \sigma = 0 \) is achieved in finite time (as is explained in, e.g., [27]) when \( \dot{u} = 0 \).

As far as the system is in the sliding mode \( \sigma = 0 \), \( x_2 = -\text{sgn}(x_1 - u)\sqrt{2F|x_1 - u|} \) is satisfied. Let us define a positive definite function \( V_2 = (x_1 - u)^2 \). Then, its time derivative can be obtained as follows:

\[
\dot{V}_2 = -2\sqrt{2F|x_1 - u|} = -2\sqrt{2FV_2^{3/4}},
\]

which means that, when \( \dot{u} = 0 \), \( x_1 = u \) is achieved in finite time as long as \( \sigma = 0 \) is satisfied.

These conditions for finite-time reaching may be violated when \( u \) is not constant, i.e., \( \dot{u} \neq 0 \) or \( \ddot{u} \neq 0 \). Stability analysis considering the effects of \( \dot{u} \) and \( \ddot{u} \) is left for future studies.