

Article

Discrete-Time Sliding Mode Filter with Adaptive Gain

Shanhai Jin ¹, Yonggao Jin ¹, Xiaodan Wang ¹ and Xiaogang Xiong ^{2,*}

¹ School of Engineering, Yanbian University, Yanji 133002, China; jinshanhai@ybu.edu.cn (S.J.); ygao@ybu.edu.cn (Y.J.); wangxiaodan@ybu.edu.cn (X.W.)

² School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

* Correspondence: xiongsg@hitsz.edu.cn; Tel.: +86-755-2603-3774

Academic Editor: Lorenzo J. Tardón

Received: 25 September 2016; Accepted: 23 November 2016; Published: 1 December 2016

Abstract: In feedback control of mechatronic systems, sensor signals are usually noisy and uncertain because of measurement errors and environmental disturbances. Such uncertainty and noise of feedback signals may cause instability of the controlled systems. This paper presents a new model-free discrete-time sliding mode filter for effectively removing noise by balancing the tradeoff between the filtering smoothness and the suppression of delay. The presented filter is an extension of a sliding mode filter (Jin et al. Real-time quadratic sliding mode filter for removing noise. *Adv. Robot.*, 2012) by including an adaptive gain, of which value is determined in a similar way to that of a first-order adaptive windowing filter (Janabi-Sharifi et al. Discrete-time adaptive windowing for velocity estimation. *IEEE Trans. Control Syst. Technol.*, 2000). The effectiveness of the presented filter is validated through numerical examples and experiments.

Keywords: adaptive gain; sliding mode filter; nonlinear filter; backward Euler discretization

1. Introduction

Sensor signals are important in feedback control of mechatronic systems. However, due to measurement errors and environmental disturbances, sensor signals are usually noisy and uncertain. Various filters are developed for obtaining smooth and reliable signals. Among them, linear filters are perhaps the first option due to their simplicity. They, however, proportionally transfer any noise component into the output. Furthermore, they produce a large phase lag, which may result in instability of feedback controlled systems, in the case of strong noise attenuation.

In order to avoid drawbacks of linear filters, nonlinear filters have been studied. For instance, median filters [1] are developed for attenuating high-frequency noise. However, it is reported [2] that they require expensive computational costs. As another example, stochastic filters, e.g., Kalman filters [3,4], are applied in some applications. However, a dynamic model of the signal source, which is not always available, is required. Moreover, their performance is related to the model accuracy.

The super-twisting algorithm [5,6] based sliding mode observers [7–9] have attracted much attention in the last decade. In these observers, finite time convergence can be theoretically realized in continuous-time analysis. However, in discrete-time implementation, the convergence accuracy depends on the sampling period, typically with finite difference [10,11]. In addition, a dynamic model is also required.

The sliding mode filter that employs a certain kind of parabolic-shaped sliding surface has been studied [12,13]. One of the major advantages of the filter is that finite time convergence of the output to the input is achieved in the case of a constant input being provided. Moreover, the filter does not

require a dynamic model. Some evaluation results [14–16] of the filter and its applications [17–19] were reported. However, the filter is prone to overshoot.

Regarding the drawback of the filter [12,13], Jin et al. [20] proposed a new model-free parabolic sliding mode filter, which is referred to as PSMF, for effectively removing noise in feedback control systems. It is reported [21] that the noise removing capability of parabolic sliding mode filter (PSMF) is almost the same as that of the second-order Butterworth low-pass filter (2-LPF), but its phase lag is smaller than that of 2-LPF. In addition, PSMF is less prone to overshoot, and it produces smaller phase lag compared with the filter [12,13]. Moreover, the discrete-time algorithm of PSMF, which is derived by using the backward Euler discretization, does not produce chattering. The effectiveness of PSMF has been experimentally validated [21,22].

This paper presents a new model-free discrete-time sliding mode filter for removing noise by balancing the tradeoff between the filtering smoothness and the suppression of delay. The presented filter is an extension of PSMF by including an adaptive gain, of which value is determined by an adaptive method inspired by a first-order adaptive windowing filter [23]. The results of numerical examples and experiments show that the presented filter performs better than PSMF.

The rest of this paper is organized as follows: Section 2 gives a brief overview of related works; Section 3 presents a new model-free discrete-time sliding mode filter; Sections 4 and 5 validate the effectiveness of the presented filter through numerical examples and experiments, respectively; and Section 6 covers concluding remarks. In addition, the Appendix provides a list of main abbreviations and notations used in the paper.

2. Overview of Related Works

2.1. Parabolic Sliding Mode Filters

In a paper [20], Jin et al. proposed a model-free sliding mode filter employing a parabolic-shaped sliding surface (PSMF), of which continuous-time representation is given as follows:

$$\dot{x}_1 = x_2, \tag{1a}$$

$$\dot{x}_2 \in -\frac{F(H+1)}{2} \operatorname{sgn}(\sigma(F, u, x_1, x_2)) - \frac{F(H-1)}{2} \operatorname{sgn}(x_2), \tag{1b}$$

where

$$\sigma(F, u, x_1, x_2) \triangleq 2F(x_1 - u) + |x_2|x_2. \tag{2}$$

Here, $u \in \mathbb{R}$ is the input, $x_1 \in \mathbb{R}$ is the output, $x_2 \in \mathbb{R}$ is the derivative of x_1 , and $F > 0$ and $H > 1$ are parameters. In addition, $\operatorname{sgn}(\cdot)$ is the set-valued signum function defined as follows:

$$\operatorname{sgn}(z) \triangleq \begin{cases} 1 & \text{if } z > 0, \\ [-1, 1] & \text{if } z = 0, \\ -1 & \text{if } z < 0, \end{cases} \tag{3}$$

where the argument $z \in \mathbb{R}$ is a scalar. It should be noted that the return value of $\operatorname{sgn}(\cdot)$ is a set instead of zero when $z = 0$. In PSMF, \dot{x}_2 can be considered as the *acceleration* of the output x_1 , and F can be seen as the gain of the *acceleration*.

Figure 1 illustrates the parabolic-shaped sliding surface (thick curve) and the trajectories of the state (x_1, x_2) (thin curves) of PSMF in x_1 - x_2 space. In PSMF, when $\sigma x_2 < 0$ and $\sigma x_2 > 0$, \dot{x}_2 satisfies $\dot{x}_2 = |F|$ and $\dot{x}_2 = |HF|$, respectively. And thus, the state (x_1, x_2) moves toward to the parabolic sliding surface in x_1 - x_2 space. Here, it should be mentioned that, in the case of $\sigma x_2 > 0$ with $H = 1$ (i.e., the filter [12,13]), the state moves parallel to the sliding surface, i.e., the state cannot reach the sliding surface in the regions of $\sigma x_2 > 0$, resulting in overshoot during the convergence. After the state

reaches the sliding surface, i.e., $\sigma = 0$, the state is constrained on the surface and it moves along the surface to the target state $(u, 0)$. It may be worth noticing that such a behavior of PSMF is similar to that of a system under a bang-bang controller.

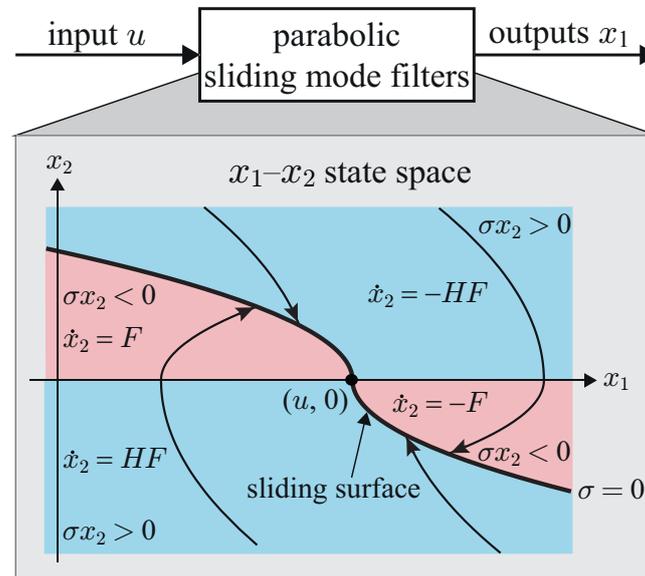


Figure 1. The parabolic-shaped sliding surface (thick curve) and trajectories of the state of parabolic sliding mode filter (PSMF) (thin curve).

The paper [20] also presented a discrete-time algorithm of PSMF, which is analytically derived by using the backward Euler discretization (i.e., by replacing x_2 by $(x_1(k) - x_1(k - 1))/T$) and the following equivalent relation [20]:

$$z - y \in \frac{b+a}{2} \text{sgn}(x - z) + \frac{b-a}{2} \text{sgn}(-z) \iff z - y = \text{clip}(\text{clip}(-b, -a, -y), \text{clip}(a, b, -y), x - y), \tag{4}$$

where $a \in \mathbb{R}$ and $b \in \mathbb{R}$ satisfy $a \leq b$, and $\text{clip}()$ is the function defined as follows:

$$\text{clip}(a, b, z) \triangleq \begin{cases} b & \text{if } z > b, \\ z & \text{if } z \in [a, b], \\ a & \text{if } z < a. \end{cases} \tag{5}$$

Specifically, the complete algorithm of PSMF is as follows (see [20] for detailed derivation):

Algorithm 1 PSMF

- 1: $x_2^* := FT \Phi\left(\frac{u(k) - x_1(k - 1)}{FT^2}\right)$
 - 2: $x_2(k) := \text{clip}(\text{clip}(x_2(k - 1) - HFT, x_2(k - 1) - FT, 0), \text{clip}(x_2(k - 1) + FT, x_2(k - 1) + FTH, 0), x_2^*)$
 - 3: $x_1(k) := x_1(k - 1) + Tx_2(k)$
 - 4: **RETURN** $x_1(k)$.
-

Here, T is the sampling interval, k is the discrete-time index, and $\Phi(\cdot)$ is the function defined as follows:

$$\Phi(z) \triangleq \text{sgn}(z) \left(\sqrt{1 + 2|z|} - 1 \right). \tag{6}$$

In Algorithm 1, $x_2^*(k)$ is the value of $x_2(k)$ that satisfies $\sigma(k) = 0$, and $x_2(k)$ is determined in order to follow $x_2^*(k)$ under the constraint of acceleration, as shown in Figure 2.

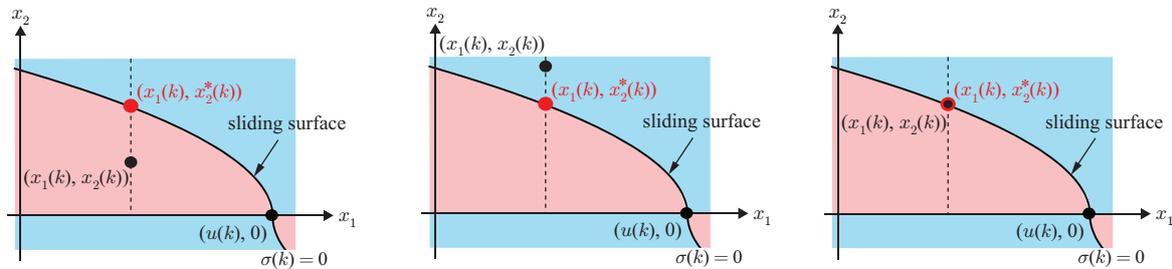


Figure 2. Relationship between $x_2(k)$ and $x_2^*(k)$.

Figure 3 shows the block diagram of the discrete-time algorithm of PSMF. One can observe that, due to the use of the backward Euler discretization and the equivalent relation (4), the algorithm is free from discontinuous functions. Thus, the numerical implementation of PSMF does not produce chattering, which has been considered one of the major problems of numerical implementation of sliding mode techniques. Such a way of realizing chattering-free implementation of discrete-time sliding modes is also reported in Refs. [24–26].

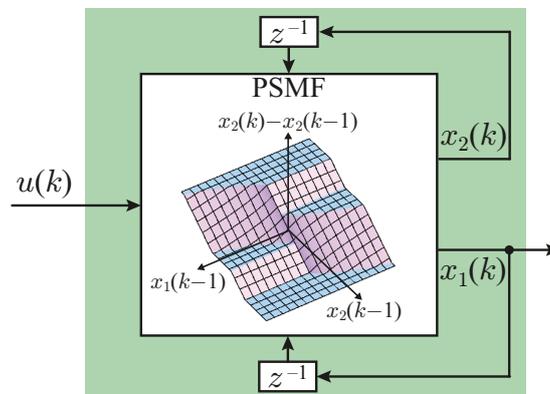


Figure 3. Block diagram of PSMF.

In Ref. [21], it is reported that the noise removing capability of PSMF is almost the same as that of 2-LPF, but its phase lag is smaller than that of 2-LPF. Moreover, compared with the filter [12,13], PSMF is less prone to overshoot, and it produces smaller phase lag. The effectiveness of PSMF has been experimentally validated [21,22].

2.2. First-Order Adaptive Windowing

In a paper [23], Janabi-Sharifi et al. proposed a discrete-time differentiator, of which the full name is “first-order adaptive windowing (FOAW)”. The basic principle of FOAW is that the window size is chosen as small for a rapid motion to instantly reflect fast varying signals, while it is chosen as large for a slow motion to obtain a smooth signal. Specifically, the complete algorithm of FOAW is as follows:

Algorithm 2 FOAW

- 1: FOR $n \in \{1, \dots, \max(k + 1, N_{\max})\}$
 - 2: $v := (u(k) - u(k - n)) / (nT)$
 - 3: FOR $i \in \{1, \dots, n\}$
 - 4: $e := u(k - i) - u(k) + iTv$
 - 5: IF $|e| > R$ RETURN $x_2(k) := v$
 - 6: END FOR
 - 7: END FOR
 - 8: RETURN $x_2(k) := v,$
-

where $u(k)$ is the input, $x_2(k)$ is the output, and $N_{\max} \in \mathbb{N}$ and $R > 0$ are parameters.

In FOAW, the current and previous input values are used for windowing, as shown in Figure 4. Specifically, the algorithm checks whether all of the points $\{u(k), u(k - 1), \dots, u(k - n - 1), u(k - n)\}$ are inside the region constructed by the two end points $u(k)$ and $u(k - n)$, and the parameter R , as shown in Figure 5. If it is the case, the window size n is further increased for providing a smoother signal. The increase of n is continued until at least one point of previous input values lies outside the region, as also illustrated in Figure 5, or n reaches its maximum value N_{\max} . Then, the last value of v obtained by the finite-time differentiation is provided as the output $x_2(k)$. In such a way, FOAW optimizes the tradeoff between the output smoothness and the suppression of delay.

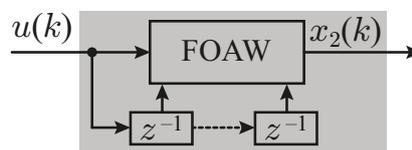


Figure 4. Block diagram of first-order adaptive windowing (FOAW).

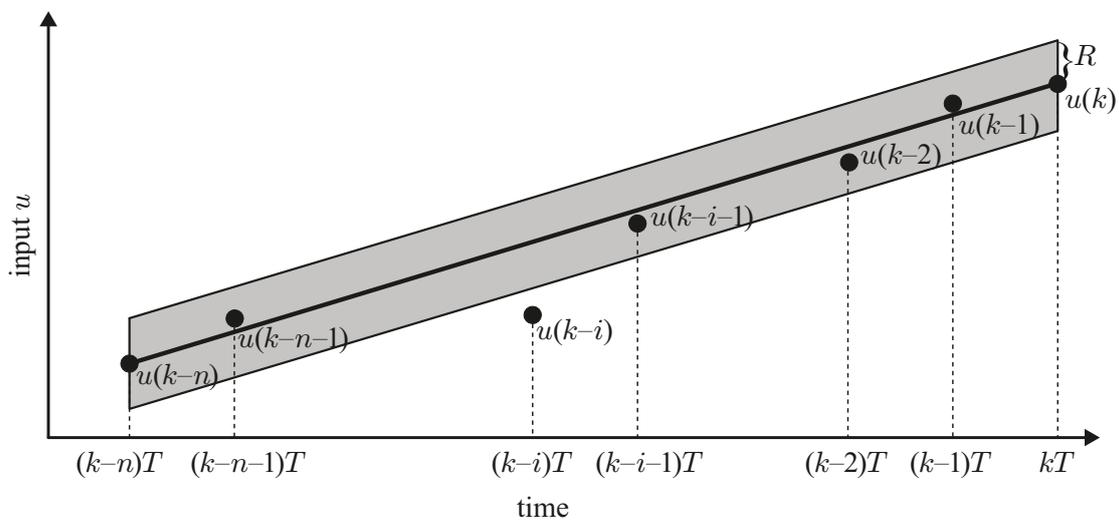


Figure 5. Windowing in FOAW.

3. Discrete-Time Sliding Mode Filter with Adaptive Gain

This section presents a new model-free discrete-time sliding mode filter, which is an extension of PSMF. The presented filter is derived through a discretization scheme inspired by the principle of FOAW.

The motivation for proposing a new filter comes from the following limitation of PSMF. That is, due to the fixed *acceleration* gain F , the output of PSMF cannot follow an input of which *acceleration* exceeds that of PSMF, whereas the output becomes sensitive to the noise contained in the input of which *acceleration* is far below that of PSMF. Towards this problem, the contribution of the presented filter is that its *acceleration* gain F adaptively changes according to the motion of input to realize balance between the output smoothness and the suppression of delay.

In order to achieve the above-mentioned balance, let us consider the following modification of PSMF. That is, for a rapid input motion, the value of F should be set large for instantly reflecting fast varying signals. On the other hand, for a slow motion, the value should be set small for obtaining smooth signals.

The remaining problem is how to adaptively determine the value of F according to the motion of the input. Towards this problem, a modified adaptive method inspired by the principle of FOAW is applied. In the light of this idea, the algorithm of an extended PSMF, which is named parabolic sliding mode filter with adaptive gain (AG-PSMF), is obtained as follows:

Algorithm 3 AG-PSMF

```

1: FOR  $n \in \{1, \dots, \min(k+1, N_{\max})\}$ 
2:    $v := (x_1(k-1) - x_1(k-n-1))/(nT)$ 
3:   FOR  $i \in \{1, \dots, n\}$ 
4:      $e := x_1(k-i-1) - x_1(k-1) + iTv$ 
5:     IF  $|e| > R$ ; EXIT LOOP
6:   END FOR
7:   IF  $i < n$ ; EXIT LOOP
8: END FOR
9:  $F^* = \frac{N_c}{n} F$ 
10:  $x_2^* := F^* T \Phi \left( \frac{u(k) - x_1(k-1)}{F^* T^2} \right)$ 
11:  $x_2(k) := \text{clip}(\text{clip}(x_2(k-1) - HF^*T, x_2(k-1) - F^*T, 0),$ 
       $\text{clip}(x_2(k-1) + F^*T, x_2(k-1) + F^*TH, 0), x_2^*)$ 
12:  $x_1(k) := x_1(k-1) + Tx_2(k)$ 
13: RETURN  $x_1(k)$  ,

```

where $N_c > 0$ is a parameter. In AG-PSMF, a modification of FOAW determines the window size through the process of steps 1–8. It should be noted here that AG-PSMF uses the previous outputs for windowing, as shown in Figures 6 and 7, while FOAW employs the current and previous inputs, as described in Section 2.2. Then, in step 9, $F^*(k)$ is obtained by applying the adaptively determined window size for balancing the tradeoff, the output smoothness and the suppression of delay. After that, in step 10, $x_2^*(k)$ is determined under the constraint of $F^*(k)$, and in step 11, $x_2(k)$ is determined in order to follow $x_2^*(k)$. Finally, in step 12, $x_1(k)$ is obtained by integrating $x_2(k)$.

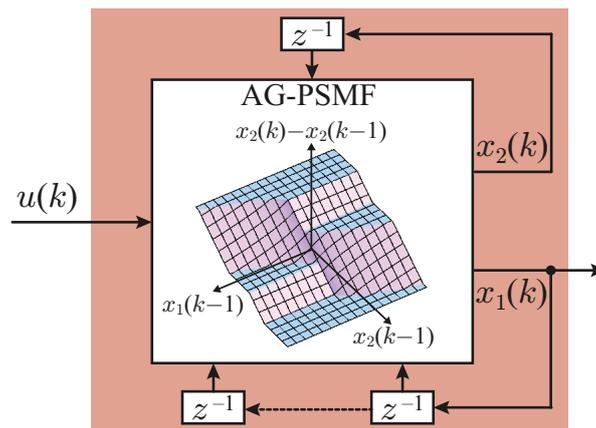


Figure 6. Block diagram of AG-PSMF.

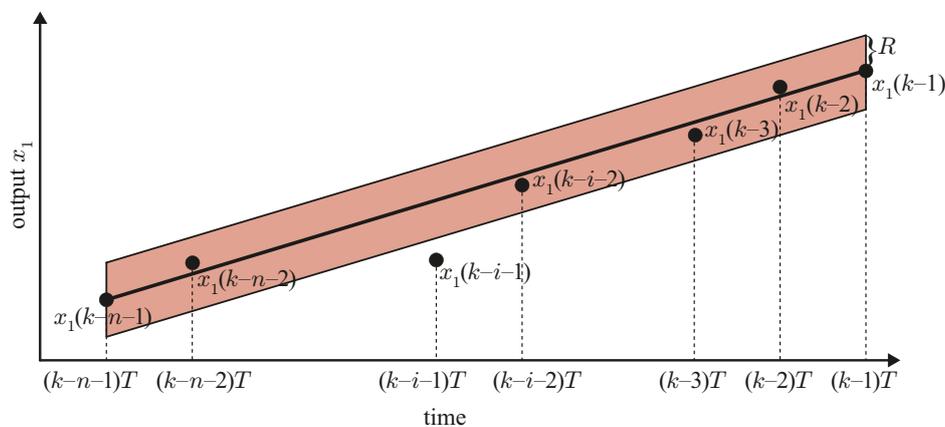


Figure 7. Windowing in AG-PSMF.

Figure 8 shows the behaviors of AG-PSMF and PSMF for the following input with step changes:

$$u = \begin{cases} 5 & \text{if } t \leq 2 \text{ s,} \\ 10 & \text{if } 2 \text{ s} < t \leq 4 \text{ s,} \\ -5 & \text{otherwise.} \end{cases} \quad (7)$$

In this case, it is shown that the convergence of AG-PSMF is faster than that of PSMF. This is because the adaptively changing gain $F^*(k)$ of AG-PSMF adjusts the shape of sliding surface according to current input and system states.

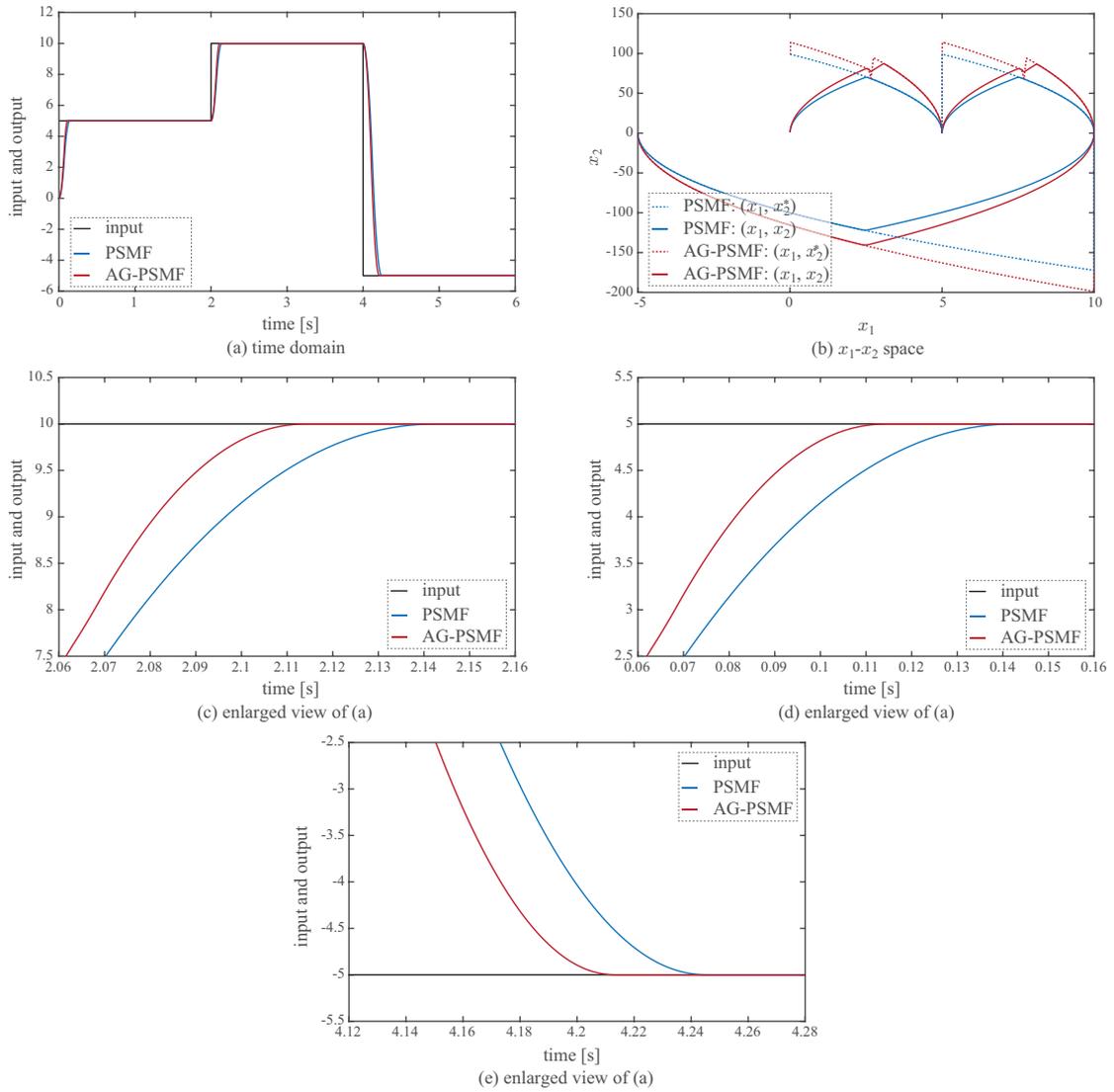


Figure 8. (a–e) Behaviors of AG-PSMF and PSMF for an input with step changes. (AG-PSMF: $F = 1000$, $H = 3$, $N_{\max} = 5$, $N_c = 4$, $R = 0.00075$; PSMF: $F = 1000$, $H = 3$; $T = 0.001$ s).

4. Numerical Examples

This section reports the performance of AG-PSMF according to the following sinusoidal input and triangular input:

$$u = \begin{cases} \sin(10t) + 0.008\epsilon(t) & \text{if } t \leq 2 \text{ s,} \\ 3\sin(10t) + 0.008\epsilon(t) & \text{if } 2 \text{ s} < t \leq 4 \text{ s,} \\ \sin(30t) + 0.008\epsilon(t) & \text{otherwise,} \end{cases} \quad (8)$$

$$u = \begin{cases} \arcsin(\sin(10t)) + 0.008\epsilon(t) & \text{if } t \leq 2 \text{ s,} \\ 2\arcsin(\sin(10t)) + 0.008\epsilon(t) & \text{if } 2 \text{ s} < t \leq 4 \text{ s,} \\ \arcsin(\sin(30t)) + 0.008\epsilon(t) & \text{otherwise,} \end{cases} \quad (9)$$

where $\epsilon \sim \mathcal{N}(0,1)$ is the unit white Gaussian noise with zero mean. Parameters $F = 1000$, $H = 3$, $N_{\max} = 5$, $N_c = 4$ and $R = 0.00075$ are used. In addition, the sampling interval $T = 0.001$ s is applied.

Figures 9 and 10 show the outputs of AG-PSMF. For comparison, the outputs of PSMF with $F = 1000$ and $H = 3$ are also included in the figures. The initial states of both filters are zeros at $t = 0$ s. The results show that both filters effectively remove the noise component for a slowly varying input, but AG-PSMF produces smaller phase lag than PSMF does, as shown in Figures 9b,c and 10b,c. For a fast varying input, i.e., the input after $t = 2$ s, the output amplitude of AG-PSMF is closer to that of signal components of inputs compared with that of PSMF, as illustrated in Figures 9d–g and 10d–g. In addition, the phase lag of AG-PSMF is smaller than that of PSMF. Such advantages of AG-PSMF is attributed to the use of adaptive gain, which optimizes the tradeoff between filtering smoothness and suppression of delay.

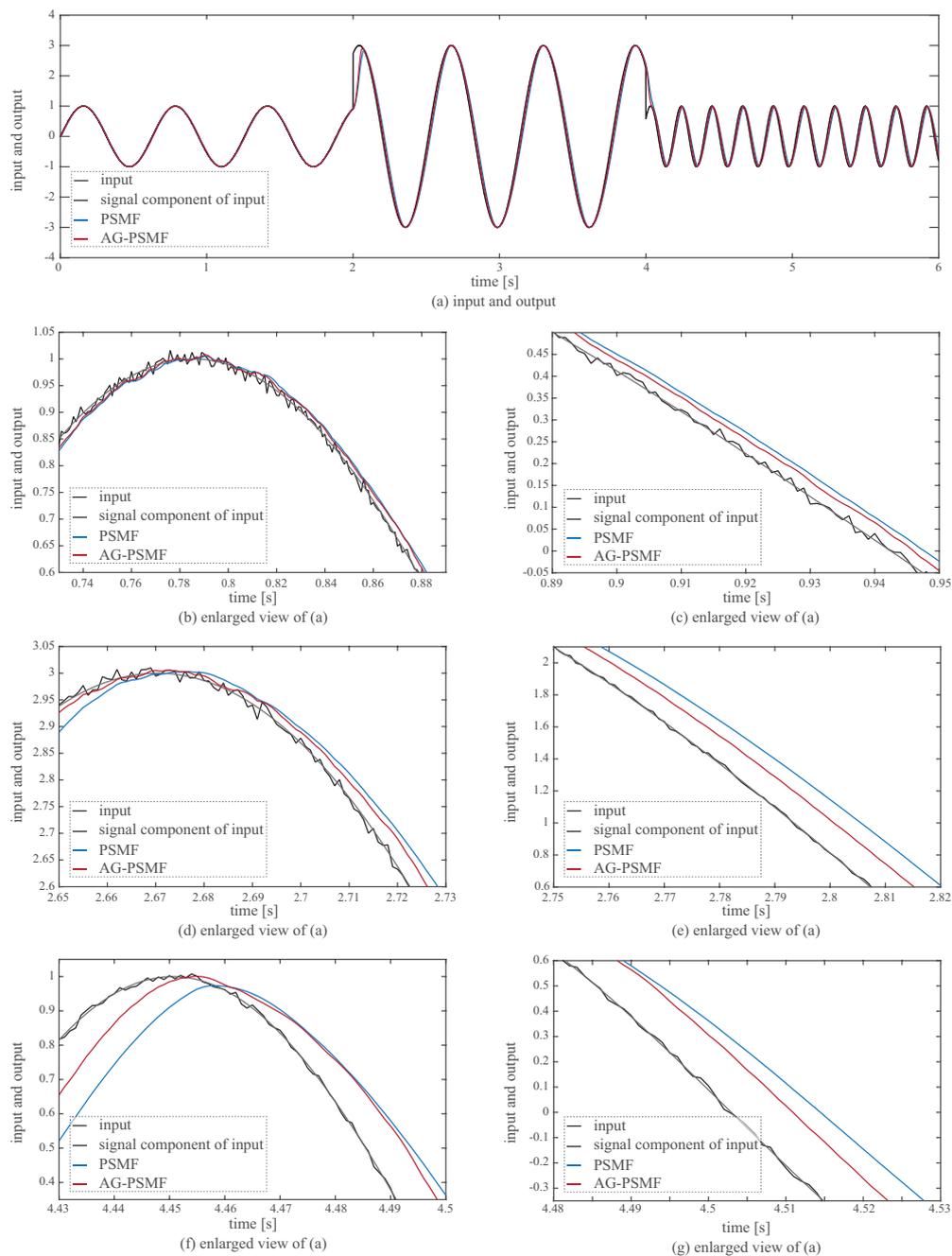


Figure 9. (a–g) Comparison between the outputs of AG-PSMF and PSMF for a sinusoidal input that is corrupted by white Gaussian noise. (AG-PSMF: $F = 1000$, $H = 3$, $N_{max} = 5$, $N_c = 4$, $R = 0.00075$; PSMF: $F = 1000$, $H = 3$; $T = 0.001$ s).

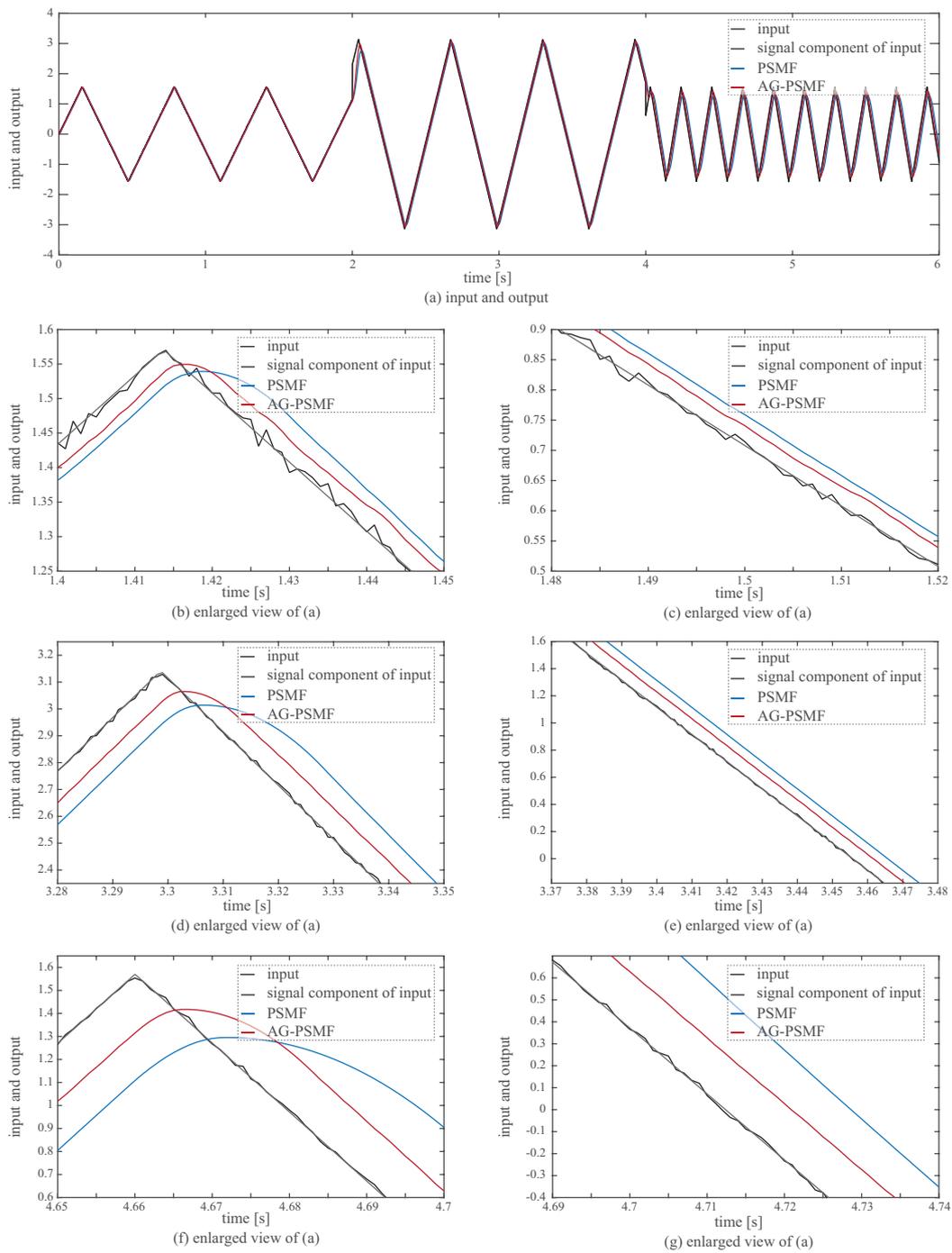


Figure 10. (a–g) Comparison between the outputs of AG-PSMF and PSMF for a triangular input that is corrupted by white Gaussian noise. (AG-PSMF: $F = 1000$, $H = 3$, $N_{\max} = 5$, $N_c = 4$, $R = 0.00075$; PSMF: $F = 1000$, $H = 3$; $T = 0.001$ s).

5. Experimental Evaluation

This section validates the effectiveness of AG-PSMF in position control. For comparison, the performances of position controls without filter and with PSMF were also evaluated. Figure 11 shows the experimental setup, which consisted of a DC motor with an optical encoder, a 16-bit motor controller and a motor driver.

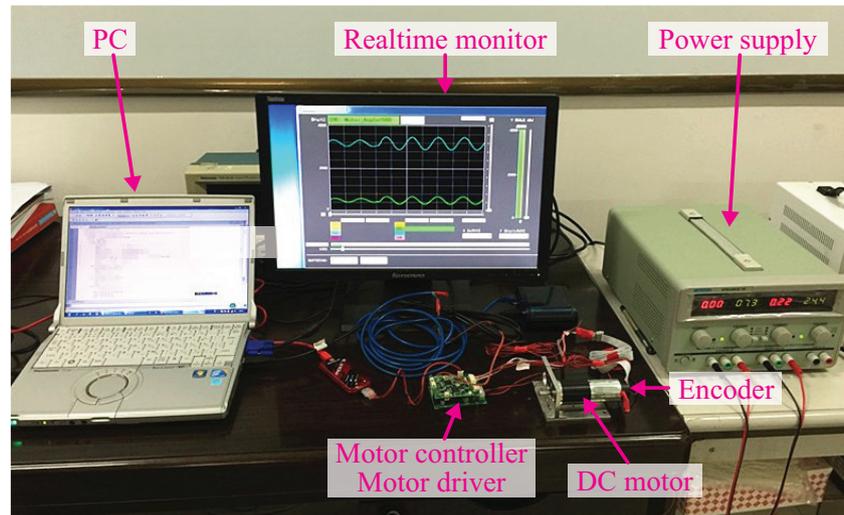


Figure 11. Experimental setup.

In the case of without a filter, the following discrete-time proportional-derivative (PD) controller was used:

$$v(k) = K_p(\theta_d(k) - \theta(k)) + K_d(\omega_d(k) - \omega(k)), \tag{10}$$

whereas the following discrete-time PD controller was employed for the cases of PSMF and AG-PSMF:

$$v(k) = K_p(\theta_d(k) - \theta_f(k)) + K_d(\omega_d(k) - \omega_f(k)). \tag{11}$$

Here, $\theta_d(k)$ is the desired trajectory, $\theta(k)$ is the current position, $\omega_d(k)$ is the desired velocity, $\omega(k)$ is the current velocity, $v(k)$ is the control voltage, $K_p > 0$ is the proportional gain, and $K_d > 0$ is the derivative gain. In addition, in the cases of PSMF and AG-PSMF, $\theta_f(k)$ is the estimated position, and $\omega_f(k)$ is the estimated velocity. Figure 12 shows the block diagrams of the PD-controlled systems used in the experiments. It should be noted that velocity signals were obtained by using the finite difference method. All experiments were implemented at sampling interval $T = 0.001$ s.

Figure 13 shows the PD-controlled results with the following desired trajectory:

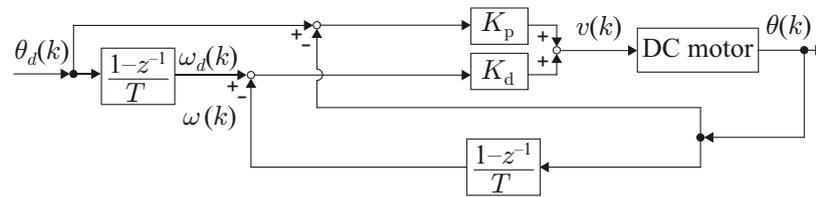
$$\theta_d(k) = \begin{cases} 90\sin(3kT) \text{ degree} & \text{if } kT \leq 6 \text{ s,} \\ 180\sin(3kT) \text{ degree} & \text{otherwise.} \end{cases} \tag{12}$$

In addition, Figure 14 illustrates the quantized position control performances through the data of the average magnitude of position error $|\theta_d - \theta|$ and the average magnitude of duty ratio (normalized control voltage v) change rate $|d'|$, which are respectively defined as follows:

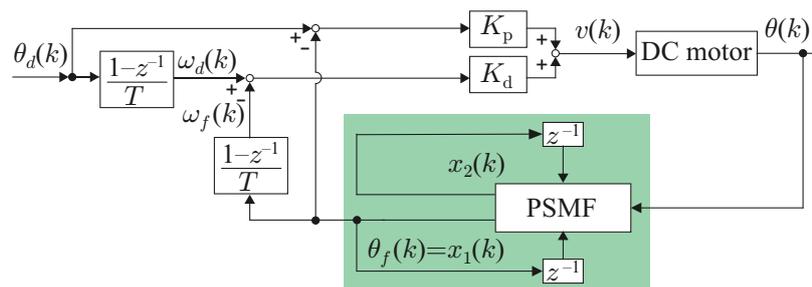
$$\text{AMP} \triangleq \frac{1}{11000 - 1000} \sum_{k=1000}^{11000} |\theta_d(k) - \theta(k)|, \tag{13}$$

$$\text{AMD} \triangleq \frac{1}{11000 - 1000} \sum_{k=1000}^{11000} \left| \frac{d(k) - d(k-1)}{T} \right|. \tag{14}$$

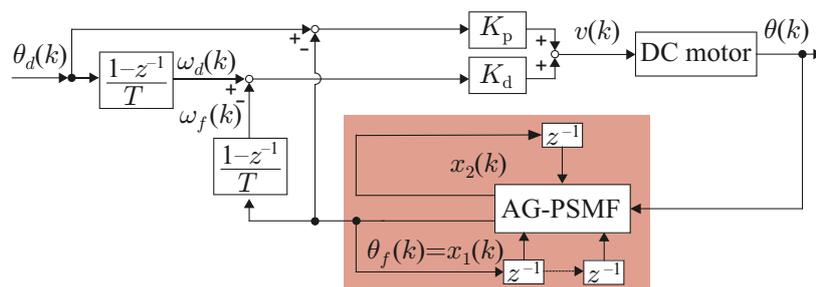
It should be mentioned that the measure AMP provides information on the controlled position, while AMD can be considered as a measure of the intensity of high-frequency vibration of the device. It should be said that, for a better position control, both AMP and AMD should be kept small.



(a) PD-controlled system without filter



(b) PD-controlled system with PSMF



(c) PD-controlled system with AG-PSMF

Figure 12. Block diagrams of PD-controlled systems. (a) without filter; (b) with PSMF; and (c) with AG-PSMF.

In the figures, one can observe that the controller (10) produced the highest AMD among the three cases. This is because finite difference produced high-frequency noise in the velocity signal by magnifying quantized encoder error contained in the position signal. As a result, there was high-frequency vibration in the device. On the other hand, the controller (11) with PSMF avoided high-frequency vibration during the period of $0 \leq t \leq 6$ s owing to the smoothed position signal. However, it resulted in instability of the device after $t > 6$ s because of the fact that, the output of PSMF cannot follow the fast varying input signal with limited *acceleration*. Compared with the above-mentioned two control schemes, the controller (11) with AG-PSMF simultaneously reduced AMP and AMD, i.e., the controlled position followed the desired trajectory effectively without producing high-frequency vibration. These results clearly show the advantage of AG-PSMF.

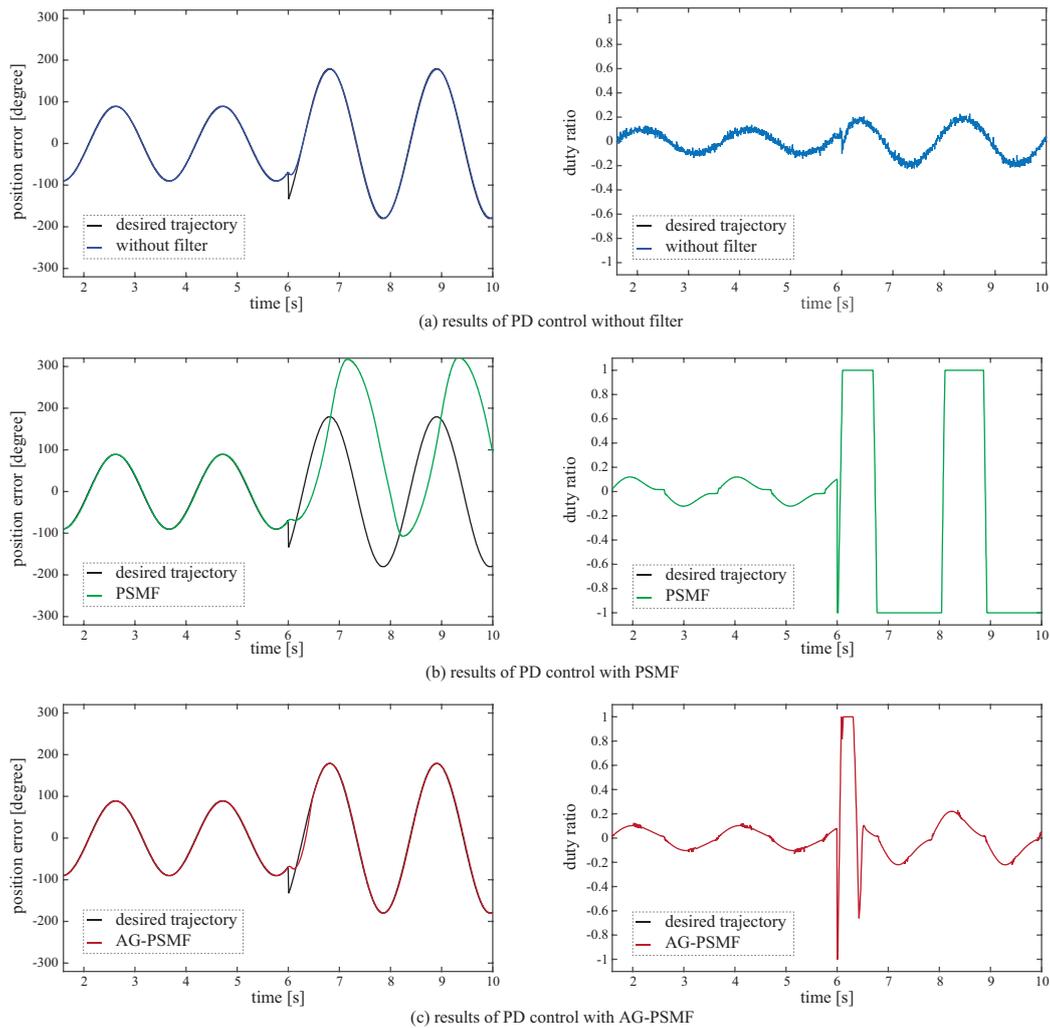


Figure 13. Results of PD control with $K_p = 100$, $K_d = 10$ and $T = 0.001$. (a) without filter (b) with PSMF ($F = 1000$, $H = 5$), and (c) with AG-PSMF ($F = 1000$, $H = 5$, $N_{\max} = 10$, $N_c = 5$, $R = 0.00008$).

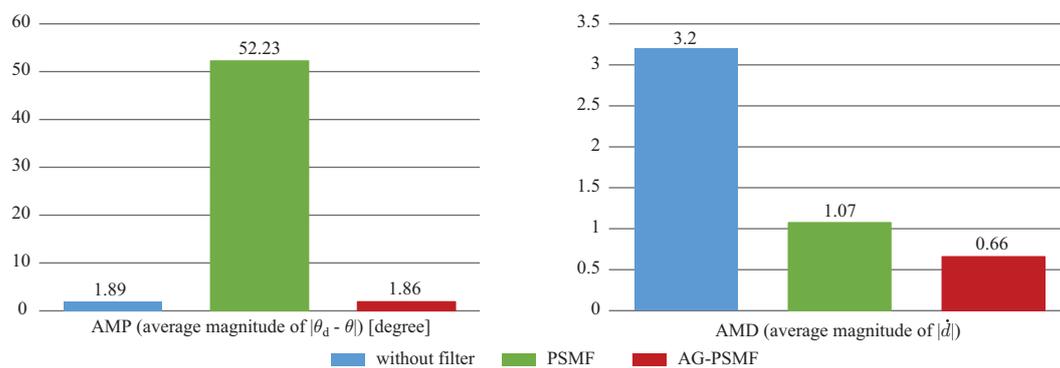


Figure 14. Position control performances achieved with $K_p = 100$, $K_d = 10$ and $T = 0.001$. (a) without filter (b) with PSMF ($F = 1000$, $H = 5$), and (c) with AG-PSMF ($F = 1000$, $H = 5$, $N_{\max} = 10$, $N_c = 5$, $R = 0.00008$).

6. Conclusions

This paper has presented a new model-free discrete-time sliding mode filter, which is named AG-PSMF, for removing noise by balancing the tradeoff between the filtering smoothness and the suppression of delay. The presented AG-PSMF is an extension of a previous parabolic sliding mode filter (PSMF) by including an adaptive gain, of which value is determined by an adaptive method inspired by FOAW. The effectiveness of AG-PSMF is validated through numerical examples and experiments.

One issue that is not addressed here but remains as a future study is to develop guidelines for selecting parameter values. In addition, the effectiveness of AG-PSMF is only validated through numerical methods, and thus theoretical validation should be conducted in future study.

Acknowledgments: This work was supported by Yanbian University Research Fund No. 413010089.

Author Contributions: Shanhai Jin conceived and wrote the paper; Shanhai Jin, Yonggao Jin and Xiaodan Wang performed simulations and experiments; Xiaogang Xiong offered useful suggestions for the preparation and writing the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2-LPF	Second-order Butterworth low-pass filter
PSMF	Parabolic sliding mode filter
FOAW	First-order adaptive windowing
AG-PSMF	Parabolic sliding mode filter with adaptive gain
u	Input of PSMF and AG-PSMF
x_1	Output of PSMF and AG-PSMF
x_2 and \dot{x}_2	Derivatives of x_1 and x_2 , respectively
F	Acceleration gain of PSMF and AG-PSMF
d	Duty ratio
θ_d	Desired trajectory
θ	Current position
ω_d	Desired velocity
ω	Current velocity
θ_f	Estimated position
ω_f	Estimated velocity
v	Control voltage
K_p	Proportional gain
K_d	Derivative gain
AMP	Average magnitude of position error $ \theta_d - \theta $
AMD	Average magnitude of the duty-ratio (normalized control voltage v) change rate $ \dot{d} $

References

1. Gallagher, N.; Gary, L. A theoretical analysis of the properties of median filters. *IEEE Trans. Acoust. Speech Signal Proc.* **1981**, *29*, 1136–1141.
2. Moshnyaga, V.; Hashimoto, K. An efficient implementation of 1-D median filter. In Proceedings of the 52nd IEEE International Midwest Symposium on Circuits and Systems, Cancun, Mexico, 2–5 August 2009; pp. 451–454.
3. Hargrave P. A tutorial introduction to Kalman filtering. In Proceedings of the IEE Colloquium on Kalman Filters: Introduction, Applications and Future Developments, London, UK, 21 February 1989; pp. 1–6.
4. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Department of Computer Science, Technical Report TR; University of North Carolina: Chapel Hill, NC, USA, 1995.
5. Levant, A. Robust exact differentiation via sliding mode technique. *Automatica* **1998**, *34*, 379–384.

6. Levant, A. Higher-order sliding modes, differentiation and output-feedback control. *Int. J. Control* **2003**, *76*, 924–941.
7. Davila, J.; Fridman, L.; Levant, A. Second-order sliding-mode observer for mechanical systems. *IEEE Trans. Autom. Control* **2005**, *50*, 1785–1789.
8. M'Sirdi, N.; Rabhi, A.; Fridman, L.; Davila, J.; Dalanne, Y. Second order sliding mode observer for estimation of velocities, wheel sleep, radius and stiffness. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; pp. 3316–3321.
9. Moreno, J.; Osorio, M. A Lyapunov approach to second-order sliding mode controllers and observers. In Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 2856–2861.
10. Solvar, S.; Le, V.; Ghanes, M.; Barbot, J.; Santomenna, G. Sensorless second order sliding mode observer for induction motor. In Proceedings of the 2010 IEEE International Conference on Control Applications, Yokohama, Japan, 8–10 September 2010; pp. 1933–1938.
11. M'Sirdi, N.; Rabhi, A.J.; Fridman, L.; Davila, J.; Delanne, Y. Second-order sliding-mode observer for estimation of vehicle dynamic parameters. *Int. J. Veh. Des.* **2008**, *48*, 190–207.
12. Han, J.; Wang, W. Nonlinear tracking-differentiator. *J. Syst. Sci. d Math. Sci.* **1994**, *14*, 177–183. (In Chinese)
13. Emaru, T.; Tsuchiya, T. Research on estimating smoothed value and differential value by using sliding mode system. *IEEE Trans. Robot. Autom.* **2003**, *19*, 391–402.
14. Han, J.; Huang, Y. Frequency characteristic of second-order tracking-differentiator. *Math. Pract. Theory* **2003**, *33*, 71–74. (In Chinese)
15. Emaru, T.; Tsuchiya, T. Research on signal tracking performance of nonlinear digital filter ESDS. *Trans. Japan Soc. Mech. Eng. Ser. C* **2004**, *70*, 2352–2359. (In Japanese)
16. Guo, B.; Zhao, Z. On convergence of tracking differentiator. *International Journal of Control* **2011**, *84*, 693–701.
17. Su, Y.; Duan, B.; Zheng, C. Nonlinear PID control of a six-DOF parallel manipulator. *IEE Proc. Control Theory Appl.* **2004**, *151*, 95–102.
18. Su, Y.; Sun, D.; Duan, B. Design of an enhanced nonlinear PID controller. *Mechatronics* **2005**, *15*, 1005–1024.
19. Peng, X.; Cheng, S.; Wen, J. Application of nonlinear PID controller in superconducting magnetic energy storage. *Int. J. Control Autom. Syst.* **2005**, *3*, 296–301.
20. Jin, S.; Kikuuwe, R.; Yamamoto, M. Real-time quadratic sliding mode filter for removing noise. *Adv. Robot.* **2012**, *26*, 877–896.
21. Jin, S.; Kikuuwe, R.; Yamamoto, M. Parameter selection guidelines for a parabolic sliding mode filter based on frequency and time domain characteristics. *J. Control Sci. Eng.* **2012**, *2012*, 923679.
22. Jin, S.; Kikuuwe, R.; Yamamoto, M. Improved velocity feedback for position control by using a quadratic sliding mode filter. In Proceedings of 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Korea, 26–29 October 2011; pp. 1207–1212.
23. Janabi-Sharifi, F.; Hayward, V.; Chen, C. Discrete-time adaptive windowing for velocity estimation. *IEEE Trans. Control Syst. Technol.* **2000**, *8*, 1003–1009.
24. Acary, V.; Brogliato, B. Implicit euler numerical scheme and chattering-free implementation of sliding mode systems. *Syst. Control Lett.* **2010**, *59*, 284–293.
25. Kikuuwe, R.; Yasukouchi, S.; Fujimoto, H.; Yamamoto, M. Proxy-based sliding mode control: A safer extension of PID position control. *IEEE Trans. Robot.* **2010**, *26*, 860–873.
26. Acary, V.; Brogliato, B.; Orlov, Y. Chattering-free digital sliding-mode control with state observer and disturbance rejection. *IEEE Trans. Autom. Control* **2012**, *57*, 1087–1101.

