A Realtime Quadratic Sliding Mode Filter for Removing Noise

* Shanhai Jin, Ryo Kikuuwe and Motoji Yamamoto (Kyushu Univ.)

jin@ctrl.mech.kyushu-u.ac.jp
{*kikuuwe, yama*}*@mech.kyushu-u.ac.jp*

This paper presents a sliding mode filter that effectively removes impulsive noise and high-frequency noise with producing much smaller phase lag than linear filters. It is less prone to overshoot than previous sliding mode filters and it does not produce chattering. In addition, it is computationally inexpensive, and thus suitable for realtime applications. The proposed sliding mode filter employs a quadratic surface as its sliding surface, which is designed so that the output converges to the input in finite time if the input value is constant. Its algorithm is derived by using the backward Euler discretization, which prevents chattering. The effectiveness of the filter was shown by experiments by using an ultrasonic sensor and an optical encoder.

Key Words: Quadratic Sliding Mode Filter, Finite Time Convergence, Backward Euler Discretization

1. Intorduction

In many robotics applications, sensor signals are corrupted by high-frequency noise. The use of a linear filter is often the first choice for removing such noise because of its simplicity, but it is also known to have some drawbacks. One is that a linear filter proportionally transfers any noise component into the output and thus it cannot remove high-amplitude impulses such as those considered as outliers. Another drawback is that it produces a phase lag in the output and thus the original shape of the input is distorted. These problems cannot be ignored in such cases where the noise has high amplitude (i.e., it is impulsive) and where the frequency range of the original signal is slightly below that of the noise signal. For example, the distance measurement obtained through an ultrasonic sensor often contains impulsive noise. As another example, the velocity signal obtained through the numerical differentiation of the position reading from an optical encoder is corrupted by high-frequency noise, while instantaneous velocity information is demanded for injecting damping into a position-controlled mechatronic system.

Some classes of nonlinear filters have been studied in order to avoid drawbacks of linear filters. For example, the median filter [1] is known to be useful for removing impulsive noise, but its computational cost is high. An adaptive windowing filter [2], as another example, is used for removing high-frequency noise caused by numerical differentiation, but its window size should not be too large for preventing unacceptable computation cost.

Some researchers study sliding mode observers for filtering. One of the major problems raised in implementing sliding mode techniques is chattering, which is high-frequency oscillation in the output value. It is known that sliding mode observers based on the super-twisting algorithm [3] do not produce chattering and do realize finite time convergence. They are, however, prone to overshoot during the convergence.

The use of a quadratic sliding surface has also been studied in the field of filtering [4][5]. One of its advantages is that, by using it, the output converges to the input in finite time when the input is constant. In particular, Emaru and Tsuchiya [5] named their quadratic sliding mode filter as "ESDS"[1] and they used it for removing impulsive noise. A problem of their filter is that it is prone to overshoot. Another problem is that the numerical error caused by their discrete-time implementation [6] produces chattering, as will be demonstrated in section 2.2.

[1]According to Emaru and Tsuchiya [5], the full form of ESDS is "the system which estimate the smoothed value and the differential value by using sliding mode system".

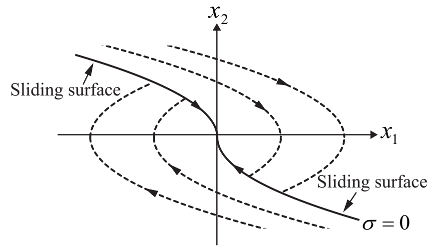


Fig. 1: Quadratic sliding surface (solid curve) and trajectories of the state (x_1, x_2) for $\sigma \neq 0$ (dashed curves) in the filter (1) with $u = 0$

The rest of this paper is organized as follows. Section 2 discusses previous work on quadratic sliding mode filters and clarifies their problems. Section 3 presents a new quadratic sliding mode filter, which performs better than previous methods. In section 4, experimental results are shown to demonstrate the advantages of the proposed filter, and in section 5 conclusions are drawn.

2. Quadratic sliding mode filters

2.1 Continuous-time expression

Let us consider the system described in the following expression:

$$\dot{x}_1 = x_2 \tag{1a}$$
$$\dot{x}_2 = -F\mathrm{sgn}(\sigma) \tag{1b}$$
$$\sigma = 2F(x_1 - u) + |x_2|x_2 \tag{1c}$$

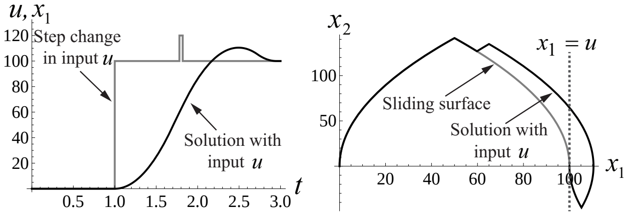
where u is an input, (x_1, x_2) is the system state, F is a positive constant, and

$$\mathrm{sgn}(z) \begin{cases} = \ \ 1 & \text{if } z > 0, \\ \in [-1, 1] & \text{if } z = 0, \\ = -1 & \text{if } z < 0. \end{cases} \tag{2}$$

This system can act as a filter with the input u and the output x_1. Han and Wang [4] used this filter for removing white noise contained in u, and Emaru and Tsuchiya [5] used it to remove impulsive noise contained in u. The sliding surface of the filter (1) is a quadratic surface, which is the set of states (x_1, x_2) that satisfy the quadratic equation $2F(x_1 - u) + |x_2|x_2 = 0$, as illustrated by the solid curve in Fig. 1.

In the filter (1), when $\sigma \neq 0$, i.e., in reaching mode, $\dot{x}_2$ takes the value of either $-F$ or F. In this period, the state (x_1, x_2) moves along a quadratic curve in the state space, as illustrated by dashed curves in Fig. 1. When $\sigma = 0 \wedge \dot{\sigma} = 0$, we have

$$2F(x_2 - \dot{u}) + 2|x_2|\dot{x}_2 = 0. \tag{3}$$



(a) Time domain (b) x_1-x_2 space

Fig. 2: Analytical solution of (1) with a step change in u hat is temporarily corrupted by a disturbance ($F = 200$)

In this case, because of $|\dot{x}_2| \leq F$, $\dot{u}$ satisfies the following condition:

$$\min(2x_2, 0) \leq \dot{u} \leq \max(2x_2, 0) \tag{4}$$

and $\dot{x}_2$ takes the following value:

$$\dot{x}_2 = -F\mathrm{sgn}(x_2)(1 - \dot{u}/x_2). \tag{5}$$

Also in this case, $x_2 = 0$ means $\dot{u} = 0$, and thus $\dot{x}_2$ takes a value between $-F$ and F. When $\sigma = 0$ and $\dot{u}$ does not satisfy (4), $\dot{x}_2$ takes either $-F$ or F. In conclusion, $\dot{x}_2$ of the filter (1) satisfies

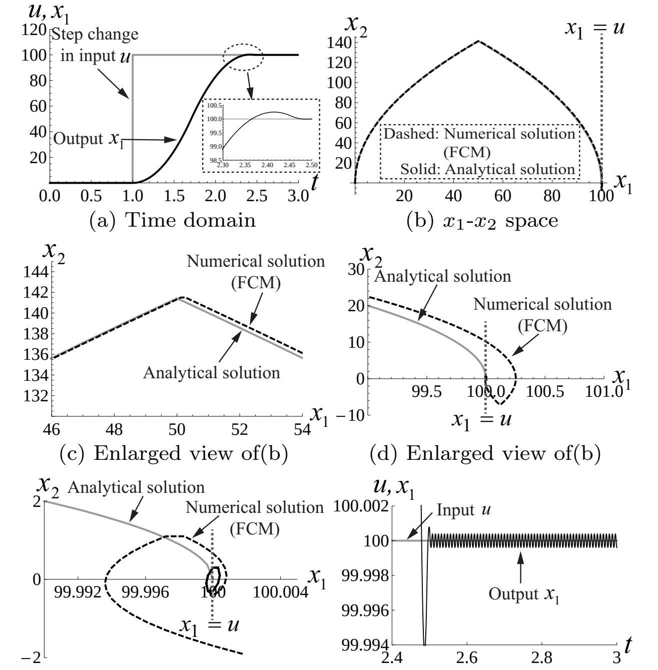
$$\dot{x}_2 \begin{cases} = F & \text{if } \sigma < 0 \lor (\sigma = 0 \land \dot{u} > \max(2x_2, 0)), \\ = -F & \text{if } \sigma > 0 \lor (\sigma = 0 \land \dot{u} < \min(2x_2, 0)), \\ \in [-F, F] & \text{if } \sigma = 0 \land x_2 = 0 \land \dot{u} = 0, \\ = -F\mathrm{sgn}(x_2)(1 - \dot{u}/x_2) & \text{otherwise.} \end{cases} \tag{6}$$

It is worth noting that, when $\sigma \neq 0 \lor (\sigma = 0 \land (\dot{u} > \max(2x_2, 0) \lor \dot{u} < \min(2x_2, 0) \lor (x_2 \neq 0 \land \dot{u} = 0))$, $\dot{x}_2$ takes either $-F$ or F, according to (6). The effect of such using extreme values is equivalent to that of a bang-bang control [7], which drives the state (x_1, x_2) from arbitrary initial states to the target state $(u, 0)$ in the minimum time under the constraint $|\dot{x}_2| \leq F$.

A problem of the filter (1) is that it is prone to overshoot. Fig. 2 shows the analytical solution of (1) with a step change in u that is temporarily corrupted by a disturbance. As the black curve shown in Fig. 2(b), the state (x_1, x_2) deviates from the sliding surface $\sigma = 0 \land x_2 > 0$ into the region $\sigma > 0 \land x_2 > 0$ by the influence of the disturbance. After the disturbance disappears, the deviated state moves in parallel to the sliding surface. This is because, in the region $\sigma > 0 \land x_2 > 0$, $\dot{x}_2$ takes the same value with that takes on the sliding surface $\sigma = 0 \land x_2 > 0$ (i.e., $\dot{x}_2 = -F$). Thus, the state returns to the sliding surface after an overshoot, i.e., after crossing the line $x_1 = u$. As a whole, if the state is in the region $\sigma x_2 > 0$, it cannot reach the sliding surface before crossing the line $x_1 = u$.

2.2 Discrete-time algorithm

According to Emaru *et al.* [6], they originally implemented the filter (1) by using 4th-order Runge-Kutta method as the discrete-time integrator. To make the computation fast, they further proposed another integration method, which they call "fast calculating method (FCM)" [6]. As to the authors' knowledge, these two are only available discrete-time algorithms of the filter (1) in the literature. A problem of FCM is that the state (x_1, x_2) cannot exactly reach the sliding surface $\sigma = 0$ due to numerical errors, and thus there occurs chattering. Fig. 3 shows the numerical solution of (1) with a step change in u obtained by FCM. We can observe that the state (x_1, x_2) slightly crosses the sliding surface and thus there occurs small overshoot, as shown in Fig. 3(c) and Fig. 3(d). Because the state (x_1, x_2) never exactly reaches the sliding surface, chattering continues around $x_1 = u$, as shown in Fig. 3(e) and Fig. 3(f).



(a) Time domain (b) x_1-x_2 space

(c) Enlarged view of(b) (d) Enlarged view of(b)

(e) Enlarged view of(b) (f) Enlarged view of(a)

Fig. 3: Numerical solution of (1) with a step change in u obtained by FCM ($F = 200$, $T = 0.001$ s)

3. Proposed Filter

3.1 Continuous-time expression

This paper presents the following sliding mode filter, which is less prone to overshoot than previous methods:

$$\dot{x}_1 = x_2 \tag{7a}$$

$$\dot{x}_2 = -F\mathrm{gsgn}(\mathrm{gsgn}(-\alpha, x_2, -1), \sigma, \mathrm{gsgn}(1, x_2, \alpha)) \tag{7b}$$

$$\sigma = 2F(x_1 - u) + |x_2|x_2 \tag{7c}$$

where $\alpha > 1$ is a constant, F is a positive constant. In addition, gsgn is a generalized signum function, which is defined as follows:

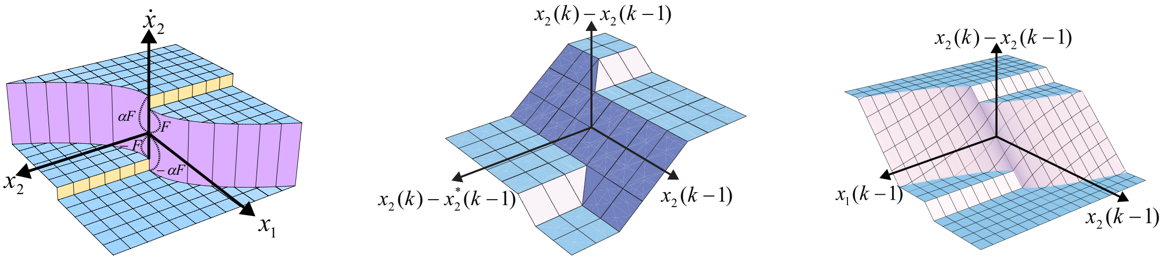
$$\mathrm{gsgn}(A, z, B) = \begin{cases} B & \text{if } z > 0, \\ [\min(A), \max(B)] & \text{if } z = 0, \\ A & \text{if } z < 0 \end{cases} \tag{8}$$

where $A \subset \mathbb{R}$ and $B \subset \mathbb{R}$ are closed intervals, $z \in \mathbb{R}$, $\min(A) \leq \max(B)$. Fig. 4(a) shows the relation between x_1, x_2 and $\dot{x}_2$. Note that $\dot{x}_2$ takes either $-\alpha F$ or αF in the region $\sigma x_2 > 0$.

Fig. 5 shows the analysis solution of (7) with a step change in u that is temporarily corrupted by a disturbance. We can observe that the state (x_1, x_2) deviates from the sliding surface into the region $\sigma x_2 > 0$ by the influence of the disturbance. According to (7), the value of $|\dot{x}_2|$ of (7) is larger than that of (1) in this region. Thus in this case, differently from (1), the deviated state returns to the sliding surface before passing the line $x_1 = u$, and there is no overshoot.

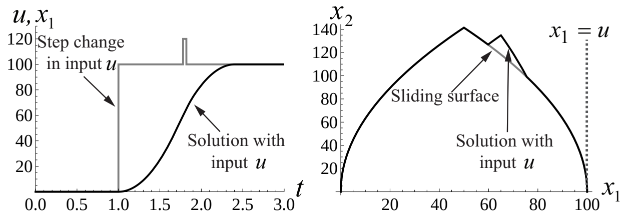
3.2 Discrete-time algorithm

In this subsection, we derive the discrete-time algorithm of the filter (7) by using the backward Euler discretization, which prevents chattering.



(a) (b) (c)

Fig. 4: **(a)** Relation between x_1, x_2 and $\dot{x}_2$; **(b)** Relation between $x_2(k-1)-x_2^*(k)$, $x_2(k-1)$ and $x_2(k)-x_2(k-1)$; **(c)** Relation between $x_1(k-1)$, $x_2(k-1)$ and $x_2(k)-x_2(k-1)$



(a) Time domain (b) x_1-x_2 space

Fig. 5: Analytical solution of (7) with a step change in u that is temporarily corrupted by a disturbance ($F = 200$, $\alpha = 2$)

Based on the backward Euler discretization, (7) can be approximated as follows:

$$x_2(k) - x_2(k-1) = -\mathrm{gsgn}(m_1(k),\ \sigma(k), m_2(k)) \quad (9a)$$

$$\sigma(k) = |x_2(k)|x_2(k) + 2F(Tx_2(k) + x_1(k-1) - u(k)) \quad (9b)$$

where $m_1(k)$ and $m_2(k)$ are determined as:

$$m_1(k) = \mathrm{gsgn}(-\alpha TF,\ x_2(k), -TF) \quad (10)$$

$$m_2(k) = \mathrm{gsgn}(TF,\ x_2(k),\ \alpha TF) \quad (11)$$

In (9), the unknown $x_2(k)$ should be determined so that it satisfies (9). The solution of (9) with respect to $x_2(k)$ can be obtained through the following procedures.

By setting $\sigma(k) = 0$, we can obtain $x_2^*(k)$, which is the value of $x_2(k)$ that satisfies $\sigma(k) = 0$, as follows:

$$x_2^*(k) = \mathrm{sgn}(x_1(k-1) - u(k))(FT - \sqrt{F^2T^2 + 2F|x_1(k-1) - u(k)|}). \quad (12)$$

Because $\sigma(k)$ is a monotonously increasing function with respect to $x_2(k)$, by using (12), (9) can be rewritten as follows:

$$x_2(k) - x_2(k-1) = -\mathrm{gsgn}(m_1(k), x_2(k) - x_2^*(k), m_2(k)) \quad (13)$$

In order to move out unknown $x_2(k)$ from the right-hand side of (13), we apply the following equivalent relation between two equations [8]:

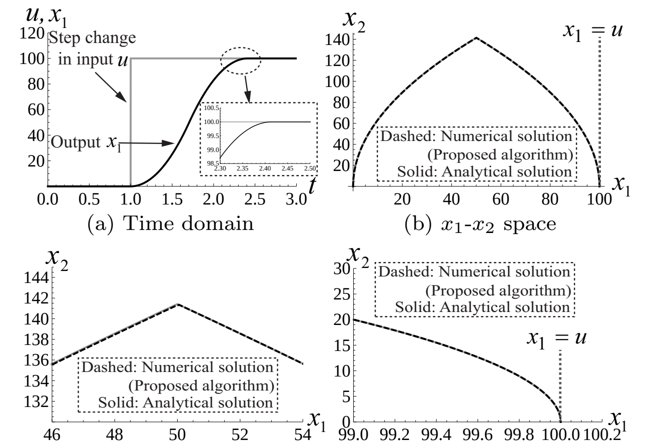
$$z - y = \mathrm{gsgn}(\mathrm{gsgn}(a, -z, b), x - z, \mathrm{gsgn}(c, -z, d)) \Leftrightarrow$$
$$z - y = \mathrm{gsat}(\mathrm{gsat}(a, -y, b), x - y, \mathrm{gsat}(c, -y, d)) \quad (14)$$

where $a \le b \le c \le d$, and gsat is a generalized saturation function [8], which is defined as follows:

$$\mathrm{gsat}(a, z, b) = \begin{cases} b & \text{if } z > b, \\ z & \text{if } z \in [a, b], \\ a & \text{if } z < a, \end{cases} \quad (15)$$

given $a \le b$. Then, we can obtain

$$x_2(k) - x_2(k-1) = -\mathrm{gsat}(n_1(k), n_0(k), n_2(k)) \quad (16)$$



(a) Time domain (b) x_1-x_2 space

(c) Enlarged view of(b) (d) Enlarged view of(b)

Fig. 6: Numerical solution of the proposed filter with a step change in u ($F = 200$, $\alpha = 2$, $T = 0.001$ s)

where $n_0(k)$, $n_1(k)$ and $n_2(k)$ are determined as:

$$n_0(k) = x_2(k-1) - x_2^*(k) \quad (17)$$

$$n_1(k) = \mathrm{gsat}(-\alpha TF,\ x_2(k-1), -TF) \quad (18)$$

$$n_2(k) = \mathrm{gsat}(TF,\ x_2(k-1),\ \alpha TF). \quad (19)$$

Fig. 4(b) shows the relation between $x_2(k-1) - x_2^*(k)$, $x_2(k-1)$ and $x_2(k) - x_2(k-1)$. Then, we can obtain $x_2(k)$ from (16), which yields

$$x_2(k) = x_2(k-1) - \mathrm{gsat}(n_1(k), n_0(k), n_2(k)). \quad (20)$$

In conclusion, the complete discrete-time algorithm of the proposed filter is obtained as follows:

$$x_2^*(k) = \mathrm{sgn}(x_1(k-1) - u(k))(FT - \sqrt{F^2T^2 + 2F|x_1(k-1) - u(k)|}) \quad (21a)$$

$$n_0(k) = x_2(k-1) - x_2^*(k) \quad (21b)$$

$$n_1(k) = \mathrm{gsat}(-\alpha TF,\ x_2(k-1), -TF) \quad (21c)$$

$$n_2(k) = \mathrm{gsat}(TF,\ x_2(k-1),\ \alpha TF). \quad (21d)$$

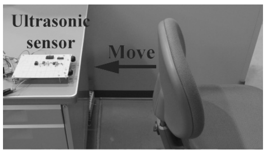
$$x_2(k) = x_2(k-1) - \mathrm{gsat}(n_1(k), n_0(k), n_2(k)) \quad (21e)$$

$$x_1(k) = Tx_2(k) + x_1(k-1). \quad (21f)$$

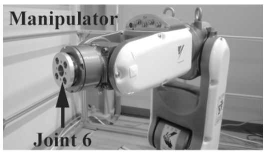
Fig. 4(c) shows the relation between $x_1(k-1)$, $x_2(k-1)$ and $x_2(k) - x_2(k-1)$, and Fig. 6 shows the numerical solution of the proposed filter with a step change in u. Note that there is no chattering in sliding mode. Such a way of avoiding chattering by using the backward Euler discretization is also reported in [8].

4. Experiment

The proposed filter was experimentally tested by using an ultrasonic sensor and an optical encoder. In both

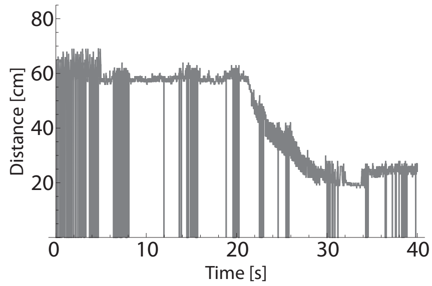


(a) Ultrasonic sensor

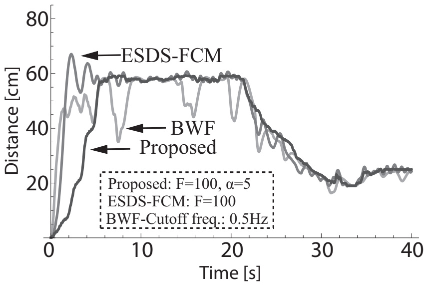


(b) Optical encoder

Fig. 7: Experiment setups



(a) Distance measurement



(b) Filtered results

Fig. 8: Experiment by using an ultrasonic sensor ($T = 0.01$s)

experiments, the parameter α of the proposed filter was chosen by try-and-error to realize as effective filtering performance as possible. The filtered output of the proposed filter was compared with those of the second-order Butterworth low-pass filter (BWF) and ESDS implemented with FCM (ESDS-FCM).

4.1 Ultrasonic sensor

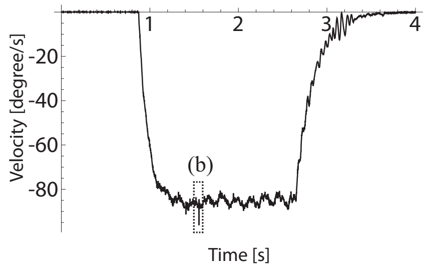
In this experiment, an ultrasonic sensor was fixed on a desk, and the distance between the ultrasonic sensor and the arc-shaped back of a chair was measured as shown in Fig. 7(a). First, the chair was in its initial position, and then the chair was moved toward the desk slowly. If the receiver did not receive the reflected wave within the sampling time of $T = 0.01$ s, the measured distance was recorded as 0 cm.

Fig. 8(a) shows the obtained distance signal, which was corrupted by impulsive noise, and Fig. 8(b) shows the outputs of the three filters. One can observe that BWF failed in removing impoulsive noise. ESDS-FCM produced overshoot, and its output was vibratory. Compared with the above two filters, the proposed filter removed impulsive noise.

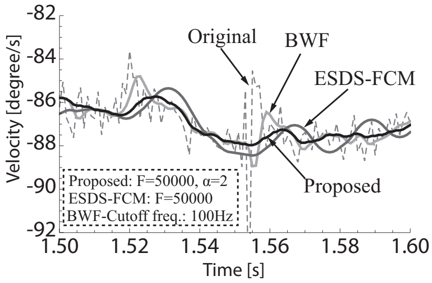
4.2 Optical Encoder

In this experiment, the joint velocity signal was obtained through the numerical differentiation of the angle reading from an optical encoder attached to the sixth joint of a manipulator, as shown in Fig. 7(b).

Fig.9 (a) shows the velocity signal obtained from the angle signal. One can observe that the velocity signal was corrupted by high-frequency noise and there was an impulse at $t = 1.554$ s. Fig. 9(b) shows that BWF smoothed high-frequency noise but failed in smoothing the impulse.



(a) Distance measurement



(b) Filtered results

Fig. 9: Experiment by using an optical encoder of a manipulator ($T = 0.001$s)

ESDS-FCM smoothed the impulse but produced vibration. The proposed filter succeeded in smoothing both the impulse and high-frequency noise.

5. Conclusion

In this paper, we have presented a quadratic sliding mode filter that effectively removes impulsive noise and high-frequency noise. The proposed filter does not produce chattering, and it is less prone to overshoot than previous sliding mode filters. In addition, its algorithm is computationally inexpensive, and thus suitable for real-time applications. The experimental results showed the effectiveness the proposed filter than prior methods.

References

[1] N. J. Gallagher and L. W. Gary, "A theoretical Analysis of the Properties of Median Filters," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 29, no.6, pp. 1136 - 1141, 1981.

[2] F. Janabi-Sharifi, V. Hayward, and C-S. J. Chen, "Discrete-Time Adaptive Windowing for Velocity Estimation," *IEEE Transactions on Control Systems Technology,* vol. 8, no. 6, pp. 1003-1009 , 2000.

[3] A. Levant, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379-384, 1998.

[4] J. Q. Han and W. Wang, "Nonlinear tracking-differentiator," *Journal of Systems Science and Mathematical Science*, vol. 14, no. 2, pp. 177-183, 1994. (in Chinese)

[5] T. Emaru and T. Tsuchiya, "Research on Estimating Smoothed Value and Differential Value by Using Sliding Mode System," *IEEE Transactions on Robotics and Automation,* vol. 19, no. 3, pp. 391-402, 2003.

[6] T. Emaru, T. Tsuchiya and K. Tanaka "Fast Calculating Method for removing the Impulsive Noise," *Transactions of the Japan Society of Mechanical Engineers. C,* vol. 70, no. 689, pp. 62-68, 2004. (in Japanese)

[7] D.E. Kirk. "The Variational Approach to Optimal Control Problems," In: *Optimal Control Theory,* R. W. Newcomb, New Jersey:Prentice-Hall, 1970, pp. 249-256.

[8] R. Kikuuwe, N. Takesue and H. Fujimoto, "A Control Framework to Generate Nonenergy-Storing Virtual Fixtures: Use of Simulated Plasticity," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 781-793, 2008.