## A Realtime Quadratic Sliding Mode Filter for Removing Noise

Shanhai Jin, Ryo Kikuuwe and Motoji Yamamoto

Department of Mechanical Engineering, Kyushu University Motooka 744, Nishi-ku, Fukuoka 819-0395, JAPAN jin@ctrl.mech.kyushu-u.ac.jp and {kikuuwe, yama}@mech.kyushu-u.ac.jp

#### Abstract

This paper presents a sliding mode filter for removing noise. It effectively removes impulsive noise and high-frequency noise with producing smaller phase lag than linear filters. In addition, it is less prone to overshoot than previous sliding mode filters and it does not produce chattering. 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 when the input value is constant. Its algorithm is derived by using the backward Euler discretization, which can be used to prevent chattering. The effectiveness of the filter was shown by experiments by using an ultrasonic sensor and an optical encoder.

*keywords*: Quadratic Sliding Mode Filter, Finite Time Convergence, Backward Euler Discretization, Chattering Avoidance, Overshoot Reduction

## 1 Introduction

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. It is, however, pointed out that its computational cost is high [2]. The peak-and-valley filter [3], which is also for removing impulsive noise, is faster but less effective than the median filter, as pointed out by [4]. An adaptive windowing filter [5], 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 computational cost.

Some researchers study the use of sliding mode techniques for filtering [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. One of the major problems raised in implementing sliding mode techniques is chattering, which is high-frequency oscillation in the output value. In order to reduce chattering, some remedies such as adaptive switching gain [6], boundary layer [7] and low-pass filtering [8] are used. However, with these remedies, the finite time convergence to the sliding mode cannot be realized in a strictly mathematical sense and the parameters must be carefully chosen considering the trade-off between the chattering attenuation and the convergence precision. It is known that sliding mode observers based on the super-twisting algorithm [9, 10, 11, 12, 13, 14, 15] realize finite time convergence in continuous-time analysis. However, the accuracy of convergence in discrete-time implementation, typically with the forward Euler discretization [13, 14, 15], depends on the sampling interval, as pointed out in [9]. In addition, they are prone to overshoot during the convergence.

The use of a quadratic sliding surface has also been studied in the field of filtering [16, 17, 18]. 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 [17, 18] 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 [19] produces chattering, as will be demonstrated in Section 2.2.

This paper presents a sliding mode filter that employs a quadratic surface as its sliding surface. The proposed filter effectively removes impulsive noise and high-frequency noise with producing smaller phase lag than linear filters. In addition, it is less prone to overshoot than previous sliding mode filters. Its algorithm is derived by using the backward Euler discretization, which can be used to prevent chattering. This algorithm is computationally inexpensive, and thus suitable for realtime applications.

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.

<sup>&</sup>lt;sup>1</sup>According to Emaru and Tsuchiya [17, 18], the full form of ESDS is "the system which estimate the smoothed value and the differential value by using sliding mode system".

## 2 Quadratic sliding mode filters

#### 2.1 Continuous-time expression of quadratic sliding mode filters

Let us consider the system described in the following expression:

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

$$\dot{x}_2 = -F \operatorname{sgn}(\sigma)$$
 (1b)

where

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

Here,  $u \in \mathbb{R}$  is an input,  $x_1 \in \mathbb{R}$  and  $x_2 \in \mathbb{R}$  are the system states, F > 0 is a constant, and sgn is the signum function defined as follows:

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

Here, note that the return value of the sgn function is a set of values instead of a single value. Therefore, strictly speaking, "=" in (1b) should be replaced by " $\in$ ", but "=" is used throughout this paper for notational simplicity. Equation (1b) should be interpreted to be equivalent to the following logical expression:

$$(\dot{x}_2 = -F \operatorname{sgn}(\sigma) \land \sigma \neq 0) \lor (\dot{x}_2 \in [-F, F] \land \sigma = 0).$$
(4)

The system (1) can act as a filter with the input u and the output  $x_1$ . Han and Wang [16] used this filter for removing white noise contained in u, and Emaru and Tsuchiya [17, 18] 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  $\sigma \neq 0$ , according to (1b),  $\dot{x}_2$  satisfies

$$\dot{x}_2 = -F \operatorname{sgn}(\sigma). \tag{5}$$

In this period, the state  $(x_1, x_2)$  moves along quadratic curves in the state space, as illustrated by dashed curves in Fig. 1. On the other hand, when  $\sigma = 0$ , i.e.,

$$2F(x_1 - u) + |x_2|x_2 = 0 (6)$$

is satisfied, the state  $(x_1, x_2)$  moves along the quadratic curve in the state space, as illustrated by the solid curve in Fig. 1.

In particular, when  $\dot{u} = 0$  and  $\sigma = 0$ , differentiating the both sides of (6) yields

$$\dot{x}_2 = -F \operatorname{sgn}(x_2). \tag{7}$$



Figure 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).



Figure 2: Analytical solution of (1) with a step change in u.

Therefore, when  $\dot{u} = 0$ ,  $\dot{x}_2$  takes the value of either -F or F, according to (5) and (7). Such a behavior of the system (1) is similar to that of a system under bang-bang control [20, Chapter 5], 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  $|\dot{x}_2| \leq F$ . Fig. 2 shows the analytical solution of (1) with a step change in u of which initial state is (0,0). In Fig. 2(b), we can observe that the trajectory of the state is symmetric in the  $x_1$ - $x_2$ space. For the former half, the filter (1) is in reaching mode, and the latter half, it is in sliding mode. This symmetry is due to the fact that the output  $x_1$  reaches the half-amplitude of the input u with  $\dot{x}_2 = F$  from its initial value, and then it reaches the input u with  $\dot{x}_2 = -F$ , as shown in Fig. 2(a).

A problem of the filter (1) is that it is prone to overshoot. Fig. 3 shows the analytical solution of (1) with a step change in u that is temporarily corrupted by a disturbance. As the solid curve shown in Fig. 3(b), the state  $(x_1, x_2)$  deviates from the sliding surface  $\sigma = 0 \wedge x_2 > 0$  into the region  $\sigma > 0 \wedge x_2 > 0$  by the influence of the disturbance. After the disturbance disappears, the deviated state moves parallel to the sliding surface. This is because, in the region  $\sigma > 0 \wedge x_2 > 0$ ,  $\dot{x}_2$  takes the same value as that is taken on the sliding surface  $\sigma = 0 \wedge x_2 > 0$  (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



Figure 3: Analytical solution of (1) with a step change in u that is temporarily corrupted by a disturbance (F = 200).

 $\sigma x_2 > 0$ , it does not reach the sliding surface before crossing the line  $x_1 = u$ .

#### 2.2 Discrete-time algorithm of quadratic sliding mode filters

According to Emaru *et al.* [19], 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)" [19]. As to the authors' knowledge, these two are only explicitly-reported 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. 4 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 by the dashed curves in Fig. 4(c) and Fig. 4(d). Because the state  $(x_1, x_2)$  never exactly reaches the sliding surface, chattering continues around  $x_1 = u$  as shown in Fig. 4(e) and Fig. 4(f).

## 3 Proposed Filter

#### 3.1 Motivation for the proposed filter

The motivation for proposing a new filter comes from the observation that there are two sources of the filter (1)'s tendency to overshoot. One is that the value of  $\dot{x}_2$  in the region  $\sigma x_2 > 0$  does not force the state  $(x_1, x_2)$  to reach the sliding surface before crossing the line  $x_1 = u$ . The second one is that numerical error and chattering are caused by improper discretization. Toward these problems, our contribution is twofold. First, we propose a modification of (1) to attract the state  $(x_1, x_2)$  toward the sliding surface even when  $\sigma x_2 > 0$ . Second, we present a discrete-time algorithm for integrating the modified version of (1) by using the backward Euler discretization, which can be used to realize exact reaching to the sliding surface.



Figure 4: Numerical solution of (1) with a step change in u obtained by FCM (F = 200, T = 0.001 s).

### 3.2 Continuous-time expression of the proposed filter

In order to force the state to be attracted by the sliding surface even when  $\sigma x_2 > 0$ , let us consider the following modification of (1):

$$\dot{x}_1 = x_2 \tag{8a}$$

$$\dot{x}_2 = \begin{cases} -\alpha F \operatorname{sgn}(\sigma) & \text{if } \sigma x_2 > 0 \\ -F \operatorname{sgn}(\sigma) & \text{if } \sigma x_2 < 0 \end{cases}$$
(8b)

where  $\alpha > 1$  is a constant and

$$\sigma \triangleq 2F(x_1 - u) + |x_2|x_2. \tag{9}$$



Figure 5: Behavior of gsgn(A, z, B).

In the filter (8), the value of  $|\dot{x}_2|$  in the case of  $\sigma x_2 > 0$  is modified from F to  $\alpha F$ . Because of this modification, in the region  $\sigma x_2 > 0$ , the state moves toward the sliding surface instead of moving parallel to it. The idea of changing the gain according to different conditions can also be found in the literature (e.g., [21, 22]).

A flaw of the expression (8) is that it does not define the value of  $\dot{x}_2$  when  $\sigma x_2 = 0$ . A more strict expression of (8) can be written as follows:

$$\dot{x}_1 = x_2 \tag{10a}$$

$$\dot{x}_2 = -Fgsgn(gsgn(-\alpha, x_2, -1), \sigma, gsgn(1, x_2, \alpha))$$
(10b)

where

$$\sigma \triangleq 2F(x_1 - u) + |x_2|x_2. \tag{11}$$

Here, gsgn is the generalized signum function defined as follows:

$$\operatorname{gsgn}(A, z, B) \triangleq \begin{cases} B & \text{if } z > 0\\ [\min(A \cup B), \max(A \cup B)] & \text{if } z = 0\\ A & \text{if } z < 0 \end{cases}$$
(12)

where  $A \subset \mathbb{R}$  and  $B \subset \mathbb{R}$  are closed intervals (which can be scalars as a special case) and  $z \in \mathbb{R}$ . This generalized signum function (12) is a set-valued function as sgn in (3) is. Fig. 5 illustrates the behavior of gsgn(A, z, B). It is worth noting that, when z = 0, (12) return the union of A, B, and all values in between. In addition, (12) reduces to (3) when A = [-1, -1] and B = [1, 1].

Fig. 6 shows the quadratic sliding surface (solid curve) and trajectories of the state  $(x_1, x_2)$  for  $\sigma \neq 0$  (dashed curves) in the filter (10), and Fig. 7 shows the relation among  $x_1$ ,  $x_2$  and  $\dot{x}_2$ . When  $\sigma x_2 \neq 0$ , (10) is equivalent to (8), and  $\dot{x}_2$  takes a single value  $(-\alpha F, -F, F \text{ or } \alpha F)$ , as shown by the horizontal surfaces in Fig. 7. When  $\sigma x_2 = 0$ , on the other hand,  $\dot{x}_2$  takes a value between the values of  $\dot{x}_2$  in adjoining regions, as shown by the vertical surfaces in Fig. 7. It is clear that (10) reduces to (1) in the case of  $\alpha = 1$  considering the properties of gsgn such as  $gsgn(-1, x_2, -1) = -1$ ,  $gsgn(1, x_2, 1) = 1$ , and  $gsgn(-1, \sigma, 1) = sgn(\sigma)$ .



Figure 6: Quadratic sliding surface (solid curve) and trajectories of the state  $(x_1, x_2)$  for  $\sigma \neq 0$  (dashed curves) in the filter (10).

Fig. 8 shows analytical solutions of the filters (10) and (1) under step-like disturbances. These results can be obtained by using simple algebra considering the fact that every solution trajectory of (10) and (1) is a sequence of parabola arcs in the state space. First, Fig. 8(a) and Fig. 8(b) show the case where a rightward sliding motion is disturbed by a positive pulse-like disturbance. The state deviates from the sliding surface due to the disturbance, and after the disturbance, it converges to the sliding surface again in the new filter (10) but does not in the conventional filter (1). In both filters, the deviation is caused by the attraction toward a temporarily displaced sliding surface with a constant *acceleration*  $\ddot{x}_1 = \dot{x}_2 = F$ . Therefore, the magnitude of the deviation increases as the disturbance magnitude increases, but only up to a particular level that is determined by the duration of the disturbance.

Second, Fig. 8(c) and Fig. 8(d) show another set of solutions in cases where negative disturbances are applied during a rightward sliding motion. The new filter (10) is influenced by the disturbance, exhibiting slower convergence, while the conventional filter (1) is not. This is because a large  $\alpha$  value can result in a stronger attraction (i.e., a larger  $|\ddot{x}_1|$ ) to the displaced sliding surface, which would be undesirable in some cases. This implies that an appropriate guideline should be sought for the choice of  $\alpha$  considering this effect.

Third, Fig. 8(e) and Fig. 8(f) show the effect of disturbances in the steady state. It is shown that the new filter (10) produces a faster resuming than the conventional filter (1). After the disturbance disappears, the  $x_2$  value of the filter (10) changes toward  $x_2 = 0$  faster than that of the filter (1) because of  $\alpha > 1$ .

#### 3.3 Discrete-time algorithm of the proposed filter

We derive the discrete-time algorithm for integrating (10) by using the backward Euler discretization, which can be used to realize exact reaching to the sliding surface.



Figure 7: Relation among  $x_1$ ,  $x_2$  and  $\dot{x}_2$ .

Based on the backward Euler discretization, (10) can be approximated as follows:

$$\frac{x_1(k) - x_1(k-1)}{T} = x_2(k) \tag{13a}$$

$$\frac{x_2(k) - x_2(k-1)}{T} = -Fgsgn(gsgn(-\alpha, x_2(k), -1), \sigma(k), gsgn(1, x_2(k), \alpha))$$
(13b)

$$\sigma(k) = 2F(x_1(k) - u(k)) + |x_2(k)|x_2(k)$$
(13c)

where k denotes the discrete time index. By using (13a), we can remove  $x_1(k)$  as follows:

$$\frac{x_2(k) - x_2(k-1)}{T} = -Fgsgn(gsgn(-\alpha, x_2(k), -1), \sigma(k), gsgn(1, x_2(k), \alpha))$$
(14a)

$$\sigma(k) = |x_2(k)| x_2(k) + 2FTx_2(k) + 2F(x_1(k-1) - u(k)).$$
 (14b)

This can be seen as a set of simultaneous equations that determines  $x_2(k)$  by using u(k),  $x_1(k-1)$  and  $x_2(k-1)$ .

Now, the solution of (14) with respect to  $x_2(k)$  is derived. First, we can simplify (14) as follows:

$$\frac{x_2(k) - x_2(k-1)}{T} = -Fgsgn(gsgn(-\alpha, x_2(k), -1), x_2(k) - x_2^*(k), gsgn(1, x_2(k), \alpha))$$
(15)

where

$$x_2^*(k) = \operatorname{sgn}(x_1(k-1) - u(k))(FT - \sqrt{F^2T^2 + 2F|x_1(k-1) - u(k)|}).$$
(16)

Here,  $x_2^*(k)$  is the value of  $x_2(k)$  that satisfies  $\sigma(k) = 0$ . Equation (15) is equivalent to (14) because  $\sigma(k)$  is a monotonously increasing function with respect to  $x_2(k)$ .

Furthermore, (15) can be rewritten as follows:

$$-(x_2(k) - x_2(k-1)) = gsgn(gsgn(-\alpha TF, x_2(k), -TF), x_2(k) - x_2^*(k), gsgn(TF, x_2(k), \alpha TF)).$$
(17)

By applying the proposition that is shown in the appendix,  $x_2(k)$  can be moved out from the right-hand side of (17) as follows:

$$-(x_2(k) - x_2(k-1)) = gsat(gsat(-\alpha TF, x_2(k-1), -TF), x_2(k-1) - x_2^*(k), gsat(TF, x_2(k-1), \alpha TF)).$$
(18)



(a) Positive disturbances in the sliding motion (time domain).





Solutions of (1) with inputs 1 and 2 (overlapping) Solution of (1) with input 3

(b) Positive disturbances in the sliding motion  $(x_1 - x_2 \text{ space}).$ 



Displaced sliding surfaces

 $(x_1 - x_2 \text{ space}).$ 

Solutions of (10) with inputs 1 and 2 (overlapping)

Solutions of (10) with inputs 1 series (10) of (10) with inputs 3
 Solutions of (1) with inputs 1, 2 and 3 (overlapping)

(d) Negative disturbances in the sliding motion

(time domain).



Figure 8: Analytical solutions of (10) and (1) with constant inputs temporarily corrupted by pulse-like disturbances ( $F = 200, \alpha = 4$ ).

Here, gsat is a generalized saturation function [23] defined as follows:

$$gsat(a, z, b) \triangleq \begin{cases} b & \text{if } z > b \\ z & \text{if } z \in [a, b] \\ a & \text{if } z < a \end{cases}$$
(19)



Figure 9: (a) Relation among  $x_2(k-1) - x_2^*(k)$ ,  $x_2(k-1)$  and  $x_2(k) - x_2(k-1)$ ; (b) Relation among  $x_1(k-1)$ ,  $x_2(k-1)$  and  $x_2(k) - x_2(k-1)$ .

where  $a \leq b$ . Fig. 9(a) shows the relation among  $x_2(k-1) - x_2^*(k)$ ,  $x_2(k-1)$  and  $x_2(k) - x_2(k-1)$ , and Fig. 9(b) shows the relation among  $x_1(k-1)$ ,  $x_2(k-1)$  and  $x_2(k) - x_2(k-1)$ . Then, we can obtain  $x_2(k)$  from (18) as follows:

$$x_2(k) = x_2(k-1) - \text{gsat}(\text{gsat}(-\alpha TF, x_2(k-1), -TF), x_2(k-1) - x_2^*(k), \text{gsat}(TF, x_2(k-1), \alpha TF)).$$
(20)

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

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

$$x_{2}(k) := x_{2}(k-1) - \operatorname{gsat}(\operatorname{gsat}(-\alpha TF, x_{2}(k-1), -TF), x_{2}(k-1) - x_{2}^{*}(k),$$
  
$$\operatorname{gsat}(TF, x_{2}(k-1), \alpha TF))$$
(21b)

$$x_1(k) := Tx_2(k) + x_1(k-1).$$
 (21c)

Fig. 10 shows the numerical solution of (10) with a step change in u obtained by algorithm (21). Note that the state exactly reaches the sliding surface and the target state (u, 0). It is also worth noting that there is no chattering in sliding mode. This kind of ways of avoiding chattering by using the backward Euler discretization is also reported in [23, 24, 25].

In addition, by setting  $\alpha = 1$ , (21) reduces to the following:

$$x_2^*(k) := \operatorname{sgn}(x_1(k-1) - u(k))(FT - \sqrt{F^2T^2 + 2F|x_1(k-1) - u(k)|})$$
(22a)

$$x_2(k) := x_2(k-1) - \text{gsat}(-TF, x_2(k-1) - x_2^*(k), TF)$$
 (22b)

$$x_1(k) := Tx_2(k) + x_1(k-1).$$
 (22c)

This algorithm can be viewed as another discrete-time algorithm of ESDS (1). In Section 4, this algorithm will be compared with (21) to demonstrate the advantage of using  $\alpha > 1$ , and it also will be compared with FCM to demonstrate the advantage of backward Euler discretization.

Fig. 11 demonstrates the tracking performances of the proposed filter (21), the second-order Butterworth low-pass filter (BWF), and ESDS implemented with the backward Euler method (ESDS-BE),



Figure 10: Numerical solution of (10) with a step change in u ( $F = 200, \alpha = 4, T = 0.001$  s).



Figure 11: Tracking performances of the proposed filter (21), BWF and ESDS-BE (T = 0.001 s).

i.e., algorithm (22). Here, the input u is a sinusoidal signal starting from u = 0 at t = 0 s, and the initial states of all filters are zeros at t = 0 s. Fig. 11 only shows the data after t = 3 s, where the outputs are in the steady state. It is observed that, ESDS-BE produces the largest phase lag among the three filters, and BWF produces larger phase lag than the propose filter.

## 4 Experiments

The proposed filter (21) was experimentally tested by using data from an ultrasonic sensor and an optical encoder. The output of the proposed filter was compared with those of the second-order Butterworth low-pass filter (BWF), ESDS implemented with FCM (ESDS-FCM), and ESDS implemented with the



(a) Experiment by using an ultrasonic sensor.



(b) Experiment by using an optical encoder.

Figure 12: Experimental setups.

backward Euler method (ESDS-BE), i.e., algorithm (22).

In both experiments,  $\alpha$  and F of the proposed filter were chosen to make the output as smooth as possible while restricting the phase lag in an acceptable range. In BWF, its cutoff frequency was chosen to make its output as close as possible to that of the proposed filter. The parameter F of ESDS-FCM and ESDS-BE was chosen equal to that of the proposed filter.

#### 4.1 Ultrasonic sensor

In the first set of experiments, an ultrasonic sensor system (transmitter: PT40-18, receiver: PR40-18, Nippon Ceramic Co., Ltd.) was fixed on a desk, and the distance between the sensor and the arc-shaped back of a chair was measured as shown in Fig. 12(a).

First, the chair was in its initial position, and then it was moved toward the desk slowly. This motion was performed twice with two different configurations of the sensor system. Specifically, the system was configured so that, when no reflection was detected within the sampling period T = 0.01 s, it recorded the minimum value (0 cm) for the first motion, or the maximum value (80 cm) for the second motion. These minimum and maximum values are the sources of impulsive noise in this experiment.

Fig. 13(a) shows the data obtained from the first motion, which is corrupted by negative impulsive noise, and Fig. 14(a) shows the data obtained from the second motion, which is corrupted by positive impulsive noise. Fig. 13(b) - Fig. 13(e) and Fig. 14(b) - Fig. 14(e) show the filtered outputs of the four filters for the data obtained from the first and second motions, respectively. In each figure, the graphs (b) to (d) are for comparing the proposed filter to conventional filters, and the graph (e) is for exhibiting the effect of backward Euler discretization.

The figures show that, in the steady state, BWF produced the most oscillatory result among the four filters, ESDS-FCM and ESDS-BE were less oscillatory, and the proposed filter was the least oscillatory. It can also be seen that, in the transient period, overshoots were suppressed by using the proposed filter. The advantage of using  $\alpha > 1$  instead of using  $\alpha = 1$  can be observed in Fig. 13(d) and Fig. 14(d), in which the proposed filter produces smaller overshoots and oscillation. Fig. 13(e) and Fig. 14(e) show that backward Euler method produced slightly smaller overshoots than FCM.

In the transient period (t = 0 s to 5 s) of Fig 13, it can be seen that the proposed filter exhibited



Figure 13: Experiment by using an ultrasonic sensor (T = 0.01 s): the first motion.

slower convergence to the correct value (about 56 cm) than the ESDS filters did. This is because the output  $x_1$  was attracted to the corrupted value (u = 0 cm) with  $\ddot{x}_1 = -\alpha F < -F < 0$  in the case of the proposed filter while it was with  $\ddot{x}_1 = -F$  in the case of the ESDS filters. A similar behavior can be observed in the early period of Fig 14, in which the output of the proposed filter stayed stationary between the corrected and corrupted values while those of ESDS filters exhibited overshoots. Thus, it should be noted that the proposed filter produces a biased output or slow convergence when the input contains dense impulsive noise, although it is still advantageous over ESDS filters in that it is less prone to overshoot and oscillation.

## 4.2 Optical Encoder

In the second set of experiments, a 6-DOF industrial manipulator (MOTOMAN-UPJ, Yaskawa Electric Corporation) shown in Fig. 12(b) was used. The input signal provided to the filters was the angular velocity signal (the numerical derivative of the angle signal) from the optical encoder attached to the



Figure 14: Experiment by using an ultrasonic sensor (T = 0.01 s): the second motion.

sixth joint of the manipulator. A smooth desired trajectory from 80 degree to -80 degree was provided to a position controller [24] to control the angle of the joint. Fig. 15(a) shows the obtained angle signal, and Fig. 15(b) shows the velocity signal obtained by the numerical differentiation. In Fig. 15(b), one can observe that the signal was corrupted by high-frequency noise and there was an impulse at t = 1.554 s. The filtered results are shown in Fig. 15(c) to Fig. 15(j), which are the enlarged views of Fig. 15(b).

The results in Fig. 15(c) to Fig. 15(f) show that BWF failed to remove the effect of the impulse, whereas ESDS-FCM, ESDS-BE and the proposed filter succeeded. It is clear that, even in the case of BWF, the effect of the impulse can be reduced by using a lower cutoff frequency, but it would cause larger phase lag. Figs. 15(d)(e)(h)(i) show that the proposed filter tracks the input signal rather smoothly and accurately compared to the ESDS filters. This can be attributed to the use of  $\alpha > 1$ , which provides smaller overshoot. The difference between ESDS-FCM and ESDS-BE is distinct in Figs. 15(f) and (j), which implies the advantage of the use of backward Euler discretization.



Figure 15: Experiment by using an encoder of an industrial manipulator (T = 0.001 s).

## 5 Conclusion

In this paper, we have presented a quadratic sliding mode filter for removing impulsive noise and highfrequency noise with producing smaller phase lag than linear filters. The proposed filter does not produce chattering, and it is less prone to overshoot than previous quadratic sliding mode filters. In addition, its algorithm is computationally inexpensive, and thus suitable for realtime applications. Experimental results showed the effectiveness the proposed filter.

One issue that is not addressed here but remained as the future topic of research is to develop guidelines for how to choose the the values of parameters F and  $\alpha$ . As another issue, it is expected that the proposed filter can be further extended for smoothing the first or higher order derivative of the input signal.

#### Acknowledgment

This work was supported in part by Grant-in-Aid for Young Scientists (B), No. 22760321, from Japan Society for the Promotion of Science.

## REFERENCES

- 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] V. G. Moshnyaga and K. Hashimoto, "An Efficient Implementation of 1-D Median Filter," in Proceedings of 52nd IEEE International Midwest Symposium on Circuits and Systems, pp. 451-454, Boise, USA, 2009.
- [3] P. S. Windyga, "Fast Impulsive Noise Removal," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 173-179, 2001.
- [4] N. Alajlan, M. Kamel and E. Jernigan, "Detail Preserving Impulsive Noise Removal," Signal Processing: Image Communication, vol. 19, no. 10, pp. 993-1003, 2004.
- [5] 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.
- [6] F. Chen and M. W. Dunnigan, "Comparative Study of a Sliding-Mode Observer and Kalman Filters for Full State Estimation in an Induction Machine," *IEE Proceedings on Electric Power Applications*, vol. 149, no. 1, pp. 53-64, 2002.
- [7] A. Alessandri, "Sliding-Mode Estimators for a Class of Non-Linear Systems Affected by Bounded Disturbances," *International Journal of Control*, vol. 76, no. 3, pp. 226-236, 2003.

- [8] I. Haskara, A. Ozguner and V. Utkin, "On Sliding Mode Observers via Equivalent Control Approach," *International Journal of control*, vol. 71, no. 6, pp. 1051-1067, 1998.
- [9] A. Levant, "Robust Exact Differentiation via Sliding Mode Technique," Automatica, vol. 34, no. 3, pp. 379-384, 1998.
- [10] J. A. Moreno and M. Osorio, "A Lyapunov Approach to Second-Order Sliding Mode Controllers and Observers," in Proceedings of 47th IEEE Conference on Decision and Control, pp. 2856-2861, Cancun, Mexico, 2008.
- [11] N. K. M'sirdi, A. Rabhi, L. Fridman, J. Davila and Y. Delanne, "Second Order Sliding Mode Observer for Estimation of Velocities, Wheel Sleep, Radius and Stiffness," in Proceedings of the 2006 American Control Conference, pp. 3316-3321, Minneapolis, USA, 2006.
- [12] J. Resendiz, W. Yu and L. Fridman, "Two-Stage Neural Observer for Mechanical Systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 10, pp. 1076-1080, 2008.
- [13] J. Davila, L. Fridman and A. Levant, "Second-Order Sliding-Mode Observer for Mechanical Systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1785-1789, 2005.
- [14] S. Solvar, V. Le, M. Ghanes, J. P. Barbot and G. Santomenna, "Sensorless Second Order Sliding Mode Observer for Induction Motor," in Proceedings of 2010 IEEE International Conference on Control Applications, pp. 1933-1938, Yokohama, Japan, 2010.
- [15] N. K. M'Sirdi, A. Rabhi, L. Fridman, J. Davila and Y. Delanne, "Second-Order Sliding-Mode Observer for Estimation of Vehicle Dynamic Parameters," *International Journal of Vehicle Design*, vol. 48, no. 3-4, pp. 190-207, 2008.
- [16] 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)
- [17] T. Emaru and T. Tsuchiya, "Research on Estimating the Smoothed Value and the Differential of the Sensor Inputs by Using Sliding Mode System (Application to Ultrasonic Wave Sensor)," *Transactions* of the Japan Society of Mechanical Engineers, Series C, vol. 66, no. 652, pp. 3947-3954, 2000. (in Japanese)
- [18] 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.
- [19] T. Emaru, T. Tsuchiya and K. Tanaka "Fast Calculating Method for Eliminating the Impulsive Noise," *Transactions of the Japan Society of Mechanical Engineers, Series C*, vol. 70, no. 689, pp. 62-68, 2004. (in Japanese)
- [20] D.E. Kirk. "Optimal Control Theory: an Introduction," Prentice Hall, 1970.

- [21] A. Levant, "Sliding Order and Sliding Accuracy in Sliding Mode Control," International Journal of Control, vol. 58, no. 6, pp. 1247-1263, 1993.
- [22] G. Bartolini, A. Ferrara and E. Usai, "Applications of a Sub-Optimal Discontinuous Control Algorithm for Uncertain Second Order Systems," *International Journal of Robust and Nonlinear Control*, vol. 7, no. 4, pp. 299-319, 1997.
- [23] 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.
- [24] R. Kikuuwe, S. Yasukouchi, H. Fujimoto and M. Yamamoto, "Proxy-Based Sliding Mode Control: a Safer Extension of PID Position Control," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 670-683, 2010.
- [25] V. Acary and B. Brogliato, "Implicit Euler Numerical Scheme and Chattering-Free Implementation of Sliding Mode Systems," Systems and Control Letters, vol. 59, no. 5, pp. 284-293, 2010.

# Appendix:

**Proposition:** The following two equations are equivalent to each other:

- z y = gsgn(gsgn(a, -z, b), x z, gsgn(c, -z, d)) and
- z y = gsat(gsat(a, -y, b), x y, gsat(c, -y, d))

where  $a \leq b \leq c \leq d$ .

#### **Proof:**

$$\begin{aligned} \text{rocf:} \\ z - y = \text{gsn}(\text{gsn}(a, -z, b), x - z, \text{gsn}(c, -z, d)) \\ \Leftrightarrow & (x - z < 0 \land ((z - y = a \land -z < 0) \lor (z - y = b \land -z > 0) \lor (z - y \in [a, b] \land -z = 0))) \\ \lor & (x - z > 0 \land ((z - y = a \land -z < 0) \lor (z - y = d \land -z > 0) \lor (z - y \in [a, d] \land -z = 0))) \\ \Leftrightarrow & (x - z < 0 \land ((z - y = a \land -y < a) \lor (z - y = b \land -y > b) \lor (z - y = [a, d] \land -z = 0))) \\ \Leftrightarrow & (x - z < 0 \land ((z - y = a \land -y < a) \lor (z - y = b \land -y > b) \lor (z - y = -y \land -y \in [a, d]))) \\ \Leftrightarrow & (x - z < 0 \land ((z - y = a \land -y < c) \lor (z - y = d \land -y > d) \lor (z - y = -y \land -y \in [a, d]))) \\ \lor & (x - z > 0 \land ((z - y \in [a, c] \land -z < 0) \lor (z - y \in [b, d] \land -z > 0) \lor (z - y \in [a, d] \land -z = 0))) \\ \Leftrightarrow & (x - z < 0 \land (z - y = g ast(a, -y, b)) \\ \lor & (x - z < 0 \land z - y = gsat(a, -y, b)) \\ \lor & (x - z > 0 \land (z - y \in [a, c] \land -z < 0) \lor (z - y \in [b, d] \land -z > 0) \lor (z - y \in [a, d] \land -z = 0))) \\ \Leftrightarrow & (x - z < 0 \land z - y = gsat(c, -y, d)) \\ \lor & (x - z > 0 \land (z - y \in [a, c] \land -z < 0) \lor (z - y \in [b, d] \land -z > 0) \lor (z - y \in [a, d] \land -z = 0)))) \\ \Leftrightarrow & (x - z > 0 \land (z - y \in gsat(c, -y, d)) \\ \lor & (x - z = 0 \land ((x - y \in [a, c] \land -z < 0) \lor (x - y \in [b, d] \land -z > 0) \lor (x - y \in [a, d] \land -z = 0)))) \\ \Leftrightarrow & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b)) \\ \lor & (z - y = gsat(a, -y, b) \land x - y < gsat(a, -y, b))$$