Methods of Software Data Filtering for Working with Sensors in the Field of Robotics

Dmitrii G. Chirkov[1], Anastasia D. Stotckaia[2]
Department of Automatic Control System
Saint Petersburg Electrotechnical University "LETI"
St. Petersburg, Russian Federation
[1]chirkofun@yandex.ru, [2]adstotskaya@etu.ru

***Abstract*—Signals from sensors usually have noise due to the different outside conditions such as vibrations, magnetic and electrical fields and others. For this reason different kind of filters can be used: hardware and software. The paper deals with software implementation of filters for MaxBotix ultrasonic sensors: median filter, moving average filter and low-pass filter. Such algorithms also can be used with other types of sensors. Choosing particular type of filter depends on different factors: signal dynamics, accuracy, loss limit and others.**

Keywords— software filtration; filters, sensors; robotic

I. Introduction

When working with sensors with analog output, there often will be a problem with the noise level of the output signal. Both hardware and software filters can be used to resolve this problem.

In this paper, it will be considered different software algorithms for filtering the signal received from the ultrasonic sensor company MaxBotix, namely: median filter, moving average filter and low-pass filter. Each of the presented filtering methods may be more effective in one situation and less effective in another, but in a particular case there was a need to preserve the dynamics of the signal, remove high-frequency noise and eliminate the possibility of significant data distortion.

II. MaxBotix ultrasonic sensor working principle

MaxBotix ultrasonic sensors send a sound signal with a frequency of 44 kHz to the object and receive the signal reflected from this object. The sensor counts down the time between sending the sound wave and receiving the reflected wave. Knowing this time and the speed of sound in the air, you can calculate the distance to the object from which the wave was reflected [1].

Such sensors can have a wide directivity pattern, which leads to the fact that the sensor sends information about the distance to the nearest object from the wave propagation zone. Also, the directivity pattern can be narrow, which allows you to identify objects with a small area. Sensors with a narrow directivity pattern give more accurate results than sensors with a wide directivity pattern. However, at the same time, sensors with a narrow directivity pattern are more expensive.

The frequency of sending data from MaxBotix ultrasonic sensors is 10 Hz, which is rather small, but it is enough to implement distance measurements under the condition of robot's immobility and the environment or at low motion speeds [1].

These sensors have several modes of sending data, as well as several types of connection. Data can be sent via the RS-232 interface in ASCII code format: RXXXX, where R is the first character followed by the distance XXXX, which is measured in centimeters [1].

There are also 2 analog modes of operation:

1. the sensor receives a signal proportional to the distance;

2. the actual signal from the microphone, for which its shape must be taken into account when converting.

The detection range of such sensors varies from 5 to 10 meters depending on the selected model. The minimum detection distance is usually 25 centimeters. If the object is closer than 25 centimeters to the plane of the emitter, the sensor may send incorrect distance information.

III. Analog signal problems with MaxBotix sensors

When working with MaxBotix sensors, there is often a problem of power stabilization. To solve this problem hardware filters should be used.

One such solution is to install a common RC filter in the sensor power circuit. Using this filter is proposed by MaxBotix itself. This filter indeed removes most of the noise, but the signal from the sensor still contains high-frequency noise, which does not allow you to use this signal for creating mechatronic systems. The connection diagram is shown in Fig. 1 [2].

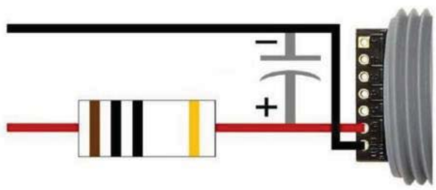

Fig. 1. Power filter for ultrasonic sensors

978-1-7281-9692-3/20/$31.00 ©2020 IEEE

IV. Software low-pass filter

The low-pass filter is based on an integrator. It removes small irregularities in the signal and also removes small fluctuations in the signal [3]. It is the fastest-performing filter of all filters discussed in this article.

Consider the formula of the simplest low-pass filter (1) [4]:

$$y^f_{(i)} = (1-K)\cdot x^{nf}_{(i)} + K\cdot y^f_{(i-1)}, \quad (1)$$

where $y^f_{(i)}$ – the current filtered value, $x^{nf}_{(i)}$ – the current non-filtered value, $y^f_{(i-1)}$ – the previous value, which is already filtered.

The K coefficient determines the degree of filter smoothing. The higher this coefficient, the more smoothed the signal will be received.

The application of a software low-pass filter with different coefficients to the signal from the sensor is shown in Fig. 2.

As can be seen from Fig. 2, the best signal filtering result is achieved at a coefficient of 0.7, since there is a small data losses and the signal shape becomes smoother (that is, without high-frequency oscillations). At the same time, there can be seen a small shift that increases the actual response time of the sensor. Using low-pass filter with other coefficients either does not create proper filtering and there are still high-frequency areas in the signal, or data losses are too large.

Table 1 provides information about data losses when using a low-pass filter.

TABLE I Data losses using a low-pass filter

Coefficient value	Filtering data loss, %
0,9	41.01%
0,7	21.44%
0,5	8,79%
0,1	1,28%

The calculation of data losses is carried out as follows: the point is selected corresponding to the reliable data with the maximum data loss relative to the original signal. Reliable data are data where there are no high-frequency areas. This point corresponds to a time value of 6.59 s.

As can be seen from Table 1, the losses using the coefficients 0.9 and 0.7 are quite large and using the coefficients 0.5 and 0.1 the losses are much smaller, but at the same time, based on the graphs in Fig. 2, these values of the coefficients do not give a sufficient level of filtration.

V. Software median filter

The median filter is a common software digital filter with a fairly simple implementation. This type of filter is based on the assumption that the noise has a pulse form (large amplitude and short duration). The advantage of this filter is that it does not introduce a significant delay in the monotonic form of the useful signal. The algorithm of this filter is as follows [5]:

1. an array with n elements is created (window array);
2. this array is filled with values from the sensor;

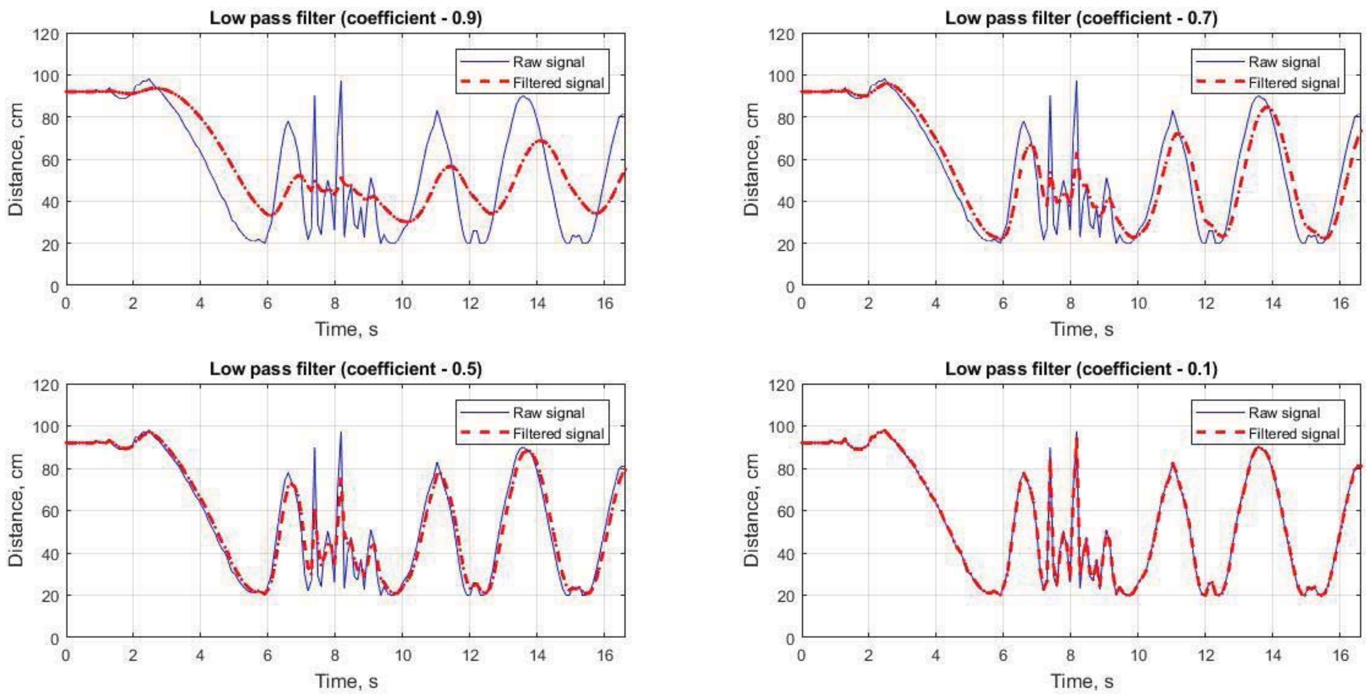

Fig. 2. Application of software low-pass filter with different coefficients

3. array elements are sorted in ascending (or descending);

4. one value is selected from the middle of the array (this is the filtered value)

Fig. 3 shows graphs comparing the raw signal from the sensor with the filtered signal by a median filter with different window values.

As can be seen from the graphs, filtered data with window values 5 and 9 have the most appropriate results. When using window values 15 and 19, there are losses of useful information (the filter coarsens the areas of the graph where the derivative sign changes, at the stationary points), which is unacceptable, since the actual value from the sensor and the filtered ones are very different.

Table 2 provides information about data losses using the median filter.

TABLE II DATA LOSSES USING MEDIAN FILTER

Window value	Filtering data loss, %
5	5,12%
9	16,67%
15	35,89%
19	61,53%

The same data losses calculation point is selected as in the case of the low-pass filter.

It can be noticed that the best result is achieved at window 5, since the data losses are negligible and the signal dynamics are preserved. Data losses with using different window values are too large.

VI. SOFTWARE MOVING AVERAGE FILTER

The algorithm of moving average filter is similar to the median filter algorithm, but instead of sorting and selecting the middle element of the array, the filter considers the average for the entire window [6, 7]. Implementing of such filter needs less time because there is no need to sort the data.

Fig. 4 shows graphs comparing the raw signal from the sensor and filtered signal with different window values.

As can be seen from the graphs, filtered data with window values 5 and 9 have the most appropriate results. Using window values 15 and 19, the filtered signal loses dynamics in areas of the chart where the dynamics are too high, and also distorts the values at the stationary points.

Table 3 provides information about data losses using the median filter.

TABLE III DATA LOSSES USING MEDIAN FILTER

Window value	Filtering data loss, %
5	7,69%
9	19,23%
15	38,46%
19	39,74%

The same data losses calculation point is selected as in the case of the low-pass filter and median filter.

The best filtering result is achieved at window 5, because at this window value there are no significant data losses and the signal dynamics are preserved.

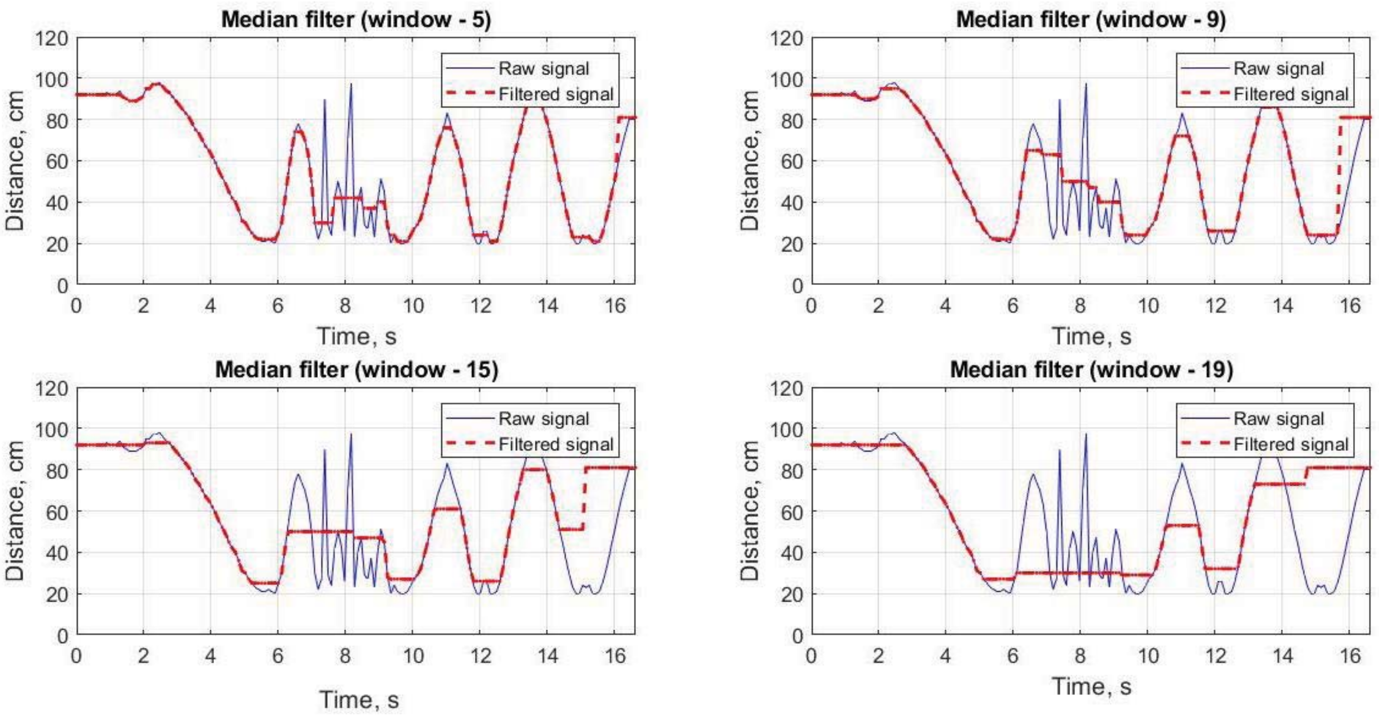

Fig. 3. Application of software median filter with different window values

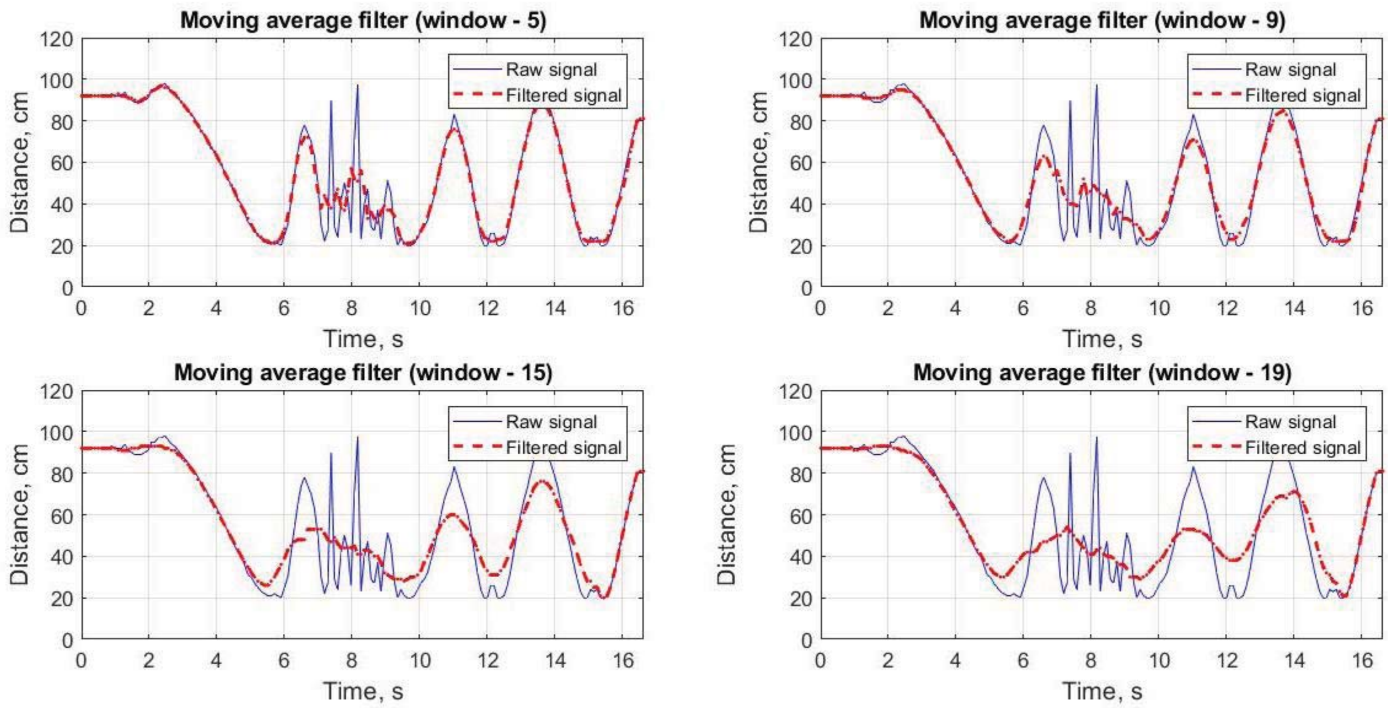

Fig. 4. Application of software moving average filter with different window values

VII. Qualitative Evaluation of the Effectiveness of Software Filters Working with MaxBotix Sensors

Based on experimental data (graphs and tables), it can be said that all filters help to remove high-frequency noise from the signal, which suggests that all these filters can be attributed to the class of low-pass filters.

Working with MaxBotix sensors, it was empirically found that a median filter with a window value that is equal to 5 is the most reasonable option to use, since the dynamics are preserved and data losses are minimal. The low-pass filter creates a "shift" between the signals, which is not allowed to work with this type of sensor. The moving average filter with window 5 copes better than the low-pass filter, but worse than the median (data loss is greater and at the same time the high-frequency area is not treated as well as with the median filter).

VIII. Conclusion

The paper considers descriptions and comparisons of three software filtration methods: low-pass filter, median filter and moving average filter. Graphs with signal forms and tables considering information about data losses are considered and described. The best filtration option is chosen.

The most appropriate filtering method to work with MaxBotix sensors is median filter with window value 5. It provides the best signal form after filtration and also the minimal data losses. Other methods can be used in different situations with different sensors.

References

[1] XL-MaxSonar-WR|WC Datasheet. Available at: https://www.maxbotix.com/documents/XL-MaxSonar-WR_Datasheet.pdf (accessed 2 March 2020)

[2] Power Supply Filter, MB7961. Available at: https://www.maxbotix.com/documents/MB7961_Datasheet.pdf (accessed 5 March 2020)

[3] Jonathan Y. Stein, Digital Signal Processing: A Computer Science Perspective. John Wiley & Sons, Inc, 2002, 588.

[4] Low-pass filter. Available at: https://en.wikipedia.org/wiki/Low-pass_filter#Simple_infinite_impulse_response_filter (accessed 6 March 2020)

[5] Median filter. Available at: https://en.wikipedia.org/wiki/Median_filter (accessed 8 March 2020)

[6] Moving average filters. Available at: https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_Ch15.pdf (accessed 12 March 2020)

[7] Walt Kester, Mixed Signal and DSP Design Techniques. Newnes/Elsevier, 2002, 6.5-6.7.