Proceedings of CONF-FMCE 2026 Symposium: Artificial Intelligence and Smart Sensing for Mechanical and Electrical Systems
DOI: 10.54254/2755-2721/2026.BA35406

A Review of Signal Processing and Noise Suppression Methods for Robot Sensors

Zihao Chen

The Glasgow Academy UESTC, Chengdu, China
17780002005@163.com

Abstract. Robot sensors undertake basic tasks including positioning, motion control and environmental detection in robotic systems. Sensor signals will be affected by multiple interference factors during collection and transmission. Common disturbances include thermal noise, electromagnetic interference, mechanical vibration and digital quantization error. These negative factors lower signal-to-noise ratio and make measured data less accurate. This paper summarizes mainstream signal processing and noise elimination techniques applied to robot sensors. It takes signal-to-noise ratio as core evaluation standard and analyzes three practical filtering algorithms, namely mean filtering, low-pass filtering and Kalman filtering. The study classifies various errors in the sensor measurement process and discusses feasible methods to implement filtering code on embedded devices. Analysis results prove that combining hardware anti-interference design and software filtering processing can effectively restrain noise and upgrade signal quality. Kalman filtering fits dynamic state calculation and real-time tracking scenes. Mean filtering and low-pass filtering are suitable for embedded hardware with limited computing space and power. The research offers usable reference schemes for sensor signal processing design in practical robot development work.

Keywords: Robot sensor, noise suppression, SNR, Kalman filter, embedded implementation

1. Introduction

Robots need sensors to obtain external environmental information. Common sensors include IMU, encoders, LiDAR, ultrasonic sensors, and force sensors. These sensors collect data such as position, speed, attitude, and contact force. The quality of sensor signals directly determines the stability and control accuracy of the robot system [1]. In practical operating environments, sensor signals are contaminated by substantial noise. Noise comes from circuit components, motor driving, power fluctuation, and mechanical movement. Noise causes signal jitter, drift, and distortion. If such noisy signals are directly adopted for control and decision-making, the robot will suffer from trajectory deviation, positioning errors and even system instability [2]. SNR is a core index to measure signal quality. It reflects the proportion of effective signals in the total signal. Higher signal-to-noise ratio (SNR) means more reliable data [3]. Software filtering is the main method to suppress noise. Common methods include mean filtering, low-pass filtering, and Kalman filtering [4]. These methods are widely used in embedded platforms such as MCU and ARM [5].

DOI: 10.54254/2755-2721/2026.BA35406

This paper focuses on four problems. First, how to use SNR to quantify the noise level of robot sensor signals. Second, what are the principles, advantages, and disadvantages of mean filtering, low-pass filtering, and Kalman filtering? Third, what are the main error sources in the process of sensor measurement and signal processing [6]? Fourth, how to implement filtering algorithms efficiently on embedded platforms [7]. This research enables engineers to select suitable filtering schemes. It provides theoretical support and engineering reference for improving the perception performance of robot systems [8].

2. Basic concepts and SNR evaluation

2.1. Sensor signal and noise characteristics

The useful signals output by robot sensors represent real physical quantities, including position, velocity, attitude, force, and torque. Noise refers to unwanted signal components that interfere with the extraction of useful information from measurements [2].

Four typical noise types exist in robot sensor systems. Thermal noise is generated by thermal motion of electronic components. It is random and belongs to white noise [2]. Electromagnetic interference originates from motors, driving circuits, and communication buses, often exhibiting periodic or pulse characteristics [9]. Mechanical vibration is produced by robot motion and ground roughness. It leads to signal oscillation [10]. Quantization errors are introduced in analog-to-digital conversion. They are related to ADC resolution and reference voltage [10].

These noises reduce the authenticity of sensor signals. They increase measurement error and affect system performance. Noise suppression is necessary in robot sensor applications [3].

2.2. SNR definition and calculation

SNR is a key metric used to evaluate the quality of a measured signal. It is defined as the ratio of useful signal power to noise power, expressed in decibels. The calculation formula is as follows:

$$\mathrm{SNR} = 10\log_{10}\frac{\mathrm{P_{signal}}}{\mathrm{P_{noise}}} = 20\log_{10}\frac{\mathrm{A_{signal}}}{\mathrm{A_{noise}}} \tag{1}$$

Where $\mathrm{P_{signal}}$ represents effective power of useful signal, $\mathrm{P_{noise}}$ represents effective power of noise, and $\mathrm{A_{signal}}$& $\mathrm{A_{noise}}$ are effective amplitudes of signal and noise [3].

In engineering practice for robot sensing systems, SNR is divided into three levels for performance interpretation: SNR below 10 dB means serious interference—the signal is basically submerged by noise; SNR between 10 dB and 20 dB means moderate noise—the signal can be used after filtering; SNR above 20 dB means good signal quality—data can be used directly with minor errors [3]. The goal of signal processing is to suppress noise and improve SNR. Effective filtering can increase SNR by more than 10 dB. It significantly improves the availability of sensor data [5].

3. Common filtering algorithms for noise suppression

3.1. Mean filtering

Mean filtering is a time-domain smoothing algorithm. It estimates the true signal by calculating the average of multiple consecutive sampling points [4]. The mathematical expression of mean filtering is given as

DOI: 10.54254/2755-2721/2026.BA35406

$$\hat{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{2}$$

where N is the length of the sliding window. More sampling points bring stronger smoothness but slower response [4].

Mean filtering has obvious advantages. It has extremely low computational complexity and is easy to implement on embedded platforms. It works well on stationary or slow-changing signals. It can effectively suppress random white noise [4]. The disadvantages are also clear. It has poor dynamic tracking ability. It cannot follow fast-changing signals timely. It has limited suppression effect on impulse noise. Improper window length may cause signal lag and distortion [4]. Mean filtering is suitable for slow-varying signals. Typical applications include temperature, air pressure and static force sensor signals. It is often used as a pre-processing step in simple robot systems [10].

3.2. Low-pass filtering

Low-pass filtering allows low-frequency signals to pass and attenuates high-frequency noise. It is one of the most widely used preprocessing methods for robot sensors [7]. A first-order analog low-pass filter is composed of a resistor and a capacitor. The cut-off frequency determines the highest frequency allowed to pass. Digital low-pass filtering is more common in embedded systems. The formula for a digital first-order low-pass filtering is:

$$y(k) = \alpha y(k-1) + (1-\alpha)x(k) \tag{3}$$

where α is the smoothing factor, a value between 0 and 1. A larger α brings stronger smoothness but introduces greater delay [7].

Low-pass filtering has many benefits, including a simple structure, low computational cost, and effective suppression of high-frequency noise. It is suitable for real-time systems [7]. However, its main drawback is the trade-off between noise suppression and signal distortion. If the cut-off frequency is set too low, useful high-frequency components will be lost; if it is set too high, noise cannot be suppressed effectively [2]. Low-pass filtering is widely used in IMU, motor speed, and current sampling. It is usually used as the first-level filtering to remove high-frequency interference before advanced processing [11].

3.3. Kalman filtering

Kalman filtering is a recursive optimal state estimation algorithm designed for linear systems with Gaussian noise. It can estimate real-time state under disturbance [5]. The algorithm runs in two steps: prediction and update. In the prediction step, the system predicts the next state and error covariance according to the model. In the update step, the algorithm uses measured values to correct the estimated state. It approaches the real state continuously [3, 5]. Key parameters of the Kalman filter include the state vector, state transition matrix, control matrix, covariance matrix, process noise and measurement noise. Appropriate parameter setting determines filtering performance [6].

Kalman filtering has significant advantages. It provides optimal estimation under linear Gaussian assumptions, exhibits strong dynamic tracking capability, and enables effective multi-sensor data fusion. It can improve SNR significantly in dynamic scenes [1, 11]. The disadvantages are also obvious. It relies on accurate system model and noise statistical characteristics. Model mismatches or parameter error reduces performance. Computation is higher than simple filtering methods [6].

DOI: 10.54254/2755-2721/2026.BA35406

Kalman filtering is widely used in robot positioning, attitude estimation and IMU data fusion. It is the preferred method for high-precision dynamic robot applications [9].

4. Error sources in robot sensor systems

4.1. Hardware errors

Hardware errors come from sensors, electronic circuits, analog-to-digital conversion units, and mechanical structures. They are the main source of noise and deviation [6].

Sensor inherent errors include zero bias, sensitivity drift, and temperature drift. These errors are inherent to the sensor itself and exist before leaving factory. They change slowly with time and environment [6]. Circuit-related errors include thermal noise, power ripples, and electromagnetic interference. Thermal noise is inevitable in electronic components. Power ripples bring periodic interference. Electromagnetic interference affects signal quality seriously [2].

Analog-to-digital converter (ADC) errors include quantization error, nonlinear error and sampling jitter. Quantization error is determined by ADC digits. Lower resolution brings larger error [9]. Mechanical errors include installation deviation, vibration, and friction. Installation misalignment causes fixed deviation. Vibration and friction produce random disturbance during movement [10].

4.2. Software errors

Software errors are produced during signal processing and algorithm execution. They are related to design and implementation [9].

Algorithm errors arise from model mismatch and parameter mismatch. Simple models cannot describe complex systems. Inappropriate filtering parameters degrade noise suppression performance [5]. Operation errors include truncation errors and rounding errors. They are common in fixed-point embedded systems. Limited data bits lower computational accuracy [7]. Time synchronization error occurs in multi-sensor systems. Different sampling frequencies and delays cause data asynchrony, introducing extra errors in data fusion [11].

4.3. Environmental errors

Environmental factors affect sensor performance directly. Temperature, humidity and light variations cause sensor drift. High temperature increases zero bias of IMU and force sensors [6].

Ultrasonic and LiDAR sensors are affected by reflection and multipath effects. Obstacle reflection changes signal propagation path. The multipath effect leads to signal overlap and measurement errors [9].

Effective noise suppression must consider all error sources. The optimal solution is to combine hardware optimization with software algorithms. Hardware reduces interference at the source. Software further suppresses residual noise [2].

5. Embedded implementation of filtering algorithms

5.1. Implementation constraints

Most robots use embedded platforms. Typical platforms include STM32, ARM and Arduino. These platforms have limited RAM, Flash and computing capacity [7].

DOI: 10.54254/2755-2721/2026.BA35406

To ensure real-time performance, filtering algorithms must meet real-time requirements. Delay should be controlled within 1 ms. Resource occupation should be as low as possible. Fixed-point arithmetic is preferred to improve efficiency. Code should be modular and easy to transplant [5].

5.2. Mean filtering implementation

Mean filtering is easy to implement on embedded systems. Developers can open a sliding window array to store N sampling points. The average is computed in each cycle. A cumulative sum can be adopted to avoid redundant calculations [4].

This method is suitable for 8-bit and 16-bit low-cost MCUs. It brings almost no computing pressure. It is widely used in low-cost robot products [7].

5.3. Low-pass filtering implementation

Low-pass filtering only requires storing the previous output value. It occupies 1 to 2 variables. The smoothing factor α is set according to sampling frequency and expected cut-off frequency [6].

Developers can adjust α to balance smoothness and response speed. It is widely applied to the preprocessing of motor current, encoder speed and IMU signals [11].

5.4. Kalman filtering implementation

Kalman filtering needs more computing resources. Platforms equipped with an FPU are recommended to accelerate floating-point calculations. For one-dimensional or low-dimensional systems, the model can be simplified to reduce computation [5].

Parameter tuning is important. Q and R matrices determine the weight between prediction and measurement. Modular C language code can be written for transplantation across platforms [5].

5.5. Typical implementation flow

A complete signal processing flow includes five steps. First, use hardware circuits for pre-filtering. Second, acquire signals via the ADC and store the collected data. Third, use mean or low-pass filtering for preprocessing. Fourth, adopt Kalman filtering for state estimation and data fusion. Fifth, output stable data to the control loop [8].

This combination gives full play to the advantages of different algorithms. It delivers excellent noise suppression performance and satisfies real-time constraints [10].

6. Discussion

Mean filtering and low-pass filtering feature low computational complexity and high real-time performance. They are suitable for static or slow-changing signals and are preferred in resource-limited systems [4].

Kalman filtering has higher complexity and better accuracy. It performs well in dynamic state estimation and is suitable for robot motion control and multi-sensor fusion scenarios [11, 12].

SNR should be taken as the core evaluation index. Filter parameters must be adjusted according to actual noise characteristics. Embedded implementation should simplify models and optimize code to ensure real-time performance [5].

Traditional filtering techniques depend on predefined noise models and show limited performance under non-Gaussian noise conditions. A fundamental trade-off exists between filtering

DOI: 10.54254/2755-2721/2026.BA35406

latency and accuracy during high-speed motion, creating challenges for real-time robot sensing. Adaptive and intelligent filtering approaches, such as wavelet-based methods, offer potential solutions to these limitations [13].

7. Conclusion

This paper conducts a comprehensive analysis of noise characteristics, practical filtering mothods, error generation mechanisms, and embedded implementation approaches for robot sensor signals. The SNR directly reflects the actual signal quality level. Engineers can select appropriate signal processing approaches based on this practical evaluation metric. Three commonly used filtering algorithms exhibit distinct performance characteristics and application scopes. Mean filtering and low-pass filtering simple computing logic. They consume minimal system resources and effectively suppress conventional random noise and high-frequency interference. These two algorithms can be readily deployed and executed on general-purpose embedded control chips. They deliver stable processing performance for static data and slowly varying sensor signals. Kalman filtering achieves estimation accuracy. It maintains excellent dynamic tracking capability for time-varying signals and supports multi-sensor data fusion, satisfying requirements for high-precision sensing and robotic closed-loop control.

Sensor measurement deviations originate from three main sources: inherent hardware imperfections, software execution errors, and dynamic external environmental influences Effective noise reduction requires coordinated hardware protection and software-based data processing. Traditional filtering techniques rely on fixed mathematical models and thus lack adaptability to complex, nonstationary noise conditions. Users must also balance signal latency against measurement accuracy in real-world applications. Future research may explore adaptive filtering algorithms and deep learning-based denoised methods. Such advanced approaches can enhance anti-interference capacity and help robot sensor systems maintain operational stability under diverse and challenging working surroundings.

References

[1] Premnath, S., Mukund, S., Sivasankaran, K., et al. (2019). Design of an autonomous mobile robot based on sensor data fusion of LiDAR, ultrasonic sensor and wheel speed encoder. *2019 9th International Conference on Advances in Computing and Communication (ICACC)*, 62–65. IEEE.

[2] Wang, Y. (2006, March 22). Sensor signal processing flow and key technologies. *21ic Electronic Network*. https://www.21ic.com/a/1001981.html

[3] He, Z., Tang, X., Li, M., et al. (2026). Enhanced indoor mobile robot localization via Lie-group IMU–UWB fusion and dual-stage Kalman filtering. *Sensors, 26*(9), 2686.

[4] Welch, G., & Bishop, G. (2006). An introduction to the Kalman filter. *UNC Chapel Hill Technical Report*.

[5] Sommer, N. (2004). Python implementation of Kalman filter for robot sensor fusion with SNR evaluation. *arXiv*, 2024. https://arxiv.org/pdf/2403.10173.pdf

[6] Tupala, A. (2025). Analyzing the impact of inertial MEMS sensor errors on sensor fusion using novel high-level model. *Aalto University AaltoDoc*, 2025.

[7] Guo, L., & Cheng, Y. H. (2018). Design of digital filters in embedded measurement systems. *Xi'an Jiaotong University Institutional Repository*, 2018. https://ir.xjtu.edu.cn/item/100169

[8] Zhao, Q. (2004). Sensor signal conditioning and filtering technology for robot joints. *Journal of Robotics*. https://www.hindawi.com/journals/jr/

[9] Chirkov, D. G., & Stotckaia, A. D. (2020). Methods of software data filtering for working with sensors in the field of robotics. *2020 XXIII International Conference on Soft Computing and Measurements*, 121–124. IEEE.

[10] Zhang, L. (2026). Balance of real-time performance and noise suppression of six-dimensional force sensor in robot control. *CSDN Library*. https://wenku.csdn.net/answer/60dajmc3q4gw

DOI: 10.54254/2755-2721/2026.BA35406

[11] Yucel, F., Akkus, O., & Dikici, Y. (2025). Optimization of filter parameters for a MEMS IMU accelerometer using an artificial immune system. In *2025 6th International Conference on Artificial Intelligence, Robotics and Control (AIRC)*, 1–6.
[12] Kwak, J., et al. (2020). Impacts of traffic signal timing optimization on vehicular fuel consumption and emissions on an urban corridor. *IEEE Transactions on Intelligent Transportation Systems*.
[13] Dell'Olio, F., et al. (2025). Wavelet-based real-time denoising for gyroscopes in robotic systems. *Politecnico di Bari Iris*. https://iris.poliba.it