JEECS (Journal of Electrical Engineering and Computer Sciences)
Vol. 10, No. 1, June 2025, pp. 9-18
e-ISSN: 2579-5392 p-ISSN: 2528-0260

Smoothing Algorithm Using Censor Distance on KRSRI ITN Malang Robot with Moving Average Method

Waradana Adikasani [1], Sotyohadi [2], M. Ibrahim Ashari [3]

[1,2,3]Institut Teknologi Nasional Malang, Jl. Raya Karanglo Km 2, Tasikmadu, Kecamatan Lowokwaru, Kota Malang, Indonesia

Article Info

Article history:
Received: 3 March 2025
Revised: 25 March 2025
Accepted: 11 April 2025

Keyword:
Smoothing Algorithm
Simple Moving Average
Distance Sensor
KRSRI Robot
Robot Navigation

Abstract

Distance sensors play a crucial role in robotic navigation systems, particularly in the Indonesian Search and Rescue Robot Contest (KRSRI). However, sensor readings often suffer from instability due to external disturbances such as robot leg movements and environmental interference. This research aims to address these issues by implementing a smoothing algorithm based on the Simple Moving Average (SMA) method to refine distance sensor readings from the GP2Y Infrared Proximity sensor, enabling the robot to detect obstacles more accurately. Experiments were conducted under various conditions, both with and without the smoothing algorithm, showing that the use of SMA significantly improves the stability and accuracy of sensor data. The results indicate that sensor readings with SMA exhibited minimal fluctuations, maintaining higher consistency around the actual measured distance, while readings without SMA showed significant variability and inaccuracies. The implementation of SMA successfully reduced measurement errors, with an average error reduction of 97.68% compared to the raw sensor data. This improvement ensures more reliable obstacle detection and navigation performance, thereby enhancing the robot's effectiveness in competitions as well as search and rescue applications in real-world environments.

Corresponding author:
Waradana Adikasani,
waradana.adikasani2002@gmail.com

DOI: https://doi.org/10.54732/jeecs.v10i1.2

1. Introduction

Robots continue to evolve with advancements in artificial intelligence, machine learning, and sensor technology, enabling them to perform increasingly complex tasks. These intelligent machines are widely utilized across various fields, including healthcare, agriculture, logistics, and space exploration. In Indonesia, one of the most renowned robotics competitions is the Kontes Robot Indonesia (KRI), which serves as a platform for students and researchers to showcase their innovations in robotics. This competition is designed to encourage technological development and problem-solving skills, supported by various academic institutions and government agencies [1].

KRSRI, or the Indonesian Search and Rescue Robot Contest, is one of the categories that focuses on rescue robots. These robots are tasked with rescuing and transporting victims to a safe zone after a disaster [2]. A legged robot is a type of robot capable of moving flexibly due to its ability to shift positions, supported by legs specifically designed as actuators. A legged robot consists of actuators on each leg, with each leg having three degrees of freedom (DOF) [3]. The robot is designed using a hexapod model, where each leg is equipped with three servos [4]. Compared to wheeled robots, hexapod robots offer superior mechanical advantages in navigating rough and uneven terrain [5]. This hexapod robot is a six-legged spider-like robot that utilizes Dynamixel or servo motors and sensors for movement and operation [6].

To improve the accuracy of distance sensor measurements, a smoothing method is applied to minimize noise and provide more stable readings. Sensor data often fluctuates due to environmental factors, electrical interference, or mechanical vibrations, which can lead to inconsistencies in measurement. One widely used technique for smoothing is the moving average method, which involves taking a set of recent observed values and computing their average to generate a more stable and reliable estimate for the subsequent period [7]. This method reduces random fluctuations, ensuring more accurate sensor readings. Its effectiveness can be evaluated using the Mean Squared Error (MSE), where a lower MSE indicates better noise reduction while preserving accuracy. In robotics applications, precise sensor data is essential for navigation, obstacle detection, and object tracking. Without smoothing, the robot may misinterpret sudden variations as obstacles, leading to unnecessary motion corrections and reduced efficiency.

In this research, the implementation of a smoothing algorithm using the moving average method will be explained for distance sensor readings on a robot used in the KRSRI competition at Institut Teknologi Nasional (ITN) Malang. This approach is intended to address the issue of distance sensor readings being obstructed by the robot's legs during movement.

2. Research Methodology

2.1. System Design

In this study, the infrared sensor plays a crucial role in the robotic system, as it detects surrounding objects by transmitting and receiving infrared signals. These signals enable the robot to perceive its environment effectively[8]. To ensure stable sensor readings, a smoothing algorithm is applied using the Moving Average method. Several research studies have demonstrated that smoothing methods are widely utilized for data forecasting. For instance, in the journal titled "Analysis of Accuracy Comparison Between Moving Average and Exponential Smoothing for Revenue Forecasting in XYZ Company," the Moving Average method was used to predict company revenue, showing that this approach yields high accuracy result [9].

Similarly, another study titled *"Implementation of an Inventory Forecasting System Using the Moving Average Method"* applied the Moving Average method to forecast inventory levels, demonstrating its effectiveness in assisting managers in making inventory-related decisions. The study showed that by using historical inventory data, the Moving Average method could provide more accurate demand predictions, reducing stock shortages and overstock issues [10].In this research, the author focuses on the smoothing algorithm by implementing the Simple Moving Average (SMA) method. The Simple Moving Average is a forecasting technique that calculates the average over a specific time period without applying any weighting [11].

This method is particularly effective for unstable data, as it does not account for trends or weight distributions. By smoothing short-term fluctuations, it provides a clearer representation of the overall pattern, making it useful for applications where sudden variations need to be minimized[12].The use of SMA in this study is justified by the nature of the robot, which requires efficient movement supported by its leg structure . In dynamic environments, where real-time adaptability is crucial, SMA provides a simple yet reliable approach to data smoothing [13].

2.2. Hardware Design

The Figure 1 below is the block diagram of the design of this tool, which aims to obtain the scheme or circuit of the tool being created.

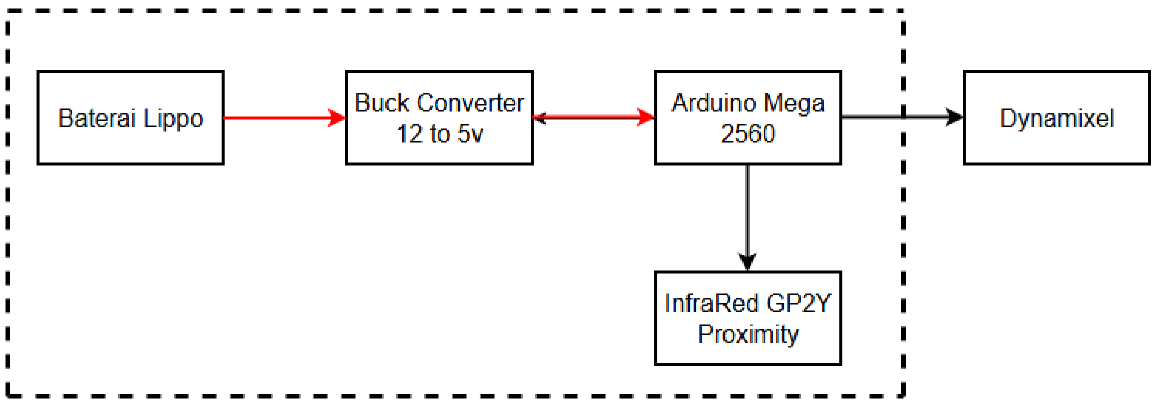

Figure 1. Hardware Design

The block diagram in Figure 1 presented illustrates an electronic system comprising key components, including a power source, microcontroller, sensor, and actuator. The system is powered by a 12V LiPo battery, which cannot be directly utilized by the Arduino Mega 2560 and the proximity sensor, necessitating voltage conversion. A buck converter is used to step down the voltage from 12V to 5V, ensuring compatibility with the microcontroller and sensor [14]. The Arduino Mega 2560 functions as the central controller, receiving input signals from the proximity infrared GP2Y sensor and processing the data to regulate the movement of the Dynamixel actuator. Designed as an open-source microcontroller, the Arduino Mega 2560 features 74 pins, including analog input pins (A0-A15), digital input/output pins (0-53), 4 UART interfaces, USB connectivity, an ICSP header, a reset button, and an extended sketch memory, facilitating its usability in various applications [15].

The proximity infrared GP2Y sensor detects objects by emitting infrared signals and measuring the reflected light to determine distance. Operating within a voltage range of 4.5V to 5.5V, this sensor can measure distances from 10 cm to 80 cm with precision[16]. Upon detecting an object, the sensor transmits data to the Arduino Mega 2560, which interprets the information and determines whether to activate the Dynamixel actuator. The Dynamixel actuator, responsible for generating motion, operates based on the control signals received from the microcontroller.

2.3. Software Design

The system Software Desain in this research, as presented in Figure 1, illustrates the smoothing algorithm process. The flowchart in Figure 2 presented the workflow of a system that reads distance sensor data, smooths the data using the Moving Average method, and evaluates the results. The process begins with the START phase, where the system is activated either by powering on the device or running the program. Next, in the System Initialization phase, the system performs initial setup, including activating the distance sensor, allocating memory for data storage, and setting parameters such as error tolerance limits and the number of samples for Moving Average calculations. The system then proceeds to Read Distance Sensor Data, where raw sensor readings are collected in specific units (e.g., centimeters or meters). Since sensor data often contains fluctuations due to environmental noise, further processing is required.

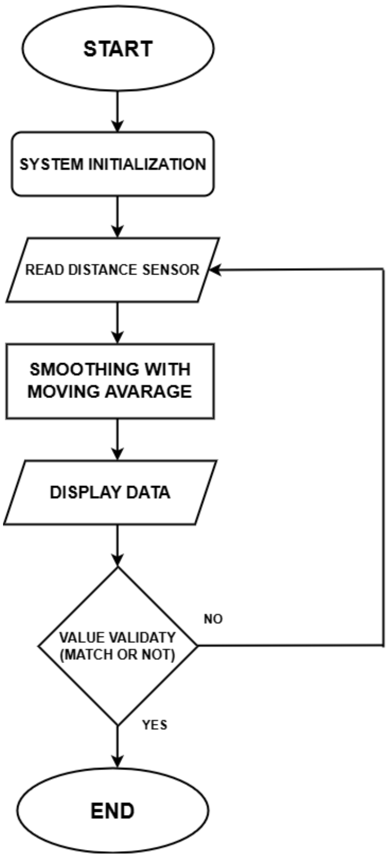

Figure 2. Software Design

The Smoothing with Moving Average phase is implemented to reduce noise by computing the average of the most recent data points, producing a more stable value. After smoothing, the processed data is displayed in the Display Smoothed Data phase, either numerically on a screen or visually as a graph in a user interface. The system then moves to the Evaluation Phase, checking whether the smoothed value meets predefined criteria, such as remaining within an acceptable range or ensuring that variations over time are not excessive. If the criteria are met (YES), the process terminates, leading to the END phase, indicating that the stabilized data is ready for further use. If the criteria are not met (NO), the system loops back to the Read Distance Sensor Data phase and repeats the process until a valid value is obtained.

2.4. Algorithm Design

Figure 3 provides an explanation of the algorithm design. The flowchart illustrates the Simple Moving Average (SMA) smoothing algorithm used for processing sensor data. The process begins with the Start node, followed by System Initialization, where necessary system components, including sensors, are set up. Next, the algorithm proceeds to Input Data, where the system receives raw sensor readings. The Read Distance Sensor Data step captures the latest measurement from the sensor. This data is then stored in a temporary buffer in the Add Data to SMA Buffer process, ensuring a sliding window of recent values is maintained. The Compute SMA Value step calculates the moving average by summing the latest N data points and dividing by N. The smoothed output is then displayed in the Display Data Smoothed step.

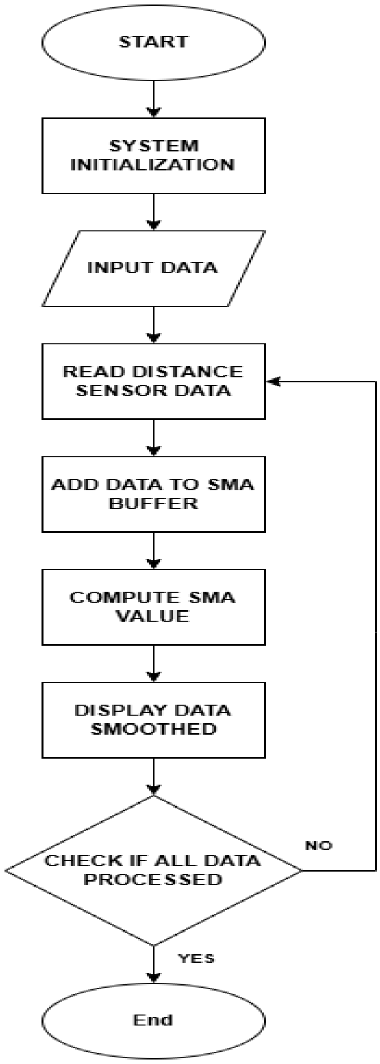

Figure 3. Algorithm Design

A decision is made at Check If All Data Processed, verifying whether the entire dataset has been processed. If Yes, the process terminates at End. If No, the algorithm loops back to Read Distance Sensor Data, ensuring continuous data acquisition and smoothing until completion. This structured approach effectively minimizes noise and fluctuations, enhancing the accuracy and reliability of sensor data interpretation.

3. Results and Discussions

3.1 Hardware and design result

Figures 4 and 5 show the robot design from the top and side views, highlighting several key components used in this research. The primary controller is an Arduino Mega, integrated with two GPDY Infrared Proximity Sensors connected according to the system requirements to support the sensor operation. This process aims to ensure that all hardware components are properly integrated and capable of performing their functions as intended by the researcher. In this experiment, two sensors are utilized to collect data and obtain the desired results.

3.2 Smoothing Algorithm

The smoothing algorithm is a technique used to reduce disturbances or fluctuations in data generated by sensors, resulting in smoother and more stable readings. In robotic systems, particularly in the use of distance sensors, readings are often affected by environmental disturbances or the robot's own movements, such as vibrations or impacts caused by the robot's leg movements. These factors can lead to unstable or inaccurate sensor readings.

The smoothing algorithm is applied to minimize these disturbances, ensuring that the sensor data obtained is more reliable for robot navigation and decision-making processes. In research involving hexapod robots, the robot's legs often become sources of interference, as they can obstruct the distance sensors during movement. Smoothing is essential to address this issue, ensuring that the robot can measure distances more accurately despite movement-induced disturbances affecting the sensors.

Figure 4 Top view of the robot

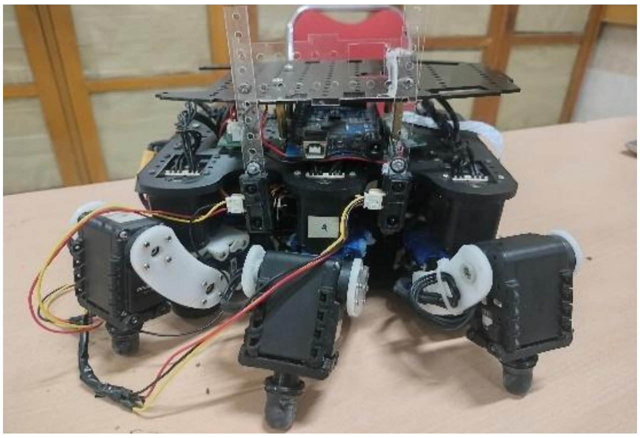

Figure 5. Side view of the robot

In this context, the smoothing algorithm processes a series of previous sensor readings and smooths the data to reduce the impact of sudden changes or extreme fluctuations. This approach allows for more consistent readings, enabling the robot to avoid obstacles or detect surrounding objects more effectively. Overall, the smoothing algorithm plays a crucial role in enhancing the accuracy and stability of sensor readings, particularly in hexapod robots that use infrared distance sensors for obstacle detection during movement across uneven or interference-prone terrain.

3.3 Simple Moving Average Method (SMA)

The Simple Moving Average (SMA) is one of the most basic and widely used smoothing techniques for reducing fluctuations in data generated by sensors. In the context of sensor readings in robotic systems, the SMA is employed to smooth sensor data, which is often affected by internal disturbances. The SMA operates by calculating the average of a set of previous observations, producing a more stable output value. This makes the SMA particularly useful in scenarios where sensor readings need to be refined to minimize noise.

In robotics, especially for robots operating on uneven terrain, sensor readings often become unstable due to vibrations or the robot's movements. For instance, in a hexapod robot equipped with infrared distance sensors, leg movements can interfere with sensor signals, leading to inaccurate readings. The SMA method is applied to smooth out these sensor data, allowing the robot to make more accurate decisions regarding navigation and obstacle detection.

The Simple Moving Average (SMA) is one of the most basic and widely used smoothing techniques for reducing fluctuations in data generated by sensors. In the context of sensor readings in robotic systems, the SMA is employed to smooth sensor data, which is often affected by internal disturbances. The SMA operates by calculating the average of a set of previous observations, producing a more stable output value. This makes the SMA particularly useful in scenarios where sensor readings need to be refined to minimize noise[17]. The SMA formula used in this research is expressed as follows:

$$SMA\ \mathrm{n} = \frac{X1+X2+\cdots.+Xn}{\mathrm{n}} \tag{1}$$

where is the Simple Moving Average at time , are the sensor readings from previous time steps, and is the number of periods used for averaging. For example, if the sensor records the following five distance measurements (in cm): 45.01, 45.20, 44.98, 45.00, 44.80
The SMA calculation for this dataset would be:

$$SMA\ \mathrm{n} = \frac{45.01+45.20+44.98+45.00+44.80}{5} = \frac{224.99}{5} = 44.998 \tag{2}$$

Thus, the smoothed distance reading is approximately 44.998 cm, reducing the impact of sudden fluctuations in raw data.

3.4 Mean Squared Error (MSE)

The Mean Squared Error (MSE) is a widely used method for evaluating the accuracy of a predictive model by measuring the average squared difference between actual and predicted values. This method is particularly useful in assessing the performance of algorithms that process sensor data, as it quantifies the level of deviation between measured and estimated values. A lower MSE value indicates a higher accuracy, meaning the predicted values are closer to the actual sensor readings.

In sensor applications, particularly in robotics, sensor readings often experience fluctuations due to noise, environmental factors, or mechanical movements. The MSE method helps determine the effectiveness of a filtering technique by evaluating how much deviation remains after processing. It is calculated using the following formula:

$$MSE = \frac{1}{n}\sum_{I=0}^{n}(yi - y"i)^2 \tag{3}$$

This Formula demonstrates how much the estimated values deviate from actual sensor readings. The MSE method provides a numerical representation of error, making it valuable for comparing different filtering techniques. However, since MSE squares the differences, it is highly sensitive to large errors, which can significantly impact the overall result. Therefore, while MSE is useful for evaluating prediction accuracy, it should be used in conjunction with other smoothing techniques, such as Simple Moving Average (SMA), to enhance the reliability of sensor data processing.

3.5 Smoothing Algorithm in Robotics

The configuration of the smoothing algorithm using the Simple Moving Average (SMA) method in this study aims to address the instability of sensor reading data caused by external disturbances, such as robot leg movements and environmental interference. By calculating the average of a series of previous data readings, the SMA method effectively reduces fluctuations and provides more stable and consistent data. This is crucial for ensuring that infrared sensor distance measurements are more accurate and reliable, particularly in the context of robots used for navigation and obstacle avoidance missions. Moreover, stable sensor data allows the robot to maintain more predictable movement patterns, reducing abrupt changes in trajectory that could otherwise compromise navigation efficiency.

The implementation of the smoothing algorithm also enhances the overall performance of the robot, enabling more effective navigation in challenging terrains. Sensor readings processed with SMA not only improve obstacle detection accuracy but also simplify the robot's control system in making decisions. By minimizing the influence of sudden sensor variations, the robot can react more efficiently to environmental conditions, reducing unnecessary corrections and improving energy efficiency. Furthermore, a more stable data stream helps in optimizing feedback control mechanisms, allowing smoother motion planning and real-time adjustments to changes in the surroundings.

Additionally, the integration of the SMA method in robotics is beneficial for reducing computational complexity compared to other filtering techniques, such as Kalman filters, which require more processing power. This makes SMA an ideal choice for low-power embedded systems commonly used in autonomous robots. Moreover, the method's adaptability allows for easy implementation in various robotic applications, ranging from industrial automation to autonomous vehicles.

3.6 Sensor Reading Results With and Without Smoothing Algorithm Implementation

In the following experiment, a comparison table is presented, showing data from a robot in a stationary position with a sensor-to-wall distance of 45 cm and 55 cm. This data compares measurement results using the Simple Moving Average (SMA) method, Mean Squared Error (MSE) method, and without any algorithm to evaluate the effectiveness of each method in stabilizing sensor readings

Table 1. Sensor reading result 45 cm

No.	With SMA (Cm)	With MSE (Cm)	Without Algorithn(Cm)	Time	Distance
1	45,01	44.50	45.02		
2	45,20	43.78	2.02		
3	44,98	44.10	2.03		
4	45.00	44.25	45.20		
5	44.80	43.90	3.03		
6	44.90	43.00	1.02		
7	45.10	44.50	45.37		
8	45.07	44.77	2.13	10 Minute	45 CM
9	45.02	43.40	2.12		
10	45.15	44.89	2.45		
11	45.77	43.10	45.02		
12	48.96	45.10	45.01		
13	45.01	45.30	3.28		
14	45.20	43.15	2.98		
15	45.12	43.16	45.45		

Table 1 presents the sensor reading results at a 45 cm distance using three different methods: Simple Moving Average (SMA), Mean Squared Error (MSE), and raw data without any algorithm. The results indicate that the SMA method provides more stable and accurate readings, effectively filtering out noise and fluctuations. The recorded values using SMA remain consistently close to the expected 45 cm, demonstrating its efficiency in smoothing sensor data. This stability is crucial for applications requiring precise distance measurements, such as robotic navigation and obstacle detection.

In contrast, the MSE method exhibits noticeable deviations from the expected values, with some readings significantly lower than 45 cm. While MSE is designed to minimize squared errors, its performance in this scenario suggests that it is less effective in handling real-time sensor fluctuations compared to SMA. The presence of lower readings indicates that MSE may introduce additional inaccuracies due to its

JEECS (Journal of Electrical Engineering and Computer Sciences)
Vol. 10, No. 1, June 2025, pp. 9-18
e-ISSN: 2579-5392 p-ISSN: 2528-0260

sensitivity to outliers and noise in the dataset. Although MSE offers an improvement over raw data, its inability to maintain consistency limits its reliability in critical applications.

The sensor readings without any smoothing algorithm reveal highly inconsistent values, including extreme variations that significantly deviate from the expected measurements. The lack of filtering results in erratic data points, highlighting the impact of environmental factors and sensor noise. These fluctuations render the raw data unreliable for precise measurement tasks.

Table 2. Sensor Reading 55 CM

No.	With SMA (Cm)	With MSE (Cm)	Without Algorithm (Cm)	Time	Distance
1	55.02	54.60	55.02	10 Minute	55 CM
2	55.19	54.20	2.20		
3	54.88	53.90	2.33		
4	55.00	54.25	55.20		
5	54.90	53.75	3.13		
6	54.70	54.55	1.12		
7	55.09	53.75	55.37		
8	55.03	53.50	2.10		
9	55.01	54.55	2.11		
10	55.10	54.30	2.44		
11	58.77	54.40	55.02		
12	58.98	55.00	55.01		
13	55.01	54.85	3.29		
14	55.17	54.70	2.90		
15	55.16	55.40	55.45		

Table 2 presents the sensor reading results at a 55 cm distance using three different methods: Simple Moving Average (SMA), Mean Squared Error (MSE), and raw data without any algorithm. The SMA method consistently produces values close to the expected 55 cm, demonstrating its effectiveness in reducing noise and maintaining data stability. The filtered values remain within a narrow range, ensuring precise measurement accuracy, which is essential for applications requiring reliable distance estimation.

On the other hand, the MSE method shows significant deviations from the expected value, with several readings below 55 cm. This suggests that MSE is more sensitive to fluctuations and outliers, leading to inconsistent results. While MSE provides an improvement over raw data, its performance is still inferior to SMA in terms of stability and accuracy. The observed variations indicate that MSE may not be the most suitable method for filtering sensor readings in real-time applications.

The raw sensor data, without any applied algorithm, exhibits extreme fluctuations and inconsistent values, making it unreliable for precise distance measurements. The presence of significant deviations highlights the impact of environmental noise and sensor inaccuracies. Compared to the other methods, SMA remains the most effective approach, as it minimizes fluctuations while maintaining values close to the actual distance. This confirms its superiority in sensor data processing, ensuring better reliability in practical implementations..

3.7 Evaluation results

The evaluation of sensor reading performance is carried out by comparing three different methods Simple Moving Average (SMA), Mean Squared Error (MSE), and raw sensor data without any filtering algorithm. The comparison is based on sensor readings taken at two fixed distances, 45 cm and 55 cm, over a duration of 10 minutes. The primary objective of this evaluation is to determine which method provides the most accurate and stable measurement while minimizing fluctuations and noise.

From Table 1 (Sensor Reading at 45 cm) and Table 2 (Sensor Reading at 55 cm), it is observed that the SMA method produces the most stable and consistent readings, with values closely matching the expected distances. SMA functions as a smoothing technique that reduces the impact of sudden fluctuations caused by sensor noise. In both datasets, the SMA values remain within a narrow range and closely align with the reference distances, confirming its effectiveness in improving measurement precision.

On the other hand, the MSE method exhibits significant deviations in the sensor readings. Many values are lower than the actual distances, particularly in Table 1, where the MSE values range between 43.10 cm and 44.89 cm, and in Table 2, where the MSE values range between 53.50 cm and 55.40 cm. This suggests that while MSE is effective in reducing large errors, it remains highly sensitive to fluctuations and does not provide the same level of stability as SMA. The tendency of MSE to generate lower values indicates a potential underestimation of distance, which could lead to inaccuracies in applications requiring precise measurements.

Meanwhile, the raw sensor readings (without any filtering algorithm) demonstrate extreme variations and instability, with some values deviating significantly from the expected distance. For instance, in Table 1, the raw readings range from 2.02 cm to 45.45 cm, while in Table 2 the values range from 1.12 cm to 55.45 cm. These inconsistencies highlight the significant impact of environmental noise and sensor inaccuracies when no data filtering method is applied. Such large variations render raw sensor data unreliable for any application requiring high precision.

Based on the evaluation, SMA remains the most effective technique for sensor data processing, as it provides the most accurate, stable, and noise-free measurements. MSE, while offering some improvements over raw data, does not perform as well as SMA due to its sensitivity to noise and tendency to underestimate values. The raw sensor data, with its high fluctuation and extreme deviations, is the least reliable method for accurate measurement. SMA is the preferred method for ensuring precise and consistent sensor readings, making it highly suitable for real-world applications that require stable distance measurement, such as robotic navigation, industrial automation, and obstacle detection systems.

4. Conclusion

Based on the objectives and results of several tests conducted, it can be concluded that the implementation of a smoothing algorithm using the Simple Moving Average (SMA) method effectively stabilizes distance sensor readings on the KRSRI robot, reducing fluctuations caused by external disturbances such as leg movements and environmental interference. The infrared proximity sensor integrated with the Arduino Mega demonstrates accurate readings at distances between 0–50 cm, with a low average error percentage. The sensor reading experiments conducted at distances of 45 cm and 55 cm showed that using the smoothing algorithm resulted in an average error percentage significantly lower than when the algorithm was not applied. The application of the SMA method successfully reduces data fluctuations, ensuring more consistent and stable readings, which improves the robot's ability to detect obstacles accurately. Sensor readings without the smoothing algorithm exhibited significant instability due to internal and external factors, highlighting the importance of data filtering for reliable sensor performance. The use of the SMA method enhances the robot’s navigation capabilities, allowing for more precise decision-making, especially in environments with obstacles or during movement. This smoothing approach supports the robot's overall performance in search and rescue tasks, enabling more efficient and accurate movement in real-world applications and competitions.

References

[1] R. A. Fatekha, B. S. B. Dewantara, and H. Oktavianto, “Sistem Deteksi Bola pada Robot Kiper Pemain Sepakbola Beroda,” *Jurnal Integrasi*, vol. 13, no. 2, pp. 127–134, 2021, doi: 10.30871/ji.v13i2.3133.
[2] R. Y. Adhitya, “Sistem Pengendalian Robot KRSRI Menggunakan Logika Fuzzy Sugeno Orde Nol,” *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 1, pp. 159–168, 2024, doi: 10.33795/elkolind.v11i1.4057.
[3] F. Faqri, W. Amaral, and J. Subur, “Implementasi Metode PID dan Inverse Kinematics pada Pergerakan Robot Berkaki Enam (Hexapod),” *SinarFe7 Seminar Nasional Fortei Regional 7*, vol. 6, no. 1, pp. 17–22, 2024.
[4] A. Ilham, A. Rahman, R. Maulana, and H. Fitriyah, “Kinematika Robot dengan Sistem Kendali Proportional Integral dan Derivative menggunakan Sensor IMU pada Robot Hexapod,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , vol. 6, no. 7, pp. 3316–3323, 2022.
[5] J. Subur, S. Suryadhi, and E. Arohman, “Penerapan Metode Fuzzy Logic-PID Sebagai Kontrol Gerak Robot Berkaki Enam (Hexapod),” *Proceedings Series on Physical & Formal Sciences*, vol. 6, pp. 98–108, 2023, doi: 10.30595/pspfs.v6i.858.
[6] R. Darwis, I. Arifianto, A. Mujadin, and S. Rahmatia, “Perancangan Robot Pemadam Api Hexapod,” *Jurnal Al-Azhar Indonesia Seri Sains dan Teknologi*, vol. 5, no. 1, pp. 1–4, 2019, doi: 10.36722/sst.v5i1.316.
[7] R. Rachman, “Penerapan Metode Moving Average Dan Exponential Smoothing Pada Peramalan Produksi Industri Garment,” *Jurnal Informatika*, vol. 5, no. 2, pp. 211–220, 2018, doi: 10.31311/ji.v5i2.3309.
[8] P. W. Ginta and R. F. Milati, “Robot Pendeteksi Dan Penghitung Jalan Berlobang Menggunakan Sensor Infra Merah Berbasis Mikrokontroler At89S51,” *Jurnal Media Infotama*, vol. 7, no. 1, pp. 69–83, 2011.
[9] M. M. Azman, “Analisa perbandingan nilai akurasi moving average dan exponential smoothing

untuk sistem peramalan pendapatan pada perusahaan XYZ," *Jurnal Sistem dan Informatika*, vol. 13, no. 2, pp. 36–45, 2019.

[10] W. Wulandari, "Implementasi Sistem Peramalan Persediaan Barang Menggunakan Metode Moving Average," *Jurnal Media Informatika Budidarma*, vol. 4, no. 3, pp. 707–714, 2020, doi: 10.30865/mib.v4i3.2199.

[11] A. Apriliani, H. Zainuddin, A. Agussalim, and Z. Hasanuddin, "Peramalan Tren Penjualan Menu Restoran Menggunakan Metode Single Moving Average," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 6, pp. 1161–1168, 2020, doi: 10.25126/jtiik.2020722732.

[12] H. B. Suryadipraja and J. Jamaaluddin, "Pengoptimalan Penggunaan Sensor Uvitron Pada Robot Berkaki Dalam Mendeteksi Keberadan Titik Api Dengan Sebuah Lilin," *SinarFe7*, vol. 2, no. 1, pp. 206–210, 2019.

[13] A. S. Samosir, N. I. Tohir, and A. Haris, "Rancang Bangun Catu Daya Digital Menggunakan Buck Converter Berbasis Mikrokontroler Arduino," *ELECTRICIAN Jurnal Rekayasa dan Teknologi Elektro*, vol. 11, no. 1, pp. 1–94, 2017, doi: https://doi.org/10.23960/elc.v11n1.2014.

[14] S. Siswanto, M. Anif, D. N. Hayati, and Y. Yuhefizar, "Pengamanan Pintu Ruangan Menggunakan Arduino Mega 2560, MQ-2, DHT-11 Berbasis Android," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 1, pp. 66–72, 2019, doi: 10.29207/resti.v3i1.797.

[15] F. A. Deswar and R. Pradana, "Monitoring Suhu Pada Ruang Server Menggunakan Wemos D1 R1 Berbasis Internet of Things (Iot)," *Jurnal Ilmiah "Technologia,"* vol. 12, no. 1, pp. 25–32, 2021, doi: 10.31602/tji.v12i1.4178.

[16] I. H. Santoso and A. I. Irawan, "Analisis Perbandingan Kinerja Sensor Jarak HC-SR04 dan GP2Y0A21YK Dengan Menggunakan Thingspeak dan Wireshark," *Jurnal Rekayasa Elektrika*, vol. 18, no. 1, pp. 43–52, 2022, doi: 10.17529/jre.v18i1.23359.

[17] L. Fauziah and F. Fauziah, "Penerapan Metode Single Exponential Smoothing dan Moving Average Pada Prediksi Stock Produk Retail Berbasis Web," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 7, no. 2, p. 159, 2022, doi: 10.30998/string.v7i2.13932.