IJRCS

ASCEE

Fish Swarmed Kalman Filter for State Observer Feedback of Two-Wheeled Mobile Robot Stabilization

Ahmad Fahmi [a,1], Marizan bin Sulaiman [a,2], Indrazno Siradjuddin [b,3,*], Anindya Dwi Risdhayanti [c,4]

[a] Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, Melaka Malaysia, Malaysia
[b] Department of Electrical Engineering, State University of Malang, Indonesia
[c] Electronic Engineering Study Program, State Polytechnic of Malang, Indonesia
[1] ahmad.fahmi.ft@um.ac.id; [2] marizan@utem.edu.my; [3] indrazno@polinema.ac.id; [4] risdhayanti@polinema.ac.id
* Corresponding Author

ARTICLE INFO

ABSTRACT

Self-balancing robots are designed to maintain an upright position without toppling over. By continuously adjusting their center of mass, they can maintain stability even when disturbed by external forces. This research aims to achieving and maintaining balance is a complex task. Self-balancing robots must accurately sense their orientation, calculate corrective actions, and execute precise movements to stay upright. Eliminating disturbances and measurement noise in self-balancing robot can enhance the accuracy of their output. One common technique for achieving this is by using Kalman filters, which are effective in addressing non-stationary linear plants with unknown input signal strengths that can be optimized through filter poles and process covariances. Additionally, advanced Kalman filter methods have been developed to account for white measurement noise. In this research, state estimation was conducted using the Fish Swarm Optimization Algorithm (FSOA) to provide feedback to the controller to overcome the effects of disturbances and noise in the measurements through the designed filter. FSOA mimics the social interactions and coordinated movements observed in fish groups to solve optimization problems. FSOA is primarily used for optimization tasks where finding the global optimal solution is desired. The results show that the use of an optimized Kalman filter with FSOA on a two-wheeled mobile robot to handle system stability reduces noise values by 38.37%, and the system reaches a steady state value of 3.8 s with a steady error of 0.2%. In addition, by using the proposed method, filtering disturbances and measurement noise in self-balancing robot can help improve the accuracy of the self balancing robot's output. System response becomes faster towards stability compared to other methods which are also applied to two-wheeled mobile robots.

1. Introduction

Technological development in the field of robots, especially mobile robots, has been rapidly increasing in the last few decades, specifically in the features of speed, accuracy, and stability. These features are benchmarks for technological advances in the field of robotics. This subsection reviews

and examines the relevant literature on self-balancing robots in accordance with research purposes. Self-balancing is a widely investigated topic due to its potential for advancing robotics, enabling practical applications, enhancing human-robot interaction, fostering research and development, and driving technological progress. The ability to maintain balance is a fundamental requirement for robots to operate effectively and safely in complex and dynamic environments. Self-balancing robots require robust and efficient control systems to maintain stability and respond to disturbances. The required control systems design that incorporate sensor integration, actuator control, feedback loops, and signal processing techniques. This process helps advance the field of control system engineering by exploring new approaches, optimizing system performance, and ensuring reliability.

Self-balancing has become a widely investigated topic in the last decade, possessing a significant role in the field of robotics and control system engineering [1]. By explaining the inverted pendulum where the self-balancing robot has a performance that resembles a pendulum, this study mainly examines the self-balancing robot model and its origins. There are several control systems that can be used on self-balancing robots. Therefore, this study describes the control system used and their respective mathematical models. The control systems used include Linear Quadratic Regulator (LQR), Pole placement, Proportional-Integral-Derivative (PID), state estimator, and Kalman filter as an estimator.

This study focuses on the efficiency of different control algorithms, such as Proportional, Integral, and Derivative (PID), pole placement, and adaptive control, in an artificial robot called Bimbo. The robot is built with modules for movement and position control. In this robot, we applied a Kalman filter to get the Roll angle of the Inertial Measurement Unit (IMU) [2]. The balance and tracking control of a two-wheeled mobile robot has been studied and analyzed. Numerical simulations to test the effectiveness of each controller has been carried out with three types of control strategies, including PID-based parallel dual feedback, serial dual feedback, and hybrid optimal control using Feed-forward PID and Feedback LQR. For different set points, the hybrid controller can achieve better performance than the other two methods. Better tracking and position balancing have been detected in the hybrid control algorithm, with zero overshoot and a lower settling time.

Recent research has explored the use of metaheuristic algorithms as state observers in deterministic systems without incomplete information about the state vector. For example, research on a mini-batch adaptive method of random search (MAMRS) has been tested on a satellite stabilization problem to find the optimal control. Besides, metaheuristic algorithms have also been used in the field of power system state estimation due to their versatility and adaptability. For example, CSA has been proposed for optimal power system state estimation, while SKF has been used to make estimations of the system's states using a population of agents. GA has also been used to improve the search speed of the parameters in motors using dynamic relationships and parameter estimation. Overall, metaheuristic algorithms are reported as effective tools for state estimation due to their ability to adapt to different problems [3]-[5].

In addition, filtering disturbances and measurement noise in self-balancing robot can help improve the accuracy of the self balancing robot's output [6]-[7]. Kalman filters are often used for this purpose since they account for non-stationary linear plants with unknown intensities of input signals [8]. Besides, the filters can be tuned using filter poles and process covariances. Additionally, improved Kalman filter methods have been developed for white measurement noise [9]. In this research, state estimation was carried out using the Fish Swarm Optimization Algorithm (FSOA) and fed back to the controller. It is expected that the designed filter will overcome the disturbance and noise measurements [10]. By combining FSOA and Kalman filters, the optimization capabilities of FSOA are utilized to search for the global optimal solution, while the state estimation abilities of the Kalman filter refine the estimation of the system's state based on available measurements [11]-[12]. This integration can be beneficial in control system optimization problems where both parameter tuning and state estimation are crucial for achieving desired outcomes [13].

2. Method

2.1. Kalman Filter

Kalman Filter is a mathematical algorithm that optimizes the use of imprecise data in a linear system with Gaussian error which continuously updates the best estimate or estimate of the state of the system [14]. Kalman Filter estimates the state of the process at a time and then takes the feedback in the form of a noise output. The Kalman filter represents the dynamic system in a state-space form, consisting of state variables, measurements, and control inputs [15]. The state variables describe the internal system states, while the measurements provide noisy or incomplete observations of the system [16]. The Kalman filter predicts the current state of the system based on the previous state estimate and the system dynamics. It utilizes the state transition matrix and control inputs (if available) to propagate the state forward in time, providing an estimate of the system's future state. The measurement update step is the core of the Kalman filter [17]. It combines the predicted state estimate with the actual measurements, taking into account the uncertainty of both the measurements and the predicted state [18].

The Kalman gain, computed from the covariance matrices of the state and measurement noise, determines the weight given to the measurements in the update process. The Kalman filter operates in a recursive manner, continuously updating the state estimate as new measurements become available. By incorporating feedback from previous estimates, the filter adapts to changing system conditions, providing real-time estimation of the system state [19].

In the Kalman Filter there is a time update (prediction equation) that can predict the value of the state vector which is then used for feedback. Calculation or measurement update or observation update is a correction equation that is used as a state vector correction. So, the Kalman Filter process of prediction and correction occurs. Kalman Filter will predict a state space and will correct the value of whether the system output is the same as the desired input [20]-[21].

The Kalman filter operates in a recursive manner, continuously updating the state estimate as new measurements become available. By incorporating feedback from previous estimates, the filter adapts to changing system conditions, providing real-time estimation of the system state [22]-[25]. The Kalman Filter shows the general problem of estimating a state in a discrete time control process governed by the equation of a stochastic linear change, in general it can be expressed in the equation of state:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{1}$$

With a measurement $z \in Rm$.

$$z_k = Hx_k + v_k \tag{2}$$

And output equation in (3).

$$y_k = Cx_k + z_k \tag{3}$$

Where:
A   = state matrix $n \times n$
B   = $n \times 1$ matrix that relates the control input to a state
C   = output matrix
x_k = process state vector $n \times 1$ at time kt
w_k = process noise vector which is assumed to be a white series with known covariance structure
v_k = measurement noise vector which is assumed to be a white series with a known
        covariance structure and zero cross-correlation with the w_k series
y_k = output vector
u_k = input vector
z_k = measurement vector at time t_k
H_k = $m \times n$ matrix which gives an ideal relationship between measurements and vectors

The random variables w_k and v_k describe process and measurement noise. These variables are assumed to be independent of one another, white (white) and with a normal probability distribution. The Kalman Filter equation is divided into two parts, namely the time update equation and the measurement update equation. The time update equations account for the future plans of a current state and the error covariance estimates to obtain an a priori estimate for the next step. The measurement update equations are used as feedback, integrating a new measurement into the a priori estimate to obtain an updated a posteriori estimate [26].

2.2. Optimize the Gain of Kalman Filter

The Kalman filter is used to estimate the unmeasured states of a system based on available measurements and a mathematical model. The observer state feedback combines the estimated states from the Kalman filter with a control law to generate control signals for the system. The gain of the Kalman filter, also known as the Kalman gain, determines the weight given to the measurements and the predicted states in the estimation process. Optimizing the Kalman gain involves finding a balance between the accuracy of the state estimation and the desired control performance [27]-[30].

The Kalman gain is typically determined by minimizing a cost function that captures the error between the estimated states and the actual states. This is often done through a process called Kalman filter design or Kalman filter tuning [31]. The specific method for tuning the Kalman gain can vary based on the system and the available information. In this research optimizing the Kalman gain using Fish Swarm Optimization Algorithm (FSO) [32].

FSO is based on the collective behavior of fish schools. It imitates the movement and interaction of fish in search of optimal solutions. Each fish represents a potential solution to the problem, and the whole school represents the population. In FSO, each fish navigates the search space using three main behaviors: attraction, repulsion, and orientation. Attraction enables fish to move toward the current best solution, repulsion helps avoid overcrowding, and orientation allows fish to explore the search space efficiently [33]. Fish in FSO communicate with each other using a chemical substance called the 'pheromone.'

Fish exchange information about the quality of solutions and use this information to update their behavior and guide their movements. FSO tends to focus more on exploration by utilizing the orientation behavior, which allows fish to explore different regions of the search space [34]. This can lead to slower convergence but potentially better exploration of the solution space. FSO has a relatively simpler implementation compared to PSO. It requires fewer parameters to tune, making it easier to implement and understand.

2.3. Proposed Method

A two-wheel balancing robot is a mobile robot that has two wheels on its right and left sides. This robot cannot be balanced without control. This two-wheeled robot is a combination of a wheeled mobile robot and an inverted pendulum system. Meanwhile, an inverted pendulum cannot be moved by itself, and it uses a gyroscope and accelerometer to sense the tilt of the vertical axis. To overcome the tilt, the controller generates a torque signal to each motor to prevent the failing system to the ground. In this study, the proposed hybrid approach (Kalman Filter and FSOA) in improving the state estimation compared to the Kalman Filter alone [35].

In addition, Kalman Filter is a mathematical algorithm that optimizes the use of inaccurate data in a linear system with Gaussian error. With Gaussian error, it continuously updates the best estimate of the system's condition [36]. Kalman Filter estimates the state of the process at a time and then takes the feedback as noise output. Kalman Filter is also equipped with a time update (prediction equation) that can predict the value of the state vector, which is further used for feedback. Calculation, measurement, or observation update is a correction equation used as a state vector correction. So, the Kalman Filter has a process of prediction and correction. Besides, it also predicts a state space and corrects the system outputs, following the desired input [37].

As a self-balancing platform, Kalman's implementation also includes the Kalman filtering process. Kalman filtering is an iterative mathematical process using a multi-dimensional matrix and input data as equations to track objects by estimating the true values of velocity and position [38]. Basically, it focuses on minimizing variation or uncertainty in continuous estimates concerning velocity and position data measurements. Meanwhile, a state matrix (multi-dimensional) is formed to store the velocity and position data of the object being tracked. The process covariance (error) matrix contains errors in the estimation process. Further, the slope angle of the filtered Kalman is compared with the complementary angle filtered to see the difference in the estimated actual angle to determine the optimal filtering method.

The Fish-Swarm Optimization Algorithm (FSOA) is used to optimize the signal that has been filtered by Kalman. Then, the signal is returned to be compared with a reference value [39]. The returned signal is expected to drive the output value, following the reference value. The block diagram is shown in Fig. 1.
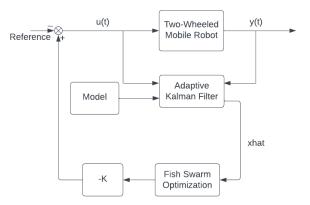

Fig. 1. Block diagram stabilizing mobile robot using Kalman Filter and fish swarm optimization algorithm

The algorithm applied to FSOA to optimize the value of the Kalman Filter is discussed in the following.

1. Initialize FSO parameters, such as the number of iterations (i), number of fishes (n), weights (w), velocities (v), and population size (x_m)
2. Initialize random fish positions in the search space (x).
3. Repeat steps 5 to 9 until the maximum iteration is reached.
4. Evaluate the objective function $F(i)$ by feeding fish positions into the Kalman filter.

$$F(i) = \sqrt{\sum_{i=1}^{n}(x_m - x)^2} \tag{4}$$

5. Calculate fitness values and select the best fish.
6. Compute the fish displacement vector by considering the weight and velocity.
7. Update new fish positions.
8. Repeat steps 5 to 9 until the maximum iteration is reached.

The best fish output is the optimal solution for improving the output from the Kalman filter. Then, the optimal output value is returned to the reference, and the equation is obtained by (5), the optimal value will be multiplied by the gain which will optimize the estimated state value. so that the robot will move according to the reference given, in this case the theta angle on the robot is set at 0 degrees.

$$u = -K\hat{x}_{opt} \tag{5}$$

2.4. Model Reference for Two-Wheeled Mobile Robot

One of the primary challenges of designing a two-wheeled mobile robot is understanding the ways the robot will behave in different scenarios. Research simulation can help model the robot's dynamics and simulate different scenarios to understand the robot's responses [40]. In this study, we simulated the robot's behavior when subjected to external disturbances, process noise, and measurement noise. For the modeling of the Two-wheeled mobile robot system, we used the parameters presented in Table 1. The modeling process then proceeded by constructing a state-space model system as the desired model reference. The model equation is displayed in (3).

Table 1. Parameter of the system

Parameter	Value
M_p	0.78
l	0.05
I_p	$\dfrac{M_p \times l}{3}$
R	0.0325
M_w	0.0775
I_w	$\dfrac{1}{2} \times M_w \times R^2$
g	9.81
R	0.0325

Then the state-space system equation can be expressed as follows:

$$\dot{X} = AX + BU$$
$$Y = CX \tag{6}$$

According to the parameters mentioned in Table 1, the state space equation is described as follows for matrices A, B, and C according to (6).

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{-2M_p^2 l^2 R^2 g}{(2M_p l^2 + I_P)(M_p R^2 + 2M_w R^2 + 2I_w) - 2M_p^2 l^2 R^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{M_p g l (M_p R^2 + 2M_w R^2 + 2I_w)}{(2M_p l^2 + I_P)(M_p R^2 + 2M_w R^2 + 2I_w) - 2M_p^2 l^2 R^2} & 0 \end{bmatrix} \tag{7}$$

$$B = \begin{bmatrix} 0 \\ \dfrac{(2M_p l^2 R + I_P R)}{(2M_p l^2 + I_P)(M_p R^2 + 2M_w R^2 + 2I_w) - 2M_p^2 l^2 R^2} \\ 0 \\ \dfrac{M_p I R}{(2M_p l^2 + I_P)(M_p R^2 + 2M_w R^2 + 2I_w) - 2M_p^2 l^2 R^2} \end{bmatrix} \tag{8}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{9}$$

2.5. The State Observability Matrix

A state observer estimates state variables based on output and control variables measurements. In this context, the concept of observability plays a crucial role. State observers can be designed if the observability conditions are fulfilled.

Observability, as mentioned earlier, refers to the ability to determine the internal states of a system based on its measurable outputs. In observer state feedback, observability is crucial for the accurate estimation of the system states by the observer or estimator. The observer uses the available output measurements and the system model to generate estimates of the internal states, which are then utilized in computing the control signals.

The observability of a system in observer state feedback can be analyzed by evaluating the observability matrix, also known as the observability Gramian. The observability matrix is constructed by selecting appropriate output variables and their time derivatives and arranging them into a matrix. The rank of this matrix determines the observability of the system. If the rank is equal to the number of states in the system, the system is said to be fully observable.

A system is said to be observable if only there is a finite time T so that the initial state $x(0)$ can be determined from the observation history $y(t)$, given control $u(t)$, $0 \leq t \leq T$. Observable Matrix (OM) is shown in (10). The system is said to be completely observable if rank $(OM) = n$.

$$OM = \begin{bmatrix} C \\ CA \\ CA^2 \\ . \\ . \\ . \\ CA^{n-1} \end{bmatrix} \qquad (10)$$

3. Results and Discussion

The system simulation process began with building a two-wheeled mobile robot model using SimScape. Then an input signal was given. Then, a plot of the system's response without control is generated. After that, we added process noise and measurement noise to test the control methods to be applied to the system. Before designing the estimator, we first verified that the system was observable. The property of observability determines whether or not we can estimate the state of the system based on its measured outputs. Similar to the process for verifying controllability, a system is observable if its observability matrix is fulfilled.

3.1. Modeling of Two-Wheeled Mobile Robot Using SimScape

Fig. 2 shows the modeling process using SimScape. The modeling involved configuring the block solver, world frame, and mechanism configuration as input from the mobile robot plant. Then, the prismatic joint, rigid transform, and revolute joint blocks were utilized to construct the chassis and cart. In the revolute joint block, the angle data was set to the sensing-position. Meanwhile, for the input signal for the mobile robot, we used an initial perturbation with an initial value of 10 and a final value of 0.
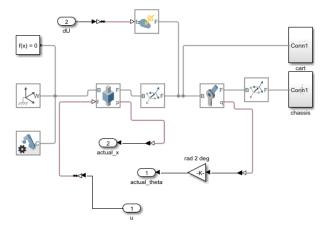

Fig. 2. Two-wheeled mobile robot using simscape

The chassis was built using a block of 3 plates with dimensions of [18 8 0.3] cm and a mass of 1/4 kg. The frame port was connected to plates 2 and 3. They were connected to a pillar with a radius of 5mm and a length of 20 cm, as shown in Fig. 3. The block chart in Fig. 3 depicts a model of 2 wheels built to function as a chassis stabilizer. The right and left wheels were constructed using a wheel body and designed with a radius of 30 mm and a length of 12.7 mm. Each of them had a mass of 48 g.
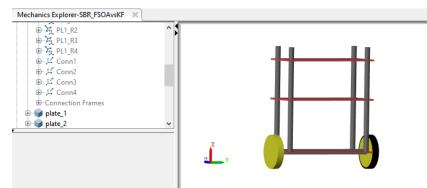

Fig. 3. Mechanics explorer of simscape

3.2. The State Observability of Two-Wheeled Mobile Robot

Equation (7) enables an individual to determine the state observability matrix of the system. The results for the state observability matrix are shown in Fig. 4. Referring to the Ob matrix in Fig. 4, rank $(Ob) = 4$ shows that the system has a completely observable state.

Fig. 4. State observability matrix (matlab result)

3.3. Modeling of Process Noise and Measurement Noise

The modeling of process noise and measurement noise utilized random signal blocks with parameters presented in Table 2. The noise will affect the motion of the chart and cause significant signal changes. However, the use of a Kalman Filter can minimize or even eliminate the noise.

Table 2. Noise parameters

No	Noise	Noise Power	Sample Time
1	Process Noise	1e-5	1e-2
2	Measurement Noise	1e-6	1e-2

3.4. Simulation Results of Two-Wheeled Mobile Robot Without Control

A simulation of an uncontrolled two-wheeled mobile robot was built using Simulink with an initial perturbation input value of 10 and a final value of 0. In this simulation, we applied Equation 1 for the model. Fig. 5 shows a block diagram of the two-wheeled robot without control, while Fig. 6 illustrates the simulation results of the uncontrolled robot. From these Figures, we can observe that the theta measurement results do not follow the reference, with the angle of -180°.
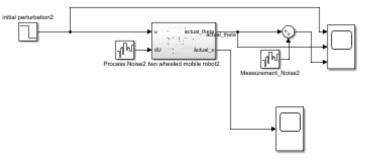

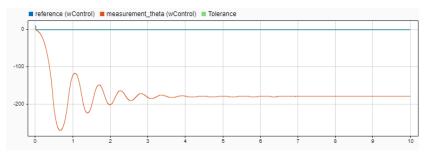**Fig. 5.** Block diagram without control

Fig. 6. Result of simulation without control

3.5. State Observer Feedback for Stabilize Two-Wheeled Mobile Robot

The main goal of observer-based state estimation is to demonstrate that the error of state estimation closes to zero, either exponentially or within a finite time period. This is typically accomplished by characterizing the error dynamics in similar means to the stability proof of control systems. This estimation is typically carried out using several methods, such as pole placement or Lyapunov functions. However, the proof of stability in observer-based state estimation generally assumes a "deterministic" environment in which measurement and process noise are not considered.

On the other hand, the Kalman filter approach concentrates on state estimation for stochastic processes where measurement and process noise are present and taken into account. The purpose of the Kalman filter is to optimize the state estimation under the existence of noise and to characterize the associated uncertainty. It does not aim to converge the estimation error to zero since it is not practical due to the stochastic nature of the system. In fact, the Kalman filter has been reported to outperform other state estimators when the assumptions of Gaussian measurement, process noise, and linear dynamics are applied. The block diagram of the system with state observer feedback using the Kalman Filter is shown in Fig. 7.
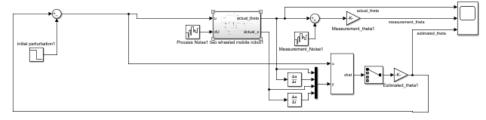

Fig. 7. Control simulation of two-wheeled mobile robot using Kalman Filter

The state-space model as a reference is described in the following.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -2.1211 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 27.5333 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 36.96 \\ 0 \\ 85.29 \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The result is shown in Fig. 8, with the actual theta represented by the green line, the measurement theta being the blue line, and the estimated theta being the red line. The actual theta represents the real value before the addition of measurement noise, while the measurement theta is the theta value after the addition of measurement and processing noise. Lastly, the estimated theta is the value generated through Kalman Filter and returned to the initial value.
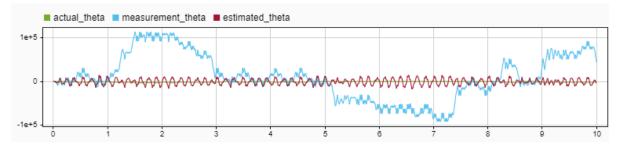

Fig. 8. Result of control simulation of two-wheeled mobile robot using Kalman filter

Then, the state estimate value generated by the Kalman filter was compared to the reference value. The results showed that noise still affects the output signal, and the output $y(t)$ does not match the reference. To address this issue, the estimated state value was optimized using the FSO algorithm and returned with a gain of -16, as illustrated in Fig. 9. The results of this optimization can be observed in Fig. 10, showing that the output response follows the reference value with reduced noise.
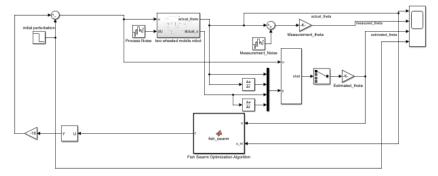

Fig. 9. Control simulation of two-wheeled mobile robot using kalman Filter-FSOA

In Fig. 10, the actual theta is visualized in a green line, the measured theta is represented by the blue line, and the estimated theta is in the orange line. The result shows that the estimated theta follows the reference with a 0.2% error steady-state.
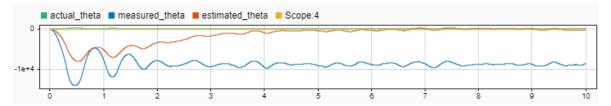

Fig. 10. Result of control simulation of two-wheeled mobile robot using Kalman Filter-FSOA

A comparison of system performance is shown in Table 3. The methods being compared were without control, state observer feedback using the Kalman Filter, as well as state observer feedback using the Kalman filter and fish swarm optimization algorithm.

Table 3. Results obtained for the proposed hybrid state observer feedback

Method	Rise time (s)	Settling time (s)	ESS (%)	MAE (%)	SNR (dB)
Without control	-	-	-180	-	-35.1625
Kalman Filter	-	-	0	5.7854e+03	-17.4808
Kalman Filter and FSOA	2	3.8	0,2	1.3866e+03	-7.7863

Table 3 shows the simulation results without control, indicating that the steady-state system error is -180% and the signal-to-noise ratio is -35.16 dB. The simulation results of the state observer feedback using the Kalman Filter showed that the steady-state error of the system is 0%, with MAE of 5.7854e+03 and a signal-to-noise ratio of -17.48 dB. The simulation results of the state observer feedback using the Kalman Filter and FSOA showed the steady state error of the system is 0.2%, with the MAE of 1.3866e+03 and the signal-to-noise ratio of -7.7863 dB.

4. Conclusion

The objective of the Kalman filter is to optimize the state estimation under the noise and characterize the associated uncertainty. It does not converge the estimation error to zero since it is not practical due to the stochastic nature of the system. The simulation results showed that the state feedback from the Kalman Filter does not match the reference model, suggesting that it is optimized using the Fish Swarm Optimization Algorithm (FSOA). Thus, the output response from the Kalman Filter and FSOA was compared to the output response from the Kalman Filter. The comparison results suggested a decrease in MAE by 61.33% and a decrease in SNR by 38.37%. The observed decrease in MAE and improvement in SNR when combining the Kalman Filter and FSOA have practical implications such as enhanced state estimation, improved system performance, noise reduction, increased stability and control, enhanced decision-making, and improved optimization performance. These implications are valuable in a wide range of applications, including control systems, robotics, autonomous systems, and various domains where accurate state estimation is crucial for achieving desired outcomes. While FSOA is an effective optimization algorithm, exploring alternative optimization algorithms can provide different perspectives and potential improvements. Algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), or Simulated Annealing (SA) can be considered as alternatives. Each algorithm has its own strengths and weaknesses, and experimenting with different optimization techniques can help find the one that suits the specific problem and system dynamics better, leading to improved state estimation accuracy.

Author Contribution: The paper conceptualization, methodology, software, and validation, were done by 4th author. The formal analysis, investigation, resources, and data curation were carried out by 2nd and 3rd authors. Meanwhile, writing original draft preparation, writing review and editing, visualization, supervision, and project administration was done by 1st author.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] X. Gao, L. Yan, Z. He, G. Wang, and I. -M. Chen, "Design and Modeling of a Dual-Ball Self-Balancing Robot," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12491-12498, Oct. 2022, doi: 10.1109/LRA.2022.3219029., https://doi.org/10.1109/LRA.2022.3219029.

[2] R. S. Martins and F. Nunes, "Control system for a self-balancing robot," *2017 4th Experiment@International Conference (exp.at'17)*, pp. 297-302, 2017, https://doi.org/10.1109/EXPAT.2017.7984388.

[3] T. Dokeroglu, A. Deniz, and H. E. Kiziloz, "A comprehensive survey on recent metaheuristics for feature selection," *Neurocomputing*, vol. 494, pp. 269-296, 2022, https://doi.org/10.1016/j.neucom.2022.04.083.

[4] Y. Q. Liang, W. Haifeng, and Z. Yi, "Attitude Control of Two Wheeled Self-Balancing Vehicle," *EPH - International Journal of Science and Engineering*, vol. 7, no. 2, 2017, https://doi.org/10.53555/eijse.v3i4.125.

[5] P. Frankovský, L. Dominik, A. Gmiterko, I. Virgala, P. Kurylo, and O. Perminova, "Modeling of Two-Wheeled Self-Balancing Robot Driven by DC Gearmotors," *International Journal of Applied Mechanics and Engineering*, vol. 22, pp. 739-747, 2017, https://doi.org/10.1515/ijame-2017-0046.

[6] A. Y. Zimit, H. J. Yap, M. F. Hamza, I. Siradjuddin, B. Hendrik, and T. Herawan, "Modelling and Experimental Analysis Two-Wheeled Self Balance Robot Using PID Controller," *International Conference on Computational Science and Its Applications*, pp. 683–698, 2018, https://doi.org/10.1007/978-3-319-95165-2_48.

[7] I. Jmel, H. Dimassi, S. H. Said, and F. M'Sahli, "Adaptive Observer-Based Output Feedback Control for Two-Wheeled Self-Balancing Robot," *Mathematical Problems in Engineering*, vol. 2020, pp. 1-16, 2020, https://doi.org/10.1155/2020/5162172.

[8] A. V. Panteleev and A. V. Lobanov "Mini-Batch Adaptive Random Search Method for the Parametric Identification of Dynamic Systems," *Automation and Remote Control*, vol. 81, pp. 2026–2045, 2020, https://doi.org/10.1134/S0005117920110065.

[9] C. Andic, A. Ozturk, and B. Turkay, "False Data Injection Attacks on CSA-Based State Estimation in Smart Grid," *2022 Global Energy Conference (GEC)*, pp. 253-257, 2022, https://doi.org/10.1109/GEC55014.2022.9986658.

[10] U. T. Abdurrahman, P. Sukamto, M. A. Sobarnas, and Mujiarto, "A Design for Self Balancing Scale Model Bicycle," *Journal of Physics: Conference Series*, vol. 1764, no. 1, p. 012171, 2021, https://doi.org/10.1088/1742-6596/1764/1/012171.

[11] A. J. Abougarair and E. S. Elahemer, "Balancing Control of Two Wheeled Mobile Robot Based on Decoupling Controller," *International Journal of Control Systems and Robotics*, vol. 3, pp. 1–7, 2018, https://doi.org/10.35778/1742-000-025-012.

[12] A. J. Abougarair, "Controllers comparison to balancing and trajectory tracking a two wheeled mobile robot," *International Robotics and Automation Journal*, vol. 5, pp. 28–33, 2020, https://doi.org/10.15406/iratj.2019.05.00168.

[13] O. Adiyatov, B. Rakhim, A. Zhakatayev, and H. A. Varol, "Sensor Reduction of Variable Stiffness Actuated Robots Using Moving Horizon Estimation," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 5, pp. 1757-1769, 2020, https://doi.org/10.1109/TCST.2019.2924601.

[14] K. Albert, K. S. Phogat, F. Anhalt, R. N. Banavar, D. Chatterjee, and B. Lohmann, "Structure-Preserving Constrained Optimal Trajectory Planning of a Wheeled Inverted Pendulum," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 910-923, 2020, https://doi.org/10.1109/TRO.2020.2985579.

[15] W. An and Y. Li, "Simulation and control of a two-wheeled self-balancing robot," *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 456-461, 2013, https://doi.org/10.1109/ROBIO.2013.6739501.

[16] M. Arvidsson and J. Karlsson, *Design, construction and verification of a self-balancing vehicle*. Chalmers University of Technology, 2012, https://odr.chalmers.se/server/api/core/bitstreams/3ca86691-df6b-4523-ad60-66c320f87d8d/content.

[17] M. O. Asali, F. Hadary, and B. W. Sanjaya, "Modeling, simulation, and optimal control for two-wheeled self-balancing robot," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 4, pp. 2008–2017, 2017, https://doi.org/10.11591/ijece.v7i4.pp2008-2017.

[18] S. Atar and A. Shaikh, "Amphibious Self-Balancing Autonomous Surveillance UGV," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, pp. 1848-1854, 2021, https://www.irjet.net/archives/V8/i10/IRJET-V8I10287.pdf.

[19] M. S. Z. Abidin, N. Baharudin, M. S. A. Mahmud, and M. K. I. A. Rahman, "Modelling and Control of DWR 1.0–A Two Wheeled Mobile Robot," *Applications of Modelling and Simulation*, vol. 1, no. 1, pp. 29–35, 2017, http://arqiipubl.com/ojs/index.php/AMS_Journal/article/view/5.

[20] S. Banerjee, B. Samynathan, J. A. Abraham, and A. Chatterjee, "Real-Time Error Detection in Nonlinear Control Systems Using Machine Learning Assisted State-Space Encoding," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 576-592, 2021, https://doi.org/10.1109/TDSC.2019.2903049.

[21] A. Barreiro, E. Delgado, J. Cuadrado, and D. Dopico, "Extended-Kalman-Filter Observers for Multibody Dynamical Systems," *Laboratorio de Ingeniería Mecánica*, 2008, http://lim.ii.udc.es/docs/proceedings/2008_06_ENOC_Extended.pdf.

[22] M. Barrimi, *et al.*, "Corticothérapie prolongée et troubles anxieux et dépressifs. Étude longitudinale sur 12 moisProlonged corticosteroid-therapy and anxiety-depressive disorders, longitudinal study over 12 months," *L'Encéphale*, vol. 39, pp. 59-65, 2013, https://doi.org/10.1016/j.encep.2012.03.001.

[23] R. Bimarta, A. E. Putra, and A. Dharmawan, "Balancing Robot Menggunakan Metode Kendali Proporsional Integral Derivatif," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 5, no. 1, p. 89, 2015, https://doi.org/10.22146/ijeis.7157.

[24] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a Stable Observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797-1812, 2017, https://doi.org/10.1109/TAC.2016.2594085.

[25] P. Brandstetter and M. Dobrovsky, "Speed estimation of induction motor using model reference adaptive system with Kalman Filter," *Advances in Electrical and Electronic Engineering*, vol. 11, no. 1, pp. 22–28, 2013, https://doi.org/10.15598/aeee.v11i1.802.

[26] Y. Celik and M. Güneş, "Designing an Object Tracker Self-Balancing Robot," *Academic Platform Journal of Engineering and Science*, vol. 6, no. 2, pp. 124–133, 2018, https://doi.org/10.21541/apjes.414715.

[27] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annual Reviews in Control*," vol. 37, pp. 89–103, 2013, https://doi.org/10.1016/j.arcontrol.2013.03.004.

[28] Y. Liu, X. Ji, K. Yang, X. He, X. Na, and Y. Liu, Finite-time optimized robust control with adaptive state estimation algorithm for autonomous heavy vehicle. *Mechanical Systems and Signal Processing*, vol. 139, p. 106616, 2020, https://doi.org/10.1016/j.ymssp.2020.106616.

[29] K. V. C. García, O. E. P. Ramírez, and C. F. R. Rodas, "Comparative Analysis Between Fuzzy Logic Control, LQR Control with Kalman Filter and PID Control for a Two Wheeled Inverted Pendulum," *Advances in Automation and Robotics Research in Latin America*, vol. 13, pp. 144–156, 2017, https://doi.org/10.1007/978-3-319-54377-2_13.

[30] B. Chen, *et al.*, "A memristor-based hybrid analog-digital computing platform for mobile robotics," *Science Robotics*, vol. 5, no. 47, pp. 1–8, 2020, https://doi.org/10.1126/scirobotics.abb6938.

[31] D. Chen, Y. Guan and L. He, "A Self-Balanced Essboard-like Mobile Robot – Essbot," *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 160-165, 2019, https://doi.org/10.1109/ROBIO49542.2019.8961631.

[32] J. Chen, "Research on Balance Cognition Based on Multi-level Heuristic Dynamic Programming of Flexible Robot," *Journal of System Simulation*, vol. 30, no. 1, pp. 147-155, 2018, https://doi.org/10.16182/j.issn1004731x.joss.201801018.

[33] R. Havangi, "An adaptive particle filter based on pso and fuzzy inference system for nonlinear state systems" *Automatika*, vol. 59, no. 1, pp. 94–103, 2018, https://doi.org/10.1080/00051144.2018.1498207.

[34] J. He and F. Gao, "Mechanism, Actuation, Perception, and Control of Highly Dynamic Multilegged Robots: A Review," *Chinese Journal of Mechanical Engineering*, vol. 33, 2020, https://doi.org/10.1186/s10033-020-00485-9.

[35] J. B. He, Q. G. Wang, and T. H. Lee, "PI/PID controller tuning via LQR approach," *Chemical Engineering Science*, vol. 55, no. 13, pp. 2429–2439, 2000, https://doi.org/10.1016/S0009-2509(99)00512-6.

[36] C. F. Hsu and W. F. Kao, "Double-loop fuzzy motion control with CoG supervisor for two-wheeled self-balancing assistant robots," *International Journal of Dynamics and Control*, vol. 8, no. 3, pp. 851–866. 2020, https://doi.org/10.1007/s40435-020-00617-y.

[37] Y. Irdayanti, R. Kusumanto, M. Anisah, N. Alfarizal, and Z. Erman, "Ultrasonic Sensor Application As A Performance Enhancement of Robot Two Wheels," *Journal of Physics: Conference Series*, vol. 1500, no. 1, 2020, https://doi.org/10.1088/1742-6596/1500/1/012007.

[38] M. R. Islam, M. R. T. Hossain, and S. C. Banik, "Synchronizing of Stabilizing Platform Mounted on a Two-Wheeled Robot," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 552–558, 2021, https://doi.org/10.18196/jrc.26136.

[39] F. Jian and J. Y. Liu, "PSO in two-wheeled self-balancing robot control research," *Advanced Materials Research*, vol. 898, pp. 534–537, 2014, https://doi.org/10.4028/www.scientific.net/AMR.898.534.

[40] S. Jiang, F. Dai, L. Li, and X. Gao, "Modeling and LQR control of a multi-DOF two-wheeled robot," *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 1964-1969, 2014, https://doi.org/10.1109/ROBIO.2014.7090624.