PARTICLE SWARM OPTIMIZATION OF A KALMAN FILTER FOR SPEED ESTIMATION OF DC MOTOR

THUẬT TOÁN TỐI ƯU BẦY ĐÀN CỦA BỘ LỌC KALMAN ĐỂ ƯỚC LƯỢNG TỐC ĐỘ CỦA ĐỘNG CƠ MỘT CHIỀU

Bui Minh Hien, Pham Duy An

University of Science and Technology of Hanoi, Vietnam Academy of Science and Technology

Ngày nhận bài: 04/11/2024, Ngày chấp nhận đăng: 20/04/2025, Phản biện: TS. Lê Trung Dũng

Abstract:

DC motors have many critical applications in our daily lives. They can be used in many fields, including electric vehicles and household appliances. Therefore, the demand for DC motor control, particularly in terms of speed, has increased over the years. To control DC motors, it is necessary to measure relevant parameters such as motor speed and armature current using sensors. However, the installation of sensors will increase the system's cost, and it is challenging for DC motors due to space and weight limitations. To overcome this problem, it is necessary to design a measurement system with fewer sensors. A new method for estimating DC motor speed without using speed sensors is described in this paper, utilizing the Kalman filter for motor speed estimation due to its resistance to external disturbances and its ability to predict states and parameters. The Kalman filter (KF) requires tuning for improved estimation, which can be a time-consuming and laborious process. For that reason, the swarm optimization algorithm is used to optimize the Kalman filter. This paper will present the application of a swarm optimization algorithm (PSO) in optimizing the Kalman filter to estimate the speed of a DC motor through simulation in MATLAB and Simulink. The simulation results have shown the effectiveness of the proposed method.

Keywords : Kalman filter, Particle Swarm Optimization, Direct Current motor

Tóm tắt:

Động cơ điện một chiều có nhiều ứng dụng quan trọng trong cuộc sống hàng ngày của chúng ta. Nó có thể được sử dụng trong nhiều lĩnh vực, bao gồm xe điện và thiết bị gia dụng. Do đó, nhu cầu điều khiển động cơ điện một chiều, đặc biệt là tốc độ của chúng, đã tăng lên trong những năm qua. Để điều khiển động cơ điện một chiều, cần phải đo lường các thông số liên quan như tốc độ động cơ và dòng điện phần ứng bằng cảm biến. Tuy nhiên, việc lắp đặt các cảm biến sẽ làm tăng giá thành của hệ thống, và gặp nhiều khó khăn đối với các động cơ điện một chiều có giới hạn về không gian và trọng lượng. Để khắc phục vấn đề này, cần thiết kế một hệ thống đo lường với số lượng cảm biến ít hơn. Một phương pháp mới để ước lượng tốc độ động cơ điện một chiều mà không sử dụng cảm biên tốc độ sẽ được mô tả trong bài báo này, đó là ước lượng tốc độ động cơ sử dụng bộ lọc Kalman, nhờ khả năng chống lại nhiễu bên ngoài và khả năng dự đoán trạng thái và thông số. Bộ lọc Kalman cần được điều chỉnh để ước lượng tốt hơn, và việc điều chỉnh bộ lọc có thể tốn nhiều thời gian và công sức. Vì lý do đó, thuật toán tối ưu bầy đàn được sử dụng để tối ưu bộ lọc Kalman. Bài báo này sẽ trình bày ứng dụng của thuật toán tối ưu bầy đàn trong việc tối ưu hóa bộ lọc Kalman để ước tính tốc độ của động cơ điện một chiều thông qua mô phỏng trong MATLAB và Simulink. Các kết quả mô phỏng đã chỉ ra tính hiệu quả của phương pháp đề xuất.

Từ khóa : Bộ lọc Kalman, Tối ưu hóa bầy đàn, Động cơ điện một chiều

1. Introduction

Back in 1832, William Sturgeon, a British scientist, created the first DC motor that had the ability to power machinery. Sturgeon's initial development was further expanded upon by Thomas Davenport, an American scientist. Davenport is known for creating the first working DC motor, which he patented in 1837. In the following years, DC motors continued to be developed and improved. Today, DC motors are vital in many industries. They are commonly used in lifting equipment such as hoists and cranes for construction, shipping, and material handling, thanks to their ability to provide precise speed control and starting torque, which is critical for raising and lowering heavy loads. DC motors are also used in milling machines and in machining industries to rotate the workpiece or cutting tool, as their ability to control the speed smoothly is suitable for precise material removal and ensures quality machining operations [1]. For transportation, it can be used to drive electric vehicles [2]. Despite their diverse applications, all these systems share one common requirement: accurate control. An incorrect movement could lead to significant issues, including damage to the machinery or even harm to its operators.

In order to achieve that demand, modern DC motors not only need proper controllers but also require good measured control data. Sensors could record data for the operation of DC motors; however, some systems have special properties so that suitable sensors might not be available. Furthermore, research for better sensors might be costly, and the implementation of new sensors in the DC motor will increase the weight and volume of the whole system. As a result of those problems, this paper will depict a new solution that uses the Kalman Filter to estimate the motor speed without the need for speed sensors [3] [4]. In this filter, the process noise covariance matrix Q and measurement noise covariance R are chosen randomly, so they require tuning so the Kalman filter can work properly. Nevertheless, the manual tuning process might be time-consuming and require more labor costs. To reduce the time and labor cost for optimizing filters, optimization algorithms like Particle Swarm Optimization, Genetic Algorithm, and Ant Colony Optimization are applied [5] [6].

This paper will focus on the application of PSO for tuning the Kalman filter in a DC motor system. It aims to demonstrate the effectiveness of the Kalman filter in precisely estimating measurements despite the noise affecting the system. Additionally, the paper will show how PSO enhances the performance of the Kalman filter.

The paper is divided into five main sections: Introduction, Development of the Kalman Filter for the Motor, Optimization of the Kalman Filter with PSO, Simulation Results, and Conclusion. The core of the paper lies in sections 2 through 4, where the system's development is discussed, and the simulation results are presented and analyzed.

2. Develop a Kalman filter for the motor

The Kalman filter is an algorithm that estimates the state of a dynamic system from noisy measurements. There are two main processes involved in the cycle: prediction and update. Firstly, the process begins with the initialization of the system's (e.g., speed, position) and the associated uncertainty (covariance). This step initializes the filter. Secondly, the filter projects the system's state and covariance into the future using the system's model. This step calculates the a priori estimate based on the previous state and control inputs, predicting where the system is heading. Thirdly, new measurements from sensors like armature current or motor speed are combined with the predicted state to get a more accurate estimate. Finally, the Kalman gain is computed and used to update the predicted state based on the sensor measurements. The state and its covariance are corrected to provide an updated, more accurate estimate, and the cycle repeats with the next time step, continually refining the state estimate as new sensor measurements arrive. The loop continues with each iteration, improving the accuracy of the estimate.

Corresponding to the two main processes, there are two types of equations involved in the algorithm: extrapolation equations and update equations. On the one hand, for extrapolation equations, there are two equations, which are the state extrapolation equation:

$$\boldsymbol{x}_{n+1,n} = \boldsymbol{A} * \boldsymbol{x}_{n,n} + \boldsymbol{B} * \boldsymbol{u}_n \ (1)$$

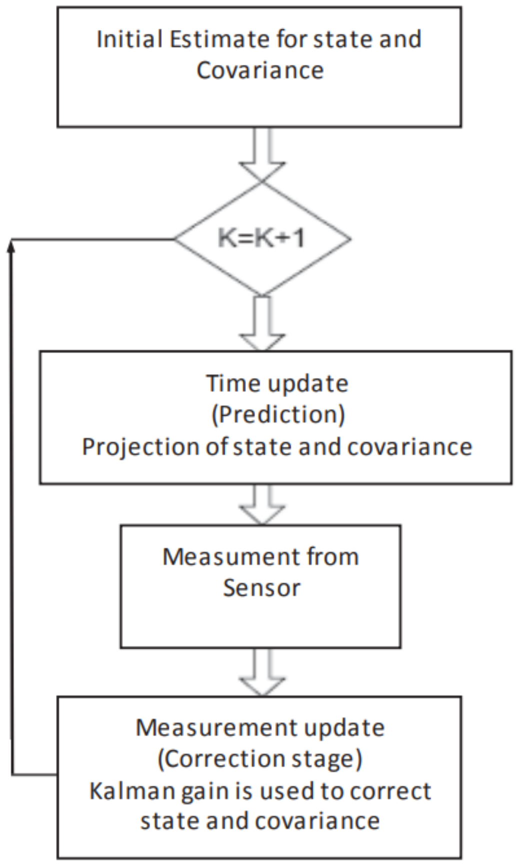

Figure 1: Kalman filter's working diagram

And covariance extrapolation equation:

$$\boldsymbol{P}_{n+1} = \boldsymbol{A}\boldsymbol{P}_{n,n}\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{Q} \ (2)$$

With A is the state transition matrix, B is the control matrix, $\boldsymbol{u}_n$ is the control variable, and Q is the process noise matrix. To obtain those equations, we must investigate a simple model of DC motor [7]:

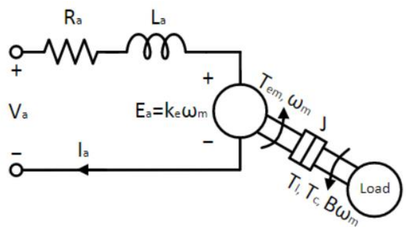

Figure 2: DC motor simple model

From the illustration of a DC motor in Figure 2, we can implement equations for state extrapolation. Firstly, we have the function of the Terminal voltage $\boldsymbol{u_a}$ as follows:

$$\boldsymbol{u_a(k) = R_a + L_a * \frac{di_a}{dt} + PSI_E * \omega_m(k)}$$

$$u_a(k) \approx R_a + L_a * \frac{i_a(k+1) - i_a(k)}{T_s} + PSI_E * \omega_m(k) \quad (3)$$

From that equation, we can infer the function for the armature current $\boldsymbol{i_a}$:

$$\boldsymbol{i_a(k+1) = \left(1 - \frac{R_a * T_s}{L_a}\right) * i_a(k) + \frac{-PSI_{E*T_s}}{L_a} * \omega_m(k) + \frac{T_s}{L_a} * u_a(k)} \ (4)$$

Secondly, the mechanical dynamics of the motor, which describe the torque and speed relationship, are given in discrete form by:

$$\boldsymbol{J\frac{\omega_m(k+1) - \omega_m(k)}{T_s} = PSI_M * i_a(k) - B * \omega_m} \ (5)$$

J is the moment of inertia of the motor's rotor, $\boldsymbol{\frac{d\omega_m}{dt}}$ is the angular acceleration, $\boldsymbol{PSI_M}$ is the torque constant, $\boldsymbol{i_a(k)}$ is the armature current, B is the viscous friction coefficient, and $\boldsymbol{\omega_m}$ is the angular velocity of the rotor. For simplicity, we could omit the friction, then, after some transformation, we have the equation for angular velocity $\boldsymbol{\omega_m}$:

$$\boldsymbol{\omega_m(k+1) = \frac{Ts * PSI_M}{J} * i_a(k) + \omega_m(k)} \ \mathbf{(6)}$$

Finally, the third equation of the state transition matrix A can be derived from the delay in the applied voltage:

$$u_a(k+1) = u_a(k) + \frac{T_s}{T_{\text{delay}}}(u_{\text{input}}(k) - u_a(k))$$

$$u_a(k+1) = \left(1 - \frac{T_s}{T_{\text{delay}}}\right)u_a(k) + \frac{T_s}{T_{\text{delay}}}u_{\text{input}}(k) \quad (7)$$

Based on equations (1), (2), and (3), we can get the state transition matrix A and the control input matrix B as below:

$$A = \begin{bmatrix} 1 - \dfrac{R_a T_s}{L_a} & -\dfrac{\Psi_E T_s}{L_a} & \dfrac{T_s}{L_a} \\ \dfrac{T_s \Psi_M}{J} & 1 & 0 \\ 0 & 0 & 1 - \dfrac{T_s}{T_{\text{delay}}} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} \dfrac{\boldsymbol{T_s}}{\boldsymbol{T}_{\text{delay}}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

On the other hand, the update equations consist of two equations, which are the state update equation:

$$\boldsymbol{x_{n,n} = x_{n,n-1} + K_n(z_n - Hx_{n,n-1})} \ (8)$$

And the covariance update equation:

$$\boldsymbol{P_{n,n} = (I - K_n H)P_{n,n-1}(I - K_n H)^T + K_n R_n {K_n}^T} \ \mathbf{(9)}$$

With: $\boldsymbol{K_n}$ is Kalman Gain, H is the Observation matrix, R_n is the Measurement noise covariance matrix, and $\boldsymbol{z_n}$ is the measurement matrix. In this system, to estimate the DC motor speed, only the armature current is directly measured, so the observation matrix H is as follows:

$$\boldsymbol{H} = [\mathbf{1\ 0\ 0}]$$

The Kalman gain is calculated by this equation:

$$\boldsymbol{K_n} = \boldsymbol{P_{n,n-1}}\boldsymbol{H^T}(\boldsymbol{H}\boldsymbol{P_{n,n-1}}\boldsymbol{H^T} + \boldsymbol{R_n})^{-1} \quad (\mathbf{10})$$

3. Optimize the Kalman filter with PSO

The Particle Swarm Optimization (PSO) is a computational optimization technique inspired by the social behavior of animals, such as birds flocking or fish schooling [9]. It was developed by James Kennedy and Russell Eberhart in 1995. The algorithm's main idea is to produce a swarm of particles that move around in the problem space, or the area that best meets their demands as determined by a fitness function, in search of their objective. PSO starts with the initialization of a swarm of particles with random positions and velocities in the search space. Each particle represents a potential solution, and they are evaluated using a fitness function. Particles track their best-known position (personal best), and the best value of personal best among particles in the swarm is tracked in the global best. The population of particles can change based on the swarm's size (N). Larger swarms will, on the one hand, cover more ground in the search area, improve their ability to explore, and lower their chance of becoming stuck in a local optimum. However, because there are more particles to assess, the convergence might be delayed as a result. On the other hand, a lower swarm size leads to a faster convergence but may not sufficiently traverse the search space and may raise the possibility of premature convergence to local optima. The new velocity and position of each particle are updated by two equations, respectively:

$$\boldsymbol{v_i}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{w} \cdot \boldsymbol{v_i}(\boldsymbol{t}) + \boldsymbol{c_1} * \boldsymbol{r_1} * \left(\boldsymbol{pbest_i} - \boldsymbol{x_i}(\boldsymbol{t})\right) + \boldsymbol{c_2} * \boldsymbol{r_2} * \left(\boldsymbol{gbest} - \boldsymbol{x_i}(\boldsymbol{t})\right) \quad (\mathbf{11})$$

$$\boldsymbol{x_i}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{x_i}(\boldsymbol{t}) + \boldsymbol{v_i}(\boldsymbol{t}+\mathbf{1}) \quad (\mathbf{12})$$

With w is the inertia weight which controls the effects of a prior velocity on its current velocity, r_1 and r_2 are random coefficients drawn from a uniform distribution of range [0,1], c_1 and c_2 are the cognitive coefficient and the social coefficient, respectively. The inertia weight w shows how the prior velocity can affect the current velocity; the higher the inertia weight, the deeper the exploration. For the cognitive and social coefficients c_1 and c_2, a high cognitive coefficient can lead to a more localized search, while a high social coefficient will lead to a broader exploration of the search space. It is crucial to balance these two parameters as it could affect the efficiency of the optimization process, for instance, a too large cognitive coefficient c_1 might lead to premature convergence, where particles become stuck in local optima because they do not sufficiently explore the global search space. The algorithm may end before finding the global optimum, resulting in a suboptimal solution.

In this paper, the PSO algorithm will be used to tune the process noise covariance matrix Q and the R measurement noise covariance matrix of the Kalman filter. If they are not well-tuned, the filter may represent noises with uncertain associated qualities.

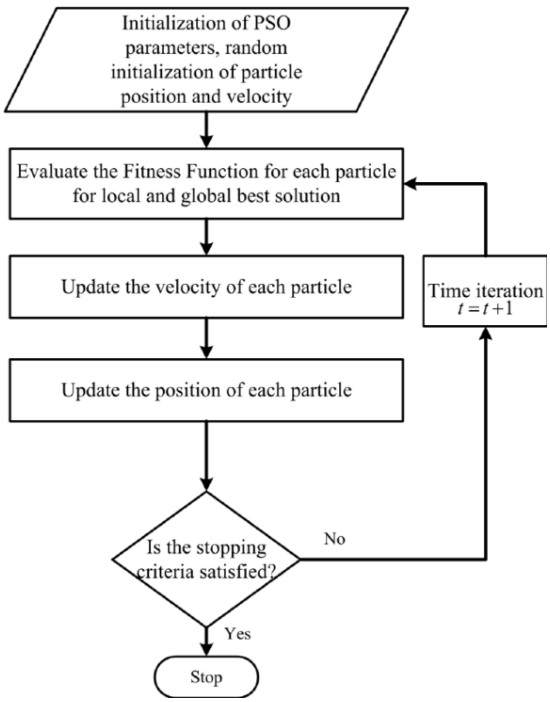

Figure 3: PSO algorithm flowchart

There have been methods to find these factors in the past, but they are frequently quite time-consuming and difficult. So, optimization methods like the genetic algorithm and PSO have been incorporated into the Kalman filter to save the time needed to estimate these parameters and to prevent needless mistakes. Firstly, the Kalman Filter section initializes state-space matrices to model the motor dynamics. It uses sensor inputs such as armature current and voltage to predict motor speed while accounting for process either over-rely on noisy measurements or trust an inaccurate prediction model too much. Although Q and R are crucial to the Kalman filter functioning, it is very challenging to identify them because they and measurement noise. The original Kalman Filter computes estimated speed using a recursive approach, continuously updating the state variables based on new sensor measurements.

Secondly, the PSO algorithm begins by initializing a swarm of particles, each representing a candidate solution for the filter parameters. These particles adjust their positions and velocities iteratively, and they are guided by their own best solutions (pbest) and the global best solution (gbest) while minimizing the estimation error. The objective function evaluates each particle's performance based on the Kalman Filter's speed estimation accuracy.

Finally, the script compares the original and PSO-optimized Kalman Filter results against the physical motor speed, which is estimated indirectly using the relationship between current and speed, derived from the motor's mathematical model, and plots the errors. Once the PSO converges, the optimized noise covariance matrices (Q_1 and R_1) are applied to the Kalman Filter, resulting in improved accuracy in speed estimation.

4. Simulation results

The DC motor is modelled in Simulink as shown in Figure 4. Both system noises and measurement noise are simulated by random number blocks whose outputs are normally (Gaussian) distributed random signals to ensure a simulation result that is close to reality. It can be seen from Figure 4 that armature current and input voltage values are transferred back to the MATLAB code for calculation. For the PSO algorithm, the number of particles in the swarm is set at 100.

The cognitive and social coefficients are both set at 2, which provides a good balance between exploration and exploitation of the algorithm. After many trials and errors, the optimal inertial weight range is chosen to be between 0.4 and 0.9. The lower and upper bounds of the search space for each dimension are also defined in the MATLAB code.

The simulation returns results as follows. After the optimization process, we have the value of the optimized process noise covariance matrix Q as below:

$$Q1 = \begin{bmatrix} 0.000075 & 0 & 0 \\ 0 & 0.011862 & 0 \\ 0 & 0 & 0.655810 \end{bmatrix}$$

The error between the actual motor speed and the estimated speed from the Kalman filter (both the original and optimized versions) is tracked through 3 values: Variance, which measures the error between the actual motor speed (omega) and the estimated speed;

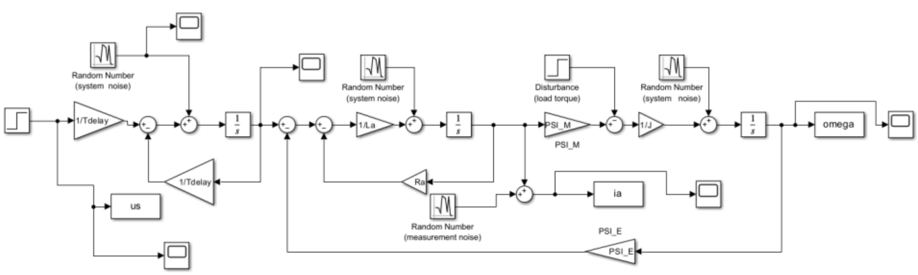

Figure 4: Simulation of DC motor in Simulink

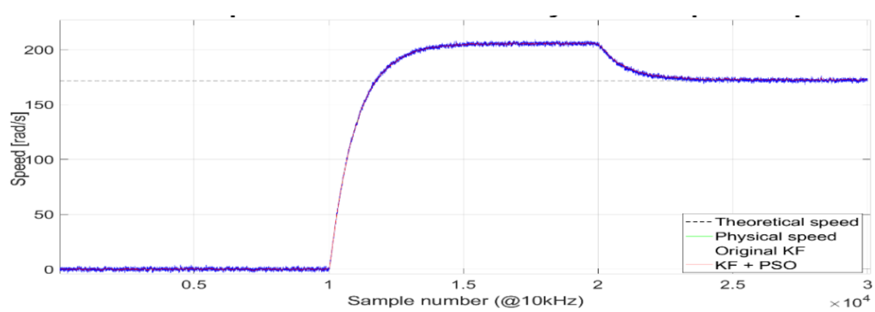

Figure 5: Theoretical Speed, Estimated Speed, Physical Speed, Optimal Speed of DC motor

Average, which is the mean value of the error between the actual motor speed (omega) and the estimated speed; Mean square error (MSE) between the estimated motor speed and the actual motor speed. Related comparisons are shown in Table 1 and Figure 5.

Table 1: Comparison between non-optimized and optimized Kalman filter

	Original KF	KF + PSO
Variance	0.4195	0.1798
Average	0.8570	0.5435
MSE	1.1540	0.4752

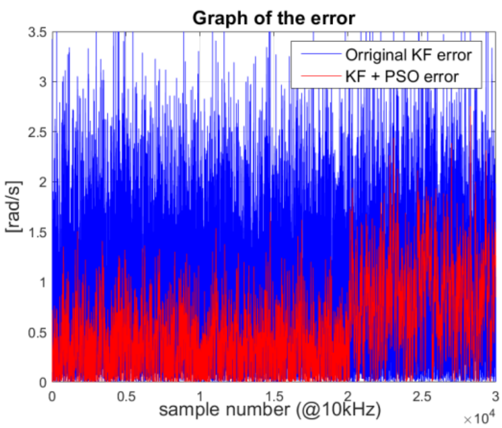

Figure 6: Error comparison between non-optimized and optimized Kalman filter

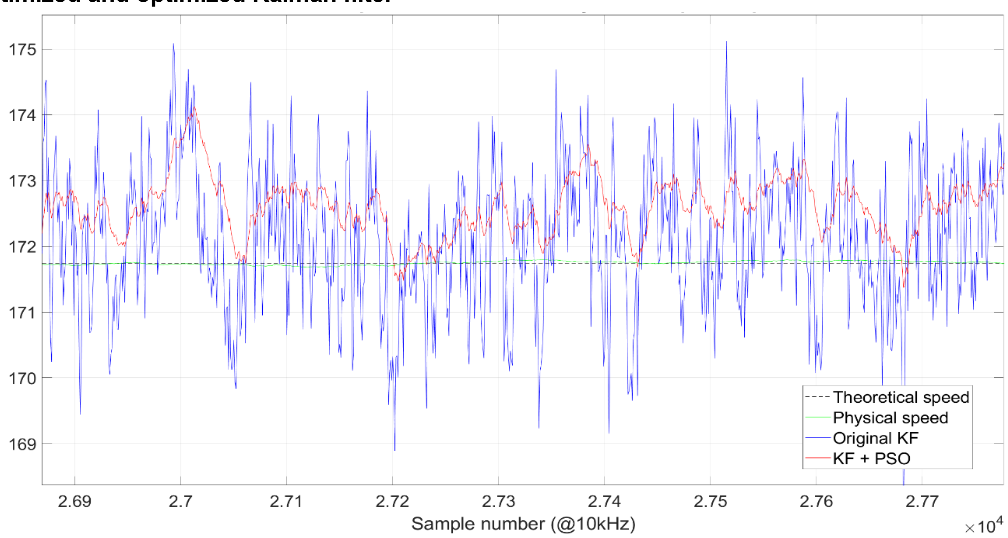

Figure 7: Zoom in figure 5

From both Table 1 and Figure 5, it is clear that the PSO has improved the efficiency of the Kalman filter noticeably, as the data recorded from the optimized filter is significantly smaller. Finally, a comparison of theoretical speed, physical speed (actual speed), non-optimized speed, and actual speed of the DC motor is shown in Figures 6 and 7. The optimal speed is closer to the physical speed than the non-optimized one. The effectiveness of the PSO algorithm is proven.

5. Conclusion

It can be inferred from this paper that accurate speed estimation plays a crucial role in the operation of a DC motor. By using the PSO algorithm to optimize the Q and R covariance matrices of the implemented Kalman filter, its effectiveness has increased significantly. This paper has successfully depicted the effectiveness of the PSO algorithm in the new method that does not require the sensor to estimate the speed of DC through simulation in MATLAB and Simulink. In the future, I plan to do further research to further improve the results obtained, as well as find ways to apply the results to devices other than DC motors.

REFERENCES

[1] S. Manglik, A. Sen, B. Singh and B. K. Panigrahi, "Low Power Sensorless PMBLDC Motor Drive with PFC for Domestic Flour Mill," 2020 3rd International Conference on Energy, Power and Environment: Towards Clean Energy Technologies, Shillong, Meghalaya, India, 2021, pp. 1-5C.C. Hwang and Y.H. Cho, "Effects of leakage flux on magnetic fields of interior permanent magnet synchronous motors," IEEE transactions on magnetics, vol. 37, no. 4, pp. 3021-3024, 2001.

[2] P. Vishnu Sidharthan and Y. Kashyap, "Brushless DC Hub Motor Drive Control for Electric Vehicle Applications," 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 2020, pp. 448-453.

[3] Q. Li, R. Li, K. Ji and W. Dai, "Kalman Filter and Its Application," 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Tianjin, China, 2015, pp. 74-77.

[4] R. P. Tripathi, P. Gangwar and A. K. Singh, "Speed Control of DC Motor with Kalman Filter and Fractional Order PID Controller," 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2021, pp. 1-4

[5] A. Polsena, Y. Kongjeen and S. Watcharakhup, "Identifying Parameter and PI Tuning of DC Motor for Predict Behavior based on PSO," 2021 9th International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 2021, pp. 97-100.

[6] M. Wei, "The Speed Control of Brushless DC Motor Based on Genetic Fuzzy PID Algorithm," 2023 3rd International Conference on Electrical Engineering and Control Science (IC2ECS), Hangzhou, China, 2023, pp. 866-872.

[7] Sivaraman Palanisamy; Sharmeela Chenniappan; A. Thaiyal Nayagi; R. Mahendran, "DC Machines," in Basic Electrical and Instrumentation Engineering, Wiley, 2021, pp.125-183,

[8] A. Khalid and A. Nawaz, "Sensor less control of DC motor using Kalman filter for low cost CNC machine," 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE), Islamabad, Pakistan, 2014, pp. 180-185.

[9] "Fundamentals of Particle Swarm Optimization Techniques," in Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems, IEEE, 2008, pp.71-87.

Biography:

Bui Minh Hien is a third-year undergraduate student at the University of Science and Technology of Hanoi (USTH), majoring in Aeronautical Engineering. He graduated from the High School for Gifted Students, Hanoi National University of Education, in 2022.

His current research focuses on linear control systems.

Pham Duy An received an engineering degree in electrical engineering from Hanoi University of Science and Technology in 2010, a master's degree in electrical engineering from the Grenoble Institute of Technology in France in 2011, and a PhD in automatic and signal processing from the University of Reims Champagne Ardenne in France in 2018. He is a lecturer in the Aeronautics Department at the University of Science and Technology of Hanoi.

His current research is focused on nonlinear control systems and optimization algorithms.