Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 10, No. 4, November, Pp. 443-454

Spoon stabilization for essential tremor patients using PID control optimized by PSO

Putri Ayu Zartika[1], Muhammad Aziz Muslim[*1], Erni Yudaningtyas[1]
Universitas Brawijaya, Indonesia[1]

Article Info
Keywords:
Essential Tremor, PID, Particle Swarm Optimization, Cohen-Coon, Kalman Filter

Article history:
Received: March 07, 2025
Accepted: June 02, 2025
Published: November 01, 2025

Cite:
P. A. Zartika, M. A. Muslim, and E. Yudaningtyas, "Spoon Stabilization for Essential Tremor Patients Using PID Control Optimized by PSO", KINETIK, vol. 10, no. 4, Nov. 2025.
https://doi.org/10.22219/kinetik.v10i4.2272

*Corresponding author.
Muhammad Aziz Muslim
E-mail address:
muh_aziz@ub.ac.id

Abstract
Essential tremor is a neurological disorder that causes uncontrollable hand tremors, interfering with daily activities such as eating. This study aims to develop a spoon stabilization system controlled by a Proportional-Integral-Derivative (PID) controller, which was tuned using Particle Swarm Optimization (PSO) and the Cohen-Coon method for performance comparison. The system utilized an inertial measurement unit to detect tremors, while a Kalman filter was used to reduce noise before a microcontroller controlled a servo motor to stabilize the spoon. The system was evaluated through simulations and hardware implementation, and its performance was assessed based on rise time, overshoot, delay time, and settling time. The results showed that the Kalman filter significantly reduced noise, lowering the average pitch angle error deviation from 1.028° to 0.037° and the roll angle error from 0.822° to 0.031°. The PSO-based tuning outperformed the Cohen-Coon method in response speed and system stability, achieving a faster rise time (0.09 s for roll, 0.34 s for pitch), a shorter settling time (0.74 s for roll, 0.59 s for pitch), and a lower delay time (0.1 s for roll, 0.15 s for pitch). However, the Cohen-Coon method resulted in a lower overshoot for the roll angle (6.08%) compared to the PSO-based tuning (11.98%). The findings suggest that implementing a PID controller optimized via PSO is a viable approach for spoon stabilization in individuals with essential tremor.

1. Introduction

Essential tremor, a chronic neurological disorder affecting approximately 2% of the population, causes tremors in the hands and arms at a frequency of 4–12 Hz, making it one of the most common movement disorders [1], [2], [3]. This condition typically manifests as postural tremors, which occur while maintaining a position against gravity, and kinetic tremors, which arise during intentional movements such as reaching or eating [4], [5]. First-line treatments commonly include medications such as propranolol or primidone. However, surveys indicate that around 35% of patients discontinue medication due to limited efficacy and adverse side effects. Furthermore, 28% of individuals have never sought medical consultation for their tremor symptoms [6]. These challenges underscore the urgent need for the development of assistive technologies aimed at enhancing patient autonomy in daily life.

Assistive technologies help individuals with disabilities perform daily tasks more independently [7]. One such application is the stabilization of eating utensils for individuals with essential tremor. In this context, a Proportional-Integral-Derivative (PID) controller can be employed to achieve system stability and mitigate tremor effects. The PID controller is widely used in industry due to its simple structure, robustness, and ease of implementation, with the added advantage of being compatible with other methods, such as optimization algorithms or filters, to enhance system performance and stability [8]. The performance of a PID controller depends on the proper tuning of three primary parameters: proportional (K_p), integral (K_i), and derivative (K_d), with common tuning methods such as Ziegler-Nichols and Cohen-Coon, the latter providing a faster response time [9], [10], [11].

Previous research [12] developed a stabilization spoon for individuals with Parkinson's disease using an MPU6050 sensor and a complementary filter to measure rotational movement. The collected data were processed by an Arduino Nano, which employed a PID controller to actuate SG90 servo motors. Experimental results demonstrated that the spoon achieved quick stabilization during slow movements. However, initial overshoots were recorded at –21.49° (pitch) and –14.89° (roll), along with oscillations ranging between –10° and 10° during severe tremor conditions. The main limitation of the study was the time-consuming manual tuning of the PID controller, which relied on a trial-and-error approach.

To address the limitations of the traditional PID tuning method, several studies have proposed the use of optimization algorithms such as Genetic Algorithm (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO). PSO is a metaheuristic algorithm inspired by the social behavior of animal swarms, with each particle representing a potential solution that explores the search space by updating its position based on its own best

experience and that of the entire swarm. This enables faster and more efficient convergence [13], [14]. A study [15] demonstrated that PID parameters optimized using PSO outperformed those obtained through SA and GA, providing faster response times, reduced overshoot, and enhanced system stability.

Research [16] developed a robotic feeding device for tremor suppression, utilizing an inertial measurement unit (IMU), two DC motors, and a PID controller optimized using a genetic algorithm. Simulation results conducted in SimMechanics demonstrated a damping efficiency of 84% to 99.41%, indicating that the spoon could be effectively stabilized during the eating process. However, the study lacked experimental validation, leaving its real-world effectiveness unconfirmed.

The MPU6050 sensor integrates an accelerometer and a gyroscope to measure hand tremors in essential tremor patients. Accelerometer data tends to have high-frequency noise, while gyroscope data experiences drift over time [17]. The Kalman Filter effectively reduces noise [17], [18], by using a mathematical algorithm that estimates system state changes in real time, providing more accurate and stable estimations [19], [20].

This study aims to develop a spoon stabilization system for individuals with essential tremor using a PID controller tuned by PSO and to compare its performance with the Cohen-Coon tuning method. The system employs an MPU6050 sensor to detect hand tremors, a Kalman filter to reduce noise, and an Arduino Nano to control two servo motors in real time. The system's effectiveness is evaluated through simulations and hardware implementation.

2. Research Method

2.1 PID Control

The PID controller is a widely used method to regulate a system to achieve the desired condition. Its structure consists of three primary parameters: proportional (K_P) which affects response speed, integral (K_i) which eliminates steady-state error, and derivative (K_d) which enhances stability and reduces overshoot. The PID controller equation, as defined in Equation 1, calculates the control variable $u(t)$ based on the system error $e(t)$, which represents the difference between the desired value (setpoint) and the actual system output [21].

$$u(t) = K_P e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{1}$$

2.2 Gimbal Model

This system consists of two joints that rotate at angles θ_1 and θ_2 along the pitch and roll axes. The pitch axis controls the spoon's up-and-down movement, while the roll axis controls its left-and-right movement. The system comprises three body frames: body (0), which is attached to the spoon handle; body (1), which connects to body (0) and rotates around the roll axis; and body (2), which connects to body (1) and rotates around the pitch axis. The movement of the end effector (body (2)) can be analyzed kinematically using the Denavit-Hartenberg (DH) convention [22], [23].

The rotation matrix between body frame (0) and body frame (1), which rotates around the roll axis, is defined in Equation 2.

$$0_{R_1} = \begin{bmatrix} cos\theta_1 & 0 & sin\theta_1 \\ 0 & 1 & 0 \\ -sin\theta_1 & 0 & cos\theta_1 \end{bmatrix} \tag{2}$$

The rotation matrix between body frame (1) and body frame (2), which rotates around the pitch axis, is defined in Equation 3.

$$1_{R_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_2 & -sin\theta_2 \\ 0 & sin\theta_2 & cos\theta_2 \end{bmatrix} \tag{3}$$

Thus, the total rotation matrix for body frames (0), (1), and (2) is defined in Equation 4.

$$0_{R_2} = \begin{bmatrix} cos\theta_1 & sin\theta_1 sin\theta_2 & sin\theta_1 cos\theta_2 \\ 0 & cos\theta_2 & -sin\theta_2 \\ -sin\theta_2 & cos\theta_1 sin\theta_2 & cos\theta_1 cos\theta_2 \end{bmatrix} \tag{4}$$

The MPU6050 sensor is placed on the end effector (body frame (2)) to measure the actual angle. The error between the desired angle and the actual sensor reading, denoted as ε_1 and ε_2, represents the difference between the setpoint angles and the measured angles for each respective axis and is defined in Equations 5 and 6.

$$\varepsilon_1 = \alpha_{1setpoint} - \alpha_1 \quad (5)$$

$$\varepsilon_2 = \alpha_{2setpoint} - \alpha_2 \quad (6)$$

From Equations 5 and 6, the rotation matrix representing the error in body frame (2) is defined in Equation 7.

$$2_{R_\varepsilon} = \begin{bmatrix} cos\ \varepsilon_1 & sin\ \varepsilon_1 sin\ \varepsilon_2 & sin\ \varepsilon_1 cos\ \varepsilon_2 \\ 0 & cos\ \varepsilon_2 & -sin\ \varepsilon_2 \\ -sin\ \varepsilon_2 & cos\ \varepsilon_1 sin\ \varepsilon_2 & cos\ \varepsilon_1\ cos\ \varepsilon_2 \end{bmatrix} \quad (7)$$

Using inverse kinematics, the relationship between the total rotation matrix and the error in body frame (2) is defined in Equation 8.

$$0_{R_\varepsilon}(\theta, \alpha) = 0_{R_2}(\theta) 2_{R_\varepsilon}(\alpha) \quad (8)$$

From Equation 8, the relationship between the error and body frame (0) can be expressed in Equation 9.

$$0_{R_2}(\theta) 2_{R_\varepsilon}(\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (9)$$

In Equation (9), r_{ij} represents the elements of the transformation matrix obtained from the multiplication of $0_{R_2}(\theta)$ and $2_{R_\varepsilon}(\alpha)$. The new motor angles required to maintain the spoon's orientation are defined in Equations 10 and 11 for the roll and pitch axes, respectively.

$$\theta_{1new} = tan^{-1}(\frac{r_{31}}{r_{33}}) \quad (10)$$

$$\theta_{2new} = sin^{-1}(r_{32}) \quad (11)$$

2.3 DC Motor Model

The transfer function of the gimbal actuator used to achieve the desired position through the controller is defined in Equation 12 [24],

$$\frac{\theta(s)}{V(s)} = \frac{K}{s[(R + Ls)(Js + b) + K^2]} \quad (12)$$

where K is the back EMF (Nm/A), L is the inductance (H), R is the resistance (Ω), J is the moment of inertia of the motor (Kg m^2/s^2), and B is the damping ratio (Nm/rad).

2.4 Kalman Filter

The Kalman filter is a set of mathematical equations designed to estimate state changes in real time by accounting for noise disturbances in sensor data. The Kalman Filter algorithm used in this system is illustrated in Figure 1. It consists of two main stages: Time Update and Measurement Update [19]. The MPU-6050 sensor reads raw data from the accelerometer and gyroscope. The accelerometer data is then converted into angles, while the gyroscope data calculates angular velocity.

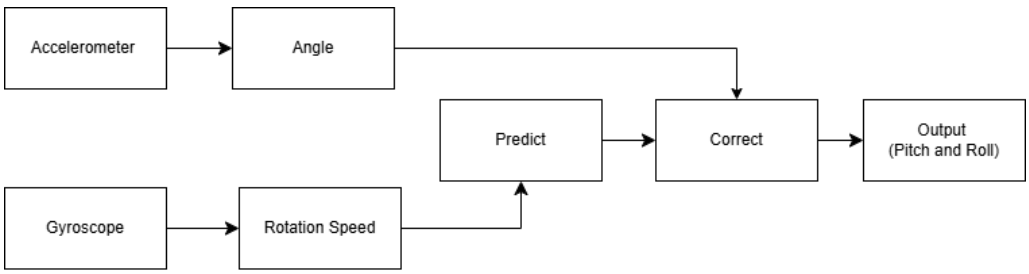


Figure 1. Kalman Filter Block Diagram

In the Kalman filter, the prediction step uses gyroscope data to estimate the sensor's position (angle) in the next step. The equations for this prediction process are given in Equations 13 and 14.

$$\hat{X}_{\bar{k}} = F\hat{X}_{k-1} + Bu_{k-1} \tag{13}$$

$$P_{\bar{k}} = FP_{k-1}F^T + Q \tag{14}$$

The correction step uses accelerometer data to refine the gyroscope-based prediction. The equations for this correction process are given in Equations 15, 16, and 17.

$$K_k = P_{\bar{k}}\,H^T(R + HP_{\bar{k}}H^T)^{-1} \tag{15}$$

$$\hat{X}_k = \hat{X}_{\bar{k}} + K_k(Z_k - H\hat{X}_{\bar{k}}) \tag{16}$$

$$P_k = (I - K_kH)P_{\bar{k}} \tag{17}$$

In these equations, $\hat{X}_k$ is the estimated state, F is the state transition matrix, u is the control input, B is the control matrix, H is the measurement matrix, Z is the measurement variable, P is the state covariance matrix, and K is the Kalman filter gain. The process noise is represented by Q, and the measurement noise is represented by R [17].

2.5 PSO Tuning Method

Particle Swarm Optimization (PSO) is a stochastic optimization technique inspired by the movement behavior of animals searching for food [23]. Unlike individuals in the Genetic Algorithm (GA), which may survive or be eliminated in the next generation, PSO particles persist throughout the process without being removed. Additionally, PSO incorporates a memory mechanism that allows each particle to retain its best solution. Its simpler structure enables faster convergence compared to GA [25], [26]. The design process of the spoon stabilization system using PID control optimized with PSO is shown in Figure 2.

The system receives an input in the form of a reference angle (setpoint), which is compared with the actual spoon position to generate an error signal. This signal is processed by the PID controller, which generates a control signal to drive the servo motor. The tuning of the PID control parameters (K_P, K_i, and K_d)) using PSO is performed in a non-real-time manner.

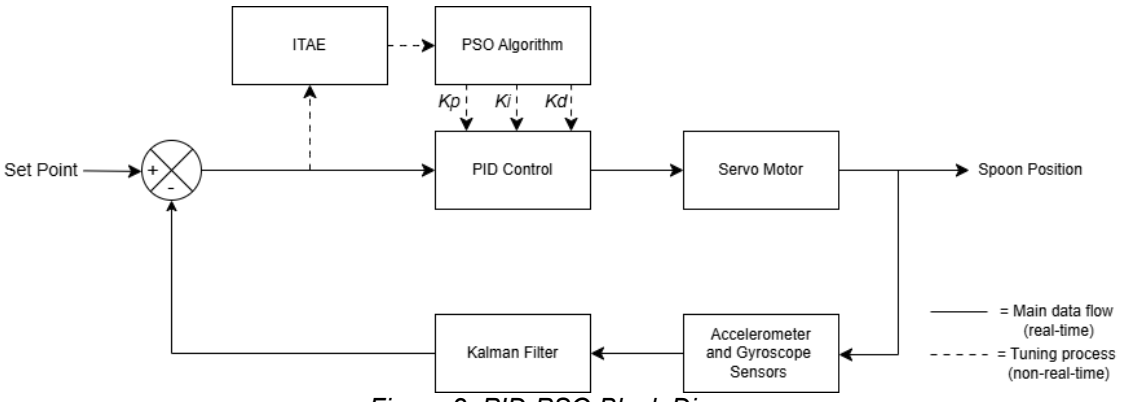


Figure 2. PID-PSO Block Diagram

PSO initializes a population of particles randomly within the search space, where each particle stores its personal best position (*Pbest*) and the global best position (*Gbest*). The position and velocity of each particle are iteratively updated as defined in Equations 18 and 19 [25],

$$v_{i,m}^{(t+1)} = w.v_{i,m}^{(t)} + C_1 * Rand * \left(Pbest_{i,m} - x_{i,m}^{(t)}\right) + C_2 * Rand * \left(Gbest_m - x_{i,m}^{(t)}\right) \tag{18}$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(i)} + v_{i,m}^{(t+1)}, i = 1, 2, \dots, n \tag{19}$$

where n is the number of particles, w is the inertia weight, C_1 and C_2 are positive constants, $Rand$ represents two random numbers between 0 and 1, t is the iteration number, $Pbest$ is the best position of particle, and $Gbest$ is the best-performing particle in the swarm.

Each particle represents a unique set of PID parameters (K_P, K_i and K_d) and generates a control signal to drive the servo motor. The performance of each particle is evaluated using the Integral of Time-weighted Absolute Error (ITAE) index, which assesses control quality based on the magnitude of error and is given in Equation 20. A lower ITAE value indicates better system performance. ITAE is widely recognized as an effective evaluation function that accounts for multiple performance aspects, including settling time, overshoot, and steady-state error [27].

$$ITAE = \int_0^\infty t|e(t)|dt \tag{20}$$

The servo motor movement results in a change in the spoon's angular position, which serves as the system output. To obtain accurate feedback, an accelerometer and gyroscope are employed to measure acceleration and angular velocity. The sensor data, which contains noise, is filtered using a Kalman Filter, producing a smoother estimation of the angular position to be used as feedback input to the system.

2.6 Cohen-Coon Tuning Method

The Cohen-Coon method is a conventional PID tuning approach that offers better steady-state response than Ziegler-Nichols, with a faster rise time and a more complex formulation [28]. The overall control mechanism is illustrated in Figure 3, where the system receives an input in the form of a reference angle (set point) which is compared with the actual spoon position to generate an error signal. This signal is processed by the PID controller to produce a control signal that drives the servo motor. The PID control parameters (K_P, K_i and K_d) are determined through the Cohen-Coon tuning method, which is performed in a non-real-time manner by analyzing the system response using the open-loop method by applying a step input signal. From the response graph, the time delay (τ_d), time constant (τ), and **system gain** (K) can be determined [11]. The **PID parameters** (K_P, K_i and K_d) are then calculated using the formulas provided in Table 1.

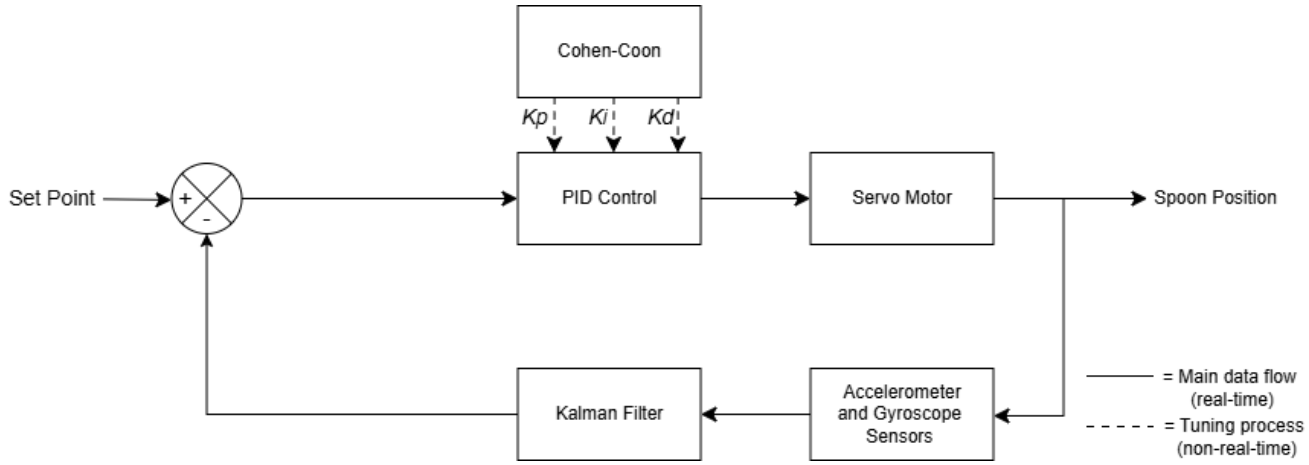


Figure 3. PID-Cohen Coon Block Diagram

Table 1. Cohen-Coon PID Parameters

Control Parameters	K_P	T_i	T_d	K_i	K_d
P	$\frac{\tau}{K\tau_d}(1+\frac{\tau_d}{3\tau})$	-	-	-	-
PI	$\frac{\tau}{K\tau_d}(0{,}9+\frac{\tau_d}{3\tau})$	$\tau_d\left(\frac{30+\frac{3\tau_d}{\tau}}{9+\frac{20}{\tau}}\right)$	-	$\frac{K_P}{\tau_i}$	-

Control Parameters	K_P	T_i	T_d	K_i	K_d
PID	$\frac{\tau}{K\tau_d}(1{,}33 + \frac{\tau_d}{4\tau})$	$\tau_d\left(\frac{32+\frac{6\tau_d}{\tau}}{13+\frac{8}{\tau}}\right)$	$\tau_d\left(\frac{4}{11+\frac{2\tau_d}{\tau}}\right)$	$\frac{K_P}{\tau_i}$	$K_P\ x\ T_d$

3. Results and Discussion

3.1 Simulation Result

The simulation in Figure 4 implements a PID controller optimized using PSO to stabilize the roll and pitch angles. A MATLAB Function block calculates the set-point based on the gimbal kinematics in the pitch and roll axes, which is then used to control the corresponding servo motors to maintain spoon stability. The tremor movement data for the roll and pitch angles used in the simulation were obtained from an MPU-6050 sensor, which was filtered using a Kalman Filter to suppress noise. Fast Fourier Transform (FFT) analysis identifies the dominant tremor frequency at approximately 4.5 Hz.

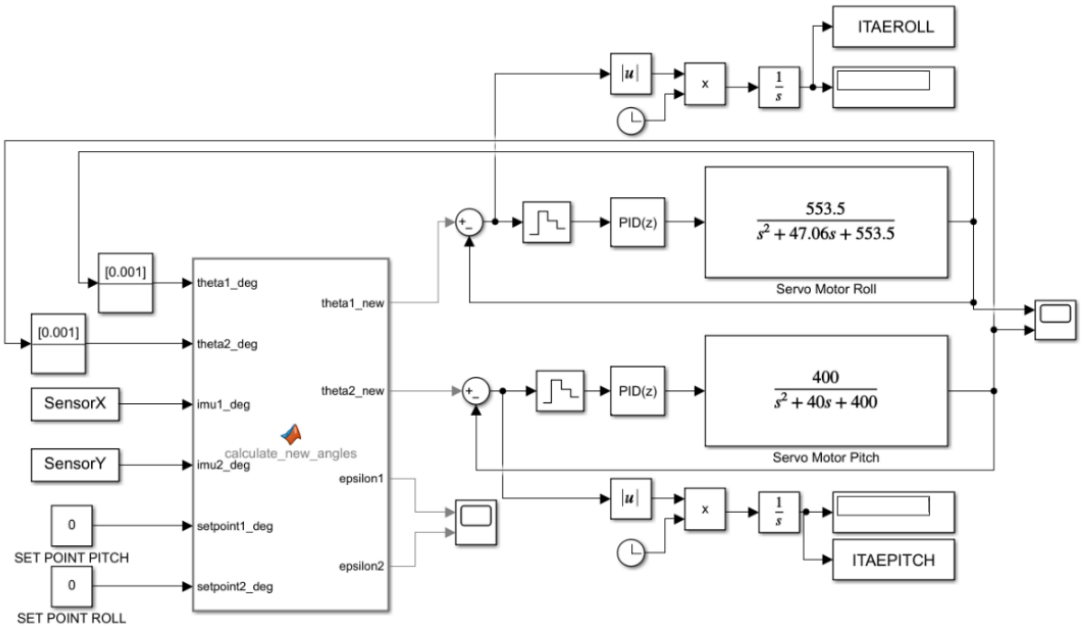


Figure 4. PID-PSO Simulation

The Particle Swarm Optimization (PSO) algorithm is applied to optimize the PID controller parameters, considering key factors that influence optimization effectiveness, as shown in Table 2.

Table 2. PSO Parameters

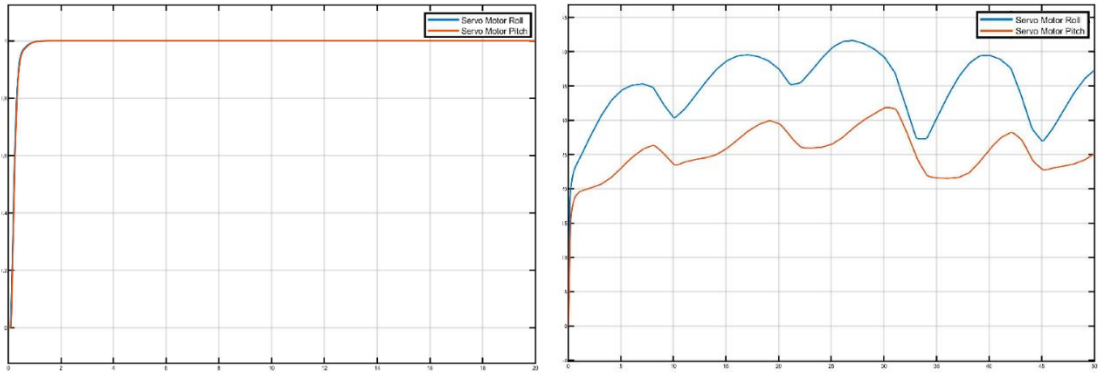
Parameter	Variable	Value
Cognitive component	C_1	1.5
Social component	C_2	1.5
Number of particles (Population)	N	50
Number of iterations	I	50
Upper Bound	U_b	5, 10, 5
Lower Bound	L_b	0.001, 0.001, 0.0001
Minimum inertia weight	W_{min}	0.4
Maximum inertia weight	W_{max}	0.9
Dimension (Number of parameters)	dim	3

The **PID parameters** (K_P, K_i, and K_d) and ITAE for the pitch and roll angles are obtained after 50 iterations, as shown in Table 3. A lower ITAE indicates a more optimal control system performance.

Table 3. PSO-Tuned PID Parameters

Controller	PID Parameter			ITAE
	K_P	K_i	K_d	
PID-PSO *Roll*	0.357661535049177	3.70696057614307	0.0001	0.04367
PID-PSO *Pitch*	0.403286597553169	3.97528929343970	0.0001	0.03862

The PID-PSO simulation results are shown in Figure 5(a) as a step response, while Figure 5(b) presents the sensor response using the MPU-6050 and Kalman Filter, illustrating the tremor movement readings captured by the sensor.



(a) (b)

Figure 5. Simulation Results: (a) Step Response, (b) System Response to Pitch and Roll Tremor Movements

Based on the system testing results using the PID-PSO controller, the system's performance response is presented in Table 6. The PSO-PID simulation demonstrated a fast and stable response for both angles. The roll angle achieved a rise time of 1.771 s and a settling time of 0.708 s, while the pitch angle had a rise time of 1.727 s and a settling time of 0.667 s, indicating that the system quickly reached stability. An overshoot of 0% and a steady-state error of 0 confirm stability without oscillations or deviations from the setpoint. Additionally, a peak time of 0 indicates that the system did not experience any overshoot before stabilizing, with low delay times of 0.229 s for roll and 0.212 s for pitch.

Table 4. Simulation Response Performance

Controller	Parameter					
	Rise Time (s)	***Max Overshoot (%)***	***Settling Time (s)***	***Peak Time (s)***	***Delay Time (s)***	***Steady State Error***
PSO-PID *Roll*	1.771	0	0.708	0	0.229	0
PSO-PID *Pitch*	1.727	0	0.667	0	0.212	0

3.2 Hardware Implementation Result

The implementation results of the stabilization spoon for essential tremor patients are shown in Figure 6. The system consists of an Arduino Nano as the microcontroller, two servo motors as actuators, an MPU6050 sensor to detect spoon bowl movements, and a battery as the power source.

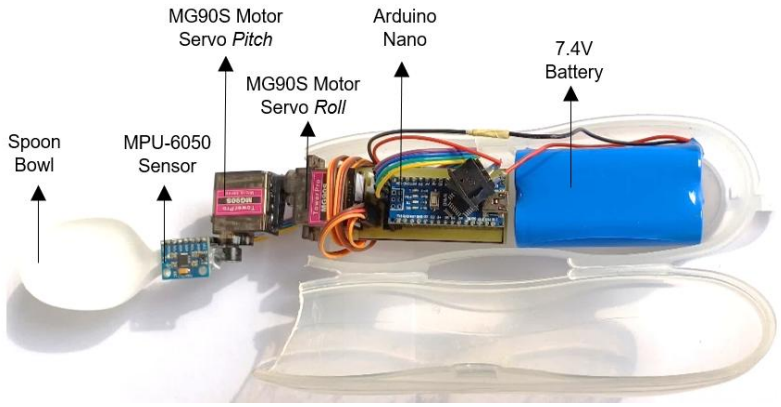


Figure 6. Stabilizing Spoon for Essential Tremor

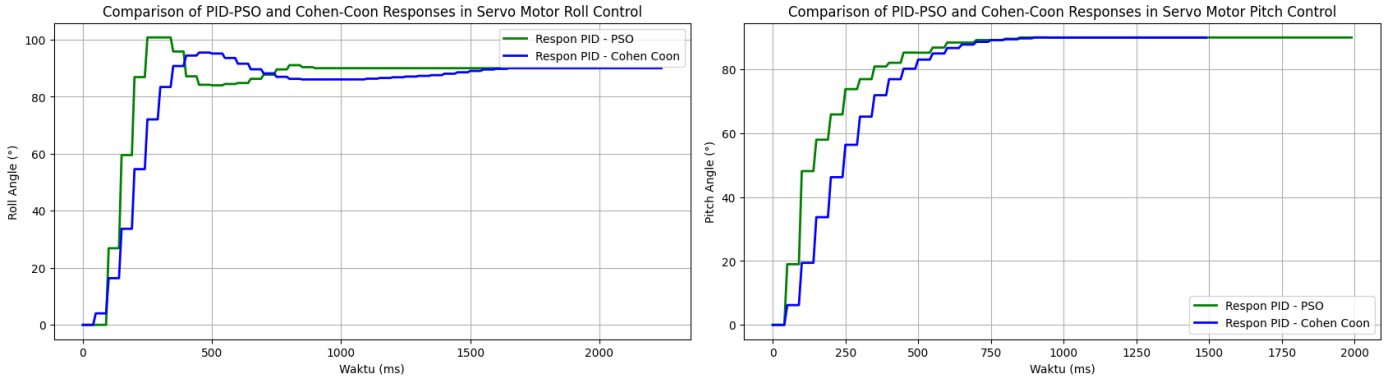
3.3 Comparison of PID controller using PSO and Cohen Coon

After analyzing the system response using the open-loop method with a step input on pitch and roll angles, the response graph is used to calculated the delay time (τ_d), time constant (τ), and system gain (K). Based on these parameters, the PID parameters (K_P, K_i and K_d) are calculated, as shown in Table 5.

Table 5. Cohen-Coon Tuned PID Parameters

Controller	PID Parameter		
	K_p	K_i	K_d
PID - Cohen Coon *Roll*	0.0175	2.45	0.000215
PID - Cohen Coon *Pitch*	0.03221	2.8938	0.0004245

Hardware testing using PID-PSO parameters showed performance differences compared to the simulation results, as the simulation does not fully represent real-world system conditions. Based on the performance response comparison in Table 6 and Figure 7, the PID-PSO Roll system demonstrated the best performance with a very fast rise time of 0.09 s and a short settling time of 0.74 s, although it experienced an overshoot of 11.98%. For the PID-PSO Pitch controller, the response was stable with no overshoot (0%) and a faster settling time of 0.59 s, along with zero steady-state error. Both controllers showed satisfactory results in terms of accuracy and speed. On the other hand, the PID-Cohen Coon Roll had a longer rise time (0.19 s) and a longer settling time (1.44 s) with a lower overshoot of 6.08%, while the PID-Cohen Coon Pitch also had a slower rise time (0.39 s) and settling time (0.69 s), with a very small overshoot of 0.33%. These hardware testing results indicate that the PID-PSO provides a faster dynamic response compared to the PID-Cohen Coon. This finding aligns with the study in [23] , which reported that PSO tuning achieves superior performance compared to other tuning methods, such as Ziegler-Nichols, Cohen-Coon, and Genetic Algorithm (GA), in camera gimbal stabilization systems.



(a) *(b)*

Figure 7. PID-PSO and Cohen-Coon Responses: (a) Servo Motor Roll, (b) Servo Motor Pitch

Table 6. Response Performance of PID-Cohen Coon and PID-PSO

Controller	Parameter					
	Rise Time (s)	***Max Overshoot (%)***	***Settling Time (s)***	***Peak Time (s)***	***Delay Time (s)***	***Steady State Error***
PID-PSO *Roll*	0.09	11.98	0.74	0.25	0.15	0
PID-PSO *Pitch*	0.34	0	0.59	0	0.1	0
PID-Cohen Coon *Roll*	0.19	6.08	1.44	0.45	0.2	0
PID-Cohen Coon *Pitch*	0.39	0.33	0.69	0.9	0.2	0

The findings suggest that implementing a PID controller optimized via PSO is a viable approach for spoon stabilization in individuals with essential tremor. Its rapid response and short settling time offer crucial advantages for supporting independent eating, thereby contributing to improved quality of life and daily autonomy.

3.4 Kalman filter

The accuracy of MPU-6050 sensor data was evaluated with and without the Kalman filter by incrementally adjusting the sensor from 10º to 180º in 10º steps for both pitch and roll angles. The results are presented in Table 7 (pitch) and Table 8 (roll).

Table 7. Kalman Filter Testing Data on Pitch Angle

Arc Angle	Pitch Angle (Without Kalman)	Pitch Angle (With Kalman)	Error Deviation (Without Kalman)%	Error Deviation (With Kalman)%
10°	10.9	10	0.9	0
20°	19.1	20.02	0.9	0.02
30°	30.33	30	0.33	0
40°	40.51	40	0.51	0
50°	48.77	50.05	1.23	0.05
60°	60.91	60	0.91	0
70°	70.32	69.82	0.32	0.18
80°	80.36	80.06	0.36	0.06
90°	89.46	90	0.54	0
100°	99	100	1	0
110°	108.86	110.2	1.14	0.2
120°	118.25	120.03	1.75	0.03
130°	131.58	130.06	1.58	0.06
140°	141.73	139.98	1.73	0.02
150°	148.55	150	1.45	0
160°	161.86	159.96	1.86	0.04
170°	171.5	169.99	1.5	0.01
180°	179.5	180	0.5	0
	Average		1.028	0.037

Table 8. Kalman Filter Testing Data on Roll Angle

Arc Angle	Roll Angle (Without Kalman)	Roll Angle (With Kalman)	Error Deviation (Without Kalman)%	Error Deviation (With Kalman)%
10°	10.68	10.08	0.68	0.08
20°	20.5	20.03	0.5	0.03
30°	30.3	30.04	0.3	0.04
40°	41.75	40	1.75	0
50°	50.21	50	0.21	0
60°	61.16	60	1.16	0
70°	71.54	70	1.54	0
80°	80.87	80	0.87	0
90°	90.8	90	0.8	0
100°	99.33	100.06	0.67	0.06
110°	109.76	110	0.24	0
120°	122	120	2	0
130°	129.43	130.1	0.57	0.1
140°	140.25	140.01	0.25	0.01
150°	151.56	150.09	1.56	0.09
160°	160.5	160.06	0.5	0.06
170°	169.7	170.09	0,3	0.09
180°	179.1	180	0.9	0
	Average		0.822	0.031

The test results showed that the average error deviation without the Kalman filter was 1.028° for the pitch angle and 0.822° for the roll angle. After applying the Kalman filter, the error significantly decreased to 0.037° for the pitch angle and 0.031° for the roll angle. This reduction in error indicates that the Kalman filter is effective in mitigating noise in sensor readings, providing more accurate angle estimations. These results are consistent with the findings of [17], which indicates that the Kalman filter has strong capabilities in handling noise by using mathematical algorithms that estimate system state changes in real-time, thereby providing more accurate and stable estimations.

4. Conclusion

The hardware implementation, utilizing an Arduino Nano, servo motors, and an MPU-6050 sensor with a Kalman filter was successfully applied. The Kalman filter significantly improved sensor accuracy by reducing the average error deviation from 1.028° to 0.037° for pitch and from 0.822° to 0.031° for roll. Simulations in Simulink demonstrated that the PID-PSO controller effectively regulated the roll and pitch angles, achieving rise times of 1.771 s and 1.727 s, low delay times of 0.229 s and 0.212 s, settling times of 0.708 s and 0.667 s, and 0% overshoot. The hardware implementation resulted in faster rise times for roll and pitch, at 0.09 s and 0.34 s respectively, along with shorter delay times. The settling times remained nearly the same, at 0.74 s and 0.59 s, although an 11.98% overshoot was observed in the roll axis. These differences are attributed to real-world system conditions that are not fully represented in the simulation. Compared to the Cohen-Coon method, PID-PSO exhibited a faster and more accurate response in mitigating tremors, despite Cohen-Coon having a lower overshoot of 6.08% for roll and 0.33% for pitch. The PID-PSO-based stabilization system demonstrates improved responsiveness and holds significant potential for assisting individuals with tremor in performing daily activities.

References

[1] P. Mcgurrin, J. Mcnames, T. Wu, M. Hallett, and D. Haubenberger, "Quantifying Tremor in Essential Tremor Using Inertial Sensors—Validation of an Algorithm," *IEEE J Transl Eng Health Med*, vol. 9, pp. 1–10, 2021. https://doi.org/10.1109/jtehm.2020.3032924

[2] J. Kim, T. Wichmann, O. T. Inan, and S. P. DeWeerth, "Fitts' Law Based Performance Metrics to Quantify Tremor in Individuals with Essential Tremor," *IEEE J Biomed Health Inform*, vol. 26, no. 5, pp. 2169–2179, May 2022. https://doi.org/10.1109/JBHI.2021.3129989

[3] S. M. Ali *et al.*, "Wearable Accelerometer and Gyroscope Sensors for Estimating the Severity of Essential Tremor," *IEEE J Transl Eng Health Med*, vol. 12, pp. 194–203, 2024. https://doi.org/10.1109/jtehm.2023.3329344

[4] J. Li *et al.*, "A Wearable Multi-Segment Upper Limb Tremor Assessment System for Differential Diagnosis of Parkinson's Disease Versus Essential Tremor," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 3397–3406, 2023. https://doi.org/10.1109/tnsre.2023.3306203

[5] A. B. Oktay and A. Kocer, "Differential diagnosis of Parkinson and essential tremor with convolutional LSTM networks," *Biomed Signal Process Control*, vol. 56, p. 101683, Feb. 2020. https://doi.org/10.1016/j.bspc.2019.101683

[6] H. V. Gupta, R. Pahwa, P. Dowell, S. Khosla, and K. E. Lyons, "Exploring essential tremor: Results from a large online survey," *Clin Park Relat Disord*, vol. 5, p. 100101, 2021. https://doi.org/10.1016/j.prdoa.2021.100101

[7] M. Al-Ayyad, H. A. Owida, A. Al-Ghraibah, and M. R. Abdullah, "Smart Assistive Spoon for people with Parkinson's Disease," in *2022 13th International Conference on Information and Communication Systems (ICICS)*, IEEE, Jun. 2022, pp. 422–425. https://doi.org/10.1109/ICICS55353.2022.9811112

[8] J. P. de Moura, J. V. da F. Neto, and P. H. M. Rego, "A Neuro-Fuzzy Model for Online Optimal Tuning of PID Controllers in Industrial System Applications to the Mining Sector," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 8, pp. 1864–1877, Aug. 2020. https://doi.org/10.1109/TFUZZ.2019.2923963

[9] S. Singh, V. Singh, A. Rani, and J. Yadav, "Optimization of PID controller based on various tuning methods," in *2023 International Conference on Power, Instrumentation, Energy and Control (PIECON)*, IEEE, Feb. 2023, pp. 1–6. https://doi.org/10.1109/PIECON56912.2023.10085805

[10] B. Verma and P. K. Padhy, "Robust Fine Tuning of Optimal PID Controller with Guaranteed Robustness," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 4911–4920, Jun. 2020. https://doi.org/10.1109/TIE.2019.2924603

[11] F. Isdaryani, F. Feriyonika, and R. Ferdiansyah, "Comparison of Ziegler-Nichols and Cohen Coon tuning method for magnetic levitation control system," *J Phys Conf Ser*, vol. 1450, no. 1, p. 012033, Feb. 2020. https://doi.org/10.1088/1742-6596/1450/1/012033

[12] N. Onasie and S. Sulaiman, "Perancangan Sendok Makan Parkinson dengan Metode PID berbasis Arduino," *Techné : Jurnal Ilmiah Elektroteknika*, vol. 22, no. 1, pp. 33–48, Apr. 2023. https://doi.org/10.31358/techne.v22i1.346

[13] A. Surana and B. Bhushan, "Designing of PSO Tuned PID Controller for Ball Balancer Arrangement and Comparative Analysis with Classical PID and Fuzzy Logic Controller," in *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, IEEE, Jul. 2021, pp. 1–5. https://doi.org/10.1109/CONECCT52877.2021.9622355

[14] R. A. Khan, S. Yang, S. Fahad, S. U. Khan, and Kalimullah, "A Modified Particle Swarm Optimization With a Smart Particle for Inverse Problems in Electromagnetic Devices," *IEEE Access*, vol. 9, pp. 99932–99943, 2021. https://doi.org/10.1109/ACCESS.2021.3095403

[15] A. Surana and B. Bhushan, "Design and Comparison of PSO, SA and GA tuned PID Controller for Ball Balancer Arrangement," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, Sep. 2021, pp. 1–5. https://doi.org/10.1109/ICECCT52121.2021.9616686

[16] B. Taşar, A. B. Tatar, A. K. Tanyıldızı, and O. Yakut, "FiMec tremor stabilization spoon: design and active stabilization control of two DoF robotic eating devices for hand tremor patients," *Med Biol Eng Comput*, vol. 61, no. 10, pp. 2757–2768, Oct. 2023. https://doi.org/10.1007/s11517-023-02886-z

[17] R. I. Alfian, A. Ma'arif, and S. Sunardi, "Noise Reduction in the Accelerometer and Gyroscope Sensor with the Kalman Filter Algorithm," *Journal of Robotics and Control (JRC)*, vol. 2, no. 3, 2021. https://doi.org/10.18196/jrc.2375

[18] S. Yuliawan, O. Wahyunggoro, and N. Setiawan, "Kalman Filter to Improve Performance of PID Control Systems on DC Motors," *IJITEE (International Journal of Information Technology and Electrical Engineering)*, vol. 5, no. 3, p. 96, Sep. 2021. https://doi.org/10.22146/ijitee.64511

[19] M. H. Setiawan, A. Ma'arif, C. Rekik, A. J. Abougarair, and A. M. Mekonnen, "Enhancing Speed Estimation in DC Motors using the Kalman Filter Method: A Comprehensive Analysis," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 10, no. 1, p. 30, Feb. 2024. http://dx.doi.org/10.26555/jiteki.v10i1.26591

[20] J. Khodaparast, "A Review of Dynamic Phasor Estimation by Non-Linear Kalman Filters," *IEEE Access*, vol. 10, pp. 11090–11109, 2022. https://doi.org/10.1109/ACCESS.2022.3146732

[21] J. Liu, T. Li, Z. Zhang, and J. Chen, "NARX Prediction-Based Parameters Online Tuning Method of Intelligent PID System," *IEEE Access*, vol. 8, pp. 130922–130936, 2020. https://doi.org/10.1109/ACCESS.2020.3007848

[22] R. N. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control (2nd Edition)*. Springer US, 2010.

[23] R. J. Rajesh and P. Kavitha, "Camera gimbal stabilization using conventional PID controller and evolutionary algorithms," in *2015 International Conference on Computer, Communication and Control (IC4)*, IEEE, Sep. 2015, pp. 1–6. https://doi.org/10.1109/IC4.2015.7375580

[24] E. S. Rahayu, A. Ma'arif, and A. Çakan, "Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor," *International Journal of Robotics and Control Systems*, vol. 2, no. 2, pp. 435–447, Jul. 2022. https://doi.org/10.31763/ijrcs.v2i2.476

[25] K. Suyanto, K. N. Ramadhani, and S. Mandala, "Deep learning modernisasi machine learning untuk big data," *Informatika*, 2019.

[26] D. P. Tripathi, M. Nayak, R. Manoj, and S. Sudheer, "Fast parameter free Smart particle swarm optimization (SPSO)," in *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, IEEE, Jan. 2021, pp. 1–7. https://doi.org/10.1109/ICNTE51185.2021.9487738

[27] M. Xu, "Control of DC adjustable speed electric vehicle based on PSO-PID algorithm optimization research," in *2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, IEEE, Nov. 2022, pp. 616–621. https://doi.org/10.1109/AUTEEE56487.2022.9994335

[28] J. J, L. D. V. Anand, and H. D, "Comparative Analysis of PID Controller Tuning Methods for Heat Exchanger Control: A MATLAB Simulation Study," in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, IEEE, Jul. 2024, pp. 1–6. https://doi.org/10.1109/APCIT62007.2024.10673460