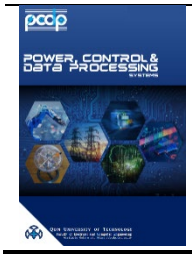
DOI: 10.30511/pcdp.2024.718236 - ISSN: 3060-8376 - Vol 1(1), 2024



POWER, CONTROL AND DATA PROCESSING SYSTEMS

Available Online at: https://pcdp.qut.ac.ir/

PD_PI Fuzzy Controller Based on PSO for Stabilization of Single-Axis Gimbal System

ARTICLE INFO

Article Type
Original Research

Authors
Seyyed Hamid Khatami[1]
Amirhossein Nakhaee[2]

[1] Department of electrical and computer engineering, Birjand University of Technology, Birjand, Iran,
h.khatami@birjand.ac.ir
[2] Department of Electrical and computer engineering, Birjand University of Technology, Birjand, Iran,
amirhosseinnakhaee@gmail.com

**** Correspondence***
Address: Department of electrical and computer engineering, Birjand University of Technology, Aviny Boulevard,
Birjand, Iran.
Phone: +98 25 36169
H.khatami@birjand.ac.ir

Article History
Received: October 16, 2024
Accepted: November 18, 2024
ePublished: December 01, 2024

ABSTRACT

Nowadays, the gimbal stabilizes the line of sight and eliminates vibration in systems such as imaging, radar line of sight, and position stabilizers. The gimbal system tries to maintain the system's current state by dealing with the changes made in the system's current state. This system reduces unwanted motion disturbances and vibrations .It is particularly beneficial in dynamic environments where maintaining stability and clarity is crucial, such as in drones, spacecraft, and aerial photography. The most common use of the gimbal system is in modern photography equipment. Additionally, gimbal stabilizers have found applications in robotics and virtual reality systems, where precision and smooth movement are essential. In this article, a single-axis gimbal stabilizer system that uses a PD_PI fuzzy controller has been investigated and the PSO algorithm has been used to optimize and calculate controller coefficients. The dynamic relations of the gimbal are described and the proposed control system based on the PD_PI fuzzy controller is optimized using the PSO algorithm. The controller coefficients have been optimized with the lowest possible settling time. The results demonstrate significant improvements in performance, particularly in minimizing response time and enhancing stability under varying conditions. The comparison of the obtained results shows that this controller has less settling time and much less overshoot than other controllers such as PID and fuzzy PID controllers.

Keywords: gimbal; stabilization; line of sight; PSO algorithm, inertial stabilization system; optimization.

1 Introduction

Many optical equipment such as lasers, imaging equipment, and radars are usually mounted on a mobile platform. In this equipment, the axis of the optical sensor must be pointed and fixed to the moving target. The need to regulate the fixed position in various applications has caused the formation and development of systems that can achieve this goal. These systems require precise control mechanisms to counteract the motion disturbances and maintain alignment with the target. The main challenge in this type of system is mainly caused by the desired or unwanted inertial movements of the devices.

These systems must have the ability to deal with these effects and minimize their impact on the system situation. The use of an inertial stabilizer system (ISP) is an appropriate solution that can solve this challenge [1]. This system can stabilize the line of sight (LOS) in the conditions of various disturbances. The most important sources of disturbance in this system are the angular movement of the platform and mass imbalance.

Most inertial stabilization systems include a gimbal device. A gimbal is a set of motors and gyroscopes that detect disturbances in the system in an inertial space using electronic sensors such as inertial measurement units (IMU). Also, by applying torque to stabilize the line of sight, disturbances can be reduced to the minimum possible amount. This system aims to compensate for vibrations, vibrations, and nonlinear disturbances such as friction, mass imbalance, dynamic mass imbalance, etc. [2].

The inertial stabilizer system is used in many different engineering applications such as weapon systems, telescopes, and cameras. Its structure and settings are designed according to the need and desired function. The first step of checking this system is checking its mechanical system. When examining many structures and mechanical mechanisms, one of the main stages of research is finding an appropriate mathematical model; A model that expresses the physical phenomena affecting the system**.** In some cases, the model must also account for external factors such as environmental conditions, which can significantly impact the system's performance. Choosing models that have enough accuracy and simplicity is always one of the important concerns of researchers and engineers.

Control knowledge is no exception to this rule. The accuracy of the control performance of a dynamic system is directly affected by the accuracy of its dynamic equation modeling. Many of the dynamic equations that describe the behavior of the system have errors that cover a wide range of variables. What adds to the complexity of these conditions in many cases is the inability to accurately measure the error factors. As a result, this problem causes that in some cases, all parts of the system are not included in its mathematical modeling [2] and its simple model is checked. Furthermore, environmental disturbances and external forces can exacerbate the inaccuracies in modeling, leading to further challenges in control system design. To model the gimbal system, in addition to the dynamic model, it is necessary to model the servo motor system and the gyroscope system. After modeling, the appropriate controller is designed and applied to the system.

The researches that have been carried out so far show that researchers have tried to control the disturbances that occur in the system by using robust linear controllers, different complex nonlinear controllers, or by combining control methods. 3]. Their goal is to provide a suitable method to identify the unknown parameters of the dynamic equation of the system and control them.

By studying the research done so far, the uncertain parameters considered for the gimbal in the dynamic equations include the components of the moment of inertia matrix and the disturbance torque matrix resulting from the center of gravity of the load not being located on the axis of rotation. Among the different control methods, the resistant control method can better control systems with uncertain parameters [4]. The advantage of the resistant control method is the lack of online calculations and the inherent resistance of the system against limited disturbances. The problem with this method is that it requires an accurate initial estimate for the range of variables.

A gimbal is an electromechanical device that is mainly used to isolate optical equipment from motion disturbances caused by the environment, such as torques caused by body movement [5]. Previous research has shown that by installing an electro-optical device on the gimbal assembly, line of sight vibrations can be reduced. Such systems are usually required to maintain stable performance and ensure accurate target tracking. These systems must maintain their stability even in case of changes in system dynamics and operational conditions. This highlights the importance of robust control strategies to adapt to varying operational scenarios and ensure consistent performance.

So far, the mathematical model and gimbal control system have been studied in many researches. Several methods have been presented to stabilize the gimbal. In [6], the accuracy of gimbal stabilization systems has been investigated. The geometric motion relations for the gimbal with two degrees of freedom have been studied in [7]. In this study, the gimbal is assumed to be in equilibrium and the gimbal components are suspended on the main axes. The equations of motion for the two-axis gimbal, assuming the imbalance of the gimbal mass, are presented in [7]. Two studies [6] and [7] have not been simulated.

Also, the stability of a gimbal with one degree of freedom (SDOF) in an environment with high vibration has been investigated in [8]. This research has minimized the vibrations caused by disturbance torques with the help of mass coefficients of imbalance of static mass and dynamic mass. Also, this method reduces the cost and reduces the execution time [8].

In [9], the gimbal motion equations are shown assuming dynamic mass imbalance. The gimbal mass distribution is calculated according to its axes. In this article, the effects of the angular velocity of the platform are considered. In [10], the

gimbal with a large size is modeled and simulated. In this research, the axis of constant height and momentary inertia is assumed to be zero. In two researches [7] and [10], the dynamic model of the height and direction axes has been obtained, assuming that the gimbal mass distribution is symmetrical concerning its axes. Therefore, the effects of inertia are very little and the gimbal model is simplified. The settings of the gimbal control system have also been investigated using different methods.

Reference [11] has investigated a proxy-based slide mode applied to a two-axis gimbal system. The controller based on the sliding mode of the two-axis gimbal, with the assumption of the axis speed of the same height and the same direction, has been presented in [12]. New design methods such as linear quadratic regulator (LQR), and linear Gaussian quadratic with loop transfer recovery control (LQG/LTR) have been used to improve the stabilization of broadband controllers [13]. This method has been used to stabilize the line of sight of vehicles [12]. A quadratic Gaussian (LQG) algorithm for simultaneous estimation and compensation of a specific type of disturbance is also presented in [14]. Types of PID controllers such as PI controller, PD, PD_PI, etc. have been investigated in [2].

To improve the performance of the gimbal controller, the fuzzy method is used in [15]. Also, in [16], the fuzzy controller has been used in the two-axis gimbal.

Optimization algorithms can work more effectively in comparison with classical methods. Due to their flexibility, they can be used in the design of controllers. Optimization methods such as genetic algorithm (GA) [17], ant colony optimization (ACO) [18], bacterial search optimization (BFO) [19], and particle swarm optimization (PSO) [20, 21] are widely used. It has been used to optimize the controllers. So far, the genetic algorithm has been used to optimize the PID controller coefficients of a simple system [22]. Also, in [23], the PSO algorithm was used to optimize the PID controller coefficients of a simple system.

This article aims to optimize the coefficients of the stabilizer controller of the gimbal system based on the PSO algorithm. The gimbal stabilizer is responsible for minimizing the angular speed of the gimbal axes after turning. For this purpose, the control system is designed to bring the angular speed of the axes to the minimum possible value (zero) after applying the rotations. Interfering torques should also be minimized. Additionally, the system's performance must remain robust under varying external disturbances and dynamic conditions to ensure stability and accuracy.

Another goal is to compare with other gimbal stabilizer control methods. This article has investigated the single-axis gimbal stabilizer system and its controller coefficients have been optimized with the aim of the lowest settling time and the lowest possible overshoot. To implement the control system, researchers try to use many different modern methods to control normal servo systems, but the PID controller and its variants are still the most used due to their simple structure, low cost, and high performance [24]. But the PID controller, when faced with nonlinear disturbances or an uncertain situation, such as what exists in the movement of drones, must be able to be adjusted while moving. In various applications, equivalent types of PID controllers, such as the PD_PI controller, are used due to greater simplicity in linearizing the controller coefficients. Fuzzy methods have also been used to improve control performance [15].

In this article, with the combination of control methods, the PD_PI fuzzy controller is used and the controller coefficients are optimized by the PSO algorithm.

2 Examining Governing Relationships and Research Methods

The relationships governing the gimbal torque are investigated considering the imbalance of the dynamic mass and the angular movement of the base. Regarding rotational motion, Newton's first law states that by applying force on the body (assuming an inertial frame), the body gains speed. In addition to this, using Newton's second law, it can be proved that if the torque force is applied to a solid object with inertia J and angle α, the torque applied is obtained according to equation (1).

$$\tau = J.\alpha \qquad Eq\ 1$$

Here, the purpose of designing the control system is to stabilize and stabilize the line of sight. To place the system stably along the line of sight after rotation and in a new position. The gyroscope measures the angular velocity of the gimbal body rotation. After calculating the rotation angle, the gimbal applies the necessary angular velocity to the stabilizer ring. The servo motor rotates in the opposite direction to the rotation of the body as necessary and keeps the position of the gimbal stable. Equation (2) shows the electrical equivalent of the motor.

$$u(t) = K_e\,\omega + Ri(t) + L\frac{di(t)}{dt} \qquad \text{Eq 2}$$

After applying the rotation, a control system is needed in order for the line of sight to be in the best possible position. In order to control stabilization, the required parameters should be obtained according to the structure of the engine and then it should be modeled [15]. Figure (1) shows the control system and components of the gimbal stabilizer ring.

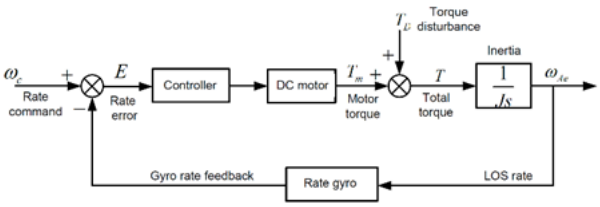


Figure 1: gimbal stabilizing ring [2]

As seen in Figure (1), the stabilizer ring includes the controller ring, DC motor, and gyroscope. ω_c is the input angular velocity and ω_{Ae} is the output angular velocity of the ring. The control loop works as a stabilizing system when ω_c and ω_{Ae} are zero

while checking the input system. Servo control can generally be divided into two basic types; The first type works with the input measurement and the system applies the actual movement that has been calculated. The second type is general servo control, which eliminates system disturbances. These disturbances can include torque disturbances to the error of incorrect parameters used.

To study the gimbal movement in one axis, two frames are considered; In Figure (2), the P frame connected to the gimbal body before moving is shown with vectors (i, j, k), and the A frame connected to the gimbal body after moving in stable mode is shown with vectors (r, e, d). The center of rotation is the same in both frames and is located at the middle point of the system. This ensures consistency in analyzing rotational dynamics.

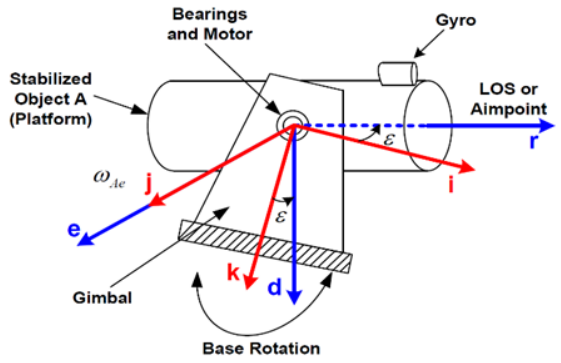


Figure 2: Single axis gimbal system [2]

The rotation takes place around the j-axis. Also, the transition between frame P to A has happened with a positive angle ε (the gimbal angle) that is made around the axis e. The frame transfer matrix from frame P to A is shown as ${}^{A}_{P}C$ in equation (3):

$$ {}^{A}_{P}C = \begin{bmatrix} \cos\varepsilon & 0 & -\sin\varepsilon \\ 0 & 1 & 0 \\ \sin\varepsilon & 0 & \cos\varepsilon \end{bmatrix} \quad \text{Eq 3} $$

Also, for each vector in the frames, angular velocity and inertia matrix are considered as follows:

$$ {}^{P}\vec{\omega}_{P/I} = \begin{bmatrix} \omega_{Pi} \\ \omega_{Pj} \\ \omega_{Pk} \end{bmatrix} \quad \text{Eq 4} $$

$$ {}^{A}\vec{\omega}_{A/I} = \begin{bmatrix} \omega_{Ar} \\ \omega_{Ae} \\ \omega_{Ad} \end{bmatrix} \quad \text{Eq 5} $$

So that ${}^{P}\vec{\omega}_{P/I}$ the angular velocity matrix of frame P includes the angular velocity of vectors (i,j,k) with values of ω_{Pi}, ω_{Pj}, and ω_{Pk}. Also ${}^{A}\vec{\omega}_{A/I}$ is the angular velocity matrix of frame A, including the angular velocity of vectors (r,e,d) with the values of ω_{Ai}, ω_{Aj}, and ω_{Ak}. Equation (6) shows the obtained inertia matrix ${}^{A}J$:

$$ {}^{A}J = \begin{bmatrix} A_r & A_{re} & A_{rd} \\ A_{er} & A_e & A_{ed} \\ A_{dr} & A_{de} & A_d \end{bmatrix} \quad \text{Eq 6} $$

In the above matrix, A_r, A_e and A_d are the moment of inertia around the axis (r,e,d) respectively. Also, A_{re}, A_{er}, A_{rd}, A_{dr}, A_{ed} and A_{de} are the moment of inertia resulting from the inertial motion of the vectors. It is clear that in one movement, the values of A_{re}, A_{er}, A_{rd}, A_{dr} are equal to each other and the values of A_{ed} and A_{de} are also equal to each other.

The output, ω_{Ae}, will be used for the control loop; Because the goal is that after the movement of the gimbal, no more torque or movement is applied to the object and the system remains stable and in a fixed state. Therefore, ω_{Ae} should be as close to zero as possible. To remove the noise caused by disturbing torques, ω_{Ae} can be measured using a gyroscope mounted on the gimbal.

Equations (7), (8) and (9) show the calculation of ${}^{A}\vec{\omega}_{A/I}$ for the transformation matrix ${}^{A}_{P}C$:

$$ \omega_{Ar} = \omega_{Pi}\cos\varepsilon - \omega_{Pk}\sin\varepsilon \quad \text{Eq 7} $$

$$ \omega_{Ae} = \omega_{Pj} + \dot{\varepsilon} \quad \text{Eq 8} $$

$$ \omega_{Ad} = \omega_{Pi}\sin\varepsilon + \omega_{Pk}\cos\varepsilon \quad \text{Eq 9} $$

Then the entered torque is calculated. According to Newton's second law and equation (1), equation (11) has been calculated:

$$ F = ma \quad \text{Eq 10} $$

$$ {}^{A}\vec{H} = {}^{A}J.\,{}^{A}\vec{\omega}_{A/I} \quad \text{Eq 11} $$

As a result, the angular momentum matrix ${}^{A}\vec{H}$ has been calculated as equation (12):

$$ {}^{A}\vec{H} = \begin{bmatrix} A_r\omega_{Ar} + A_{re}\omega_{Ae} + A_{rd}\omega_{Ae} \\ A_{re}\omega_{Ar} + A_e\omega_{Ae} + A_{de}\omega_{Ae} \\ A_{rd}\omega_{Ar} + A_{de}\omega_{Ae} + A_d\omega_{Ad} \end{bmatrix} = \begin{bmatrix} H_r \\ H_e \\ H_d \end{bmatrix} \quad \text{Eq 12} $$

Having the momentum matrix, the input torque is obtained:

$$ \vec{T} = \frac{d}{dt}\left({}^{A}\vec{H}\right) + {}^{A}\vec{\omega}_{A/I} \times {}^{A}\vec{H} \quad \text{Eq 13} $$

$$ \vec{T} = \begin{bmatrix} \dot{H}_r + \omega_{Ae}H_d - \omega_{AdH_e} \\ \dot{H}_e + \omega_{Ad}H_r - \omega_{Ar}H_d \\ \dot{H}_d + \omega_{Ar}H_e - \omega_{Ae}H_r \end{bmatrix} \quad \text{Eq 14} $$

As a result, equation (15) shows the torque (created on the motor shaft) on the e-axis:

$$ T_m = \dot{H}_e + \omega_{Ad}H_r - \omega_{Ar}H_d \quad \text{Eq 15} $$

The disturbance torques on the system (T_D) are shown in equation (16):

$$ \begin{aligned} T_D = {} & (A_d - A_r)\omega_{Ar}\omega_{Ad} + A_{rd} \\ & + (\omega_{Ar}{}^2 - \omega_{Ad}{}^2) \\ & - A_{de}(\dot{\omega}_{Ad} - \omega_{Ae}\omega_{Ar}) \\ & - A_{re}(\dot{\omega}_{Ar} - \omega_{Ae}\omega_{Ad}) \end{aligned} \quad \text{Eq 16} $$

The body of the gimbal represents the motor load that is connected to the motor axis. This body, which is modeled based on its moment of mobility, depends on its dimensions and position with respect to the axis of rotation. Here, a model is proposed to show the body. For the proposed body, mass M=1Kg, $J_l = 9.8 \times 10^{-3} kgm^2$and radius r=14cm are considered. Here it is assumed that $\omega = 2\,rad/sec$ in 35 msec. Therefore, the engine must have conditions $\alpha = 56\,rad/sec^2$. The torque applied to the motor is shown in equation (18):

$$ J_L = \frac{1}{2}Mr^2 = 9.8 \times 10^{-3} Kg.m^2 \quad \text{Eq 17} $$

$$ T = J.\alpha = 9.8 \times 10^{-3} \times 56 = 5.5Nm \quad \text{Eq 18} $$

$$ P_{out} = T.\omega = 5.5 \times 2 = 11W \quad \text{Eq 19} $$

This system can be considered as a system (MISO) with two inputs (voltage applied to the motor and output torque) and one output or a system (SISO) with one input (body angular velocity) and one output [2]. The choice between MISO and SISO modeling depends on the level of complexity and control precision required for the application.

As discussed in the physical analysis, ω_{Ae} is considered as the output of the system. ω_{Ae} is the amount of angular velocity after movement, caused by applying the input angular velocity. Therefore, ω_C is considered as the input angular velocity and ω_{Ae} as the output angular velocity. To apply torque or rotation to the system, a servo motor is needed. It is also necessary to ensure the output status. As a result, a closed loop system including gyroscope is needed in these conditions. To model the engine, first the required parameters were calculated according to the structure of the engine and then it was modeled. Figure (3) shows the electrical equivalent of a DC motor:

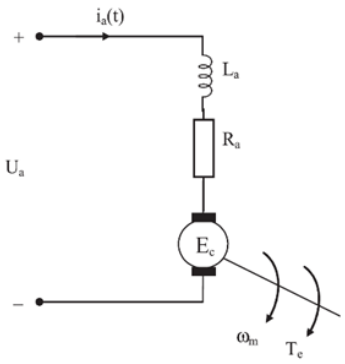


Figure 3: Electrical circuit representation of the motor

$$u(t) = K_e\omega_m + R_a i(t) + L_a \frac{di(t)}{dt} \quad \text{Eq 20}$$

$$T_m(t) = K_{TM} i(t) \quad \text{Eq 21}$$

$$T_m(t) = J_m^* \dot{\omega_m} + a_m^* \omega_m + T_D \quad \text{Eq 22}$$

In these equations, u(t) is the time function of motor terminal voltage and i(t) is the time function of motor terminal current. m is the mass of the object, R is the input resistance of the terminal, L is the input inductance of the terminal, K_e is the return electric force of the motor and K_{TM} is the torque constant. $K_e\omega_m$ shows the position of the motor and T(t) shows the torque entered. The physical effects of the object are also considered. The forces are calculated based on Newton's second law (equation (10). The acceleration equation (23) is created, which shows:

$$\dot{\omega_m} = \frac{-\frac{K_e K_{TM}}{R_a}\omega_m + \frac{K_{TM}}{R_a}u(t) - a_m^*\omega_m - T_D}{J_m^*} \quad \text{Eq 23}$$

A control system for the designed motor has been proposed, whose input is the voltage function u(t) and its output is the motor movement with angular speed ω_m and torque T_m [2]. Figure (4) shows the servo motor control system. In this model, additional moments caused by inertia are also considered in the model.

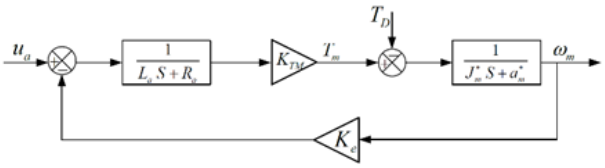


Figure 4: Servo motor control diagram [16]

Table 1: Engine specifications [16]

parameter	value
u_a	27 V
(no load speed) ω_{nl}	303 RPM
R_a	45 Ω
L_a	0.003 H
K_{TM}	0.85 Nm/A
K_e	0.85 V/(rad/sec)
J_m	0.0017 kg/m^2

In this ring, $J_m{}^*$ is the inertia of the rotor and $a_m{}^*$is the damping ratio of the motor which is calculated in equations (24) and (25).

$$J_m{}^* = J_m + J_l \quad \text{Eq 24}$$

$$a_m{}^* = a_m + a_l \quad \text{Eq 25}$$

where J_m is the inertia of the motor, J_L is the inertia of the load of the structure, a_m is the damping effect of the motor and a_l is the damping effect of the load of the structure.

In examining the obtained physical model, first, if the system is motionless, the values of ω_{Pi}, ω_{Pj}, and ω_{Pk} will be zero. As a result, equation (26) shows the transformation function of the system regardless of T_D.

$$G_m(s) = \frac{\omega_m}{u_a(s)} = \frac{K_{TM}}{(L_a s + R_a).(J_m^* + a_m^*) + K_e K_{TM}} \quad \text{Eq 26}$$

Here, the NORTHROP GRUMMAN engine is used and the conversion function is calculated with the given engine specifications [25]. In table (1), the specifications of this engine are given. Equation (27) shows the conversion function of the engine system by replacing its specifications:

$$a_m^* = 0$$

$$G_m(s) = \frac{\omega_m}{u_a(s)} = \frac{24637.68}{s^2 + 1500s + 20942} \quad \textit{Eq 27}$$

Also, in order to detect vibrations and angular velocity, to calculate and apply the amount of torque required to stabilize the system, a gyroscope sensor is needed. Gyroscope is used in feedback to sample the output and apply it to the input of the system. Figure (5) shows the conversion function and gyroscope control system:

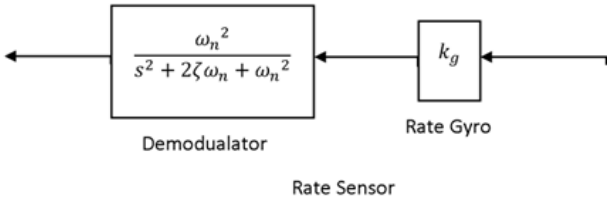


Figure 5: Gyroscope control diagram

A 475T gyroscope from US Dynamics company is used here. The conversion function of the gyroscope is calculated

according to the characteristics of the engine and the gyroscope. In table (2), the specifications of this gyroscope are given [26].

Table 2: Specifications of the gyroscope [16]

characteristic	amount
input ratio (interval)	±40 to ±100 °sec
output	AC or DC
scale	Buyer's desired specifications
natural frequency	20 to 140 Hz
ratio	0.4 to 1

To calculate the conversion function of the gyroscope, the damping ratio is ξ=0.7, the natural frequency of the system f_n=50 Hz, and the gyroscope coefficient is considered to be K_n=1. According to the specifications of the gyroscope, its conversion function is calculated in equation (28) [27].

$$G_{Gyro} = \frac{\omega_n^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)} = \frac{2500}{(s^2 + 70S + 2500)} \quad \text{Eq 28}$$

The investigations reveal that the PD_PI controller outperforms others with improved performance and shorter settling time. The proposed PSO based PD_PI fuzzy controller enhances the PD_PI controller by incorporating fuzzification methods. To further minimize settling time and system overshoot, the controller's coefficients have been optimized using the PSO algorithm.

In the examination of the physical model, it was shown that the most influential variable in the single-axis gimbal system is ω_{Pj}. Whenever ω_{Pj} increases, the response of the system increases. According to the governing mathematical relations obtained from the study of physical behavior and the use of the motor and gyroscope control system, the control loop of the gimbal stabilizer is shown in figure (8).

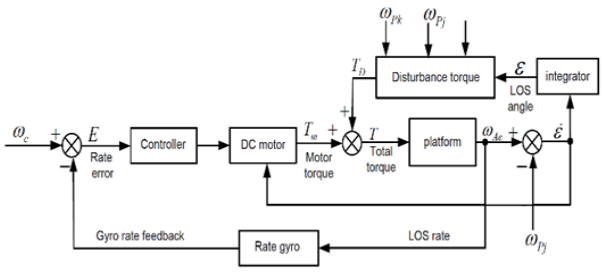


Figure 6 : Gimbal stabilizer control ring

The PID controller is one of the most widely used controllers due to its optimal, reliable and simple performance. For the accurate performance of the PID controller, its coefficients must be calculated and adjusted correctly. For this purpose, there are various methods, the most well-known and simplest of which is the Ziegler-Nichols (Z-N) method. This method can be considered as a traditional method based on modeling and control. Ziegler-Nichols regulation rules for use in closed loop systems are associated with a significant reduction of disturbances. The design criterion of this method is based on the damping ratio of the amplitude in three periodic periods [28]. The output of the PID controller is in the form of equation (29):

$$y(t) = K_P e + \frac{K_I}{s} e + K_D es \quad \text{Eq 29}$$

In this equation, e is the error function and K_D, K_I and K_P are the PID controller coefficients. The classical structure of PID controller can be transformed into different forms by using methods such as cascade control. These equivalent forms can be more appropriate in practice. The reason for using PD_PI is some forms of PID controller, PI, PD and PD_PI controllers. The application of the PD_PI controller is simpler and the linearization of the coefficients in it is easier.

In this equation, e represents the error function, while K_D, K_I and K_P are the coefficients of the PID controller. The classical PID controller structure can be modified into alternative forms, such as cascade control, to better suit practical applications. Among these variations, including PI, PD, and PD_PI controllers, the PD_PI controller stands out for its simplicity and ease of coefficient linearization, making it a more practical choice in many cases.

Also, so far, the fuzzy method has been used to reduce overshoot [26]. Fuzzy controller includes fuzzification, fuzzy table (knowledge table), fuzzy rules and difuzzification [26].

In this article, PD_PI fuzzy controller is used for gimbal stabilization (e-axis stabilization). Figure (7) shows the PD-PI phase controller.

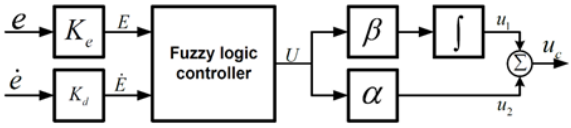
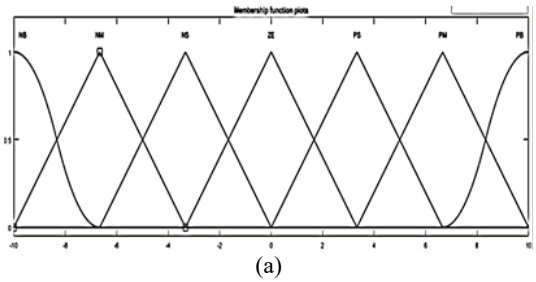
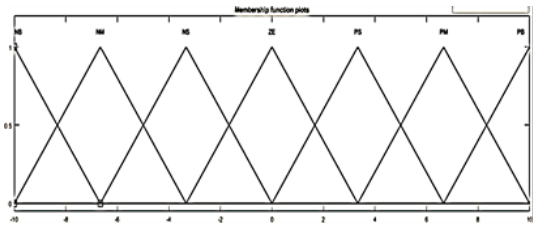


Figure 7 : PD-PI phase controller

In figure (7), the input function is (E, E˙) and the output function is (U). Also, the input and output membership functions for fuzzy variables are shown in Figure (8).



(a)



(b)

Figure 8: Input (a) and output (b) membership functions for fuzzy variables

The rules of PD_PI fuzzy controller are shown in table (3). The used membership functions, NL, NM, NS, ZR, PS, PM and PL

represent large negative, medium negative, small negative, zero, small positive, medium positive and large positive respectively.

Membership functions are defined in the closed interval [-1,1]. Rules are defined based on performance; when the system output error is far from the intended output; That is, e = PL and e˙ = ZR, the output value of U is chosen equal to the value of PL to reduce the system error and make the system stable. This approach ensures that the system responds effectively to large deviations, improving overall stability and accuracy.

Table 3: Fuzzy controller rules PD_PI

e/e’	NL	NM	NS	ZR	PS	PM	PL
NL	LN	LN	LN	LN	MN	SN	ZE
NM	LN	LN	LN	MN	SN	ZE	SP
NS	LN	LN	MN	SN	ZE	SP	MP
ZR	LN	MN	SN	ZE	SP	MP	LP
PS	MN	SN	ZE	SP	MP	LP	LP
PM	SN	ZE	SP	MP	LP	LP	LP
PL	ZE	SP	MP	LP	LP	LP	LP

Also, for example, if e = ZR and e˙ = NM, the value of U = NM. When both e and e˙ are zero (which is the desired state and the system does not need a control input), the output U is selected equal to the value of ZR.

In this article, the PSO optimization algorithm is used to calculate the PD_PI fuzzy controller coefficients. The coefficients have been optimized by the PSO algorithm for the lowest settling time and the lowest possible overshoot.

Then the fuzzy PD_PI coefficients are adjusted with PSO. The PSO algorithm is one of the most widely used optimization algorithms due to its simple structure and suitable speed. This method is designed based on the collective behavior of particles, such as birds gathering to find food. The PSO algorithm works by having an initial population and possible responses.

The movement of the particles towards their best-known position in the search space and also the best position in the entire categories. When better positions are discovered, particle routing is done. This process is repeated and finally the desired answer to the problem will be discovered. This method places a set of particles (probable answers) in the multidimensional search space by randomly selecting the speed and position. The random position of the particles is considered as the best position to start, and then the speed of the particles is updated based on the experience of the particles (by applying the cost function) of the other population [29].

Also, the cost function is a function that values the particles relative to the optimal point. This function is defined for optimization, the lowest settling time and the lowest possible overshoot. The minimum value of this function is the optimization goal. Each value for the controlling coefficients represents a particle. The best position of the particle among all visited positions (cost function applied to it) is the best position of the group (P_{best}). The best position of the particle among all groups of particles is the best overall position (G_{best}). Figure (9) shows the PSO algorithm. In this algorithm, n is the number of particles in the group and i=1,2,3.....n is the particle number. After the first movement of the particles, in the first iteration, a new velocity and position is considered for each particle. The new speed is used to determine the speed and direction of the particle.

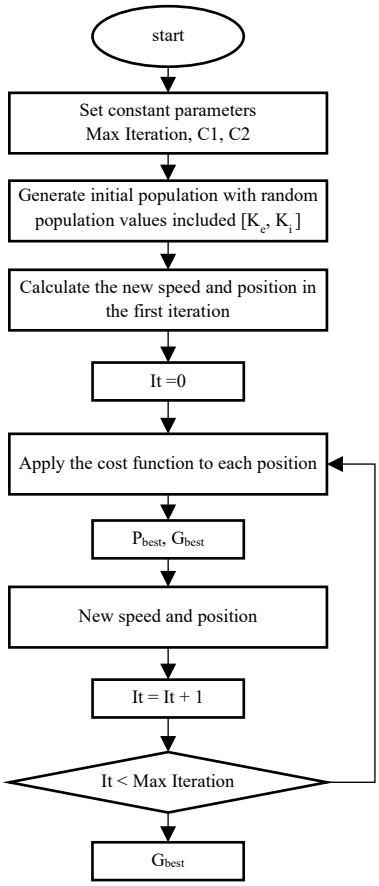


Figure 9: PSO algorithm

The speed is calculated by equation (30).

$$V_{i,m}{}^{t+1} = W * V_{i,m}{}^{t+1} + C1 * rand * \left(Pbest_{i,m} - X_{i,m}^{t}\right) + C2 * rand * \left(Gbest_{m} - X_{i,m}^{t}\right) \quad \text{Eq 30}$$

In this equation, m = 1,2,3.....d where d is the number of parameters, t is the number of each iteration, $V_{i,m}{}^{t}$ is the speed of each particle in that iteration round, W is the weighting coefficient , C_1 and C_2 are constant numbers to accelerate (reach the answer) of the algorithm, rand is a random number between zero and one, P_{best}i is the best previous position of the particle in the group and G_{best} is the best position in the whole group of particles. After each iteration, all particles move to the previous best position and find their own best positions. Equation (31) shows the new position of the particle after moving.

$$X_{i,m}{}^{t+1} = X_{i,m}^{t} + V_{i,m}{}^{t+1} \quad \text{Eq 31}$$

In this equation, $X_{i,d}{}^{t}$ shows the current position of the particle in that repetition period. Each particle contains the coefficients k_e, k_I, etc. in the form $[k_P, k_I, k_D, k_e]$. The constant values of the PSO algorithm are considered as follows. 100 is considered for population and maximum repetition. The values of C_1 and

C_2 are 1.5. Because the range of coefficients is not clear, the range of particle values is not limited.

To improve the speed of finding the optimal coefficients, the weighting coefficient (W) is calculated for each particle in the population during each iteration. This coefficient, which is not constant within a single iteration, helps the algorithm discover the optimal coefficients in fewer iterations. Initially, a high weighting coefficient causes the particles to spread out across a wider search space, allowing the algorithm to explore a larger area. In the later iterations, the weighting coefficient is decreased to prevent excessive scattering of particles, enabling the algorithm to focus more precisely around the optimal solution.

The highest weighting coefficient is considered for the first population particle and the lowest weighting coefficient is considered for the last population particle. Equation (32) shows the formula for calculating the weighting coefficient for each particle in the repetition round in which it is located.

$$W = \left(\left(\frac{-0.8 * i}{pop}\right) + 0.9\right) \quad \text{Eq 32}$$

Pop shows the number of particles (population, considered equal to 100 particles in this article). To optimize the coefficients in this article, first, the angular velocity ω_{Pj} is considered zero. So that the online controller can control the conditions for different angles; First, four coefficients $[k_P, k_I, k_D, k_e]$ are considered. The initial population is considered to be 1000. The algorithm obtains the controlling coefficients. The obtained k_P and k_D coefficients remain constant and the population is set to 100. Two controller parameters k_P and k_I should be optimized for each angle ω_{Pj} in the interval [10,0]. Their desired value is searched and calculated in the two-dimensional search space. Therefore, in the two-dimensional search, the population with the size of 100 is randomly assigned. Also $[x_{i,1}, x_{i,2}]$ and $[V_{i,1}, V_{i,2}]$ are considered as initial position and speed.

The highest weighting coefficient is considered for the first population particle and the lowest weighting coefficient is considered for the last population particle. Equation (32) shows the formula for calculating the weighting coefficient for each particle in the repetition round in which it is located.

The way of working at this stage is that first, the initial position is considered randomly for the initial population; Then the initial speed is calculated for the initial performance of each point. A point with the minimum value of the cost function is calculated as P_{best} (the best position in the group) and G_{best} is the best position in the group of particles (the lowest value of the cost function).

In this article, the cost function is a function of the steady state error, the maximum overshoot in percentage, the rising time and the settling time of the system. The contribution of these partial functions in the main performance of the cost function is determined by a scale factor that depends on the choice of the designer.

Equation (33), the cost function is considered, which shows the lowest overshoot and the lowest settling time [30].

$$F = \big(1 - exp(-\beta)\big)(MP + E_{SS}) + \big(exp(-\beta)\big)(T_S - T_r) \quad \text{Eq 33}$$

In this equation, F is the output of the cost function, MP is the maximum jump in percentage, T_S is the settling time, T_r is the rising time, and β is the scale factor that depends on the designer's conditions. In this article, the value of β = 0.01 is considered. The PSO algorithm can be repeated with a thousand iterations (or even more). The final optimal point is G_{best}. Figure (10) shows the block diagram of PD_PI fuzzy controller coefficients adjustment based on the PSO algorithm. Figure (11) shows the optimization process of coefficients.

The cost function is simulated in Matlab software. The controller coefficients (k_P, k_I, k_D, k_e) are the input values and the calculated value of equation (33) is the output of the cost function. This cost function has been used to evaluate the system performance with the input parameters k_e and k_I.

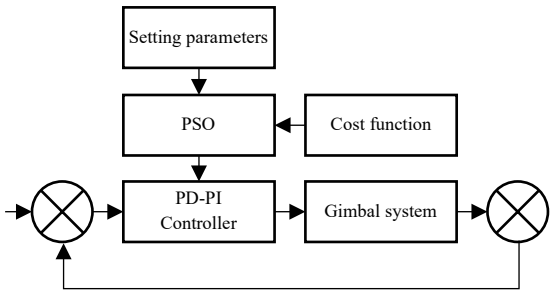


Figure 10: PD_PI fuzzy controller based on PSO

3 Results

Before the simulation, the simulation hypotheses are considered, so that the objectives are determined.

It is concluded from equation (16) that if the base is motionless, the disturbance torque is zero ($\omega_{Pi}=\omega_{Pj}=\omega_{Pk}=0$).

The features of the gimbal mechanism according to Figure (2) show that the angle ω_{Pj} is the most effective parameter in the disturbance torque T_D. Therefore, this parameter is used in the setting and testing function.

The total inertia of the motor is equal to $J_m + J_l = 11.5 * 10^{-3}(kg.m^2)$.

To consider the effect of the angular movement of the base, the amount of $\dot{\varepsilon}$ in equation (8) should be given to the feedback motor through the electric constant of the motor (EMF constant).

In the simulation tests, the following values are considered. The entry angle is ω_c=10 degrees per second; While ω_{Pj} changes from zero to 10 degrees per second.

The input coefficient k_e affects the integral and proportional coefficients. The adjustment parameters are controlled based on the design process and are sometimes adjusted through testing. To dynamically control the response of the system in the test angle range of 0 to 10 degrees, two parameters k_e and k_i are adjusted online. Other parameters are fixed. These parameters are obtained using the PSO algorithm and used in

the system. Equations (34) and (35) show the function of variable coefficients k_e and k_i:

$$ki = -0.007233 * {\omega_{Pj}}^2 - 0.06103 * \omega_{Pj} + 16.63 \quad \text{Eq 34}$$

$$ke = -0.001433{\omega_{Pj}}^2 - 0.07613 * \omega_{Pj} + 0.9167 \quad \text{Eq 35}$$

The constant coefficient values are given in table (4):

Table 4: Fixed coefficients of the controller

parameter	k_p	K_d
amount	0.2670	0.0559

Based on what was done, the complete simulator model of the servo control system has been simulated in Matlab software; the Servo control system is shown in figure (8).

First, the working principles of the gimbal system, which depend on Newton's second law, have been explained. Then the control system is built using the gimbal model obtained from equations (15) and (16). It can be shown precisely how the closed-loop control system creates a torque in the motor that is equal to the disturbance torque. Therefore, the rotation of the object is prevented.

The comparison of the time response of the servo control system for normal PI and fuzzy controllers and the proposed controller with optimal coefficients has been done. The optimal performance of the proposed servo system includes the least settling time and the least overshoot possible. Overshoot within the error range of 2%, permanent error is zero and its settling time is as low as possible.

Naturally, the required performance criteria are defined according to the application in which the gimbal system is used. The mentioned criteria indicate the conditions that are generally observed in each control system.

To evaluate the efficiency of the proposed controller, the responses of the system have been calculated for different values of the base angular velocity ω_{Pj}. Some of the responses of the system are shown in figures (11), (12), (13), (14) and (15). The efficiency of the PSO-based PD_PI controller can be seen in comparison with the normal PI.

All three controllers achieve the desired rise time without steady-state error, with results that are nearly identical. To validate the performance of the proposed controller, a comparison is made using the overshoot percentage and settling time parameters, as shown in Tables (5) and (6). The overshoot percentage is a direct indicator of the gimbal system's stability, while the settling time reflects the time required for the system's response to remain within a specified range (two percent) of the final value. This range is highlighted by two red lines in the figures.

The results show how increasing the angular speed has a negative effect on the performance of the gimbal system. This increase causes more problems and slows down the response when using a normal PI controller. Fuzzy controller has less overshoot and settling time problems. But the proposed controller has the problem of overshooting and very little settling time. Although the fuzzy controller has been able to reduce the overshoot percentage; But it does not have a favorable sitting time. As can be seen, the PD_PI fuzzy controller based on the proposed PSO, in addition to the significant increase in speed, has also reduced the overshoot to the lowest value (in the range of two percent which is the acceptable permanent error range).

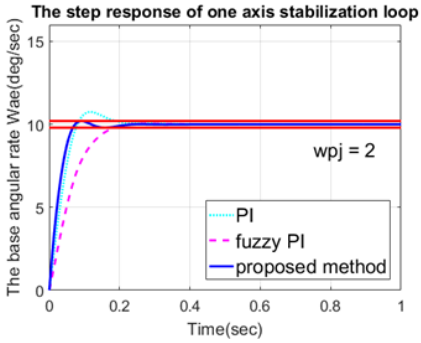


Figure 11: Step response per input $\omega_{Pj} = 2$ deg/sec

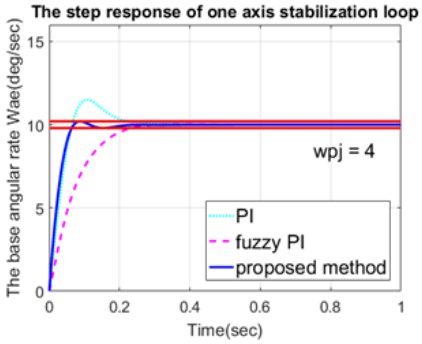


Figure 12: Step answer per input $\omega_{Pj} = 4$ deg/sec

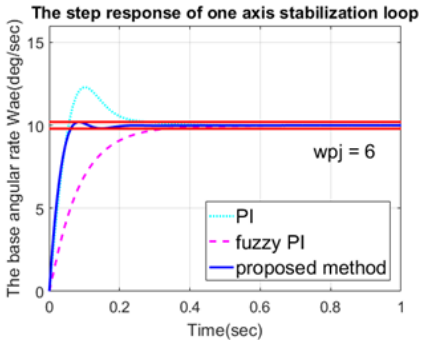


Figure 13: Step answer per input $\omega_{Pj} = 6$ deg/sec

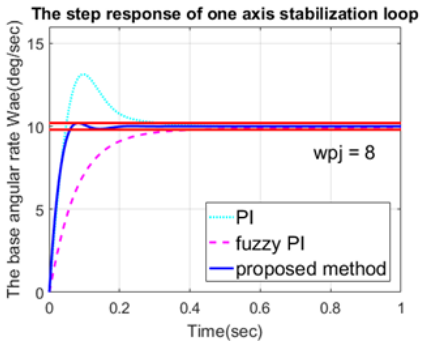


Figure 14: Step answer per input $\omega_{Pj} = 8$ deg/sec

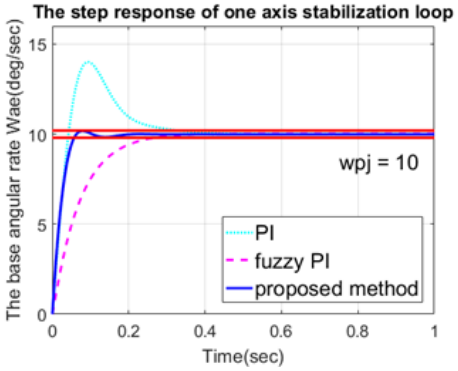


Figure 15: Step answer per input $\omega_{Pj} = 10$ deg/sec

Table 5: Overshoot percentage of each controller

PSO PD_PI	PSO PI	Fuzzy PID	PI	ω_{Pj} (deg/sec)
2.000	2.374	0.011	1.037	0
2.002	1.890	0.411	7.469	2
2.006	1.166	0.275	14.91	4
2.001	0.847	0.000	22.948	6
2.005	0.959	0.000	31.363	8
1.786	1.516	0.292	40.056	10

Table 6: The sitting time of each controller

PSO PD_PI	PSO PI	Fuzzy PID	PI	ω_{Pj} (deg/sec)
0.069	0.083	0.146	0.099	0
0.066	0.082	0.179	0.180	2
0.063	0.081	0.226	0.217	4
0.060	0.080	0.313	0.258	6
0.058	0.078	0.368	0.290	8
0.056	0.074	0.455	0.310	10

In addition to the advantage of high speed and zero overshoot, PD_PI phase controller based on PSO can be used for test angles (ω_{Pj}) greater than ten (up to ω_{Pj}=14) degrees/second. The values of its online controller parameters for this range are considered as equations (36) and (37).

$$ki = -0.05749 * {\omega_{Pj}}^2 + 1.242 * \omega_{Pj} + 8.795 \quad \text{Eq 36}$$

$$ke = -0.002495 * \omega_{Pj}^2 - 0.05776 * \omega_{Pj} + 0.825 \quad \text{Eq 37}$$

These coefficients are also calculated with the help of PSO algorithm. The result is shown in figure (16).

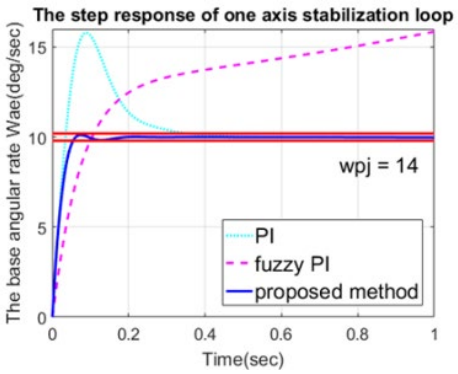


Figure 16: Step answer per input $\omega_{Pj} = 14$ deg/sec

This controller has shown very good stability in this range, which is a great advantage over other controllers.

4 Conclusion

In this article, a single-axis gimbal system has been introduced and its mathematical model has been investigated using Newton's laws with respect to the base angle and balance.
Then, the stabilizing loop is built and the PD_PI fuzzy controller coefficients are optimized with the help of PSO algorithm and then adjusted online. The proposed controller can be easily configured. The control system is simulated using Matlab Simulink.
A comparative study was conducted to test the performance of the proposed controller. The obtained results have proven the performance of the controller well. The proposed controller can maintain the change of its stability conditions and show a quick response with the least overshoot compared to other controllers and especially the regular PI controller. Also, the proposed controller has better stability for an angular speed greater than the test range. Another important advantage of this controller is the use of lower processing resources than the usual fuzzy controller in addition to a favorable response.

Disclosure of Potential Conflicts of Interest

The Authors declare that there is no conflict of interest.

Reference

[1] M. K. MASTEN, "Inertially stabilized platforms for optical imaging systems," in IEEE Control Systems Magazine, vol. 28, no. 1, pp. 47-64, Feb. 2008.
[2] Havangi, R. and S.H. Khatami, Simultaneous identification and tracking of objects using deep learning. Journal of Machine Vision and Image Processing, 2023. 10(3): p. 79-92
[3] Sasaki, Takahiro, et al. “Robust Attitude Control Using a Double-Gimbal Variable-Speed Control Moment Gyroscope.” Journal of Spacecraft and Rockets, vol. 55, no. 5, Sept. 2018, pp. 1235–47.
[4] Jayachitra, A., and R. Vinodha. “Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor.” Advances in Artificial Intelligence, vol. 2014, Dec. 2014, pp. 1–8.
[5] Ravindra, S., 2008: “Modeling and Simulation of the Dynamics of a Large Size Stabilized Gimbal Platform Assembly”. Asian International Journal of Science and Technology in Production and Manufacturing, Vol. 1, pp. 111-119.
[6] Rue, A. K., 1974: “precision stabilization systems”. IEEE Trans. Aerospace and Electronic Systems. AES-10, pp. 34-42.
[7] Ekstrand, B., 2001: “Equation of Motion for a Two Axes Gimbal System”. IEEE Trans. On Aerospace and Electronic Systems, Vol. 37, pp. 1083-1091.
[8] Daniel, R., 2008: “Mass properties factors in achieving stable imagery from a gimbal mounted camera”. Published in SPIE Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications V. 6946.

[9] Özgür, H., Aydan, E., and İsmet E., 2011: "Proxy-Based Sliding Mode Stabilization of a Two-Axis Gimbaled Platform". Proceedings of the World Congress on Engineering and Computer Science, San Francisco, USA (WCECS), I.

[10] Ravindra, S., 2008: "Modeling and Simulation of the Dynamics of a Large Size Stabilized Gimbal Platform Assembly". Asian International Journal of Science and Technology in Production and Manufacturing, Vol. 1, pp. 111-119.

[11] Khodadadi, H., 2011: "Robust control and modeling a 2-DOF Inertial Stabilized Platform". International Conference on Electrical, Control and Computer Engineering, Pahang, Malaysia.

[12] Smith, B. J., Schrenck, W. J., Gass, W. B, and Shtessel, Y. B., 1999: "Sliding mode control in a twoaxis gimbal system". in Proc. IEEE Aerospace Applicat. Conf, Vol. 5, pp. 457–470.

[13] Willian, B., and Steven, P. T., 1989: "Optimal motion stabilization control of an electrooptical sight system". Proc. SPIE Conference, Vol. 1111, pp. 116–120.

[14] Hullender, L. Bo, and DeRenzo, D., M., 1998: "Nonlinear induced disturbance rejection in inertial stabilization systems". IEEE Trans, Vol. 6, pp. 421–427.

[15] Abdo, Maher Vali, Ahmad Toloei, Alireza. (2013). Fuzzy Stabilization Loop of One Axis Gimbal System. International Journal of Computer Applications. 77. 6-13. 10. 5120/13372-0975.

[16] Maher Mahmoud Abdo, Ahmad Reza Vali, Ali Reza Toloei, Mohammad Reza Arvan. (2014). Stabilization loop of a two axes gimbal system using self-tuning PID type fuzzy controller. ISA Transactions, Volume 53, Issue 2, Pages 591-602.

[17] S. M. G. Kumar, R. Jain, N. Anantharaman, V. Dharmalingam, and K. M. M. S. Begam, "Genetic algorithm based PID controller tuning for a model bioreactor," Indian Institute of Chemical Engineers, vol. 50, no. 3, pp. 214–226, 2008.

[18] I. Chiha, N. Liouane, and P. Borne, "Tuning PID controller using multiobjective ant colony optimization," Applied Computational Intelligence and Soft Computing, vol. 2012, Article ID 536326, 7 pages, 2012.

[19] V. Rajinikanth and K. Latha, "Bacterial foraging optimization algorithm based pid controller tuning for time delayedunstable systems," Mediterranean Journal of Measurement and Control, vol. 7, no. 1, pp. 197–203, 2011.

[20] M. Zamani, M. Karimi-Ghartemani, N. Sadati, and M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization," Control Engineering Practice, vol. 17, no. 12, pp. 1380–1387, 2009.

[21] V. Rajinikanth and K. Latha, "Identification and control of unstable biochemical reactor," International Journal of Chemical Engineering Applications, vol. 1, no. 1, pp. 106–111, 2010.

[22] Jayachitra, A., and R. Vinodha. "Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor." Advances in Artificial Intelligence, vol. 2014, Dec. 2014, pp. 1–8.

[23] Latha, K. & Rajinikanth, Venkatesan & Surekha, P.. (2013). PSO-Based PID Controller Design for a Class of Stable and Unstable Systems. ISRN Artificial Intelligence. 2013. 10.1155/2013/543607.

[24] Lee, T.H. & Huang, Sunan & Tang, K.Z. & Tan, K.K. & Al-Mamun, Abdullah. (2003). PID control incorporating RBF-neural network for servo mechanical systems. Comptes Rendus Mathematique - C R MATH. 2789 - 2793 Vol.3. 10.1109/IECON.2003.1280689.

[25] Fujita, H., and Sasaki, J., 2010: "Torque Control for DC Servo Motor using Adaptive Load Torque Compensation". Proceedings of the 9thWSEAS international conference on System science and simulation in engineering, pp. 454-458.

[26] Lee, H. -P. and Yoo, I. -E., "Robust control design for a two-axis gimbaled stabilization system", in Aerospace Conference, IEEE. (2008), 1-7.

[27] Kurukowa, H., A Geometric Study of Control Moment Gyroscopes, PhD Thesis, University of Tokyo, 1998.

[28] Yucelen, T., Kaymakci, O., Kurtulan, S. (2006). Self-Tuning PID Controller Using Ziegler-Nichols Method for Programmable Logic Controllers. IFAC Proceedings Volumes, 39, 11-16.

[29] Koohi and V. Z. Groza, "Optimizing Particle Swarm Optimization algorithm," 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), Toronto, ON, 2014, pp. 1-5, doi: 10. 1109/CCECE. 2014. 6901057.

[30] Vincent, A. K. and Nersisson, R., "Particle swarm optimization based PID controller tuning for level control of two tank system", in Materials Science and Engineering Conference Series, 2017, vol. 263, no. 5, p. 52001.