



**RESEARCH ARTICLE**

**EVALUATION OF CONTROLLER PARAMETERS ON THE TWIN ROTOR MULTIPLE  
INPUT MULTIPLE OUTPUT SYSTEM USING BUTTERFLY-BASED PARTICLE SWARM  
OPTIMIZATION**

Ali Can ÇABUKER<sup>1,\*</sup>, M. Nuri ALMALI<sup>2</sup>, İshak PARLAR<sup>3</sup>

<sup>1</sup>Van Yuzuncu Yil University, Faculty of Engineering, Department of Electrical and Electronics Engineering, 65080, Van, [alicancabuker@gmail.com](mailto:alicancabuker@gmail.com), ORCID: 0000-0003-2011-2117

<sup>2</sup>Van Yuzuncu Yil University, Faculty of Engineering, Department of Electrical and Electronics Engineering, 65080, Van, [mnal@yyu.edu.tr](mailto:mnal@yyu.edu.tr), ORCID: 0000-0003-2763-4452

<sup>3</sup>Van Yuzuncu Yil University, Faculty of Engineering, Department of Electrical and Electronics Engineering, 65080, Van, [ishakparlar@yyu.edu.tr](mailto:ishakparlar@yyu.edu.tr), ORCID: 0000-0002-3383-8091

Receive Date:02.11.2022

Accepted Date: 02.03.2023

**ABSTRACT**

Studies on the control of nonlinear systems with metaheuristic algorithms are increasing day by day. It is one of the nonlinear systems in the Twin rotor multiple input multiple output (TRMS) system, which emerged as a prototype of helicopters. This system has two control angles horizontally and vertically. In this study, the yaw and pitch angle control parameters of the TRMS system were found using both traditional and butterfly-based particle swarm optimization (BFPSO) method. In experimental studies, reference values of main propeller and tail propeller angles were tried to be reached in TRMS with fractional order proportional-integral-derivative (FOPID), proportional-integral-derivative (PID) and tilt-integral-derivative (TID) controllers.

**Keywords:** TRMS, Pitch Angle, Yaw Angle, BFPSO, PID, FOPID, TID.

**1. INTRODUCTION**

The use of metaheuristic algorithms for systems with nonlinear behavior to reach reference values is gaining momentum today. In the studies carried out, algorithms that give better results are developed, inspired by the behavior changes of animals. These algorithms can be based not only on the behavior of animals, but also on biologically based algorithms such as the immune system. Particle Swarm Optimization (PSO), which is one of the most basic types of swarm optimization, has also been developed over time and has allowed different types of optimization algorithms [1]. Another optimization type created using the PSO algorithm is BFPSO [2, 3]. These optimization types are used in many areas such as renewable energy applications and TRMS systems [4]. In addition to swarm-based optimizations, biological-based algorithms such as genetic algorithms are also used in the control of nonlinear systems [5, 6]. The aim of this study is to estimate the controller coefficients for

nonlinear systems of traditional controllers and metaheuristic optimization methods. Hybrid optimization types have been developed to improve the performance of the two optimization types, usually finding solutions using their populations. With the developed hybrid metaheuristic optimization types, coefficients can be found for control methods such as PID and FOPID. The coefficients obtained can provide a very effective success in the control of non-linear systems such as the twin rotor mimo system. Since non-linear systems are very sensitive, the control coefficient search range should also be found with precision [7, 8].

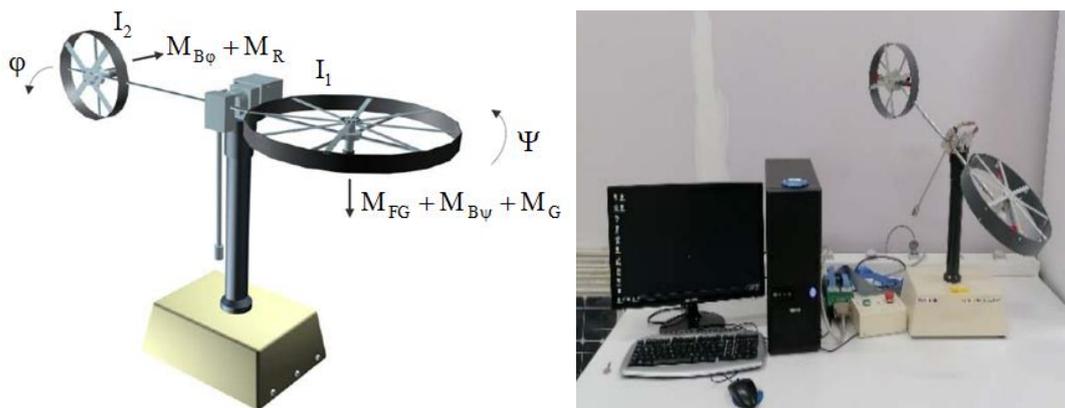
The remainder of the paper is arranged into several sections. Section 2 introduces configuration and theoretical analysis of the TRMS and BFPSO method. Experimental results of traditional control methods and BFPSO method-based controllers are shown in tables in section 3. Finally, in section 4 draws the main conclusions of the paper.

## 2. MATERIAL AND METHOD

### 2.1. TRMS

TRMS, which is produced as a prototype of helicopters, reaches the desired reference angle values with the speed of the DC motors fixed on the ground and on it, unlike the helicopter. When comparing TRMS to helicopter in general terms, TRMS is not capable of flying like a helicopter and does not include cyclic control. Also, in TRMS, a fixed beam determines the equilibrium position.

TRMS is a very difficult device to control due to the coupling dynamics that these angles create against each other, as well as the pitch and yaw angles [9-11]. The TRMS rotor aerodynamics have an angular velocity, which in turn turns the aerodynamic torques. Mechanical-electrical model and experimental setup of TRMS is given in Figure 1.



**Figure 1.** Mechanical-electrical model and experimental setup of TRMS.

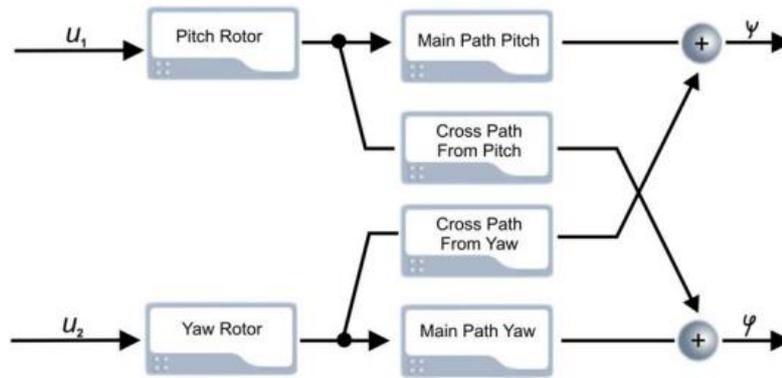
The dynamical equations of TRMS are produced based on Newton's second law. The dynamic equation in the vertical plane is given in Eq.1.

$$M_v = J_v \frac{d^2 \alpha_v}{dt^2} \quad (1)$$

$M_v$  is the sum of the moment components and  $J_v$ , is the sum of the moments of inertia about the horizontal axis. Moments of the repulsive forces in the horizontal plane as well as in the vertical plane can be expressed as in Eq.2.

$$M_{h1} = l_t F_h(w_t) \cos \alpha_v \quad (2)$$

In Figure 2, the blocks in the system structure of TRMS's main path pitch, main path yaw and coupling dynamics cross path from pitch and cross path from yaw, which are cross-connection dynamics are given [12].



**Figure 2.** Decoupling Dynamics of TRMS.

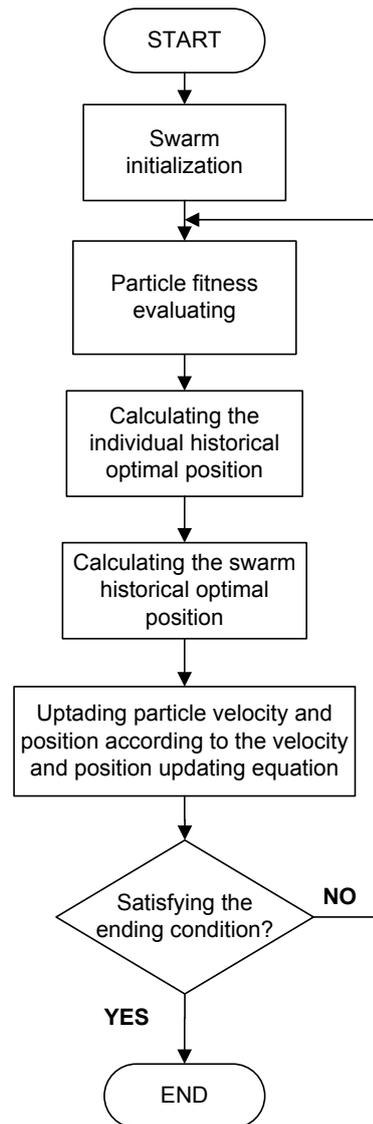
The cross-link dynamics shown in Figure 2 cause control difficulties in TRMS, as Pitch and Yaw angles affect each other [13-14]. Transfer function of TRMS was given Eq. 3 and Eq. 4.

$$G_v = \frac{1.359}{s^3 + 0.997s^2 + 4.786s + 4.278} \quad (3)$$

$$G_h = \frac{3.6}{s^3 + 6s^2 + 5} \quad (4)$$

## 2.2. Particle Swarm Optimization

Particle swarm optimization algorithm is one of the oldest and basic algorithms based on swarm intelligence. It has been used in many applications since its first appearance in 1995. Particle swarm optimization method can be used to solve constrained, nonlinear and multi-objective optimization problems [15, 16]. The flowchart of PSO is given in Figure 3.



**Figure 3.** Flowchart of the particle swarm optimization algorithm.

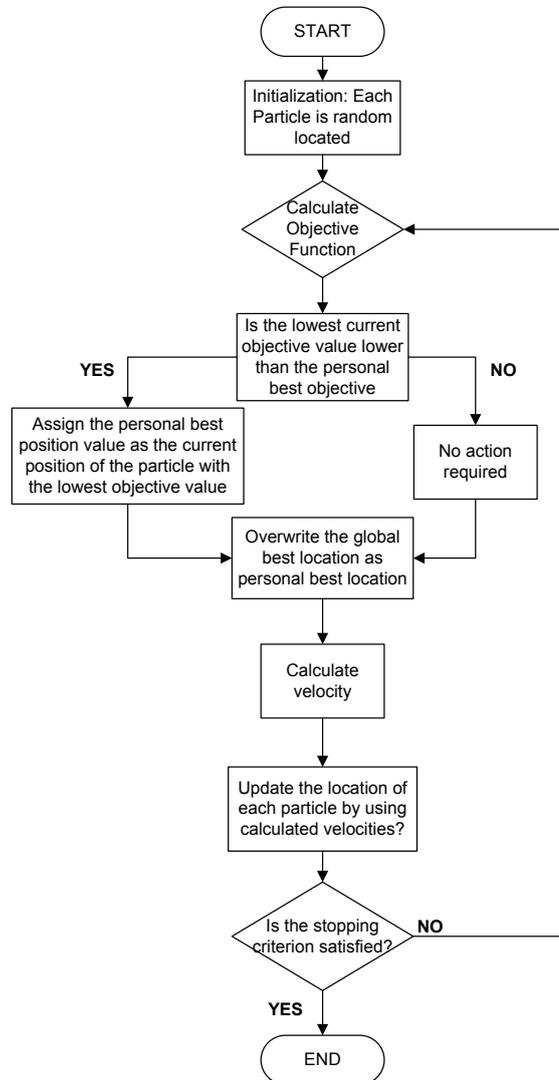
The basic mathematical expressions of particle swarm optimization are given in equations 5 and 6. These equations will be rearranged later for butterfly-based particle swarm optimization;

$$v_{k+1} = w * v_k + c_1 r_1 (pbest_k - currentposition) + c_2 r_2 (gbest_k - currentposition) \quad (5)$$

$$x_{(k+1)} = x_k + V_{k+1} \tag{6}$$

### 2.3. Butterfly-Based Particle Swarm Optimization

In this study, we used butterfly-based particle swarm optimization, which was developed as a hybrid. Particle swarm optimization (PSO) consists of certain stages. The flow diagram of the BFPSO is given in Figure 4 [17-19].



**Figure 4.** Flow Diagram of the BFPSO.

PSO phases begin by randomly placing each particle in the swarm. An objective function is then calculated for each part, comparing whether the calculated objective value is lower than the particle's personal best. If it is not lower and no action is taken, the personal best position value is assigned as the current position of the particle with the lowest objective value [20-22].

After these actions, the personal best position is updated by overwriting the global best position. Next; the velocity of each particle is calculated, the position of each particle is updated using the calculated velocity. Finally, it is questioned whether the obtained values are sufficient to stop the loop. The velocity and motion formulas of particle swarm optimization are given in Eq.5 and Eq.6.

In equations  $w$  is the inertia weight,  $r_1$  and  $r_2$  are two random numbers between (0, 1).  $c_1$  and  $c_2$  are the cognitive and social scaling parameters. However, in order to achieve butterfly-based particle swarm optimization, it is necessary to place the equations of butterfly optimization in the cycle of particle swarm optimization. The velocity equation of the particle swarm optimization is rewritten according to the butterfly optimization. The velocity equation for butterfly-based particle swarm optimization is given in Eq.7.

$$v_{k+1} = w * v_k + S_k(1 - P_k)c_1r_1(pbest_k \text{ currentposition}) + p_k c_2 r_2 (gbest_k - \text{currentposition}) \quad (7)$$

$S_k$  and  $P_k$  are sensivity and probability in the velocity equation are given in Eq. 8 and Eq. 9 [2].

$$s_k = \exp - (Iter_{max} - Iter_k) / Iter_{max} \quad (8)$$

$$P_k = FIT_{gbest,k} / -(FIT_{lbest,k}) \quad (9)$$

We choose parameters of BFPSO are  $c_1=2$ ,  $c_2=2$ ,  $w=0.8$ .  $Iter_{max}$  in equation 8 indicates the maximum number of iterations, and  $Iter_k$  indicates the number of iterations at the time of the loop and  $Iter_{max} = 1000$ . In equation 9,  $FIT_{best}$  = Fitness of local best solutions,  $FIT_{gbest}$  = Fitness of global best solutions [2].

#### 2.4. TID (Tilt-Integral-Derivative) Controller

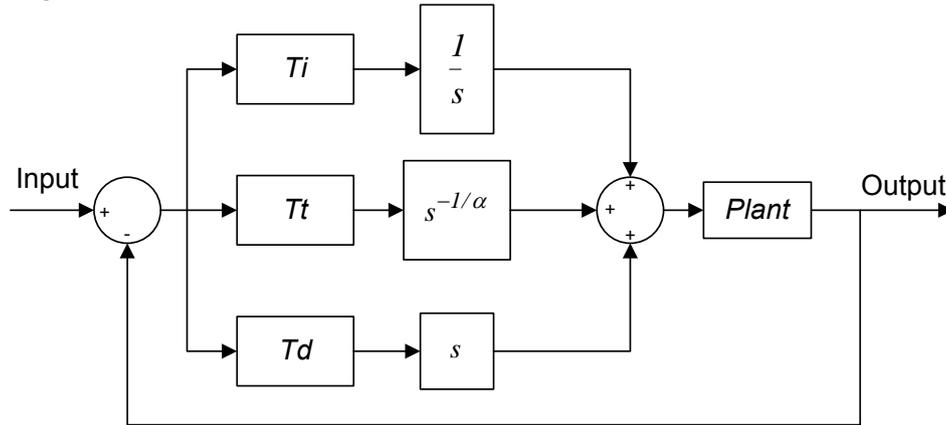
Introduced as a tilt controller (TID), it provides feedback gain as a function of frequency that is tilted or shaped by the gain frequency of a traditional balancing unit. For the tilt compensator,  $\alpha$  is a real number, usually between 2 and 3. The  $\alpha$  value in the TID controller optimizes the  $K_p$  value [17,24].

The traditional expression of the TID Controller is given in Eq. 10.

$$C(s) = T_i S^{-1} + T_t S^{-1/\alpha} + T_d S \quad (10)$$

The TID controller retains many of the advantages of the traditional PID controller due to its easy of adjustment. Also, the TID controller has three parallel paths with adjustable properties, just like the

general block diagram of the PID controller [25]. The general block diagram of the TID controller is given in Figure 5.



**Figure 5.** TID Controller general block diagram.

### 2.5. Fractional Order Proportional Integral Derivative (FOPID)

Fractional Proportional Differential Integral controller (FOPID), unlike classical PID controller, includes fractional degrees in its mathematical equation. Due to this feature, it provides a more effective control of the systems compared to the classical PID controller. Mathematical expression of fractional order PID controller is given in Eq.11.

$$FOPID = Kp + Ki. s^{-\lambda} + Kd. s^{\mu} \quad (11)$$

$\lambda$  and  $\mu$  in Eq. 11 represent the fractional power of integral and differential control.

### 2.6. IAE, ISE, ITAE and ITSE Error Performance

Specific tests are carried out to examine the fault performance of the twin rotor multi-input multi-output system. These tests are integral absolute error (IAE), integral squared error (ISE), integral time squared error (ITSE), and integral time absolute error (ITAE). Mathematical expressions of error performance measures are given in Eq.12-15 [26-28].

$$ISE = \int_0^{\infty} e^2(t)dt \quad (12)$$

$$ITSE = \int_0^{\infty} e^2(t)tdt \quad (13)$$

$$IAE = \int_0^{\infty} |e(t)|dt \quad (14)$$

$$ITAE = \int_0^{\infty} t|e(t)|dt \quad (15)$$

### 3. RESULTS

In this section, the effects of the coefficients obtained by using traditional control methods and BFPSO method on the controllers are discussed one by one. All experimental results are detailed. The rise time, settling time and overshoot times obtained from the graphs are calculated and presented in tables. Finally, the error performance analyzes between the results obtained with the traditional and BFPSO methods were evaluated. In this way, it can be easily seen how much performance is achieved with the traditional and BFPSO method.

PID, FOPID and TID controllers coefficients obtained by butterfly-based particle swarm optimization are given in Table 1-3.

**Table 1.** Coefficients of PID controller with BFPSO method.

<b>BFPSO</b>	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>
<b>PID<math>\phi</math></b>	0.2354	0.4329	0.3250
<b>PID<math>\psi</math></b>	0.2175	0.6207	0.3681

**Table 2.** Coefficients of FOPID controller with BFPSO method.

<b>BFPSO</b>	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>	<b><math>\lambda</math></b>	<b><math>\mu</math></b>
<b>FOPID<math>\phi</math></b>	0.3291	0.2397	0.2470	0.98	1.0295
<b>FOPID<math>\psi</math></b>	0.1942	0.1563	0.1530	1.0125	0.9755

**Table 3.** Coefficients of TID controller with BFPSO method.

<b>BFPSO</b>	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>	<b><math>\alpha</math></b>
<b>TID<math>\phi</math></b>	0.1512	0.4898	0.2248	0.3179
<b>TID<math>\psi</math></b>	0.1010	0.3838	0.1693	0.8936

PID, FOPID and TID controller coefficients obtained by trial and error method are given in Table 4-6. These coefficients are kept within the boundary conditions in which the TRMS system can operate.

**Table 4.** Coefficients of traditional PID controller.

<b>Traditional</b>	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>
<b>PID<math>\phi</math></b>	3	2.5	1.5
<b>PID<math>\psi</math></b>	2.5	7.5	10

**Table 5.** Coefficients of traditional FOPID controller.

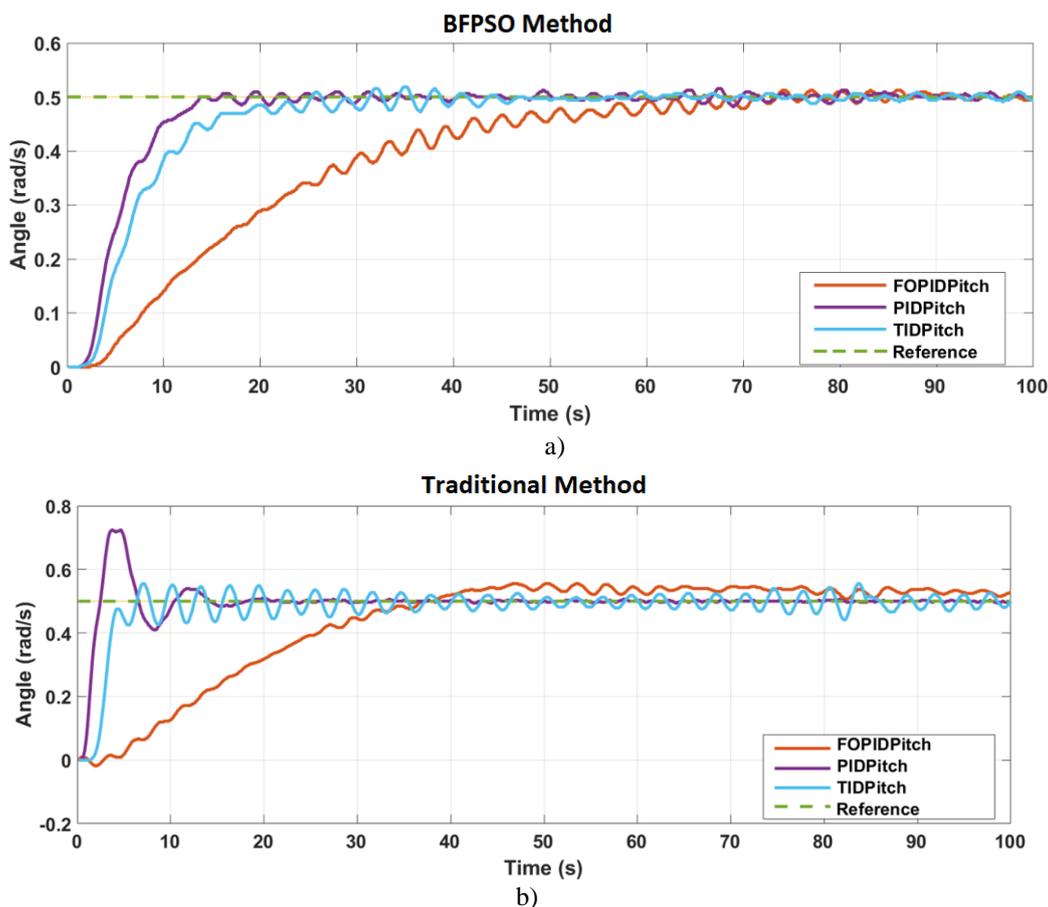
<b>Traditional</b>	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>	<b><math>\lambda</math></b>	<b><math>\mu</math></b>
<b>FOPID<math>\phi</math></b>	0.25	0.10	0.05	1.2	1.90

<b>FOPID<math>\psi</math></b>	0.20	0.10	0.05	1.2	1.8
-------------------------------	------	------	------	-----	-----

**Table 6.** Coefficients of traditional TID controller.

Traditional	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>	<b><math>\alpha</math></b>
<b>TID<math>\phi</math></b>	0.20	0.75	0.25	0.5
<b>TID<math>\psi</math></b>	0.75	0.50	0.25	0.9

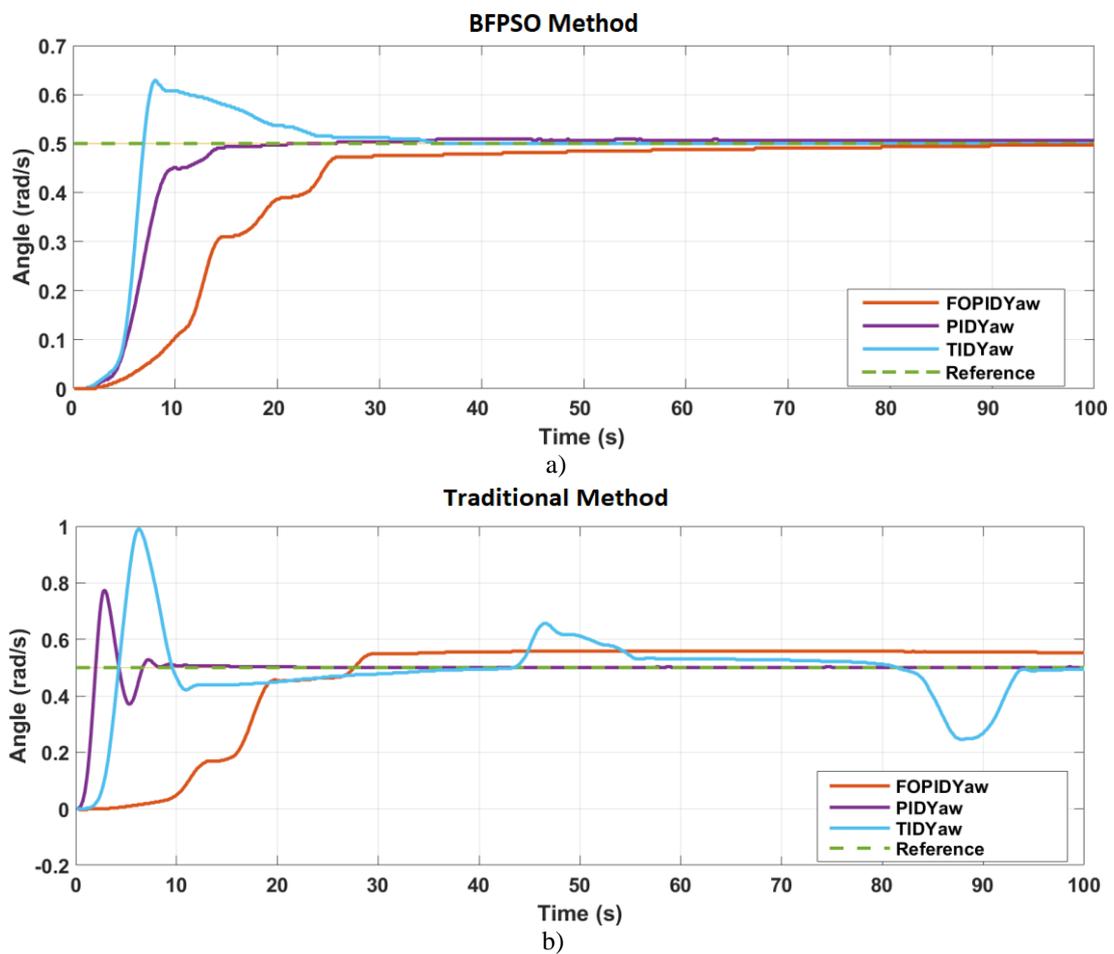
The pitch and yaw angles in TRMS were experimentally obtained using the coefficients of BFPSO. Pitch responses of PID, FOPID and TID controllers with BFPSO method and traditional method show in Figure 6.



**Figure 6.** Pitch responses of PID, FOPID and TID controllers a) BFPSO method and b) traditional method.

As can be seen from the graphs obtained by using the coefficients obtained by the BFPSO method, it is seen that the pitch angles on all controllers have significantly improved in the amount of rise, settling and overshoot.

The pitch and yaw angles in TRMS were experimentally obtained using the traditional coefficients. Yaw responses of PID, FOPID and TID controllers with BFPSO method and traditional method show in Figure 7.



**Figure 7.** Yaw responses of PID, FOPID and TID controllers a) BFPSO method and b) traditional method.

As can be seen from the graphs obtained by using the coefficients obtained by the BFPSO method, it is seen that the yaw angles on all controllers have significantly improved in the amount of rise, settling and overshoot. It has also been observed that it prevents serious overshoots in yaw angles.

The error performance analyzes of pitch and yaw angles on the PID, FOPID and TID controllers of the coefficients obtained using the BFPSO and traditional method are given in Table 7-8.

**Table 7.** Error analyzes performance of PID, FOPID and TID controllers with BFPSO method.

<b>BFPSO</b>	<b>ISE</b>	<b>ITAE</b>	<b>IAE</b>	<b>ITSE</b>
<b>PID<math>\phi</math></b>	1.42	44.59	4.013	15.75
<b>PID<math>\psi</math></b>	1.003	31.22	3.192	10.39
<b>FOPID<math>\phi</math></b>	2.842	113.8	8.376	113.8
<b>FOPID<math>\psi</math></b>	3.218	192.1	10.85	175.9
<b>TID<math>\phi</math></b>	1.337	27.48	4.062	11.77
<b>TID<math>\psi</math></b>	1.3	45.4	4.338	28.99

**Table 8.** Error analyzes performance of traditional PID, FOPID and TID controllers.

<b>Traditional</b>	<b>ISE</b>	<b>ITAE</b>	<b>IAE</b>	<b>ITSE</b>
<b>PID<math>\phi</math></b>	0.373	4.429	1.329	1.52
<b>PID<math>\psi</math></b>	0.4107	14.78	1.796	0.4485
<b>FOPID<math>\phi</math></b>	3.49	326.9	11.89	191.7
<b>FOPID<math>\psi</math></b>	4.26	463.6	16.22	463.5
<b>TID<math>\phi</math></b>	1.789	282.9	7.677	68.22
<b>TID<math>\psi</math></b>	0.716	97.31	3.585	16.77

The rise time, settling time, overshoot and maximum values of the controllers in the yaw and pitch angles with the BFPSO and traditional method are given in Table 9-12.

**Table 9.** Rise time, settling time, overshoot and maximum values of yaw with BFPSO method.

<b>BFPSO</b>	<b>Rise Time</b>	<b>Settling Time</b>	<b>Overshoot</b>	<b>Maximum Value</b>
<b>PID<math>\phi</math></b>	8.4934	22.1729	0.1061	0.5093
<b>FOPID<math>\phi</math></b>	20.8403	68.0384	0	0.5000
<b>TID<math>\phi</math></b>	2.8283	37.1837	0.1289	0.6289

**Table 10.** Rise time, settling time, overshoot and maximum values of pitch with BFPSO method.

<b>BFPSO</b>	<b>Rise Time</b>	<b>Settling Time</b>	<b>Overshoot</b>	<b>Maximum</b>
--------------	------------------	----------------------	------------------	----------------

				<b>Value</b>
<b>PID<sub>ψ</sub></b>	8.3709	95.676	0.0154	0.5154
<b>FOPID<sub>ψ</sub></b>	43.4094	88.9760	0.0123	0.5123
<b>TID<sub>ψ</sub></b>	11.5126	99.0780	0.0185	0.5185

**Table 11.** Rise time, settling time, overshoot and maximum values of yaw angle with traditional method.

<b>Traditional</b>	<b>Rise Time</b>	<b>Settling Time</b>	<b>Overshoot</b>	<b>Maximum Value</b>
<b>PID<sub>φ</sub></b>	1.034	9.474	0.2731	0.7731
<b>FOPID<sub>φ</sub></b>	17.3	28.83	0.0584	0.5584
<b>TID<sub>φ</sub></b>	1.66	93.43	0.4910	0.9910

**Table 12.** Rise time, settling time, overshoot and maximum values of pitch angle with traditional method.

<b>Traditional</b>	<b>Rise Time</b>	<b>Settling Time</b>	<b>Overshoot</b>	<b>Maximum Value</b>
<b>PID<sub>ψ</sub></b>	1.321	16.87	0.2240	0.7240
<b>FOPID<sub>ψ</sub></b>	27.92	99.04	0.0553	0.5553
<b>TID<sub>ψ</sub></b>	1.71	99.77	0.0553	0.5553

#### 4. CONCLUSION

The computer on which applications are performed on TRMS has i3 2nd generation 8 GB ram and used the TRMS owned by the university and the estimated budget of the TRMS is 250.000 Turkish liras. When the times to reach the reference values are compared; In yaw angles, the rise time was improved with the TID controller 2.8283 seconds, the PID controller 8.4934 seconds, and the FOPID controller 20.84 seconds. While there is a difference of about 6 seconds between the PID controller and the TID controller, the TID controller provides an advantage in rising time and a disadvantage in settling time. The TID controller has a high overshoot, while the PID controller settles to the reference value about 15 seconds ago. For the pitch angle, the PID controller responded faster than the TID controller for about 8 seconds at rise time and 4 seconds at settling time. On the contrary, in terms of overshoot, the FOPID controller performed faster than the other controllers with almost zero overshoot in yaw angle and 0.0123 seconds overshoot in pitch angle despite the slow response.

In the light of the obtained data, it has been observed that the PID controller gives better results than the FOPID and TID controllers for the pitch and yaw angle. In the graph obtained for the pitch angle, it was observed that the controllers could not provide sufficient resistance against the gravity and therefore experienced oscillations while reaching the reference value. It was seen that the most successful response against gravity was obtained with the PID controller. Among the control methods, FOPID was determined to be the controller that gave the slowest response in reaching the reference

value that is why error performance is high and the TID controller responds quickly at rise time, it makes a very slow movement at settling time. Butterfly-based particle swarm optimization gives better results than classical particle swarm optimization in terms of sensitivity and possibilities. High sensitivity prevents overshoot and deterioration in the control of TRMS. Because of that BFPSO was selected for the simulation of TRMS.

#### **ACKNOWLEDGEMENTS**

The authors thank the TRMS Application Unit at Van Yuzuncu Yil University Electronics Laboratory for some of the data presented in this article.

#### **REFERENCES**

- [1] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks, 4, 1942-1948.
- [2] Bohre, A. K. Agnihotri, G. and Dubey, M. (2014). Hybrid butterfly based particle swarm optimization for optimization problems. In 2014 First International Conference on Networks & Soft Computing, 172-177.
- [3] Bohre, A. K. Agnihotri, G. and Dubey, M. (2015). The butterfly-particle swarm optimization (Butterfly-PSO/BF-PSO) technique and its variables. International Journal of Soft Computing, Mathematics and Control, IJSCMC, 4, 3.
- [4] Mathi, D. K. and Chinthamalla, R. (2020). A hybrid global maximum power point tracking method based on butterfly particle swarm optimization and perturb and observe algorithms for a photovoltaic system under partially shaded conditions. International Transactions on Electrical Energy Systems, 30, 10.
- [5] Agrawal, A. K. (2013). Optimal Controller Design for Twin Rotor MIMO System, Doctoral dissertation.
- [6] Mustapha, S. Fayçal, K. M. and Mohammed, S. (2015). Application of artificial immune algorithm-based optimisation in tuning a PID controller for nonlinear systems. International Journal of Automation and Control, 9, 3, 186-200.
- [7] Ting, T. O. Yang, X. S. Cheng, S. Huang, K. (2015). Hybrid metaheuristic algorithms: past, present, and future. Recent advances in swarm intelligence and evolutionary computation, 71-83.
- [8] Khanduja, N. Bhushan, B. (2021). Optimal design of FOPID Controller for the control of CSTR by using a novel hybrid metaheuristic algorithm. Sādhanā, 46, 2, 1-12.

- [9] TRMS, T. R. M. (2010). System Control Experiments Manuel. 33-949S: Feedback Instruments Ltd. Sussex, UK.
- [10] Tiwalkar, R. G. Vanamane, S. S. Karvekar, S. S. and Velhal, S. B. (2017). Model predictive controller for position control of twin rotor MIMO system. In 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI, 952-957.
- [11] Chalupa, P. Přikryl, J. and Novák, J. (2015). Modelling of twin rotor MIMO system. Procedia Engineering, 100, 249-258.
- [12] Wijekoon, J. Liyanage, Y. Welikala, S. and Samaranyake, L. (2017). Yaw and pitch control of a twin rotor MIMO system. In 2017 IEEE International Conference on Industrial and Information Systems, ICIIS, 1-6.
- [13] Chaudhary, S. and Kumar, A. (2019). Control of Twin Rotor MIMO system using 1-degree-of-freedom PID, 2-degree-of-freedom PID and fractional order PID controller. In 2019 3rd International conference on Electronics, Communication and Aerospace Technology, ICECA, 746-751.
- [14] Katoch, S. Chauhan, S. S. and Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80, 5, 8091-8126.
- [15] Wang, D. Tan, D. and Liu, L. (2018). Particle swarm optimization algorithm: an overview. Soft computing, 22, 2, 387-408.
- [16] El-Shorbagy, M. A. Hassanien, A. E. (2018). Particle swarm optimization from theory to applications. International Journal of Rough Sets and Data Analysis, IJRSDA, 5, 2, 1-24.
- [17] Jaen-Cuellar, A. Y. de J. Romero-Troncoso, R. Morales-Velazquez, L. and Osornio-Rios, R. A. (2013). PID-controller tuning optimization with genetic algorithms in servo systems. International Journal of Advanced Robotic Systems, 10, 9, 324.
- [18] Meena, D. C. and Devanshu, A. (2017). Genetic algorithm tuned PID controller for process control. In 2017 International Conference on Inventive Systems and Control, ICISC, 1-6.
- [19] Khuwaja, K. Tarca, I. C. and Tarca, R. C. (2018). PID controller tuning optimization with genetic algorithms for a quadcopter. Recent Innovations in Mechatronics, 5, 1, 1-7.
- [20] Abukan, Y. Almalı, M. N. Çabuker, A. C. and Parlar, İ. (2022). Determining The PID Parameters of The TRMS System Using PSO. 1ST INTERNATIONAL CONFERENCE ON ENGINEERING AND APPLIED NATURAL SCIENCES, Konya, Türkiye, 10 - 13 Mayıs 2022, 97-102.

- [21] Abdulhussein, K. G. Yasin, N. M. Hasan, I. J. (2021). Comparison between butterfly optimization algorithm and particle swarm optimization for tuning cascade PID control system of PMDC motor. *International Journal of Power Electronics and Drive Systems*, 12, 2, 736.
- [22] El Hajjami, L. Mellouli, E. M. Berrada, M. (2019). Optimal PID control of an autonomous vehicle using Butterfly Optimization Algorithm BOA. In *Proceedings of the 4th international conference on big data and internet of things*, 1-5.
- [23] Esgandanian, A. and Daneshvar, S. (2016). A comparative study on a tilt-integral-derivative controller with proportional-integral-derivative controller for a pacemaker. *International Journal of Advanced Biotechnology and Research, IJBR*, 7, 3, 645-650.
- [24] Aidoud, M. Feliu-Battle, V. Sebbagh, A. Sedraoui, M. (2022). Small signal model designing and robust decentralized tilt integral derivative TID controller synthesizing for twin rotor MIMO system. *International Journal of Dynamics and Control*, 1-17.
- [25] Lurie, B. J. (1994). Three-parameter tunable tilt-integral-derivative (TID) controller.
- [26] Yusoff, W. A. W. Yahya, N. M. and Senawi, A. (2006). Tuning of Optimum PID Controller Parameter Using Particle Swarm Optimization Algorithm Approach. *Fakulti Kejuruteraan Mekanikal University Malaysia Pahang*.
- [27] Faisal, R. F. and Abdulwahhab, O. W. (2021). Design of an adaptive linear quadratic regulator for a twin rotor aerodynamic system. *Journal of Control, Automation and Electrical Systems*, 32, 2 404-415.
- [28] Bahramipour-Esfahani, R. Nasri, M. Tabatabaei, S. M. (2021). Designing a Metaheuristic Multi-objective Fractional-order PID Controller for TRMS system. *Computational Intelligence in Electrical Engineering*, 12, 2, 91-112.

## APPENDIX I

**Table 13.** TRMS nonlinear model parameters [7].

Parameters	Value
$I_1$ - Moment of inertia of pitch rotor	$6,8 \cdot 10^{-2} kg \cdot m^2$
$I_2$ - Yaw rotor moment of inertia	$2 \cdot 10^{-2} kg \cdot m^2$
$a_1$ - Static characteristic parameter	0,0135
$b_1$ - Static characteristic parameter	0,0924
$a_2$ - Static characteristic parameter	0,02
$b_2$ - Static characteristic parameter	0,09
$M_G$ - Gyroscope momentum	0,32 Nm
$B_{1\psi}$ - Friction momentum function parameter	$6 \cdot 10^{-3} Nm/rad$
$B_{2\psi}$ - Friction momentum function parameter	$1 \cdot 10^{-3} Nm/rad$
$B_{1\varphi}$ - Friction momentum function parameter	$1 \cdot 10^{-1} Nm/rad$

$B_{2\phi}$ - Friction momentum function parameter	$1 \cdot 10^{-2} Nm/rad$
$K_{gy}$ - Gyroscope momentum parameter	$0,05 rad/s$
$k_1$ - 1. Engine gain	1,1
$k_2$ - 2. Engine gain	0,8
$T_{11}$ - 1. Engine denominator parameter	1,1
$T_{10}$ - 1. Engine denominator parameter	1
$T_{21}$ - 2. Engine denominator parameter	1
$T_{20}$ - 2. Engine denominator parameter	1
$T_p$ - Cross-reaction momentum parameter	2
$T_0$ - Cross-reaction momentum parameter	3,5
$k_c$ - Cross-reaction momentum gain	-0,2

---