JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES
(ISSN : 1304-0448)
July 2021 Volume 14 Number 2
jast.hezarfen.msu.edu.tr

Research

Chaotic Particle Swarm Optimisation Algorithm Based Quadrotor Control

M. Buğrahan ARTUÇ[1], İsmail BAYEZİT[2*]

[1] *Istanbul Technical University, Faculty of Aeronautics and Astronautics, 34467 Sariyer, Istanbul, Turkey*, artucm@itu.edu.tr, https://orcid.org/0000-0002-6787-9916
[2] *Istanbul Technical University, Faculty of Aeronautics and Astronautics, 34467 Sariyer, Istanbul, Turkey*, bayezit@itu.edu.tr, https://orcid.org/0000-0001-9345-5108
* *Corresponding Author*

Article Info

Received: *April 13, 2021*
Accepted: *July 14, 2021*
Online: *July 26, 2021*

Keywords: IRIS Quadrotor, Chaotic PSO, PID Control, Trajectory Tracking, Swarm Intelligence

Abstract

Unmanned aerial vehicles are currently used for civil and military purposes. Today, quadrotor vehicles are highly important example of non-conventional underactuated rotary-wing platforms. In addition, quadrotors include complex nonlinearities compared to conventional aerial vehicles due to internal instability. Therefore, Chaotic Particle Swarm Optimisation Algorithm is employed to overcome and regulate this issue. In this study, a nonlinear system model of the IRIS+ quadrotor is used and the classical PID control is designed using CPSO algorithm. Chaotic Particle Swarm Algorithm is used to emerge a stable reference model for robust adaptive control of the experimental testbed.

To Cite This Article: M.B. ARTUÇ, İ. BAYEZIT, "Chaotic Particle Swarm Algorithm Based Quadrotor Control", Journal of Aeronautics and Space Technologies, Vol. 14, No. 2, pp. 261-267, July. 2021.

Kaotik Parçacık Sürü Algoritması Tabanlı Quadrotor Kontrolü

Makale Bilgisi

Geliş: *13 Nisan 2021*
Kabul: *14 Temmuz 2021*
Yayın: *26 Temmuz 2021*

Anahtar Kelimeler: IRIS Quadrotor, Kaotik Particle Swarm Optimisation, PID Kontrol, Yörünge Takibi, Sürü Zekası

Özet

İnsansız hava araçları günümüzde sivil ve askeri amaçlarla kullanılmaktadır. Bugün, quadrotor hava araçları konvansiyonel olmayan yüksek öneme sahip döner kanatlı hava araçlarına örnek olarak verilebilir. Çok rotorlu hava araçları durum uzay modeline göre tam olarak modellenemediğinden konvansiyonel hava araçlarına göre nonlineeritesi yüksektir. Bu sebeple, Kaotik Parçacık Sürü Algoritması (CPSO) bu durumun üstesinden gelmek için kullanılabilir. Bu çalışmada doğrusal olmayan IRIS+ quadrotor hava aracının sistem modeli elde edilmiştir. Daha sonra klasik kontrol PID metodu CPSO algoritması yardımıyla kararlı bir referans model ortaya çıkarmak için uygulanmıştır. Çalışma yardımıyla dayanıklı uyarlamalı kontrolcüler ilgili referans çalışma üzerine gelecekteki deneysel çalışmalarda kullanılacaktır.

1. INTRODUCTION

Today, many different unmanned aerial vehicles are used for defence and business related operations. One of the highly used aerial research testbeds is quadrotor type small aerial vehicle. There are wide approximation methods in order to control quadrotor vehicles with different approaches [1,2]. However, it is significantly challenging problem in respect to optimality. Quadratic regulator problem is solved to address this phenomenon, previously. But, translational motion dynamics cannot be modelled conveniently for quadrotor type vehicles using state-space form due to lack of state controllability and dependencies on weight matrixes as in [3,4]. Therefore, generic and available solutions are employed in order to improve the model and control system quality. For this reason, heuristic methods such as particle swarm algorithm and genetic algorithm are used to deal with this problem [5-7]. In this paper, nonlinear quadrotor model is used to adjust gain values of Linear Time Invariant (LTI) based compensator structure.

The quadrotor nonlinear model is generally obtained by using Newton-Euler method. It is available in models obtained by Lagrange mechanics, but it is mainly used in systems with constraints on dynamics and cannot provide a solution for energy loss forces such as friction. As in the simulation model, the nonlinear system model can be reorganized as in state space form by means of linearization techniques. However, there may be a chaos situation due to the smallest difference in the starting point or unconsidered assumptions in reality [8]. In this paper, Chaotic Particle Swarm Algorithm (CPSO) based PID tuning method is used to stabilize quadrotor states. As a result, we are capable of using the reference model implementing robust adaptive control design for real flight tests or testbench experiment [8].

PID control system, which is tuned by Ziegler Nichols methods and Matlab Tuner toolbox including parametric relation method, is practically acceptable for real-time applications. However, tuning the linear control coefficients with this method are not useful for nonlinear applications. This method does not provide an optimal solution due to not structuring the system in the state-space form. On the other hand, systems including sequential rotation and translation plant cannot be used in full-state feedback form [9]. For this reason, deterministic and heuristics methods have been developed. In deterministic methods, there is a very complex mathematics for nonlinear models. As a result, genetic algorithm approach as in [10] and particle swarm optimization methods are used as the tuning method for PID coefficients detailed in [5].

Actually, there are several heuristic based approaches and methods for tuning controllers of quadrotors [20, 21]. In reference [20], Ant Colony Optimization (ACO), Particle Swarm Optimization, and Artificial Bee Colony Optimization methods are compared to design efficient PID controllers for multirotor vehicles. According to the aforementioned research, ACO algorithm is one of the best algorithms in order to optimize transient response compared to other techniques given in [20]. However, results are not presented for the full-state behaviour of the multirotor in this paper. Other nature-inspired algorithms which are called Genetic Algorithm, Evolution Strategies, Differential Evolutionary and Cuckoo Search are compared in reference [21]. Genetic Algorithm is the best optimization method to tune controller gains although Evolutions Strategies response desirable way without rotation response because of sensitivity in [21]. However all these references do not mention perspective from chaos theory despite using of chaos numbers as proved in reference [22]. According to the study in [22], chaotic maps is one of the best method to increase efficiency of the optimization method. Additionally, research provide 10 chaotic map and impact of these maps to biogeography based optimization. Moreover, it can be said that chaotic probability can increase efficiency for Genetic Algorithms and Particle Swarm Optimization according to reference in [11]. CPSO algorithm is also used for Fault Tolerant Control for powerful optimal stability [23]. However, path tracking performance and full-state effect is not provided in [23].

In this paper, the optimum control of a nonlinear quadrotor model using PID control is achieved. In order to get rid of the mathematical complexity given in deterministic approaches, Chaotic Particle Swarm Optimization method is used as the tuning method for control coefficients [11]. The efficiency of the developed algorithm is compared with the Matlab Control Toolbox SISO tuner results. SISO tuner is generally used to design linear control systems for linear/nonlinear systems within Matlab/Simulink environment. In this approach, nonlinear dynamic equations of the vehicle system are just isolated and linearized for controller design. The CPSO and SISO tuner methods are used to compare the time response performance of reference flight mission path. We combine using modified LQR as given in [9] in order to implement adaptive control approach for reference model as given in [12-15].

2. SYSTEM MODEL

The system model is extracted as a nonlinear model and widely used coordinate systems are presented as in Figure 1.

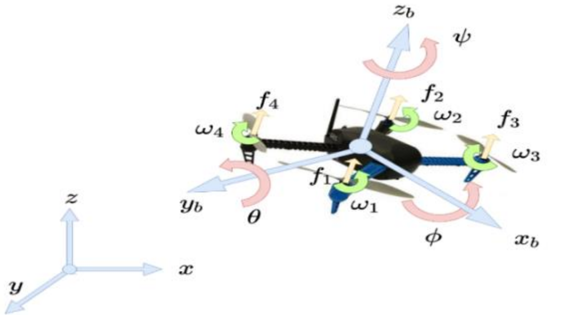

Figure 1. IRIS+ Quadrotor Kinematics

2.1 Quadrotor Kinematics

There are two configurations available in the literature for the quadrotor mathematical model: plus '+' and cross 'x' configurations. In this study, 'x' configuration based modeling of quadrotor vehicle is used. In order to mathematically model the quadrotor, two different axis sets are defined. In this way, a transformation matrix is defined for the rotation between axes [16].

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\varphi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{1}$$

In Eq. 1, three-dimensional rotation matrix is identified in order to convert translational states of quadrotor from body to inertial axis.

2.2 Quadrotor Dynamics

For the system model, the quadrotor is rigid, symmetrical, the origin of the body center and body coordinate system coincides, the propellers are rigid and the thrust and creep are proportional to the square of angular velocity. At the same time, by incorporating gyroscopic effects, linear and cyclic dynamic model are obtained by Newton Euler mechanics [5].

$$\begin{aligned}
\ddot{x} &= \frac{U_1}{m}(cos\phi sin\theta cos\psi + sin\phi sin\psi) \\
\ddot{y} &= \frac{U_1}{m}(cos\phi sin\theta cos\psi - sin\theta cos\psi) \\
\ddot{z} &= \frac{U_1}{m}(cos\phi cos\theta) - g \\
\ddot{\phi} &= \frac{l}{I_{xx}}U_2 - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \frac{I_{zz}}{I_{xx}}\dot{\psi}\dot{\theta} - \frac{I_{yy}}{I_{xx}}\dot{\psi}\dot{\theta} \\
\ddot{\theta} &= \frac{l}{I_{yy}}U_3 + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \frac{I_{xx}}{I_{yy}}\dot{\psi}\dot{\phi} - \frac{I_{zz}}{I_{yy}}\dot{\psi}\dot{\phi} \\
\ddot{\psi} &= \frac{l}{I_{zz}}U_3 + \frac{J_r}{I_{zz}}\dot{\Omega}_r + \frac{I_{yy}}{I_{zz}}\dot{\theta}\dot{\phi} - \frac{I_{xx}}{I_{zz}}\dot{\phi}\dot{\theta}
\end{aligned} \tag{2}$$

The dynamical equations of a quadrotor platform can be seen in Eq. 2.

2.3 Control Motor Mixer Matrix

The control input vector is calculated by multiplying rotor angular velocities by aerodynamic coefficients in Eq. 3. This vector contains impulse, wobble, pitch, and deviation moments. The subsystem, which is turned into a matrix, is the matrix form of the input from the rotor to the quadrotor input.

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ -K_f & K_f & K_f & -K_f \\ K_f & -K_f & K_f & -K_f \\ -K_M & -K_M & K_M & K_M \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (3)$$

Rotor transfer function can be given as input speed to output speed. The effect of the rotor can be added to the system by placing the rotor transfer function between these two systems.

2.4 Calculation of Aerodynamic Coefficients

Aerodynamic coefficients can be obtained either experimentally or theoretically. The airfoil shape of the propellers is taken from NACA 0009 and the aerodynamic data is derived from the coefficients according to the height during the cycle.

Propeller produces control inputs for the system. These inputs are forces and moments. Propeller provides this effect with wings. This produces friction force with the airfoil propulsion in the wing section. This impulse and friction can be defined as thrust and torque coefficient. In Eq. 4, C_T and C_Q can be reached by using the momentum and blade element theories [17].

Lift-curve slope is Cl_α and θ_0 is the blade collective pitch angle per each motor, induced flow is represented as λ per rotor, zero lift drag coefficient of the used airfoil in each motor is C_{d0}. Number of blades in each motor is N. Rotor radius of each blade is R. Chord length is used as c. According to NACA 0009 profile data: $Cl_\alpha = 2.87$, $C_{d0} = 0.012$, $C_{l0} = 0$.

Inflow ratio λ can be calculated as $= \sqrt{\frac{C_T}{2}}$. Solidity factor is $= \frac{N*c}{\pi R}$. Blade collective pitch is as follows:

$\theta_0 = \frac{6C_T}{\sigma C_{l\alpha}} + \frac{3}{2}\sqrt{\frac{C_T}{2}}$.

Torque coefficient is calculated as follows:

$C_Q = \frac{1}{2}\sigma\left(\frac{\sqrt{2}(C_T)^{3/2}}{\sigma} + \frac{C_{d0}}{4}\right)$.

As a result, we are able to calculate C_T and C_Q, accordingly in Eq. 4.

In Eq. 4, ρ, A, C_T and C_Q, R, and Ω_i are represented air density, blade area, aerodynamic coefficients, radius of blades, and angular velocity of each rotor, respectively. The aerodynamic force and moments rely on the geometry of the propeller and air density. In our IRIS+ quadrotor model, the air density can be considered as constant due to the limited maximum altitude. Then, equations can be reduced to $f_i = K_f \omega_i^2$ and $m_i = K_M \omega_i^2$ for each rotor.

$$\begin{aligned} C_T &= \frac{1}{2}\sigma Cl_\alpha\left(\frac{\theta_0}{3} - \frac{\lambda}{2}\right) \\ C_Q &= \frac{1}{2}\sigma\left(\frac{\lambda Cl_\alpha \theta_0}{3} - \frac{\lambda Cl_\alpha^2}{2} + \frac{C_{d0}}{4}\right) \\ K_f &= \frac{1}{2}\rho A C_T R^2 \quad K_M = \frac{1}{2}\rho A C_Q R^3 \end{aligned} \quad (4)$$

3. IMPLEMENTATION

Particle Swarm Optimisation algorithm is one of the best algorithms having different type used in motion planning and control [5,18]. PSO is first discovered in 1995. Developing over time has been developed into the final form. In this algorithm, finding the global minimum point is done with particles. According to the deterministic methods, it aims to approach global minimum or maximum in all directions rather than searching from a single point. It detects particles at randomly selected locations within the search space as individuals of a flock of birds. The randomness is represented with continuous uniform random numbers for chaotic PSO. Iteration represents progression along the velocity vector for each individual [19].

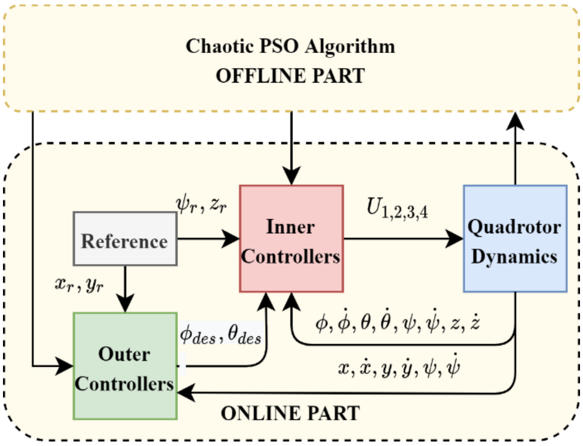

Figure 2. Implementation of Chaotic Particle Swarm Optimisation Algorithm

Actually, there are several methods to obtain global point for nonlinear problems such as Chaotic Particle Swarm Optimization, Chaotic Genetic Algorithm and their normal version compared and tested for some nonlinear system such as Lorenz, Tent and Henon system models. It can be seen that Chaotic Particle Swarm Algorithm is the fastest solution to converge global point of the nonlinear models [11]. Therefore, this approach can be used in order to address adjusting PID gains by using uniform random function reflecting chaotic probability. As seen in Algorithm 2, Tent Map for chaotic probability provides chaotic array. The third- and fourth-line code in Algorithm 1 refers to sequential random numbers by using "MaxIt" and "np" corresponding to maximum iteration and number of particles. Also "rand(1)" giving random number is just enable algorithm 2 to initial point. Finally, this can be seen several chaotic type maps as in [22].

Algorithm 1: Chaotic Particle Swarm Optimization Algorithm

Input: *Objective Function*
Output: Global Point
Initialize the position and velocities of particles and find fitness, p_{id}^{k} *and* g_{id}^{k} *values.*
n*=1;*
C_{r_1} *=**ChaoticRand(** rand(1),MaxIt, np)*
C_{r_2} *=**ChaoticRand(** rand(1), MaxIt, np)*
for it*=1:Maximum Iteration*
decrease w value to avoid local points
  for i*=1: number of particles*
    C_{r_1} **(n)=** C_{r_1}
    C_{r_2} **(n)=** C_{r_2}
    Update Velocity and Position Using Equation (5)
    Apply lower and upper bounds for position's regions which correspond to control gain to prevent over increasing of gains
    Set each PID gains equal to particle's positions.
    Run Nonlinear Model in chapter 2 and its Controller Blocks according to Objective Function.
    Set each cost of the position equal to value of fitness
    Update best personnel position & cost
    Update Global Best Cost
    n*=**n**+1*
  end
    If it satisfy stopping condition, stop algorithm
end

In Eq. 5, it stores the best position "p_{id}^{k}" for each iteration in terms of performance in the flock and the position "x_{id}^{k}" in which each individual performs best during iteration. According to this information, particle swarm is targeted to reach the global point "g_{id}^{k}" by means of the calculated velocity "v_{id}^{k}" as given in Eq. 5. Additionally, w, c_1, and c_2 help the approximation speed to global best value of the controller gain selection space. C_{r_1} and C_{r_2} are approximate uniform random functions to define chaotic probability. Performance is a determined value function. This function can also be selected as the total square error for the control algorithm.

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k}$$
$$v_{id}^{k+1} = w * v_{id}^{k} + c_1 C_{r_1}\left(p_{id}^{k} - x_{id}^{k}\right) + c_2 C_{r_2}\left(g_{id}^{k} - x_{id}^{k}\right) \quad (5)$$

As it is seen in Figure 2 and Algorithm 1, Implementation of CPSO algorithm is presented. We assign two objective functions in order to obtain optimal control coefficients within Inner and Outer Loop of our quadrotor control system. CPSO defined in Algorithm 1 aims to reduce ISE functions given in Eqs. 6 and 7 using Chaotic Tent Map as given in Algorithm 2.

Algorithm 2: Sequential Chaotic Number Generation using Tent Map

function *Tentmap=**ChaoticRand**(initial point, Maximum Iteration, number of particles)*
  Maximum Iteration*=* ***Maximum Iteration *number of particles***
  x*(1)=initial point % it can be from 0 to 1*
  for n*=1:Maximum Iteration*
    if *x(n)<0.7*
    x(n+1)=x(n)/0.7;
    end
    if *x(n)>=0.7*
    x(n+1)=(10/3)(1-x(n));*
    end
    Tentmap=x(n)
  end
end

Eq. 6 is used to determine altitude and attitude behaviour of control coefficients. In Figure 3, the decrease of the cost during the iteration is illustrated. After 40 iterations, capability of the control structure can be stabilized and any differences cannot be observed significantly. For the horizontal position, Eq. 7 is set to tune the control structures. As it can be seen in Figure 4, iteration number is enough to reach safe and fast coverage for reference input signal compared to inner control loop.

Eqs. 6 and 7 are Integral Square Errors (ISE) represent the total square error calculation for angular, altitude and x-y position states of quadrotor. Table 1 shows optimal control parameters by means of our CPSO Algorithm and standard Matlab Optimisation tools.

$$ISE_1 = \int_0^{\infty} \left(e_z^{\,2}(t) + e_\phi^{\,2}(t) + e_\theta^{\,2}(t) + e_\psi^{\,2}(t)\right) dt \quad (6)$$

$$ISE_2 = \int_0^{\infty} \left(e_x^{\,2}(t) + e_y^{\,2}(t)\right) dt \quad (7)$$

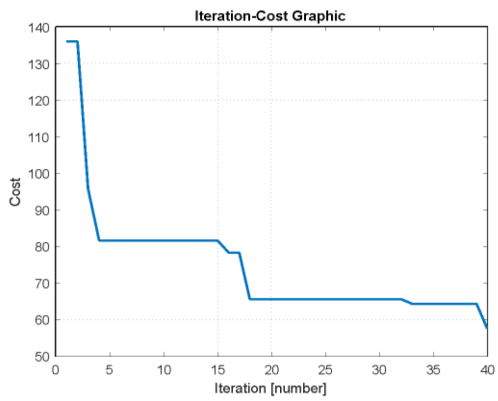

Figure 3. ISE1 Iteration Cost for Inner Loop Control

As it is seen in Fig. 3 and Fig. 4, the cost decreases during the iteration. Optimal results are achieved with 40 iterations for 24 particles in ISE1 and 21 iterations for 10 particles in ISE2.

3.1 Inner Loop Controller

The system response of the controllers in the inner loop to the angle input of 1 meter and 30 degrees can be seen in Figure 5. The prominent contribution is the settling time for PID controllers adjusted with the CPSO algorithm. The settling time of the system response has been reduced and it is achieved with fewer peak values especially for pitch and yaw angles.

Firstly, rise time is highly faster in CPSO based control of quadrotor vehicle. Another contribution is the response of rotational dynamics having much faster settling time for CPSO tuning. This method also provides a better solution for peak value compared to the SISO Tuner tool. Although same peak values can also be observed for altitude and roll angle responses in Figure 5, convergence to the reference input is much better in CPSO based control design algorithm.

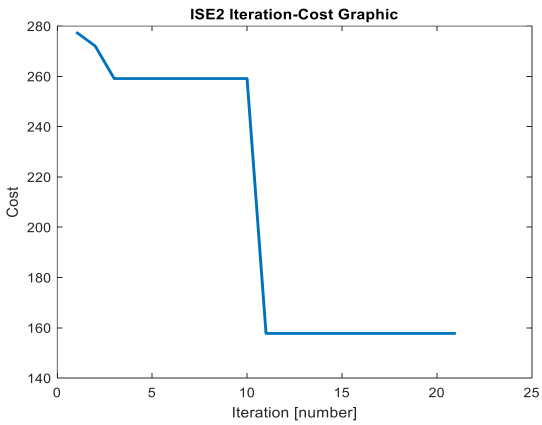

Figure 4. ISE2 Iteration Cost for Outer Loop Control

3.2 Outer Loop Controller

Step responses of the outer loop controllers are shown

in Figure 6. The PID controllers obtained via CPSO algorithm for the inner loop controller, have produced better step response than the PID structures set with SISO Tuner tool. Figure 5 compares the response of the x and y positions to the step input of the outer loop controller. Finally, the detailed three dimensional flight performance for reference path tracking is provided in Section 3.3.

CPSO algorithm reflects the efficiency according to the rise time and settling time of the x position tracking in Figure 6. However, Matlab SISO tuner gives better performance than the CPSO method in terms of overshoot as shown in Figure 6. Also, SISO tuner do not settles in the given time as compared with CPSO based control for x position tracking. This outcome can also be seen in Section 3.3.

Table 1. PID Gains obtained by SISO Tuner & CPSO

MATLAB SISO			CPSO Method		
Roll Controller			**Roll Controller**		
Kp	Ki	Kd	Kp	Ki	Kd
0,178	0,094	0,085	74,993	250	19,724
Pitch Controller			**Pitch Controller**		
Kp	Ki	Kd	Kp	Ki	Kd
0,13	0,06	0,07	217,5	250	26,5896
MATLAB SISO			**CPSO Method**		
Yaw Controller			**Yaw Controller**		
Kp	Ki	Kd	Kp	Ki	Kd
0,23	0,109	0,12	99,067	100,824	4,8458
Z-Position Controller			**Z-Position Controller**		
Kp	Ki	Kd	Kp	Ki	Kd
125	85	63	250	250	83,775
X-Position Controller			**X-Position Controller**		
Kp	Ki	Kd	Kp	Ki	Kd
0,48	0,1	0,9	4,995	5	4,325
Y-Position Controller			**Y-Position Controller**		
Kp	Ki	Kd	Kp	Ki	Kd
0,8	0,1	1,44	5	0,051	3,0911

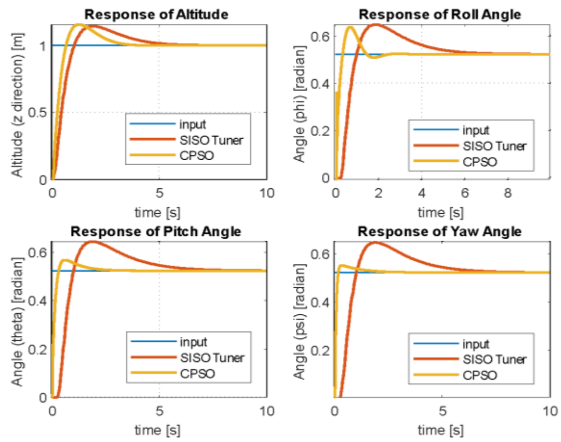

Figure 5. Performance of the inner loop controllers

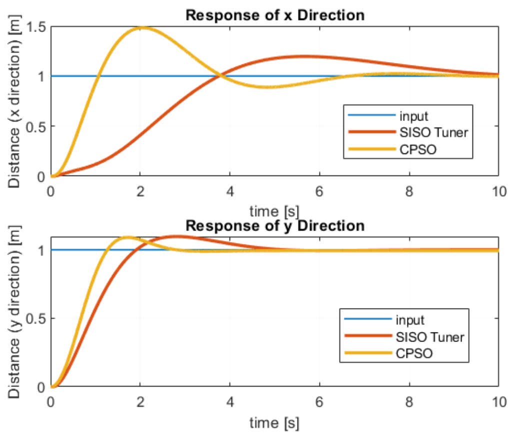

Figure 6. The performance of outer loop controllers

3.3 Quadrotor Path Tracking Performance

The behaviour of the quadrotor for a specified reference input is shown in Figure 7. The blue line is the system reference input. As it can be seen in this figure, the PID controllers designed via the CPSO algorithm have much better performance. Figure 7 shows that the PID controller set with the CPSO algorithm has a better system response than a given trajectory. In Figure 7, reference input after reaching 0.5 meter high goes to the x direction with one meter and starts to track one meter radius circle path.

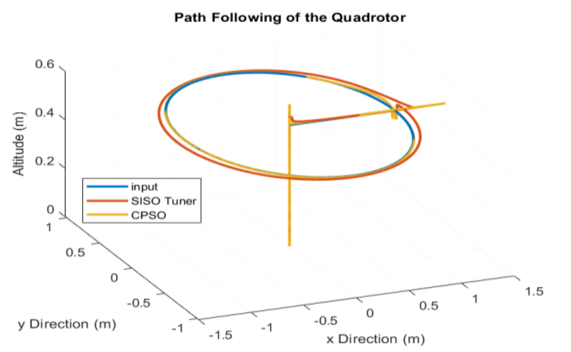

Figure 7. Quadrotor path tracking performance

The PSO algorithm provided better time response results than the SISO Tuner as shown in Figure 8. In particular, a better reference tracking is achieved with the PID coefficients obtained by the CPSO algorithm. Because x and y direction cannot be followed for the nonlinear system by SISO tuner based PID method as requested. This is due to the nonlinearities among the angle of the rotational dynamics.

Although z response of the CPSO algorithm partially corrupts at cross maneuvers of the trajectory as in Figure 8, altitude controller having faster response compared to the SISO tuner method. In contrast, the performance of CPSO based design can be improved with higher iteration and much better controller performance can be obtained.

4. CONCLUSION

The aim of this study is to compare the CPSO method and SISO Tuner method in order to achieve trajectory tracking for IRIS quadrotor. Therefore, the nonlinear quadrotor system model is obtained via flight mechanical parameters of the system. Nested PID controllers are designed for 6 degrees of freedom of quadrotor vehicle. The controllers are designed using the SISO Tuner method and the Chaotic Particle Swarm Optimisation Algorithm. In this work, CPSO algorithm yielded sufficient results for three dimensional flight control of nonlinear quadrotor vehicle behaviour.

The aim of this study is to make real-time studies on IRIS+ quadrotor model using PID coefficients determined by CPSO algorithm. In this respect, it is a practical data source in terms of operating time and optimum performance. After obtaining stable dynamics, CPSO algorithm can be expressed for other techniques, such as weight of the Q and R matrix and modified LQR as given in [9]. Therefore, algorithm can be extended with the robust adaptive control techniques in future targets of this research.

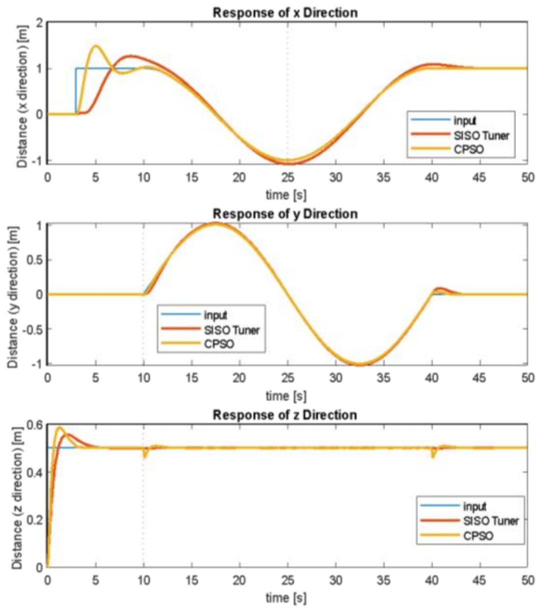

Figure 8. Variation of trajectory response over time

5. REFERENCES

[1] K. Turkoglu, et al. "PID Parameter Optimization of an UAV Longitudinal Flight Control System", *World Academy of Science, Engineering and Technology*, vol. 45, pp. 340-245, 2008.

[2] Ş. Kurak, M. Hodzic, "Control and Estimation of a Quadcopter Dynamical Model", *Periodicals of Engineering and Natural Sciences*, vol. 6, pp. 63-75, March 2018.

[3] E. Okyere, et al. "LQR Controller Design for Quadrotor Helicopters", *The Journal of Engineering*, vol. 17, pp. 4003-4007, January 2019

[4] C. Luis, J. Le Ny, "Design of a Trajectory Tracking Controller for a Nano-Quadcopter", *arxiv.org*, [Online]. Available: https://arxiv.org/abs/ 1608.05786. [Accessed: March 13, 2021].

[5] N. H. Abbas, A. R. Sami, "Tuning of PID Controllers for Quadcopter System using Hybrid Memory based Gravitational Search Algorithm–Particle Swarm Optimization", *International Journal of Computer Applications*, vol. 172, pp. 9-18, August 2017.

[6] V. Brito, et al. "Quadcopter Control Approaches and Performance Analysis", in proceedings of *International Conference on Informatics in Control, Automation and Robotics, ICINCO 2018, Portugal*, pp. 86–93.

[7] M. Erdem, “Quadrotor Trajectory Tracking Control Using Reinforcement Learning”, Master Thesis of Istanbul Technical University, Turkey, 2019.

[8] M. Hancer, R. Bitirgen, and I. Bayezit, “Designing 3-DOF Hardware-In-The-Loop Test Platform Controlling Multirotor Vehicles”, *IFAC-Papers Online*, vol. 51, pp. 119-124, January 2018.

[9] R. Guardeño, M. J. Lopez, and V. M. Sánchez, "MIMO PID Controller Tuning Method for Quadrotor based on LQR/LQG Theory," *Robotics*, vol. 8, pp. 36, May 2019.

[10] A. Goel, R. Kumar, and S. Narayan, "Design of MRAC augmented with PID controller using genetic algorithm", *in proceedings of IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2016, Delhi, India July 4-6, 2016*.

[11] R. Hosseinpourfard, M. M. Javidi, "Chaotic PSO using the Lorenz System: An Efficient Approach for Optimizing Nonlinear Problems", *Çankaya University Journal of Science and Engineering*, vol. 12, pp. 40-59, 2015.

[12] F. Jurado, and R. Hernández. "Decentralized MRAC with Integral Action for Attitude Control of a Quadrotor UAV", *in proceedings of IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC 2018, Ixtapa, Mexico, Nov. 14-16, 2018.*

[13] A.P. Nair, N. Selvaganesan, and V. R. Lalithambika, "Lyapunov Based PD/PID in Model Reference Adaptive Control for Satellite Launch Vehicle Systems", *Aerospace Science and Technology*, vol. 51, pp. 70-77, April 2016.

[14] F. Jurado, et al. "Decentralized direct MRAC for attitude control of a quadrotor UAV", *in proceedings of 14th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2017, Mexico City, Mexico, October 20-22, 2017*.

[15] M. Navabi, H. Mirzaei, "Robust Optimal Adaptive Trajectory Tracking Control of Quadrotor Helicopter", *Latin American Journal of Solids and Structures*, vol. 14, pp. 1040-1063, June 2017.

[16] W. Z. Fum, “Implementation of Simulink Controller Design on IRIS+ Quadrotor,” Master Thesis of Naval Postgraduate School Monterey, US, 2015.

[17] G. J. Leishman, *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2006.

[18] V. Çavuş, A. Tuncer, "İnsansız Hava Araçları İçin Yapay Arı Kolonisi Algoritması Kullanarak Rota Planlama" *Karaelmas Science and Engineering Journal*, vol. 7, pp. 259-265, 2017.

[19] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", *in proceedings of IEEE International Conference on Neural Networks, ICNN 1995, Perth, WA, Australia, Nov. 27-Dec. 1, 1995.*

[20] A. Kaba, “A Comparative Study on the Tuning of the PID Flight Controllers Using Swarm Intelligence”, *International Journal of Aviation Science and Technology*, vol. 1, pp. 80-91, 2020.

[21] L. Abdou, and G. Hossam-Eddine, "Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms." *Evolutionary Intelligence*, vol. 14, pp. 61-73, November 2019.

[22] S. Saremi, S. Mirjalili, and A. Lewis. "Biogeography based Optimisation with Chaos." *Neural Computing and Applications*, vol. 25, pp. 1077-1097, April 2014.

[23] W. Jun, Y. Xiong-Dong, and T. Yu-Yang, "Fault-Tolerant Control Design of Quadrotor UAV Based on CPSO." In the proceeding of *IEEE 4th International Conference on Control Science and Systems Engineering, ICCSSE 2018, Wuhan, China, August 21-23, 2018,* pp. 279-283.

VITAE

M. Buğrahan ARTUÇ received his B.Sc. degree in mechanical engineering from Istanbul Technical University (ITU) in 2018. He completed the double major degree on aeronautical engineering at ITU in 2019. Then, he started graduate degree at faculty of aeronautics and astronautics and completed his M.Sc. degree in aeronautics and astronautics engineering programme at ITU in 2021. He is currently a Ph.D. student, research and teaching assistant in the same department at Istanbul Technical University.

Dr. İsmail BAYEZİT received his B.Sc. from electronics engineering and control system engineering at Istanbul Technical University (ITU), Turkey as a double major student in 2005 and 2006, respectively. He received M.Sc. degree at ITU control and automation engineering in 2009. Ismail received his Ph.D. degree as mechanical engineer from University of Waterloo (UW), Canada in 2014. His main research focus includes collaborative flight, space technologies, such as reaction wheels and control moment gyros, connected and autonomous vehicles, model-based design and control of systems. He is currently working as an assistant professor and department vice chair at aeronautics engineering of Istanbul Technical University.