Anatolian Journal of Computer Sciences

Volume: 3 No: 1 2018 © Anatolian Science

pp: 14-28 ISSN:2548-1304

Multi-Parameter Optimization of Sliding-Mode Controller for Quadcopter Application

İsmail Can DİKMEN [1], Teoman KARADAĞ [2], Celaleddin YEROĞLU [3]
[1]Inonu University, Electrical Electronics Engineering, Malatya, TURKEY (can.dikmen@inonu.edu.tr)
[2]Inonu University, Electrical Electronics Engineering, Malatya, TURKEY (teoman.karadag@inonu.edu.tr)
[3]Inonu University, Computer Engineering, Malatya,TURKEY (c.yeroglu@inonu.edu.tr)

Submission Date: 10.06.2018
Date of Acceptance: 06.07.2018

Abstract— For many years, quadcopters are quite popular in the academic field because of its structural simplicity. However, this property comes out the problem of designing an effective controller. Designing a controller for quadcopter is rather complicated because tuning of the controller parameters of multi-rotor structure to achieve a desired performance for agility, flying efficiency and immediate reaction is a challenging problem. To deal with such a difficulty, Ant Colony Optimization (ACO), Invasive Weed Optimization (IWO) and Firefly Optimization (FO) algorithms are used to obtain optimal parameters of Sliding Mode Controller (SMC). SMC is used for both attitude and position control of the quadcopter. By taking into consideration all six variables with different number of parameters (total number of parameters to be optimized are nineteen). This makes it a complicated tuning problem. In this numerical study, performance results of optimization algorithms are compared with respect to convergence rate and cost function.

Keywords :*quadcopter, ant colony optimization, firefly optimization, invasive weed optimization, sliding mode control*

1. Introduction

The main advantage of a quadcopter comparing to a helicopter is the fixed rotor propulsion. The roll, pitch and yaw angles of quadcopter can be changed by altering the throttle measure of each rotor. On a quadcopter, four rotors are placed on the edges of cross shaped body frame. Opposing rotors rotate in the same direction. (S. Gupte, Paul Infant Teenu Mohandas and J. M. Conrad, 2012). To ensure the movement on roll axis(φ), left and right rotors, in respect to the front end, change their speed as to one another. Same as roll axis; to ensure the pitch movement(θ) front and rear rotors change their speed comparing to one another. To provide a movement on the yaw axis(ψ) rotors in the same direction change their speed in inverse proportion to the rotors in opposite direction. So, an unbalance occurs on the yaw axis forces while total trust remains the same. Angles and body frame axis of a sample quadcopter can be seen in Figure1.

For controlling quadcopters, many different control strategies applied since it becomes a popular academic research topic. Most of these algorithms and quadcopter mathematical modelling can be found in (Lebao Li, Lingling Sun and Jie Jin, 2015; I. C. Dikmen, A. Arisoy and H. Temeltas, 2009) and references therein. Some of the control strategies are PID, LQR, LQG, H∞, sliding mode, feedback linearization, robust, model predictive, back stepping, adaptive, fuzzy logic, nested saturation, neural network, reinforcement learning, iterative learning, memory and learning based intelligent controllers. To run these algorithms effectively, control parameters must be optimized in the range of usage and physical specifications (P. D. Sheth and A. J. Umbarkar, 2015).

There are many inspirations of the invention of heuristic optimization algorithms. Some of them are biology-based, while some are music-based (Bo Xing, Wen-Jing Gao, 2014). Variety of optimization algorithms successfully applied to the parameter optimization problem of the quadcopter (Bouabdallah, 2007) (Gustavsson, 2015) (SundarapandianVaidyanathan, 2017). In this study, ACO, IWO and FO algorithms are applied to the stability and position control problem of quadcopter separately. Total 19 parameters of the controller; 3 for each pitch, roll and yaw axis, 3 for X position, 3 for Y position and 2 for altitude (Z) are optimized to achieve desired performance. Simulation results of ACO, IWO and FO optimization algorithms are compared according to colony population and cost function efficiency.

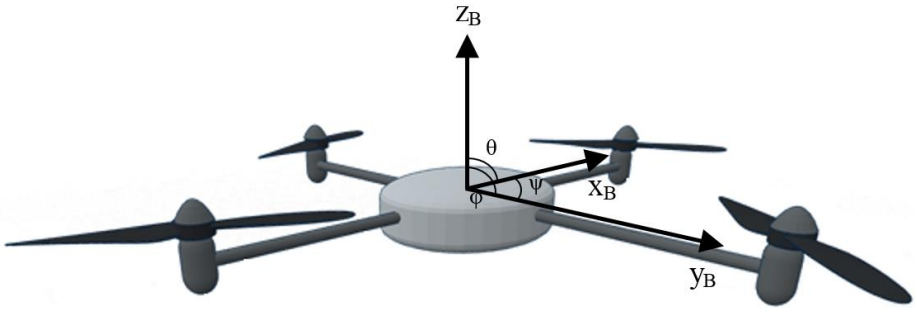

Figure1. Quadcopter's body frame axis and Euler angles.

2. Methodology

2.1. Quadcopter Modelling

Quadcopter can only be controlled by changing rotors speeds independently. Thereby, changing forces; torques and moments can be generated on the body. The quadcopter model used in this paper is shown in Figure1. In the figure, x_B, y_B and z_B are unit vectors with respect to the body frame. φ,θ and ψ are Euler angles that represents roll, pitch and yaw angles. Let $V_i$be speed and τ_i be trust of ith rotor. Total trust applied to the system is

$$U_4 = \tau_1 + \tau_2 + \tau_3 + \tau_4 \tag{1}$$

The roll, pitch and yaw moments can be achieved by varying the trust of rotors respectively as following,

$$U_1 = l(\tau_4 - \tau_2) \tag{2}$$

$$U_2 = l(\tau_3 - \tau_1) \tag{3}$$

$$U_3 = -\tau_1 + \tau_2 - \tau_3 + \tau_4 \tag{4}$$

The trust $(\tau_i)$and torque (D_i) produced by propellers can be defined by square of rotor speed which is denoted by Ω^2.

$$\left.\begin{matrix} \tau_i = b\Omega_i^2 \\ D_i = d\Omega_i^2 \end{matrix}\right\} (b, d: constant) \tag{5}$$

One can write control signals of quadcopter as

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \\ b & b & b & b \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{6}$$

where, U_4 is for altitude control input, U_1, U_2, U_3 are for roll, pitch and yaw angle control inputs respectively. The dynamic model is derived using Lagrange formulism under the assumptions (Bouabdallah, 2007):

- The structure of the body is rigid and symmetrical.
- Propellers are rigid.
- Thrust and drag are proportional to the square of rotor speed.
- Center of mass and body frame origins are coinciding.

The Lagrangian ($\mathcal{L}$) can be shown as potential energy (E_{Pot}) subtracted from the sum of translational(E_{Tr})and rotational energy(E_{Pot}).

$$\mathcal{L}(q_i, \dot{q}_i) = E_{Tr} + E_{Rot} - E_{Pot} \tag{7}$$

$$\Gamma_i = \begin{bmatrix} F \\ \tau \end{bmatrix} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \tag{8}$$

where, $\dot{q}_i$ represents generalized coordinates and Γ_i represents generalized forces and torques. Quadcopter's angular motion equations can be written as follows;

$$\ddot{\varphi} = \frac{\left(I_y - I_z\right)}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r}{I_x}\dot{\theta}\Omega_r + \frac{l}{I_x}U_1 \tag{9}$$

$$\ddot{\theta} = \frac{\left(I_z - I_x\right)}{I_y}\dot{\varphi}\dot{\psi} + \frac{J_r}{I_y}\dot{\varphi}\Omega_r + \frac{l}{I_y}U_2 \tag{10}$$

$$\ddot{\psi} = \frac{(I_x - I_y)}{I_z}\dot{\varphi}\dot{\theta} + \frac{1}{I_z}U_3 \tag{11}$$

Translational equation of motions can be written as;

$$\ddot{x} = \frac{U_4}{m}(\cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi) \tag{12}$$

$$\ddot{y} = \frac{U_4}{m}(\cos\varphi \sin\theta \sin\psi - \sin\varphi \cos\psi) \tag{13}$$

$$\ddot{z} = \frac{U_4}{m}(g - \cos\varphi \cos\psi) \tag{14}$$

The symbols used above are given in Table I.

Table 1. List of symbols used in modeling

Symbol Definitions			
φ	Roll angle	J_r	Propeller inertia
θ	Pitch angle	l	Arm length
ψ	Yaw angle	I	Inertia moment
Ω	Rotor speed	b	Trust factor
Ω_r	Disturbance	d	Drug factor
m	Total mass	g	Gravitational acceleration

2.2. Simulation

For this paper, full control of a quadcopter simulation is prepared in MATLAB R2015. An IMU can provide linear acceleration and angular velocity information rather fast, more than 1000 Hz. (Gustavsson, 2015). Therefore, typical sampling frequency is taken as 200Hz, that is, the sensor delay is taken as 5 milliseconds for the simulation.

2.3. Sliding Mode Controller Design(SMC)

To obtain straightforward and robust control structure, Lyapunov stability method can be used for SMC algorithm (Sundarapandian Vaidyanathan, 2017). For ease of terminology and expressing equations, control method is taken into two sections. First one is the attitude controller and second one is the position controller. The state vector of the simulation consists of both attitude and position state vectors.

Attitude controller:

Suppose X_{att} represents the state vector of attitude controller.

$$X_{att} = \left[\varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\psi}\right] \tag{15}$$

Hereφ, θ, ψ are roll, pitch and yaw;$\dot{\varphi}, \dot{\theta}, \dot{\psi}$ are angular velocities respectively. So, one can write the state vector as

$$X_{att} = [x_1, x_2, x_3, x_4, x_5, x_6] \tag{16}$$

$$\dot{x}_1 = x_2, \qquad \dot{x}_3 = x_4, \qquad \dot{x}_5 = x_6 \tag{17}$$

Then, a sliding surface should be determined. For this, the order of switching functions should be less than the order of the plant. To provide these requirements the sliding surface should be chosen as (Samir Bouabdallah, Roland Siegwart, 2005);

$$s = x_2 + Cx_1 \tag{18}$$

$$\dot{s} = \dot{x}_2 + C\dot{x}_1 = A\dot{x}_2 + BU + Cx_2 \tag{19}$$

After choosing the sliding surface, control law can be stated. This control law has to converge the system to the sliding surface. When it reaches the sliding surface it must remain on the surface. In order to achieve this, a Lyapunov function is chosen as;

$$V = \frac{1}{2}s^2 \tag{20}$$

It's time derivative is, $\dot{V} = s\dot{s}$, from Eq.19

$$\dot{V} = s[A\dot{x}_2 + BU + Cx_2] \tag{21}$$

$$A\dot{x}_2 + BU + Cx_2 = 0 \tag{22}$$

Here, $U(x) \sim \beta(x)$.

$$\beta(x) = -\frac{A\dot{x}_2 + Cx_2}{B} \tag{23}$$

Stability is provided if

$$U \begin{cases} < \beta(x), & s > 0 \\ = \beta(x), & s = 0 \\ > \beta(x), & s < 0 \end{cases} \tag{24}$$

So, this is ensured by using the control law

$$U = \beta(x) - Ksign(s) \qquad K > 0 \tag{25}$$

Signum function in Eq.25 causes chattering on the controller so the system is affected negatively. To overcome this case, the most common way is to replace signum function with an equivalent one (Sundarapandian Vaidyanathan, 2017). In our case, we preferred to replace the control function with hyperbolic tangent function. Hence the input function becomes;

$$U = \beta(x) - Ktanh(\lambda s) \qquad \lambda > 0 \tag{26}$$

We are able to adjust the smoothness of hyperbolic tangent function by changing λ parameter. In this study λ is taken 0.5. In order to implement SMC to quadcopter dynamics, the sliding surfaces for roll, pitch and yaw angles are defined respectively as,

$$s_{\varphi} = \left(\dot{\varphi}_{ref} - \dot{\varphi}\right) + \mathrm{C}_1\left(\varphi_{ref} - \varphi\right) \tag{27}$$

$$s_{\theta} = \left(\dot{\theta}_{ref} - \dot{\theta}\right) + \mathrm{C}_3\left(\theta_{ref} - \theta\right) \tag{28}$$

$$s_{\psi} = \left(\dot{\psi}_{ref} - \dot{\psi}\right) + \mathrm{C}_5\left(\psi_{ref} - \psi\right) \tag{29}$$

From Eq.23, Eq.26 and Eq.27; input control function for roll angle is derived as follows,

$$\beta_1(x) = -\frac{a_1\dot{\theta}\dot{\psi} + a_2\dot{\theta}\Omega_r + C_2\dot{\varphi}}{b_1} \tag{30}$$

$$U_1 = \frac{a_1\dot{\theta}\dot{\psi} + a_2\dot{\theta}\Omega_r + C_2\dot{\varphi}}{b_1} - K_1 tanh\left(\lambda s_{\varphi}\right) \tag{31}$$

The input functions U_2 and U_3 for roll and yaw angle control can be derived with the same steps from Eq. 20, 22 and 23 as below,

$$\beta_2(x) = -\frac{a_3\dot{\varphi}\dot{\psi} + a_4\dot{\varphi}\Omega_r + C_4\dot{\theta}}{b_2} \tag{32}$$

$$U_2 = -\frac{a_3\dot{\varphi}\dot{\psi} + a_4\dot{\varphi}\Omega_r + C_4\dot{\theta}}{b_2} - K_2 tanh(\lambda s_{\theta}) \tag{33}$$

$$\beta_3(x) = -\frac{a_5\dot{\theta}\dot{\varphi} + C_6\dot{\psi}}{b_3} \tag{34}$$

$$U_3 = -\frac{a_5\dot{\theta}\dot{\varphi} + C_6\dot{\psi}}{b_3} - K_3 tanh\left(\lambda s_{\psi}\right) \tag{35}$$

Table2. Variables used in controller design

Variable Definitions			
a_1	$(I_y - I_z)/I_x$	a_5	$(I_x - I_y)/I_z$
a_2	$-J_r/I_x$	b_1	$1/I_x$
a_3	$(I_z - I_x)/I_y$	b_2	$1/I_y$
a_4	J_r/I_y	b_3	$1/I_z$

Position controller:

Suppose X_{pos} represents the state vector of position controller.

$$X_{pos} = [x, \dot{x}, y, \dot{y}, z, \dot{z}] \tag{36}$$

where x, y, $z$are positions with respect to earth frame; $\dot{\varphi}, \dot{\theta}, \dot{\psi}$ are angular velocities respectively. So, one can write the state vector as

$$X_{pos} = [x_7, x_8, x_9, x_{10}, x_{11}, x_{12}] \tag{37}$$

$$\dot{x}_7 = x_8, \qquad \dot{x}_9 = x_{10}, \qquad \dot{x}_{11} = x_{12} \tag{38}$$

From here on same step can be applied until the control inputs obtained. To do that first sliding surfaces should be determined.

$$s_x = (\dot{x}_{ref} - \dot{x}) + C_7(x_{ref} - x) \quad (39)$$

$$s_y = (\dot{y}_{ref} - \dot{y}) + C_{10}(y_{ref} - y) \quad (40)$$

$$s_z = (\dot{z}_{ref} - \dot{z}) + C_{13}(z_{ref} - z) \quad (41)$$

U_x,$U_y$and $U_z$are the control inputs for position and altitude. Accordingly, one can write control inputs as,

$$U_x = \frac{m}{U_4}\left(K_4 tanh(\lambda s_x) + C_8 s_x + \ddot{x}_d + C_9(\dot{x}_d - \dot{x})\right) \quad (42)$$

$$U_y = \frac{m}{U_4}\left(K_5 tanh(\lambda s_y) + C_{11} s_y + \ddot{y}_d + C_{12}(\dot{y}_d - \dot{y})\right) \quad (43)$$

$$U_z = \frac{m}{\cos\varphi \cos\theta}\left(K_6 tanh(\lambda s_z) + C_{14} s_z + g + \ddot{z}_d + C_{15}(\dot{z}_d - \dot{z})\right) \quad (44)$$

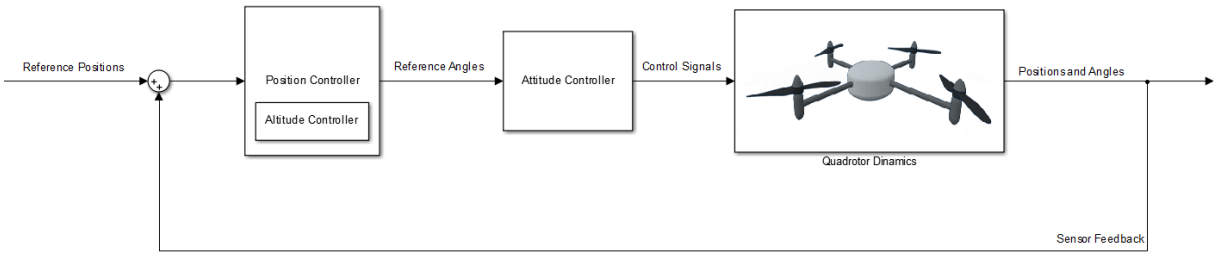

Figure 2. Blog diagram of the SMC controller

2.4. Ant Colony Optimization Algorithm

Some problems such as scheduling, vehicle routing, timetabling, are considered as real-world problems and they are solved with the help of heuristic algorithms. This algorithm is first introduced by Marco Dorigo in his Ph.D. thesis (Dorigo, 1992) and it is inspired by foraging behavior of ants. The common behaviors of ants are mathematically modeled for the algorithm. Similar to searching of ants for the food, they start to explore the surrounding area around their nest randomly. When an ant finds a food source, it takes some portion of food and then carries it back to their nest. On the way back to the nest, ant leaves a pheromone track on the ground. Pheromone quantity on the ground is proportional to the quality and amount of the food found. So, this trail gives information and guides other ants to the food source. This pheromone based, circuitous communication between ants helps them finding the shortest track between the food and their nest (Marco Dorigo, Mauro Birattari, Thomas Stützle, 2006). This communication technique is known as stigmergy (Abraham, Grosan, & Ramos, 2006).

Table3.ACO Pseudo Code

ACO Pseudo Code
Initialize **While** *Termination conditions are not met* **do** *For all population* *Generate paths and calculate fitness* *Pheromone update* *Update paths* **end** **end**

2.5. Invasive Weed Optimization

This optimization algorithm is first proposed by Mehrabian and Lucas (Mehrabian AR, Lucas C., 2006;). It is based on the natural selection rule. For the selection some prominent characteristic properties of weeds, namely adaptation, robustness, aggressiveness, and invasion, became dominant. In general, natural intrusion is an event in which the groups of life forms (for example weeds) move to new place for better conditions and contend with local populaces (Nanako Shigesada, Kohkichi Kawasaki, 1997). In fact, it is not a new discovery anyway, it is a standout amongst the most vital effects on the world's environments (Shibu Jose, Harminder Pal Singh, Daizy Rani Batish, Ravinder Kumar Kohli, 2013). Additionally, it can be utilized as an essential system in designing effective optimization algorithms (I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, 2012).

Table4.IWO Pseudo Code

IWO Pseudo Code
Initialize **While** *Termination conditions are not met* **Do** *For all population* *Calculate Fitness* *Produce Seeds* *Distribute Seeds* **end** *Competitive Exclusion* **end**

2.6. Firefly Algorithm

This algorithm is first proposed by Xin-SheYang (Yang, 2008) and inspired by nature. It is based on flashing behavior of fireflies. To do that three rules are to be set. These are;

- All fireflies are unisex
- Attractiveness is proportional to their brightness
- Brightness of a firefly is determined by its fitness.

In this algorithm first fireflies are distributed in the search space in a random manner. Interaction between fireflies is calculated considering their individual brightness, environment and distance between. The attractiveness is inversely proportional to the distance between two individuals. If there is no other firefly brighter than one firefly it will move randomly.

Table5.FA Pseudo Code

FA Pseudo Code
Initialize **While** *Termination conditions are not met* **do** *For all population* *Evaluate Light Intensity* *Calculate Attractiveness* *Move* **end** **end**

3. Implementation

3.1. Control

Sliding mode controller is used as low level controller for attitude control and used as high level controller for position control. Thus the performance of this controller under noise and disturbance is needed to be studied in order to present the effective control of quadcopter. When the effective control is ensured then parameter optimization is to be studied. All simulations are run with optimum parameters for 38 seconds. Running time period is chosen as this because the completion of a one full round of reference paths takes 38 seconds as shown in Figure 3.

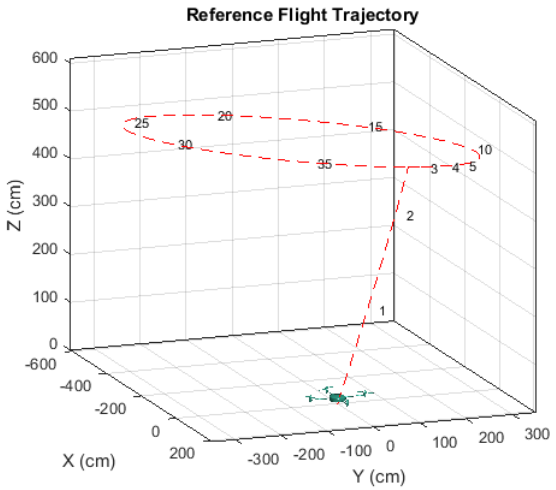

Figure 3. Reference Flight Trajectory and Time on the Spot

3.1.1. Low level controller with noise

Sensor noise is added to the feedback signal on the model as shown in Figure 4. Noise power is taken as 0.1 with 0.001 sampling time. Noise added simulation results for attitude controller is shown in Figure 5.

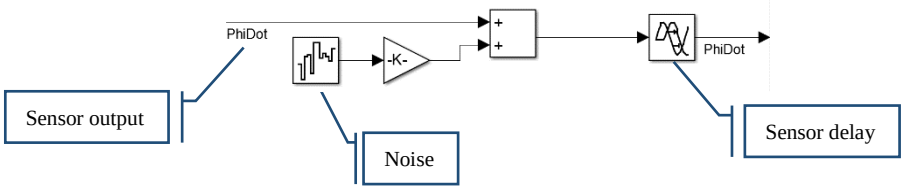

Figure 4. Noise and sensor delay application on the model.

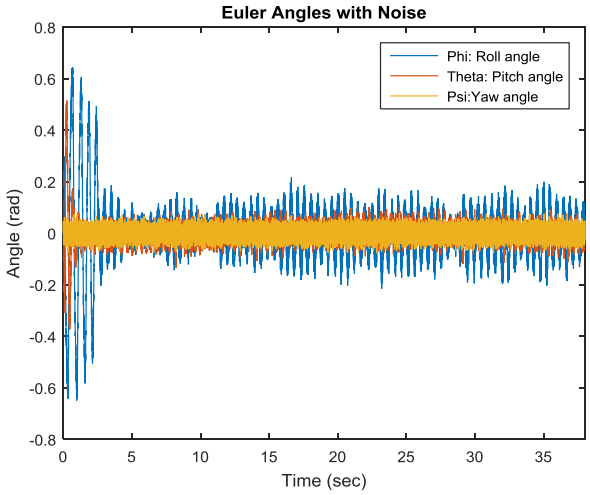

Figure 5. Attitude controller with noisy sensor feedback.

3.1.2. High level controller with noise

Sensor noise is added to the feedback signal on the model as shown in Figure 6. Noise power is taken as 0.1 with 0.001 sampling time. Noise added simulation results for position controller is shown in Figure 7.

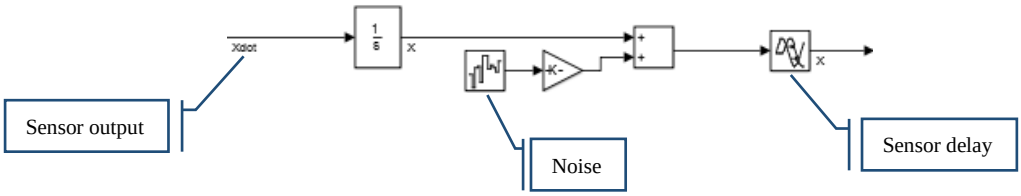

Figure 6. Noise and sensor delay application on the model.

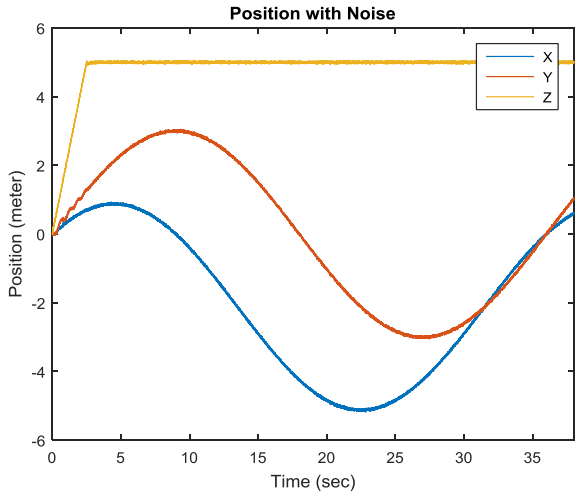

Figure 7. Position controller with noisy sensor feedback.

3.1.3. Low level controller with disturbance

Attitude controller's performance under disturbance is presented in Figure 8. Disturbance is applied to pitch angle twice on 10th and 25thsecond. Amplitude of the disturbance signal is chosen as 1. It can be seen that controller can stabilize the system in less than two seconds.

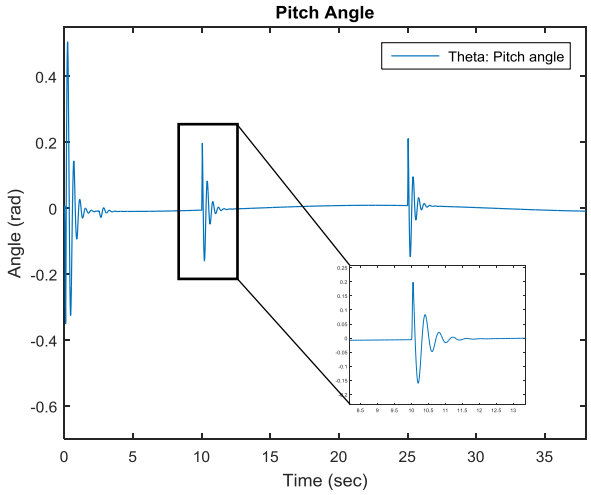

Figure 8. Attitude controller with disturbance.

3.1.4. High level controller with disturbance

Position controller's performance under disturbance is presented in Figure 9. Disturbance is applied to X positiontwice on 10th and 25th second. Amplitude of the disturbance signal is chosen as 1. It can be seen that controller can stabilize the system in less than one second.

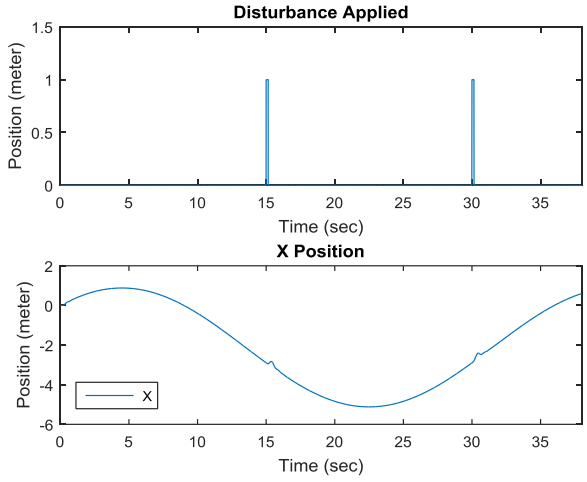

Figure 9. Position controller with disturbance.

1.1. Optimization

A cost function (CF) should be defined to apply these algorithms to the problem. In our case, cost function is chosen as following squared error fitness function,

$$CF = \frac{1}{3}\left[\left(\sum_{k=1}^{N} e_x(k)\right)^2 + \left(\sum_{k=1}^{N} e_y(k)\right)^2 + \left(\sum_{k=1}^{N} e_z(k)\right)^2\right] \quad (45)$$

where position error values are represented by e_x, e_y and e_z. To calculate such a cost function a reference flight trajectory is generated. That is shown in Figure 2. It can be seen where the quadcopter should be on a specific time. Time scale in the figure is given in seconds. All algorithms are executed on the same trajectory in order to satisfy the equality of comparison terms. Twelve parameters of the model are tuned in total. C parameters, presented at the end of Section II-B, C_7, C_{10} and C_{13} are belong to the sliding surfaces. C_8, C_9, C_{11}, C_{12}, C_{14} and C_{15} are belong to $\beta(x)$ functions. K parameters are belong to $tanh(\lambda x)$ functions. All parameters are optimized within the range of [0-10] due to hardware limitations.It is observed that population of 25 individuals can givesatisfactory results by means of time and cost function efficiency. Also 50 iterations are enough to observe that the algorithm reached a satisfactory result which was proven by means of convergence rates and converged values. Thus, for time and processing efficiency, all results presented in this paper were evaluated with 50 iterations. The parameters used for the algorithms are given in Table6. Under these conditions the quadcopter simulation run in accelerated mode for 40 seconds. After takeoff, the vehicle should settle on an altitude of 5 meters and follows an elliptic trajectory for a one complete round.

Table6. Parameters Used in Algorithms

ACO
Sample size: 40
Intensification factor: 0.5
Deviation-Distance ratio: 1
IWO
Initial population size: 10
Min / Max number of seeds: 0 / 10
Variance reduction exponent: 3
Standard deviation initial value: 0.8
Standard deviation final value: 0.001
FA
Light absorption coefficient: 1
Attraction coefficient base value: 2
Mutation coefficient: 0.2
Mutation coefficient damping ratio: 0.98
Uniform mutation range: 0.5

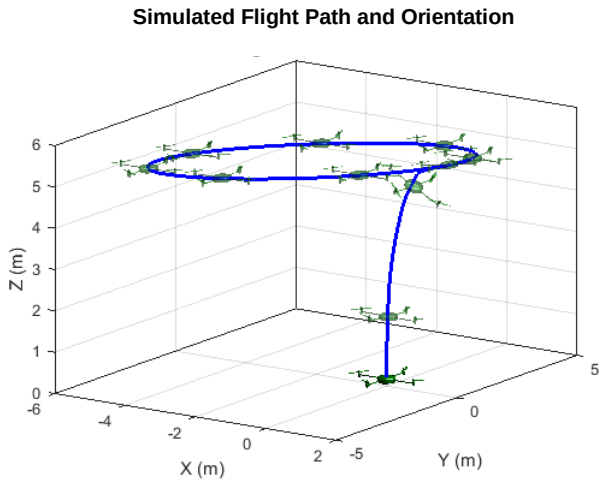

Figure 10. Simulated Flight Path of the Vehicle and its Orientation

2. Results

According to the results obtained all three algorithms displayed satisfactory performances. Simulated flight trajectory and vehicle's orientation can be seen on Fig 3. During the simulation position errors are decreased to 1-2 cm. This is achieved by optimizing control parameters with the presented algorithms. Comparison of convergence rate of algorithms is shown in Figure 11, where FA algorithm converged to the smallest value with the highest convergence speed. This makes it ideal to run it for small number of iterations.

All algorithms are performed under equal conditions. The difference between iteration time, total time spent and best costs achieved are shown in Table 8. FA has the best performance however it needs the highest processing power.

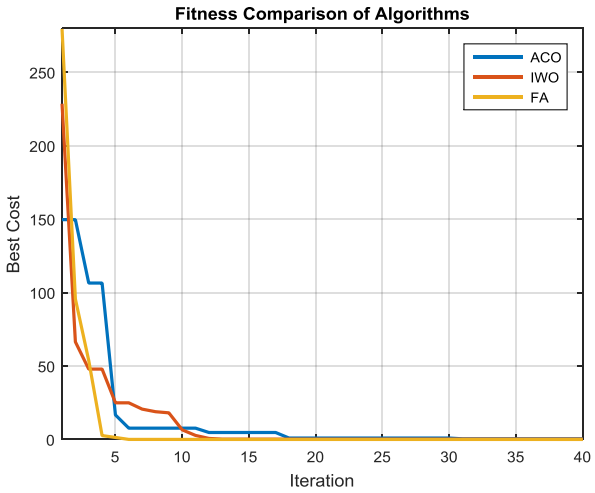

Figure 11. Fitness Comparison of the algorithms on SMC

Position errors with optimized parameters are shown in figures 12-14.

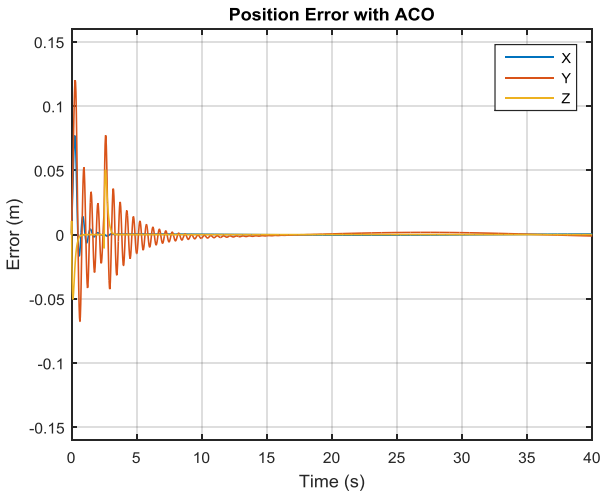

Figure 12. Position Error with ACO

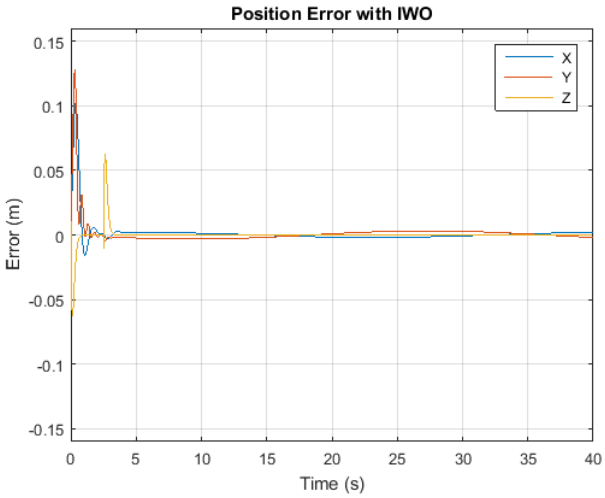

Figure 13. Position Error with IWO

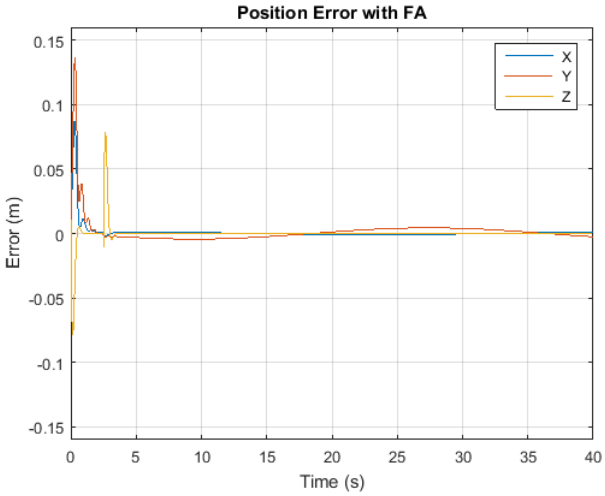

Figure 14. Position Error with FA

Table 7. Time and Best Cost Comparison of the Algorithms

	Average Time Per Iteration	Total Time	Best Cost
ACO	00:00:30	00:25:33	0.4303
IWO	00:01:03	00:53:19	0.2031
FA	00:03:36	03:00:44	0.0006

3. Conclusion

It is observed that using heuristic optimization algorithms for the quadcopter's parameter optimization problem provides satisfactory results. As seen in Figure12, Figure13and Figure14, position errors decrease to centimeter scale. This indicates that high precision control of quadcopter can be achieved by optimizing control parameters. In comparison of algorithms FA presented the best effectiveness. To run such an algorithm in real-time, it will need more computation power comparing to IWO and ACO. To overcome this problem, algorithm can be run on a separate processor at some reasonable time intervals. As seen in Figure 11, convergence rate of FA is good enough to run it for around five or six iterations. With a high computation power this may take small enough time to run it real time.

For future studies real-time optimization of the control parameters can be studied to provide high precision control under highly variable conditions like atmospheric changes (gusty wind, humidity changes, temperature changes etc.), near field explosions, getting shot and even electromagnetic warfare conditions.

4. References

A. Baldini, L. Ciabattoni, R. Felicetti, F. Ferracuti and A. Monteriù. (2017). Particle Swarm Optimization Based Sliding Mode Control Design: Application to a Quadrotor Vehicle. In *Applications of Sliding Mode Control in Science and Engineering* (pp. 143-169). Springer International Publishing.

Abraham, A., Grosan, C., & Ramos, V. (Eds.). (2006). *Stigmergic Optimization* (Vol. 31). Berlin: Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-540-34690-6

Bo Xing, Wen-Jing Gao. (2014). *Innovative Computational Intelligence A Rough Guide to 134 Clever Algorithms.* Switzerland: Springer International Publishing.

Bouabdallah, S. (2007, February). PhD. Thesis. *Design and control of quadrotors with application to autonomous flying*. Lausanne, Sweden: EPFL.

Dorigo, M. (1992). PhD. Thesis. *Optimization, learning and natural algorithms*. Milano, Italy: Politecnico di Milano.

F. Yacef, O. Bouhali, M. Hamerlain, and A. Rezoug. (October 29-31, 2013). PSO optimization of Integral Backstepping Controller for Quadrotor Attitude Stabilization. *Proceedings of the 3rd International Conference on Systems and Control.* Algiers, Algeria.

Gustavsson, K. (2015). Master Thesis. *UAV Pose Estimation using Sensor Fusion of Inertial, Sonar and Satellite Signals*. Uppsala: Uppsala University.

I. C. Dikmen, A. Arisoy and H. Temeltas. (2009). Attitude control of a quadrotor. *4th International Conference on Recent Advances in Space Technologies*, (pp. 722-727). Istanbul.

I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino. (2012). Biological invasion–inspired migration in distributed evolutionary algorithms. *Information Sciences, 207*, 50-65.

Lebao Li, Lingling Sun and Jie Jin. (2015). Survey of advances in control algorithms of quadrotor unmanned aerial vehicle. *IEEE 16th International Conference on Communication Technology (ICCT)*, (pp. 107-111). Hangzhou.

Marco Dorigo, Mauro Birattari, Thomas Stützle. (2006, December). Ant Colony Optimization. *IEEE Computational Intelligence Magazine, 1*(4), 28-39.

Mehrabian AR, Lucas C. (2006;). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics, 1*(4), 355–366.

Nanako Shigesada, Kohkichi Kawasaki. (1997). *Biological Invasions: Theory and Practice.* Tokyo: Oxford University Press.

P. D. Sheth and A. J. Umbarkar. (2015). Constrained Optimization Problems Solving Using Evolutionary Algorithms: A Review. *International Conference on Computational Intelligence and Communication Networks (CICN),* (pp. 1251-1257). Jabalpur.

S. Gupte, Paul Infant Teenu Mohandas and J. M. Conrad. (2012). A survey of quadrotor Unmanned Aerial Vehicles. *Proceedings of IEEE Southeastcon*, (pp. 1-6). Orlando, FL.

Samir Bouabdallah, Roland Siegwart. (2005). Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. *IEEE International Conference on Robotics and Automation (ICRA).* Barcelona, Spain: IEEE International.

Shibu Jose, Harminder Pal Singh, Daizy Rani Batish, Ravinder Kumar Kohli. (2013). *Invasive Plant Ecology.* Boca Raton: CRC Press .

Sundarapandian Vaidyanathan, C.-H. L. (Ed.). (2017). *Applications of Sliding Mode Control in Science and Engineering*. Cham, Switzerland: Springer International Publishing.

T. T. Mac, C. Copot, T. T. Duc and R. De Keyser. (2016). AR.Drone UAV control parameters tuning based on particle swarm optimization algorithm. *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, (pp. 1-6). Cluj-Napoca. doi:10.1109/AQTR.2016.7501380

Yang, X.-S. (2008). *Nature-Inspired Metaheuristic Algorithms.* Frome, UK: Luniver Press.