International Journal on
Advanced Science
Engineering
Information Technology

Vol.7 (2017) No. 4
ISSN: 2088-5334

Modelling and PSO Fine-tuned PID Control of Quadrotor UAV

Aminurrashid Noordin#1, Mohd Ariffanan Mohd Basri#2, Zaharuddin Mohamed#3 and
Amar Faiz Zainal Abidin*4

#Faculty of Electrical Engineering, Universiti Teknologi Malaysia, UTM Johor Bahru, 81310, Johor, Malaysia

*Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia
E-mail: 1aminurrashid@utem.edu.my, 2ariffanan@fke.utm.my, 3zahar@fke.utm.my, 4amarfaiz@utem.edu.my

***Abstract*—This paper describes nonlinear dynamics model of x-configuration quadrotor using Newton-Euler modelling technique. To stabilize quadrotor attitude (roll (ϕ), pitch (θ), yaw (ψ)) during hovering, a PID controller is proposed. There is individual PID controller for each roll, pitch, yaw and z where 12 parameters consist of kp, ki, and kd are fine-tuned using particle swarm optimization algorithms. From the simulation, the sum absolute error fitness function give the best optimize result where quadrotor achieve zero steady state error for hovering with 18.9% overshoot, and 4.42s settling time. Accordingly, for attitude stabilization, roll angle, pitch angle, and yaw angle converge to the set point, zero approximately with settling time 2.76s, 0.1s and 3.2s respectively.**

***Keywords*— UAV; PID; PSO: quadrotor; Newton-Euler.**

I. Introduction

Recently, due to advances in micro-electro mechanical technology (MEMs) [1][2][3] and rapid prototyping technology, research community and DIY hobbyist tend to focus on unmanned aerial vehicles (UAV) which is very promising vehicle for navigations, surveillances, and as well as educational purposes [4][5][6][7].

With simplest electronics and mechanical structures design, quadrotor UAV is an aerial vehicle that has capabilities in vertical take-off and landing (VTOL), omni-directional flying, and easy hovering performances in limited spaces always being considered in research.

The quadrotor is an under-actuated and dynamically unstable system which possess with complex behaviours. Many presented work in literatures use ‘+’ configuration and simplified model, where non-linear effect is neglected. Several literatures have mentioned of proportional-integral-derivative (PID) control a quadrotor [8][9][10] but using linearize model.

The main issues addressed in this paper is a fine-tuned PID controller for highly nonlinear quadrotor, x-configuration model focus on attitude stabilization during altitude control. Since there are 12 parameters to be tuned, a particle swarm optimization is chosen to amend control parameters; kp, ki, and kd to the best optimal values for the preferred control response.

II. Material and Method

A. Quadrotor Model

Quadrotor is a type of helicopter that can be controlled by varying the rotor speeds. It is an under-actuated, dynamic vehicle with four input forces and six output coordinates. Quadrotor, composed of four rotors with symmetrically arrangement where two diagonal motors (1 and 2) are running in the same direction whereas the others (3 and 4) in the other direction to eliminate the anti-torque [7][11][12]. As shown in Fig. 1, there are two basic types of quadrotor configurations; a plus configuration and a cross configuration [13][14][15].

Quadrotor is a helicopter category composed of four rotors with symmetrically structure in either ‘x’ mode configuration or ‘+’ mode configuration. To eliminate anti-torque force during maneuver, two diagonal motors (1 and 2) are set to run in the same direction (anti-clockwise) while (3 and 4) in the other direction (clockwise). This paper present ‘x’ mode configuration quadrotor as illustrated in Fig. 1, where the coordinate systems of two reference frames describe the dynamics of a quadrotor; an earth fixed initial reference frame, {E} and a body fixed reference frame {Q} located at the center of gravity (COG) of quadrotor body frame which is a rigid body in free motion with six degree of freedom (DOF) consist of three translational and three rotational.

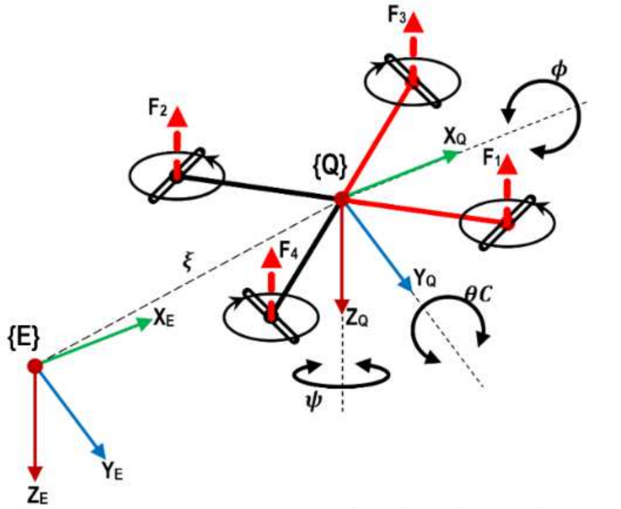

Fig. 1 Inertial coordinate systems and body-fixed frame for X configuration quadrotor

For the modelling, the following assumption is defined for simplification [16]:

1. The quadcopter is assumed as a rigid body.
2. The quadcopter's structure is assumed as symmetric with respect to the XY-axis.
3. The centre of mass and the origin of the body fixed frame are coinciding.
4. The propellers are considered as rigid; no blade flapping occurs.
5. The four propellers work under the same conditions at any time, meaning that thrust coefficient, $\boldsymbol{b}$ and reaction torque coefficient, $\boldsymbol{d}$ are the same for all propellers.
6. The generalized coordinates for the quadrotor based on Fig. 1 can be described as follow

$$\xi = [x, y, z]^T \in \mathbb{R}^3 \tag{1}$$

$$\eta = [\phi, \theta, \psi]^T \in \mathbb{R}^3 \tag{2}$$

where, vector ξ, Eq. (1) denotes the position of the quadrotor relative to inertial frame, vector η, Eq. (2) denotes the attitude of the quadrotor. The relation of body fixed reference frame, {Q} respect to earth fixed initial reference frame, {E} satisfy as $\{Q\}^T = R_T \times \{E\}^T$. Eq. (3) defines the rotation matrix $\boldsymbol{R}_T$, where, S and C stands for trigonometric operators 'sin' and 'cos' respectively.

$$\boldsymbol{R}_T - \begin{bmatrix} C\psi C\theta & -S\psi C\phi + C\psi S\theta S\phi & S\psi S\phi + C\psi C\phi S\theta \\ S\psi C\theta & C\psi C\phi + S\psi S\theta S\phi & -C\psi S\phi + C\phi S\psi S\theta \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \tag{3}$$

From general Newton-Euler translational and rotational dynamics, the quadrotor dynamics, is described as Eq. (4) and Eq. (5), where, g is gravitational coefficient, E_z vector matrix of z-axis defined as $[0\ 0\ 1]^T$, U_1 is total thrust force generated by four rotors. I is the moments of inertia for the quadrotor, a diagonal matrix 3-by-3 and defined as $I = diagonal[I_{xx}\ I_{yy}\ I_{zz}]^T$. J_r is rotor inertia, $\boldsymbol{\Omega}_d$ is total rotor speeds generated from the two pairs of rotor. U_2, U_3 and U_4 are total torque, τ related to quadrotor as of total summation of Coriolis torque, τ_c, and Gyroscopic torque, $\boldsymbol{\tau_g}$ and quadrotor body frame torque, $\boldsymbol{\tau_u}$.

$$m\ddot{\xi} = mgE_z + U_1 R_T E_z \tag{3}$$

$$I\ddot{\eta} = -\eta \times I\eta - J_r(\eta \times E_z)\Omega_d + [U_2\ U_3\ U_4]^T \tag{4}$$

Finally, from Eq. (3) and Eq. (4), the final equation for quadrotor translation dynamics and rotational dynamics can be formulated as

$$\begin{aligned} \ddot{x} &= \frac{U_1}{m}(S\psi S\phi + C\psi S\theta C\phi) \\ \ddot{y} &= \frac{U_1}{m}(-C\psi S\phi + S\psi S\theta C\phi) \\ \ddot{z} &= g + \frac{U_1}{m}(C\theta C\phi) \end{aligned} \tag{5}$$

$$\begin{aligned} \ddot{\phi} &= \left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right)\dot{\psi}\dot{\theta} - \left(\frac{J_r\Omega_d}{I_{xx}}\right)\dot{\theta} + \left(\frac{l}{I_{xx}}\right)U_2 \\ \ddot{\theta} &= \left(\frac{I_{zz}\quad I_{xx}}{I_{yy}}\right)\dot{\psi}\dot{\phi} + \left(\frac{J_r\Omega_d}{I_{yy}}\right)\dot{\phi} + \left(\frac{l}{I_{yy}}\right)U_3 \\ \ddot{\psi} &= \left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right)\dot{\theta}\dot{\phi} + \left(\frac{1}{I_{zz}}\right)U_4 \end{aligned} \tag{6}$$

Table 1 shows x-configuration quadrotor's parameter used in this research which is obtained through system identification. Fig. 2 shows the hardware configuration of the laboratory quadrotor UAV system used in this research.

TABLE I
X-QUADROTOR PARAMETER

Specification	Parameter	Unit	Value
Quadrotor mass	m	kg	1.033
Lateral moment arm	l	m	0.225
Thrust coefficient	b	Ns^2	2.8625x10-7
Drag coefficient	d	Nms^2	4.4212x10-10
Rolling moment of inertia	I_{xx}	kgm^2	0.0183
Pitching moment of inertia	I_{yy}	kgm^2	0.0183
Yawing moment of inertia	I_{zz}	kgm^2	0.0385

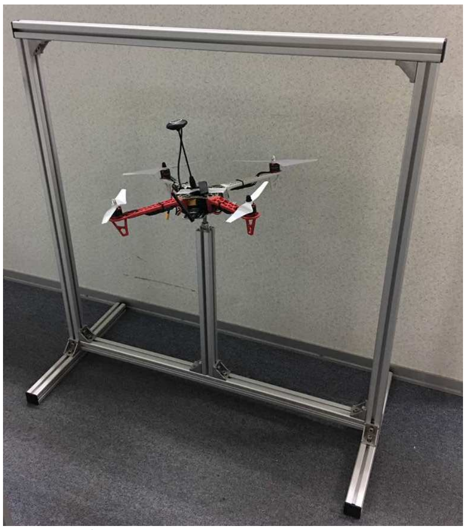

Fig. 2 x-configuration Laboratory Quadrotor UAV System

B. PID Control

PID controller is a classical controller that has proven to be robust and tremendously beneficial in many linear or non-linear applications. The PID design are pointed out in many references, such as [10][17][18][19][20]. The controller attempts to minimize the error over time by adjustment of a control variable u(t). The mathematical representation of PID controller is given as

$$u(t) = k_p e(t) + k_i \int e(t) + k_d \frac{d}{dt} e(t) \tag{7}$$

where, $\boldsymbol{u(t)}$ is the input signal and the error signal $\boldsymbol{e(t)}$ is defined as

$$e(t) = desired\_input(t) - actual\_output(t) \tag{8}$$

On the other hand, a PID controller continuously calculates an error value $\boldsymbol{e(t)}$ then applies a correction based on proportional, integral, and derivative terms as shown in Fig. 3 (a). The proposed controller for attitude and altitude stabilization in this simulation is shown as Fig. 3 (b), where each of references input have its own PID controller to control highly nonlinear quadrotor model.

The main objective of this simulation is to design controller which makes x-quadrotor's attitude stabilized during hovering or taking off. The PID parameters are tuned by particle swarm optimization (PSO) method and Sum Square Error (SSE) is chosen as it fitness functions.

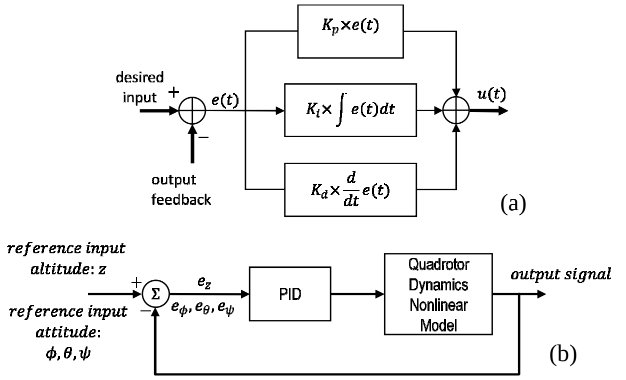

Fig. 3 PID in Nonlinear Quadrotor's Attitude and Altitude Control a) PID Structure b) Quadrotor Control Block Diagram

C. Particle Swarm Optimization (PSO)

PSO was firstly introduced by Kennedy and Eberhart in 1995, based on fish schooling and bird flocking movement behaviour [21][22][23][24]. In PSO algorithm, particles will move randomly in any possible direction towards it global best and current best location using its present velocity which later then being updated randomly. Hence, the used of real-number randomness and global communication between particles will increase efficiency in swarm.

Let x_i and v_i is the position vector and velocity vector for particle i, respectively. There are 12 parameters of PID controller to be tuned, therefore the proposed model of this application is as shown in Eq. (9).

$$x_i = [k_p^z \; k_i^z \; k_d^z \; k_p^\phi \; k_i^\phi \; k_d^\phi \; k_p^\theta \; k_i^\theta \; k_d^\theta \; k_p^\psi \; k_i^\psi \; k_d^\psi] \tag{9}$$

For this simulation, the swarm particles, x_i and particles' velocity, v_i are initially set randomly using Eq. (10) and Eq. (11), respectively, where l_b (lower boundary) and u_b (upper boundary) is limitation space or regions set for the particles, r_1, and r_2 is random real-number between [0 1].

$$x_i = l_b + (u_b - l_b) \times r_1 \tag{10}$$

$$v_i = r_2 \tag{11}$$

In this application, the objective is to find minimum error of the fitness function. There are four fitness function which are as shown in Eq. (12) – Eq. (15).

$$f_1(x) = Fitness_1 = SSE_z = \sum e_z^2(t)dt \tag{12}$$
$$f_2(x) = Fitness_2 = SSE_\varphi = \sum e_\phi^2(t)dt \tag{13}$$
$$f_3(x) = Fitness_3 = SSE_\theta = \sum e_\theta^2(t)dt \tag{14}$$
$$f_4(x) = Fitness_4 = SSE_\psi = \sum e_\psi^2(t)dt \tag{15}$$

Where the fitness of the particle is evaluated based on the responses by sum square error (SSE). The personal best of an agent is updated based on the mathematical statement in Eq. (16) as

$$x_{t+1}^* - \begin{cases} x_t^i, f_1(x_t^i) < f_1(X_t^*) \; AND \; f_2(x_t^i) < f_2(X_t^*) \; AND.. \\ ..f_3(x_t^i) < f_3(X_t^*) \; AND \; f_4(x_t^i) < f_4(X_t^*) \\ x_t^*, otherwise \end{cases} \tag{16}$$

While the global best is updated based on the mathematical statement in Eq. (17) as

$$g_{t+1}^* = \begin{cases} x_t^i, f_1(x_t^i) < f_1(g_t^*) \; AND \; f_2(x_t^i) < f_2(g_t^*) \; AND.. \\ ..f_3(x_t^i) < f_3(g_t^*) \; AND \; f_4(x_t^i) < f_4(g_t^*) \\ g_t^*, otherwise \end{cases} \tag{17}$$

The new velocity vector [22] for this PSO is determined by the following formula

$$v_i^{t+1} = w_i \times v_i + \alpha \times r_3 \times (g^* - x_i^t) + \beta \times r_4 \times (x^* - x_i^t) \tag{18}$$

where, an inertia function, $\boldsymbol{w_i}$ as Eq. (19) is used to update the velocity, $\boldsymbol{v_i^{t+1}}$ in every new iteration $(t+1)$ per Eq. (20) [25][26], $\boldsymbol{r_3}$ and $\boldsymbol{r_4}$ is random real-number between [0 1], the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the social coefficient and personal coefficient which is constant and typically set as two [22].

$$w_i = 0.4 + 0.5 \times \frac{N_i - i}{N_i} \tag{19}$$

The new position then is updated by

$$x_i^{t+1} - x_i + v_i^{t+1} \tag{20}$$

Stopping criteria used in this application is maximum iteration where, once the iteration reached the maximum value set, the simulation will stop and the best result obtained is display.

III. Result and Discussion

In this simulation, initially, ϕ, θ, and ψ are set at 0.2 radian. The simulation times is set from 0 to 20s and updates every 100Hz. Desired ϕ, θ, and ψ are set to 0 radian, while desired z is set at 10 meters. Number of iteration is set fixed at 100. The fitness function or so-called objective functions is set all high priority using logical AND (&&) in MATLAB programming to have better performance of very nonlinear quadrotor system compared to [24] which analyses each one separately. The data is recorded few times with difference number of particles being set. Overall performances are measured using sum square error (SSE). Table 2 shows the first setting with SSE fitness function for z-axis, roll (ϕ), pitch (θ), and yaw (ψ). Fig. 4 shows the z-axis performance after each optimizing process. Based on this figure, the suitable PID parameter for altitude performance of quadrotor during hovering is best at Case 6 with percentage of Overshoot (%OS) at 0.86, settling time, Ts at 0.83s and final value at 10.09 as stated in Table 3. Fig. 5, Fig. 6, and Fig. 7 show the roll (ϕ), pitch (θ), and yaw (ψ) angle after each optimizing process, respectively. Overall, from the figures and per Table 3, it clearly shows that performance of SSE on Case 1 is preferable where, the attitude control converges faster to the set point, 0 at 0.42 seconds for roll, and 0.70 seconds for pitch while the altitude settling at 10.12 seconds even though with 1.2% error.

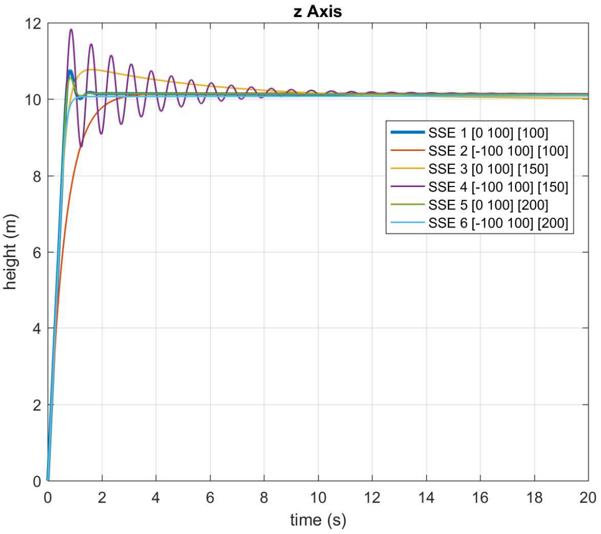

Fig. 4 z-axis performance after optimizing using SSE fitness function

The simulation is then being continued with difference mix fitness function, difference number of particles, difference lower boundary. But still overall performances are measured using sum square error (SSE). The chosen results are tabulated as shown in Table 4. Result from Table 4 and Table 5, clearly shows that fitness function, Sum Absolute Error (SAE) and the combination of fitness function Mean Absolute Error (MAE) and fitness function Sum Absolute Error (SAE) give zero steady state error for hovering which means quadrotor can reach the desired position (10 meters) compared to others with 0.4% to 1.4% error. MAE + SAE have better settling time, Ts at 3.06s compared to SAE with 4.42s but the percentage of Overshoot (OS%) MAE+SAE is nearly double of SAE. For attitude performances, it clearly shows that fitness function SAE give fastest settling time, 2.75s for roll (ϕ), 0.10s for pitch (θ), and 3.19s for yaw (ψ) angle. Fig. 8, Fig. 9, Fig. 10 and Fig. 11 show the comparison performances between mix fitness function for altitude and attitude for quadrotor during hovering, respectively.

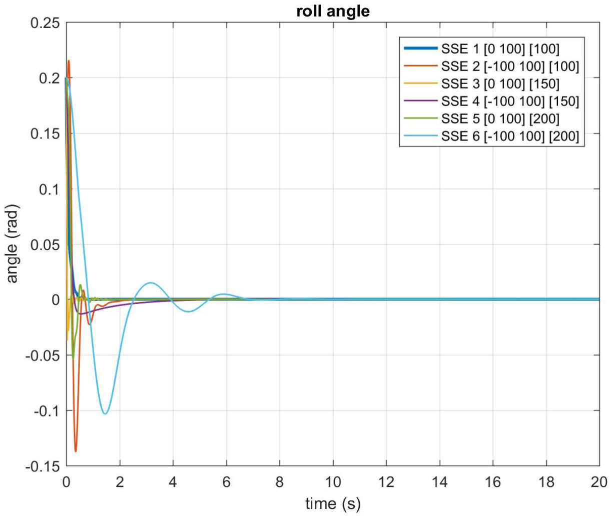

Fig. 5 Roll, ϕ performance after optimizing using SSE fitness function

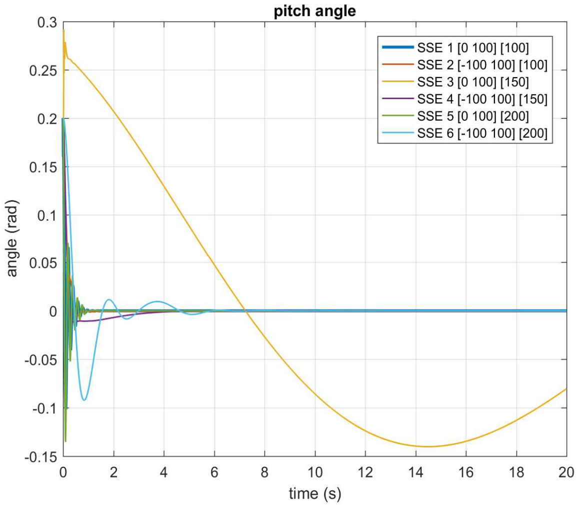

Fig. 6 Pitch, θ performance after optimizing using SSE fitness function

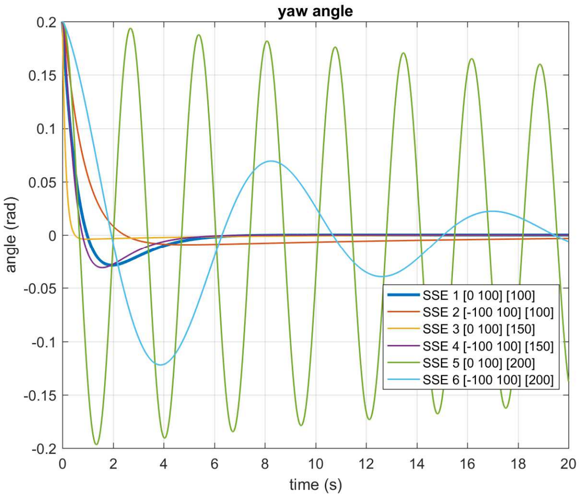

Fig. 7 Yaw, ψ performance after optimizing using SSE fitness function

TABLE II
SUM SQUARE ERROR FITNESS FUNCTION

	Case	**1**	**2**	**3**	**4**	**5**	**6**
z	**Ts**	1.04	2.05	8.74	10.58	1.06	0.83
	OS	7.59	1.49	7.73	18.37	5.62	0.86
	Final Value	10.12	10.14	10.02	10.14	10.12	10.09
roll	**Ts**	0.42	1.54	0.30	2.52	0.61	6.08
	OS	inf					
	Final Value	0.00	0.00	0.00	0.00	0.00	0.00
pitch	**Ts**	0.70	0.66	NaN	2.73	0.85	4.35
	OS	inf					
	Final Value	0.00	0.00	-0.08	0.00	0.00	0.00
yaw	**Ts**	5.17	16.96	0.41	4.50	NaN	NaN
	OS	inf					
	Final Value	0.00	0.00	0.00	0.00	0.00	0.00
Ts = settling time, OS = overshoot							

TABLE III
SYSTEM PERFORMANCES FOR SSE

Case	**Fitness Function**	**SSE**	**SSE**	**SSE**	**SSE**

Case 1
{lb: 0}
{ub: 100}
{Np: 100}
		z	ϕ	θ	ψ

	K_p	91.34	55.30	88.26	21.72
	K_i	0.94	0.58	0.10	11.66
	K_d	8.15	6.62	0.76	10.94
	SSE Performance	0.60	0.00	0.00	0.00

Case 2
{lb: -100}
{ub: 100}
{Np: 100}
		z	ϕ	θ	ψ

	K_p	86.07	14.48	36.88	0.79
	K_i	0.49	24.38	0.65	0.06
	K_d	53.24	0.80	4.70	0.66
	SSE Performance	11.52	0.00	0.00	0.00

Case 3
{lb: 0}
{ub: 100}
{Np: 150}
		z	ϕ	θ	ψ

	K_p	57.71	92.06	0.99	65.53
	K_i	10.62	11.37	0.40	11.27
	K_d	17.49	0.39	11.05	8.32
	SSE Performance	0.72	0.00	0.09	0.00

Case 4
{lb: -100}
{ub: 100}
{Np: 150}
		z	ϕ	0	ψ

	K_p	74.33	28.96	0.80	01.34
	K_i	1.18	17.48	0.29	0.78
	K_d	0.76	3.31	0.76	0.58
	SSE Performance	0.01	0.00	0.00	0.00

Case 5
{lb: 0}
{ub: 100}
{Np: 200}
		z	ϕ	θ	ψ

	K_p	91.84	14.38	84.87	0.38
	K_i	0.98	0.29	0.05	64.72
	K_d	10.14	0.94	0.67	11.92
	SSE Performance	0.64	0.00	0.00	0.00

Case 6
{lb: -100}
{ub: 100}
{Np: 200}
		z	ϕ	θ	ψ

	K_p	84.39	0.78	1.65	0.28
	K_i	-1.33	0.73	1.17	0.54
	K_d	17.65	0.27	0.43	1.03
	SSE Performance	4.65	0.00	0.00	0.00
lb = lower boundary, ub = upper boundary, N p= number of particles					

TABLE IV
PERFORMANCE WITH DIFFERENCE FITNESS FUNCTION COMBINATION

SSE
{lb: -100}
{ub: 100}
{Np: 150}
	Fitness Function	**SSE**	**SSE**	**SSE**	**SSE**

		z	ϕ	θ	ψ
	K_p	74.33	28.96	0.80	01.34
	K_i	1.18	17.48	0.29	0.78
	K_d	0.76	3.31	0.76	0.58
	SSE Performance	0.01	0.00	0.00	0.00

SAE
{lb: 0}
{ub: 100}
{Np: 100}
	Fitness Function	**SAE**	**SAE**	**SAE**	**SAE**

		z	ϕ	0	ψ
	K_p	90.32	94.82	98.78	40.51
	K_i	55.73	55.92	25.19	39.83
	K_d	21.19	16.87	7.69	15.74
	SSE Performance	0.033	0.00	0.00	0.00

SAE + SSE
{lb: 0}
{ub: 100}
{Np: 150}
	Fitness Function	**SAE**	**SSE**	**SSE**	**SSE**

		z	ϕ	θ	ψ
	K_p	79.45	93.89	13.66	10.29
	K_i	3.22	32.01	0.27	58.44
	K_d	1.90	12.26	3.77	17.94
	SSE Performance	0.05	0.00	0.00	0.00

MAE + SSE
{lb: -100}
{ub: 100}
{Np: 350}
	Fitness Function	**MAE**	**SSE**	**SSE**	**SSE**

		z	ϕ	0	ψ
	K_p	98.46	38.46	24.29	5.10
	K_i	-0.86	0.56	35.25	0.85
	K_d	79.74	34.31	10.55	2.23
	SSE Performance	54.22	0.02	0.00	0.00

MAE + SAE
{lb: 0}
{ub: 100}
{Np: 1050}
	Fitness Function	**MAE**	**SAE**	**SAE**	**SAE**

		z	ϕ	θ	ψ
	K_p	87.36	58.17	46.43	0.09
	K_i	94.11	0.35	0.20	0.04
	K_d	21.18	0.60	0.96	0.24
	SSE Performance	0.01	0.00	0.00	0.00
lb = lower boundary, ub = upper boundary, N p= number of particles					

TABLE V
SYSTEM PERFORMANCES FOR MIX FITNESS FUNCTIONS

		SSE 4	***SAE**	**SAE + SSE**	**MAE + SSE**	**MAE + SAE**
z	Ts	10.58	4.42	7.43	3.00	3.06
	OS	18.37	18.93	16.93	0.39	30.20
	Final Value	10.14	10.00	10.12	10.04	10.00
roll	Ts	2.52	2.75	0.33	3.06	0.96
	OS	inf				
	Final Value	0.00	0.00	0.00	0.00	0.00
pitch	Ts	2.73	0.10	0.75	2.65	0.68
	OS	inf				
	Final Value	0.00	0.00	0.00	0.00	0.00
yaw	Ts	4.50	3.19	12.66	8.53	17.66
	OS	inf				
	Final Value	0.00	0.00	0.00	0.00	0.00
Ts = settling time, OS = overshoot						

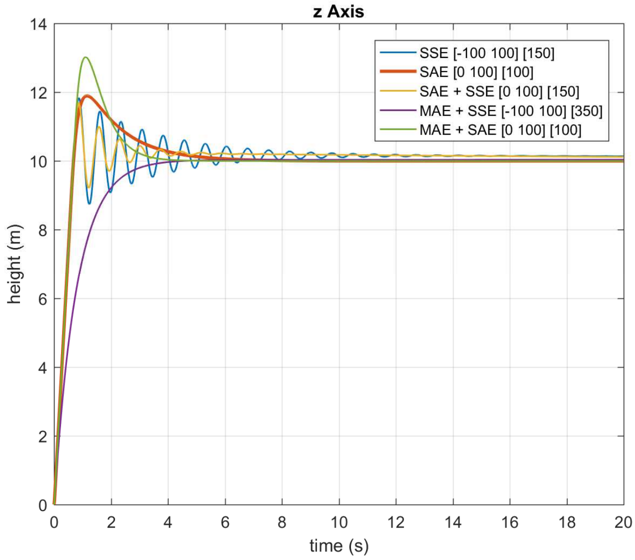

Fig. 8 z-axis performance after optimizing using mix fitness function

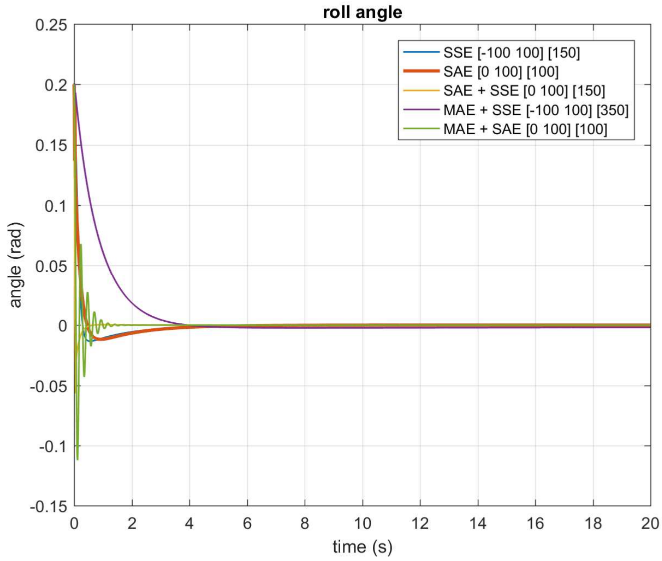

Fig. 9 Roll, ϕ angle performance after optimizing using mix fitness function

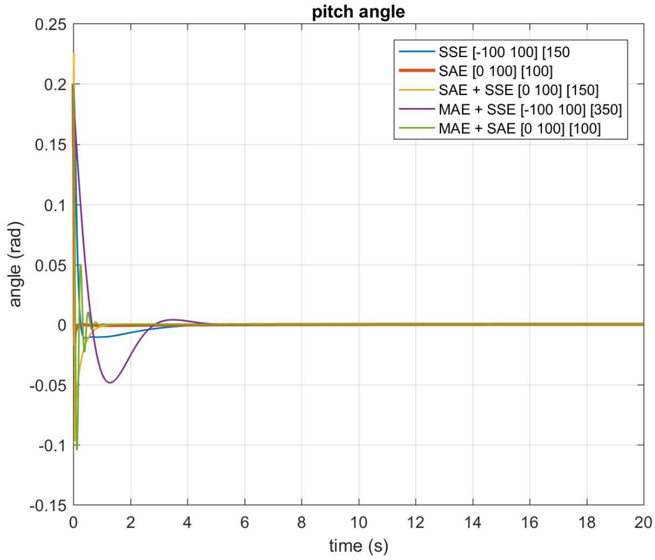

Fig. 10 Pitch, θ angle performance after optimizing using mix fitness function

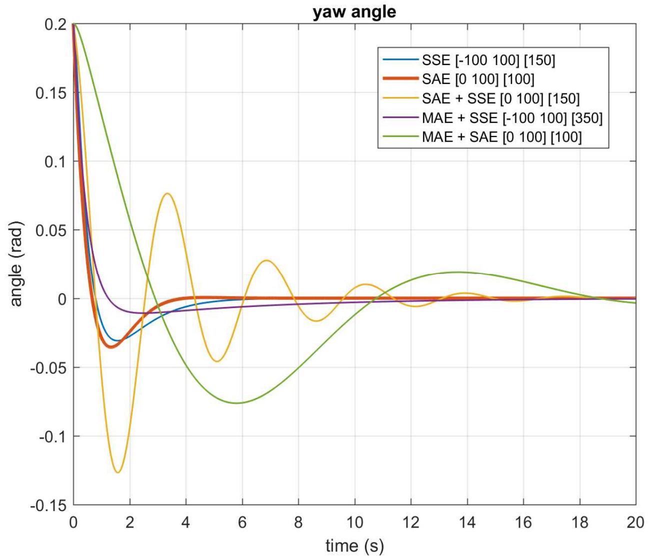

Fig.11 Yaw, ψ angle performance after optimizing using mix fitness function

IV. CONCLUSIONS

In this paper, a fined-tuned PID control of quadrotor UAV using PSO algorithm is proposed. The quadrotor used in this simulation is x-configuration types and a Newton-Euler modelling is used to describe quadrotor nonlinear dynamics model. A standard accelerate PSO is used in tuning the PID parameter where four type fitness functions; sum square error, sum absolute error; sum absolute error + sum square error, mean absolute error + sum square error, and mean absolute error + sum absolute error, has been tested to see quadrotor performances during attitude and altitude control for the desired set point. The obtained result shows that sum absolute error fitness function gives the best PID parameters with lower overshoot, better settling time and reach set point value compared to others. In the other hands, the higher the number of particles, the longer algorithm takes to complete it iteration but the result sometimes is not even better from the previous run.

ACKNOWLEDGMENT

The authors would like to thank Universiti Teknologi Malaysia Malaysia (UTM) under the Fundamental Research Grant Scheme (R.J130000.7823.4F761), Research University Grant (Q.J130000.2523.15H39), Universiti Teknikal Malaysia Melaka (UTeM), and Ministry of Higher Education for supporting this research.

REFERENCES

[1] J. Chang, J. Cieslak, A. Zolghadri, J. Dávila, and J. Zhou, “Quadrotor attitude estimation with gyroscope bias reconstruction capabilities,” *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 260–265, 2016.

[2] V. Kriz and P. Gabrlik, “UranusLink-Communication protocol for UAV with small overhead and encryption ability,” in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2015, vol. 48, no. 4, pp. 474–479.

[3] P. J. Bristeau, F. Callou, D. Vissière, and N. Petit, “The Navigation and Control technology inside the AR.Drone micro UAV,” in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2011, vol. 18, no. PART 1, pp. 1477–1484.

[4] Z. R. Mahayuddin, H. M. Jais, and H. Arshad, “Comparison of Human Pilot (Remote) Control Systems in Multirotor Unmanned

Aerial Vehicle Navigation," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 1, pp. 132–138, 2017.

[5] J. Ajmera and V. Sankaranarayanan, "Point-to-point control of a quadrotor: Theory and experiment," in *IFAC-PapersOnLine*, 2016, vol. 49, no. 1, pp. 401–406.

[6] J. Bazin, T. Fields, and A. J. Smith, "Feasibility of In-Flight Quadrotor Individual Motor Thrust Measurements," *AIAA Atmos. Flight Mech. Conf.*, no. January, pp. 1–12, 2016.

[7] D. B B V L and P. Singh, "A survey on design and development of an unmanned aerial vehicle (quadcopter)," *Int. J. Intell. Unmanned Syst.*, vol. 4, no. 2, pp. 70–106, 2016.

[8] A. L. L. Salih, M. Moghavvemi, H. a. F. a F. Mohamed, and K. S. S. Gaeid, "Modelling and PID controller design for a quadrotor unmanned air vehicle," *IEEE Int. Conf. Autom. Qual. Test. Robot.*, vol. 1, pp. 1–5, 2010.

[9] R. García, F. R. Rubio, and M. G. Ortega, "Robust PID Control of the Quadrotor Helicopter," *IFAC Conf. Adv. PID Control*, 2012.

[10] F. Goodarzi, D. Lee, and T. Lee, "Geometric nonlinear PID control of a quadrotor UAV on SE (3)," *Control Conf. (ECC), 2013 Eur.*, pp. 3845–3850, 2013.

[11] J. Ghommam, L. F. Luque-Vega, B. Castillo-Toledo, and M. Saad, "Three-dimensional distributed tracking control for multiple quadrotor helicopters," *J. Franklin Inst.*, vol. 353, no. 10, pp. 2344–2372, 2016.

[12] D. Domingos, G. Camargo, and F. Gomide, "Autonomous Fuzzy Control and Navigation of Quadcopters," *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 73–78, 2016.

[13] S. Norouzi Ghazbi, Y. Aghli, M. Alimohammadi, and A. A. Akbari, "Quadrotors unmanned aerial vehicles: A review," *Int. J. Smart Sens. Intell. Syst.*, vol. 9, no. 1, pp. 309–333, 2016.

[14] X. Zhang, X. Li, K. Wang, and Y. Lu, "A survey of modelling and identification of quadrotor robot," *Abstract and Applied Analysis*, vol. 2014. 2014.

[15] A. R. Partovi, A. Zong, Y. Kevin, H. Lin, and B. M. Chen, "Development of a Cross Style Quadrotor," *AIAA Guid. Navig. Control Conf.*, no. August, pp. 1–23, 2012.

[16] M. A. Mohd Basri, A. R. Husain, and K. A. Danapalasingam, "Enhanced Backstepping Controller Design with Application to Autonomous Quadrotor Unmanned Aerial Vehicle," *J. Intell. Robot. Syst. Theory Appl.*, vol. 79, no. 2, pp. 295–321, 2014.

[17] J. Li and Y. Li, "Dynamic analysis and PID control for a quadrotor," *2011 IEEE Int. Conf. Mechatronics Autom.*, pp. 573–578, 2011.

[18] S. Liu, Z. Hou, and J. Zheng, "Attitude adjustment of quadrotor aircraft platform via a data-driven model free adaptive control cascaded with intelligent PID," *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, pp. 4971–4976, 2016.

[19] B. Kada, *A new methodology to design sliding-PID controllers: Application to missile flight control system*, vol. 2, no. PART 1. IFAC, 2012.

[20] M. Nguyen Duc, T. N. Trong, and Y. S. Xuan, "The quadrotor MAV system using PID control," *2015 IEEE Int. Conf. Mechatronics Autom. ICMA 2015*, pp. 506–510, 2015.

[21] M. Shatnawi, M. F. Nasrudin, and S. Sahran, "A New Initialization Technique in Polar Coordinates for Particle Swarm Optimization and Polar PSO," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 1, pp. 242–249, 2017.

[22] X. S. Yang, *Engineering Optimization An Introduction with Metaheuristic Applications*. John Wiley & Sons, 2010.

[23] H.-K. Tran and J.-S. Chiou, "PSO-Based Algorithm Applied to Quadcopter Micro Air Vehicle Controller Design," *Micromachines*, vol. 7, no. 9, p. 168, 2016.

[24] T. T. Mac, C. Copot, T. T. Duc, and R. De Keyser, "AR . Drone UAV control parameters tuning based on particle swarm optimization algorithm," *IEEE Int. Conf. Autom. Qual. Testing, Robot. (AQTR), Cluj-Napoca*, pp. 19–21, 2016.

[25] A. Adam, A. F. Z. Abidin, Z. Ibrahim, A. R. Husain, Z. M. Yusof, and I. Ibrahim, "A Particle Swarm Optimization Approach to Robotic Drill Route Optimization," *2010 Fourth Asia Int. Conf. Math. Model. Comput. Simul.*, pp. 60–64, 2010.

[26] M. S. Muhammad, K. V. Selvan, S. M. W. Masra, Z. Ibrahim, and A. F. Z. Abidin, "An Improved Binary Particle Swarm Optimization Algorithm for DNA Encoding Enhancement," *2011 IEEE Symp. Swarm Intell. SIS*, pp. 60–67, 2011.