Journal of Mechatronics, Electrical Power, and Vehicular Technology 11 (2020) 1-10

Journal of Mechatronics, Electrical Power, and Vehicular Technology

e-ISSN: 2088-6985
p-ISSN: 2087-3379

www.mevjournal.com

Swarm control of an unmanned quadrotor model with LQR weighting matrix optimization using genetic algorithm

Endra Joelianto [a, c, *], Daniel Christian [b], Agus Samsi [c]

[a] *Instrumentation and Control Research Group - Institut Teknologi Bandung*
Jl. Ganesha No. 10, Bandung, 40132, Indonesia

[b] *Externship Researcher of Engineering Physics Study Program - Institut Teknologi Bandung*
Jl. Ganesha No. 10, Bandung, 40132, Indonesia

[c] *Engineering Physics Study Program - Institut Teknologi Bandung*
Jl. Ganesha No. 10, Bandung, 40132, Indonesia

Received 30 April 2020; accepted 21 June 2020; Published online 30 July 2020

Abstract

Unmanned aerial vehicle (UAV) quadrotors have developed rapidly and continue to advance together with the development of new supporting technologies. However, the use of one quadrotor has many obstacles and compromises the ability of a UAV to complete complex missions that require the cooperation of more than one quadrotor. In nature, one interesting phenomenon is the behaviour of several organisms to always move in flocks (swarm), which allows them to find food more quickly and sustain life compared with when they move independently. In this paper, the swarm behaviour is applied to drive a system consisting of six UAV quadrotors as agents for flocking while tracking a swarm trajectory. The swarm control system is expected to minimize the objective function of the energy used and tracking errors. The considered swarm control system consists of two levels. The first higher level is a proportional – derivative type controller that produces the swarm trajectory to be followed by UAV quadrotor agents in swarming. In the second lower level, a linear quadratic regulator (LQR) is used by each UAV quadrotor agent to follow a tracking path well with the minimal objective function. A genetic algorithm is applied to find the optimal LQR weighting matrices as it is able to solve complex optimization problems. Simulation results indicate that the quadrotors' tracking performance improved by 36.00 %, whereas their swarming performance improved by 17.17 %.

Keywords: unmanned aerial vehicle (UAV); quadrotor model; swarm model; proportional – derivative (PD) controller; linear quadratic regulator (LQR); optimization problems; genetic algorithm.

I. Introduction

Unmanned aerial vehicles (UAVs) are a type of aircraft that continues to develop because of its ability to fly without a human pilot. With the development of technology, especially for unmanned aircraft driven by propellers, the interest of researchers to acquire air vehicles with four rotors (quadrotor) is increasing [1][2][3]. The potential use of UAV quadrotors is progressing rapidly and extensively, ranging from simple flying to performing difficult tasks such as carrying equipment to places that are not easily accessible or are dangerous. For example, a quadrotor can be used to bring first-aid equipment to victims who are in locations that are difficult to reach, to find and notify about the location of earthquake victims in a building, and to take aerial photographs of areas that are difficult to visit.

A UAV quadrotor has several advantages over conventional propeller planes, which are only driven by one or two rotors. The propeller on the opposite side of a quadrotor rotates in the opposite direction so that the gyroscopic effect, torque, and moment on the axis of the plane tend to be more balanced [1]. As a result, quadrotor aircraft can maneuver better and can approach obstacles without having to fear to crash [2]. Moreover, the quadrotor's ability to fly vertically and float makes this type of aircraft very suitable for flying in areas that have many obstacles.

Research on quadrotors and how to fly them has received considerable attention in many years. Bresciani [3] in his thesis has modelled a quadrotor

* Corresponding Author. Tel: +62-22-2504424; Fax: +62-22-2506281
E-mail address: ejoel@tf.itb.ac.id

doi: https://dx.doi.org/10.14203/j.mev.2020.v11.1-10

by explaining the influence of aerodynamic forces acting on it and modelling its components. Sudiyanto *et al.* [1] have developed quadrotor modelling by using the first-principle approach. One of the problems in UAV quadrotors is the limited payload available, as this depends on lift force of the rotors. In many practical applications, the payload problem can be solved by using many quadrotors [4]. Advances in UAV technologies and improvement of sensor capabilities have led to group operation of quadrotors, which is important for applications that require large coverage areas such as search and rescue missions, inspection and detection, and surveillance reconnaissance [5].

Swarming is a group behaviour that is found in nature. This group behaviour can be found in organisms that live in groups such as bacteria, birds, fish, and ants. Swarm behaviour can arise through various mechanisms that are usually out of the instinct of the organism [6]. Several studies on models and controls for moving together have also been carried out, on mobile robots [7][8][9], autonomous helicopter [10][11], and quadrotor [12][13][14]. In [15], a modelled multi-agent swarm movement of a system to be able to follow the desired path has been investigated.

Coordination of movements between two or more quadrotors is needed to optimize the work to be performed by the quadrotors. Therefore, the swarm coordination of quadrotors is a very interesting topic to study. Research on quadrotor swarm has been carried out in previous studies [16][17][18]. In [17], quadrotor swarming is considered using the leader–follower formation approach. In [18], the particle optimization approach is used to make the optimal formation control for a group of quadrotors. Some studies did not consider linear quadratic regulator (LQR) controllers, although they are well-known controllers that have been used in various applications, such as in controller systems, up until now [19][20][21]. The use of an LQR controller on a quadrotor is only for maneuver control, not yet for swarm motion [22][23].

The selection of LQR weighting matrices in the cost function is a major obstacle that makes it difficult to obtain the minimum cost function performance or desired control response. In the diagonal weighting matrix, the number of selected weight values is proportional to the order of the system plus the number of system control signals. Weight selection is often done by trial and error or using the desired response based on the analysis method [24][25]. Recently, many studies have proposed using the genetic algorithm (GA) method to obtain the LQR weighting matrices, which results in desired control performances more quickly and easily [26][27][28].

This paper considers a swarm control system of many quadrotors with two levels of the control loop and also defines simulation showing the coordination of six UAV quadrotor models to move and to form flocks. The design of the swarm control system is used in three-dimensional space simulations. Given the length limit of the article, the swarm center reference model is assumed to have been optimally provided by the higher-level control. This paper then focuses on designing the tracking control of the optimal swarm center reference model with LQR to minimize tracking errors. The problem of selecting the optimal weighting matrices of LQR is difficult to handle, but in this paper, the problem is solved by implementing a GA in order to produce the optimal performance of the UAV quadrotor swarm control system, which minimizes swarm tracking control errors in terms of the root-mean-square error (RMSE).

II. Materials and Methods

A. Quadrotor mathematical model

The working principle of a quadrotor is the balance of lift and torque produced by the rotation of four rotors. The lifting force is obtained from the acceleration of air around the blades that occurs because of the rotation of the rotors. Accelerated air displacement produces forces in the opposite direction from the air displacement. If the force exceeds the weight of the quadrotor, then the quadrotor can move vertically. The forward and reverse movements, as well as the left and right movements, can be obtained with the combination of the four propellers. For example, the forward movement results from the rotation of the two front propellers, which is faster than that of the propellers on the backside. This combination produces the resultant force and torque that make the quadrotor move forward.

Figure 1 shows the main components of a UAV quadrotor and the lifting force generated by the rotors. In addition to lifting, a quadrotor also works by balancing the torque produced by the four propellers. Setting the rotation speed of the rotating blades opposite can adjust the quadrotor rotation. In general, maneuvers that can be carried out by a quadrotor are shown in [29]. The lift force and torque of the blades are greatly influenced by the type of blades used. Calculation of lift and torque can be explained through the momentum theory and propeller element theory.

The momentum theory can be used to determine the relationship between lift, speed, and power in the propeller [1]. The principle of energy conservation is used to formulate the lift force in a floating state by including the efficiency of the rotor. Based on Newton's second law for translational motion with the fixed vehicle mass, the transformation matrix of the Earth axis, the body axis, and angular movements, the equilibrium moment equations are obtained. Using the translational motion with fixed vehicle mass in the form of Cartesian coordinates and kinematic triad relations, the following nonlinear model of a quadrotor is obtained using Equations (1) to (9):

$$\dot{u}_B = \frac{1}{m} \cdot \sum F_{xB} - g \cdot \sin\theta + r \cdot v_B - q \cdot w_B \quad (1)$$

$$\dot{v}_B = \frac{1}{m} \cdot \sum F_{yB} + g \cdot \sin\phi \cdot \cos\theta - r \cdot u_B + p \cdot w_B \quad (2)$$

$$\dot{w}_B = \frac{1}{m} \cdot \sum F_{zB} + g \cdot \cos\phi \cdot \cos\theta + q \cdot u_B + p \cdot v_B \quad (3)$$

$$\dot{p} = \frac{1}{J_{xx}} \cdot \sum M_x - \frac{(J_{zz} - J_{yy})}{J_{xx}} \cdot q \cdot r \quad (4)$$

$$\dot{p} = \frac{1}{J_{xx}} \cdot \sum M_x - \frac{(J_{zz} - J_{yy})}{J_{xx}} \cdot q \cdot r \quad (5)$$

$$\dot{p} = \frac{1}{J_{xx}} \cdot \sum M_x - \frac{(J_{zz} - J_{yy})}{J_{xx}} \cdot q \cdot r \quad (6)$$

$$\dot{p} = \frac{1}{J_{xx}} \cdot \sum M_x - \frac{(J_{zz} - J_{yy})}{J_{xx}} \cdot q \cdot r \quad (7)$$

$$\dot{p} = \frac{1}{J_{xx}} \cdot \sum M_x - \frac{(J_{zz} - J_{yy})}{J_{xx}} \cdot q \cdot r \quad (8)$$

$$\dot{\psi} = (q \cdot \sin\phi + r \cdot \cos\phi) \cdot \sec\theta \quad (9)$$

Equations (1) to (9) show the quadrotor motion state variables consisting of three body velocity variables (u_B , v_B , w_B), three angular velocity variables (p, q, r), and three angular variables (ϕ, θ, ψ). The detail derivation and the notation description can be found in [1]. The quadrotor mathematical model can then be written in the linear time-invariant state space equation after being linearized in the specified flying conditions as Equations (10) and (11):

$$\dot{x}(t) = Ax(t) + Bu(t);\ x(t_0) = x_0 \quad (10)$$

$$y(t) = Cx(t) + Du(t) \quad (11)$$

where $x \triangleq \{du\ dv\ dw\ dp\ dq\ dr\ d\phi\ d\theta\ d\psi\}$, $x \in R^n$, denote state space variables, and $u \triangleq \{d\Omega_{R1}\ d\Omega_{R2}\ d\Omega_{R3}\ d\Omega_{R4}\}$, $u \in R^m$, describe input variables; y denotes the output, $y \in R^l$; and the matrices are $A \in R^{nxn}$, $B \in R^{nxm}$, $C \in R^{nxl}$, and $D \in R^{lxl}$. The initial state condition at the time t_0 is denoted by x_0. The components of the matrix $\{A,B,C,D\}$ of the linear time-invariant quadrotor model in Equations (10) and (11) used in this paper are obtained when flying it at a speed of 5 m/s as follows [1].

B. Swarm model

Each swarm agent, in this case, a quadrotor, is assumed to move simultaneously and know the relative position of its members. The equation of motion of individual i by following [30] is given by the following Equation (13):

$$\dot{x}_i = -\nabla_{x_i}\rho(x_i) + \sum_{j=1, j\neq i}^{N} f(x_i - x_j), i = 1, \cdots, N \quad (13)$$

where $x_i \epsilon R^n$ denotes the position of individual i, $\rho(\cdot): R \to R$ is an odd function, and N denotes the number of individual swarm members. In the second term, $f(\cdot)$ is a function that describes the mutual attraction and repulsion that occur between agents as the following Equation (14) [31][32]:

$$f(y) = -y\left[a - b\exp^{(-\frac{||y||^2}{c})}\right] \quad (14)$$

where a, b, and c are positive constants and $||y||$ represents the distance between agents. Equation (14) shows that attraction dominates on long distances, whereas repulsion dominates on close distances. The function determines the direction and movement of each agent and prevents collisions between agents.

In practice, the value $\rho(\cdot)$ represents the attractant, repellent, or neutral profile that determines where the swarm agents will move together, for example, if there are hot spots in an area or the source of leakage of toxic chemicals [30]. The negative $\rho(\cdot)$ value represents the attractant profile where the swarm agents will gather, whereas the positive $\rho(\cdot)$ value indicates the repellent profile where the swarm agents will move away. The neutral condition is denoted by $\rho(\cdot) = 0$. Distance is

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.1008 & -0.1285 & -9.8034 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.9314 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1285 & 9.8034 & 0 & -4.9983 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.9314 & 0 & 0 & 0.0867 & 0 \\ 0 & 0 & 0 & -0.0531 & 4.9983 & 0.2520 & 0 & 0 & 0 & -1.1869 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.0017 & 0.0022 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.0004 & 0.0008 & -0.0014 & 0.0030 \\ 0 & 0 & 0 & 0 \\ 0.0086 & 0.0111 & -0.0111 & -0.0111 \\ -0.0013 & -0.0014 & -0.0014 & -0.0014 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D = 0 \quad (12)$$

affected by repulsion, and its magnitude is affected by parameters a, b, and c. There is a position where the magnitudes of attraction and repulsion are balanced. The magnitude of the dominant attractant for $||y|| > \delta$, and that for the dominant repellent for $||y|| < \delta$. From (14), it can be seen that a sign change occurred at that time using Equation (15):

$$y = \delta = \begin{cases} y = 0 \\ \|y\| = \sqrt{c \ln \frac{b}{a}} \end{cases} \quad (15)$$

It should also be noted that the change in sign will only occur if the value of $b > a$; thus, the condition will be obtained when the attraction function changes to the repulsion function. If this condition is not met, then there will be no change in the value of the function, resulting in a collision between swarm agents. By using the attraction/repulsion model as in Equation (14), for $t \to \infty$, the swarm model will stop moving [30]. This shows that at time instant $\bar{t}$, the agents in the swarm will be in a position where the attraction and repulsion functions will be balanced.

The cohesiveness of swarm agents can be influenced by the selected model. Movement between agents can be done by referring these agents to the swarm center as it will always be in the same position [31]. Another alternative is to refer to the position of an agent to other agents. Moreover, based on the model used, swarm cohesiveness can also be influenced by the nature of the attraction and repulsion functions. Attractions and repulsions in the swarm model can be local, when they occur only because they reached a certain threshold, or global when they are felt by the agent in whatever position the agent is located.

Based on the theorem developed by Gazy and Passino [31], it can be said that if a swarm system has only an attraction function, then the swarm agents will converge to a point that is the center of the swarm. By contrast, if the system has only a repulsion function, the swarm agents will move away from the central point to an infinite position. On this basis, it can be concluded that the best model that can be used is one that has the attraction function dominating at large distances to prevent swarm agents from dispersing and the repulsion function dominating at close distances to prevent these agents from colliding with each other [30].

C. Quadrotor tracking

Quadrotor tracking is expected to produce very small quadrotor position errors against the desired target. The technique commonly used is to add an integrator to an error or to use a derivative of an error [25]. Error state variables are described as Equation (16):

$$e(t) \triangleq x_{ref}(t) - x(t) \quad (16)$$

If derived from time, then the following Equation (17) is obtained:

$$\dot{e}(t) = \dot{x}_{ref}(t) - \dot{x}(t) \quad (17)$$

If the reference does not change (tracking target at a fixed point), then the derivative of x_{ref} is 0, so $\dot{e}(t) = -\dot{x}(t)$. From (17), the path tracking equation can use the following general Equation (18):

$$\dot{e}(t) = -\eta \dot{x}(t) \quad (18)$$

where η represents a constant that determines the weight of the tracking in the cost function. In the form of a matrix, the above equation can be written as Equation (19):

$$\dot{e}(t) = \begin{bmatrix} \dot{e}_x(t) \\ \dot{e}_y(t) \\ \dot{e}_z(t) \end{bmatrix} = \begin{bmatrix} -\eta \dot{x}_x(t) \\ -\eta \dot{x}_y(t) \\ -\eta \dot{x}_z(t) \end{bmatrix} \quad (19)$$

Substituting $\dot{x}_x = u$, $\dot{x}_y = v$, and $\dot{x}_z = w$ yields Equation (20):

$$\dot{e}(t) = \begin{bmatrix} \dot{e}_x(t) \\ \dot{e}_y(t) \\ \dot{e}_z(t) \end{bmatrix} = \begin{bmatrix} -\eta u(t) \\ -\eta v(t) \\ -\eta w(t) \end{bmatrix} \quad (20)$$

Based on the theorem developed [30], for an ideal system, the swarm center point will not move when the agents are flocking. Therefore, when given a target, the movement of the swarm center should be a straight line from the initial swarm center position toward the target [31].

D. A two-level swarm control system

The swarm control system consists of two levels. The lower-level controller uses a discrete LQR to control the movement of an individual quadrotor so that it can follow the desired tracking path. On the other hand, the higher-level controller is a proportional – derivative (PD) type controller that controls the movement between swarm members by producing paths that must be followed by each quadrotor. The swarm control system block diagram followed [11] is shown in Figure 2.

1) Lower-level control

The lower-level controller uses a discrete LQR. This controller is used to control the movement of each quadrotor against the trajectory produced by the high-level controller. A full state feedback gain K_{LQR} of the LQR is obtained by varying the weighting matrices $Q = Q^T \geq 0$ and $R = R^T > 0$ on a cost function defined as Equation (21):

$$J_\infty = \sum_{k=1}^{\infty} [x^T(k)\, Qx(k) + u^T(k) R u(k)] \quad (21)$$

with discretized dynamic equations of each quadrotor (10) given by Equation (22):

$$x(k+1) = Ax(k) + Bu(k) \quad (22)$$

Optimal quadratic control signal $u(k)$, by using the tracking definition and discretized error (16) above, is defined as Equation (23):

$$u(k) = -K_{LQR} e(k) \quad (23)$$

with a full state feedback gain matrix K_{LQR} that fulfils the Equation (24):

$$K_{LQR} = (R + B^T S B)^{-1} B^T S A \quad (24)$$

The S matrix is the steady-state solution of the algebraic Riccati equation in Equation (25):

$$S = Q + A^T SA - A^T SB(R + B^T SB)^{-1} B^T SA \quad (25)$$

The Riccati equation is solvable if the pair $\{A, B\}$ is stabilizable and there is no unobservable mode on the unit circle of the pair $\{Q, A\}$.

Two adjustable variables are sought to optimize the cost function (21) of the lower-level controller, namely, the Q and R matrices. The Q matrix has row and column sizes of a number of state variables in the system, whereas the matrix R has a number of rows of input variables. The movement of a controlled quadrotor agent will be determined based on a closed-loop system in the state space function in Equation (26):

$$x(k+1) = \left(A - BK_{LQR}\right)x(k) + K_{LQR}x_{ref}(k)$$
$$y(k) = Cx(k) \quad (26)$$

The closed-loop system (26) is asymptotically stable utilizing the solvability of the Riccati Equation (25) [25].

2) Higher-level control

The higher-level controller in the swarm model used in this paper implements a PD controller. The attraction and repulsion functions used follow equation (14). In addition to the above functions, the swarm agent movement is also influenced by a repulsion function that occurs if the agent is too close. This function is represented in Equation (27):

$$u' = k_r \exp\left(\frac{-\frac{1}{2}\|x_i - x_j\|^2}{r_s^2}\right) \quad (27)$$

where $k_r > 0$ is the magnitude of the repulsion and $r_s > 0$ is the size of the area around the agent. The difference between agents is too far; that is, when $\left\|x_i - x_j\right\|$ is very large relative to r_s, then the function will be zero. Let us respectively define the swarm center and the average velocity, which are generally time-varying [33], as Equation (28):

$$\bar{x} = \frac{1}{N}\sum_{i-1}^{N} x^i \text{ and } \bar{v} = \frac{1}{N}\sum_{i-1}^{N} v^i \quad (28)$$

The dynamics of the error system for position and velocity of each agent i are defined as Equation (29):

$$e_{i,p} = x_i - \bar{x} \text{ and } e_{i,v} = v_i - \bar{v} \quad (29)$$

In the end, the movement of the agent in the swarm model used in this paper is determined by the Equation (30) in [33].

$$u_i = -M_i k_a e_{i,p} - M_i k_b e_{i,v} - M_i k_v v_i$$
$$+M_i k_r \sum \exp\left(\frac{\frac{1}{2}\|e_{i,p} - e_{j,p}\|}{r_s^2}\right)\left(e_{i,p} - e_{j,p}\right) -$$
$$M_i k_f \left(\nabla J_p(x_i) - d_f^i\right) \quad (30)$$

Here, it is assumed that each agent has mass M_i and velocity v_i with a bounded sensing noise signal d_f^i. The scalar value $k_a > 0$ denotes a reinforcement of attraction indicating how aggressive the agent is to aggregate. The scalar value $k_r > 0$ is a reinforcement of repulsion that indicates how much agent i will move away from agent j.

In this paper, the developed swarm model has the characteristic of having no leader among swarm members. Swarm member dynamic equations are modelled as in Equation (14). The state of agent i was defined with x_i $(i = 1, \ldots, N)$, and it is assumed that the agent moves in the n -dimension in Euclidean space, with the equation of motion of each agent determined by the combination of the following equations:

$$\begin{aligned} \dot{x}_i &= v_i \\ m\dot{v}_i &= u_i \end{aligned}; x_i, v_i, u_i \in R^n, i = 1, \ldots, N \quad (31)$$

where, respectively, x_i, v_i, u_i, and m_i are the position, speed, control input, and mass of the agent.

The main goal of the swarm model control design is to control the dynamic movement of all swarm members to the desired position. The dynamic movement of the agent is influenced by the following: (i) the distance and velocity between the agents, and (ii) the attraction and repulsion on the designed profile. In this paper, the control protocol is a modification from [33] for the agent modelled as Equation (32):

$$u_i = -M_i k_a e_{i,p} - M_i k_b e_{i,v} - M_i k_v v_i$$
$$+M_i \sum_{i=1.i\neq j}^{N}(a-b)\exp\left(\frac{||e_{i,p} - e_{j,p}||}{c}\right)\left(e_{i,p} - e_{i,p}\right) -$$
$$M_i k_f \left(\nabla J_p(x_i)\right) \quad (32)$$

It is assumed that the agent is equipped with sensors and programs having the mass of M_i and the speed of v_i. The proportional gain $k_a > 0$ is a reinforcement that affects the aggressiveness of the agent to aggregate. The gain $k_r > 0$ functions as a parameter that affects the repulsion that occurs between agents. Both k_a and k_b are proportional gains, whereas k_v and k_f are attenuation of speed and derivative strengthening to follow the desired agent movement profile. The model in (32) is modified from (30), by adding the attraction force between the agent and the value $M_i = 1$ (to simplify the control system). The swarm control system diagram block is designed as in Figure 2.

III. Results and Discussions

A. Determination of trial and error parameters

The mathematical equation model implemented in the simulation uses the parameters of reference [1] discretized with 0.01 s sampling time. Searching for the optimal values of Q and R using the trial and error method, the initial parameters are simulated using predetermined Q and R matrices as Equations (33) and (34):

$$\begin{aligned} Q_1 &= 10 \times I \\ Q_2 &= 100 \times I \\ Q_3 &= 1000 \times I \\ Q_4 &= 10000 \times I \end{aligned} \quad (33)$$

$$R_1 = 0.1 \times I$$
$$R_2 = 0.01 \times I \quad (34)$$
$$R_3 = 0.001 \times I$$
$$R_4 = 0.0001 \times I$$

where I denote the identity matrix with an appropriate size of Q and R.

The combination of the parameters Q and R from Equation (33) and (34) is then simulated for tracking a quadrotor on the trajectory function defined by $l(t)$ using Equation (35):

$$l_x(t) = \begin{cases} x_0 + \frac{t}{20}, \text{for } 0 < t \le 1000 \\ x_0 + 50, \text{for } 1000 < t \le 2000 \\ x_0 + \left(50 - \frac{t-2000}{20}\right), \text{for } 2000 < t \le 3000 \\ x_0, for\ 3000 < t \le 3999 \end{cases}$$

$$l_y(t) = \begin{cases} y_0, \text{for } 0 < t \le 1000 \\ y_0 + \frac{t-1000}{20}, \text{for } 1000 < t \le 2000 \\ y_0 + 50, \text{for } 2000 < t \le 3000 \\ y_0 + \left(50 - \frac{t-3000}{20}\right), \text{for } 3000 < t \le 3999 \end{cases}$$

$$l_z(t) = \begin{cases} z_0 + \frac{t}{20}, \text{for } 0 < t \le 1000 \\ z_0 + 50, \text{for } 1000 < t \le 2000 \\ z_0 + (50 - \frac{t-2000}{20}), \text{for } 2000 < t \le 3000 \\ y_0, \text{for } 3000 < t \le 3999 \end{cases} \quad (35)$$

The performance of the tracking error is measured using the RMSE method. The best weight matrix results are obtained for $Q = 1000I$ and $R = 0.001I$, which have an RMSE value of 7.122 m. Simulation results are shown in Figure 3.

B. Parameter optimization using genetic algorithm

GA is one area of research that has received considerable attention until now [34][35][36]. GAs can help produce a solution of complex functions on the basis of the selection process that occurs in nature. They work on the basis of the initial population that can be linked to certain variables. In general, this initial population contains randomly constructed binary combinations 1 and 0. This binary combination corresponds to the value of the variable to be found. In this paper, by following the GA, the mating probability of each chromosome is determined using the following Equation (36):

$$probability = \frac{fitness}{total\ fitness} \times 100\% \quad (36)$$

The parameters used in the GA are as follows: the number of chromosomes, which is 20; the number of genes per chromosome, 1,000; and the number of generations produced, 30. The chromosomes containing binary numbers are encoded into real-valued individuals in the specified interval of [100; 1,000]. The process of reproduction of selected chromosomes has a crossover probability of 0.6. Chromosomes in a population have a mutation probability of 0.5 %. Next, the GA is applied to find the Q_1 to Q_{12} parameters in the matrix Q defined as Equation (37):

$$Q = \begin{bmatrix} Q1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Q2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q12 \end{bmatrix} \quad (37)$$

and the value of R_v on the matrix R defined as Equation (38):

$$R = R_v \times I^{4\times4} \quad (38)$$

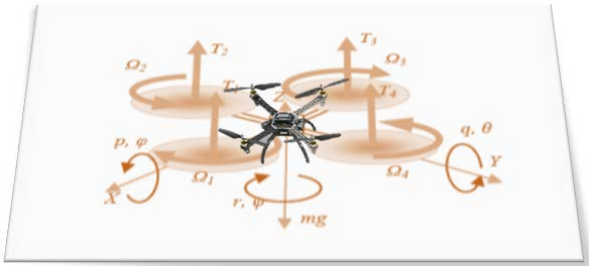
The fitness function of the GA uses the quadrotor motion RMSE of the trajectory tracking according to the trajectory described in Equation (35). The accomplishment of the GA results in the minimum RMSE of the tracking error with the parameters Q and R as mentioned in Equation (39):

The control parameters obtained from the GA are then simulated in tracking. The results of the simulation can be seen in Figure 4. Simulation results show an RMSE value of 4.5429 m. It can be concluded that the GA improves quadrotor tracking performance by 2.5791, or by 36 %. It can be seen in the simulation results of trajectory tracking using the best trial and error parameters (Figure 3a) that the movement of quadrotor has a considerable difference to the path (RMSE 7.122 m). In addition, there is a slight delay so that when the path turns the quadrotor does not immediately follow. After optimization using GA (Figure 4a), it can be seen that

$$Q = \begin{bmatrix} 1.21e+7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11.75e+5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.67e+04 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30.24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 51.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 46.84 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38.85 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 98.49 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 68.99 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.58e+02 \end{bmatrix}$$

$$R_v = 8.14 \times 10^{-5} \quad (39)$$

(a)



(b)

Figure 1. (a) Quadrotor at the Artificial Intelligence, Control and Automation Laboratory-ITB; (b) The dynamics of a quadrotor

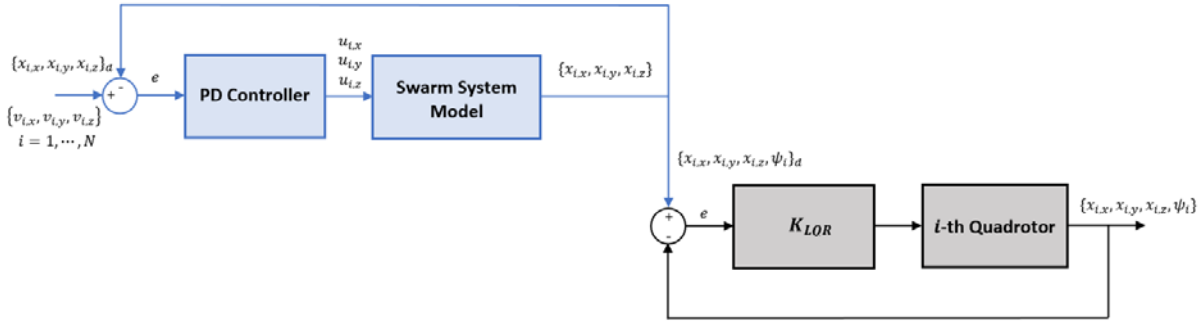
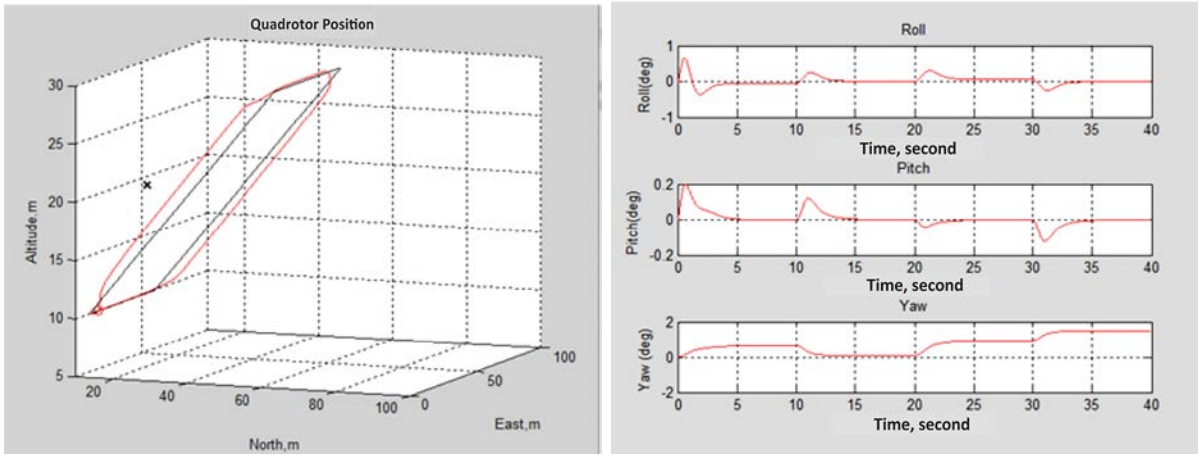


Figure 2. Higher-level and lower-level controllers in a swarm control system



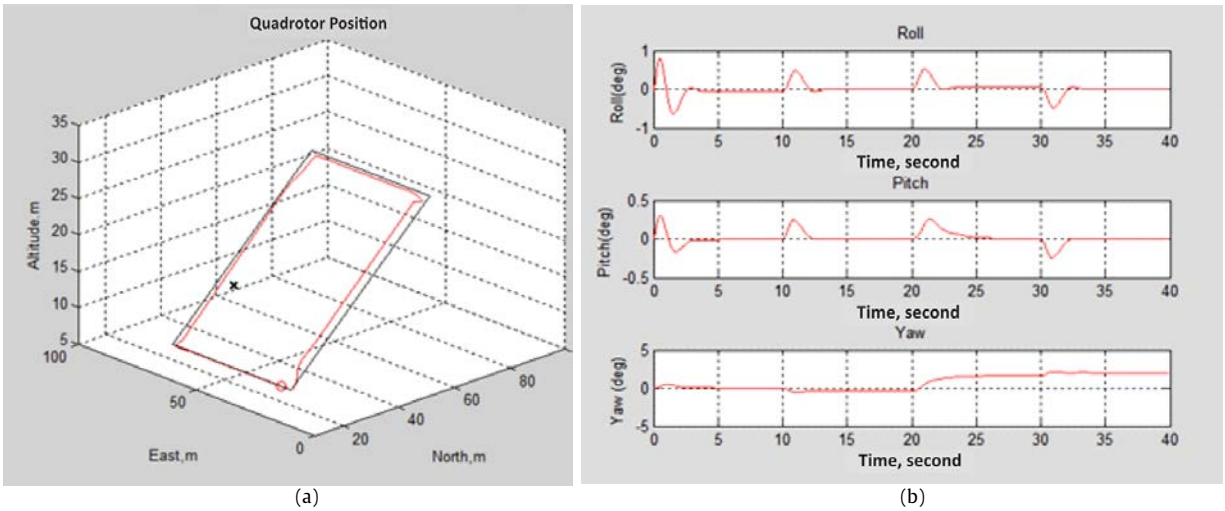
(a) (b)

Figure 3. (a) Path tracking using LQR parameters based on the trial and error method; (b) Pitch angle, roll, and yaw tracking simulation with LQR parameters based on the trial and error method

the quadrotor movement is closer to the tracking path.

C. Swarm simulation

The higher-level controller has seven parameters that must be chosen properly to produce swarming with fast aggregation. The selection of these parameters requires a separate discussion. In this paper, it is assumed that these parameters have been obtained properly using GA. The optimal higher-level control parameters for swarm models and flocking simulations are specified as $a = 1$, $b = 10$, $c = 3$, $k_a = 0.7779$, $k_b = 4.3194$, $k_v = 2.4428$, and $k_f = 1.4214$. The trajectory of the swarm center using the above parameters is shown in Figure 5. Flocking simulation results using these parameters are shown in Figure 6. The movement of UAV quadrotor agents in a swarm is shown by colors: red,



(a) (b)

Figure 4. (a) Path tracking using LQR parameters based on the genetic algorithm; (b) Pitch angle, roll, and yaw tracking simulation with LQR parameters based on the genetic algorithm

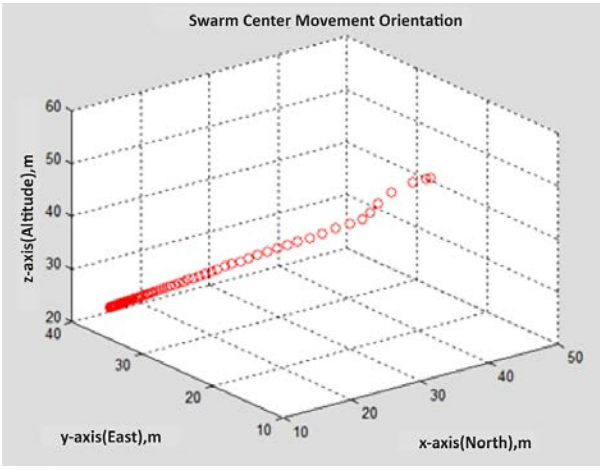
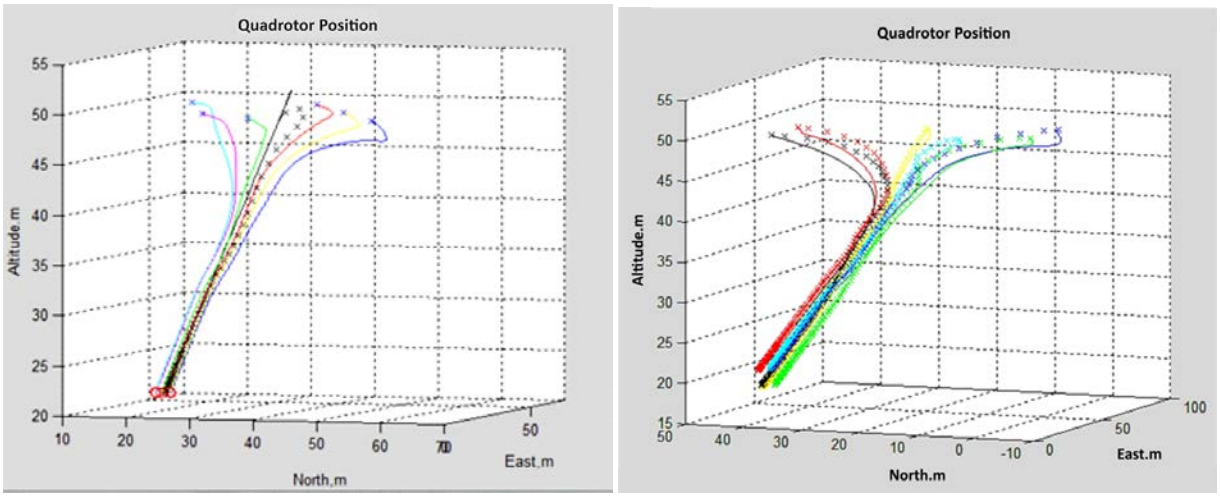


Figure 5. Swarm center movement using optimal parameters by the genetic algorithm

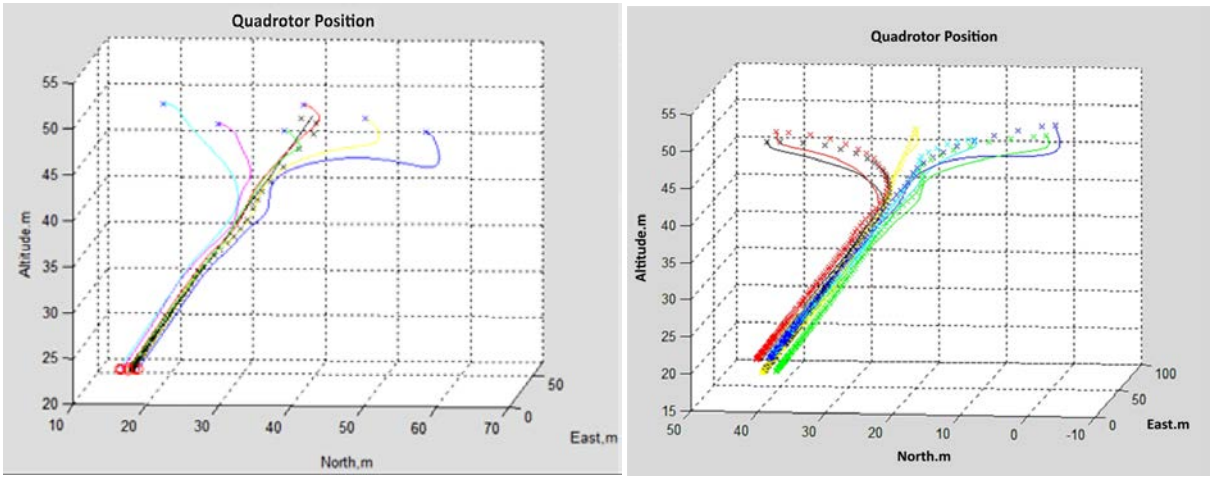


(a) (b)

Figure 6. (a) Swarm agent movement using LQR parameters from the trial and error method; (b) Swarm agent movements and tracking trajectories using LQR parameters from the trial and error method

yellow, blue, green, light blue, and purple. In addition, the center of the swarm is represented by a black "×" sign, and a steady-state line is represented by a black line. RMSE is calculated by looking at the difference between the distance of the swarm center movement and the line equation when the swarm system has reached a steady-state.

The simulation application of LQR parameters for swarm system simulation uses the best trial and error results, namely, $Q = 1000 \times I^{12\times12}$ and $R = 0.001 \times I^{4\times4}$ shown in Figure 6. The LQR using the trial and error method produces an RMSE of the swarm model of 0.2365 m. After that, a swarm model simulation is performed using the LQR



(a) (b)

Figure 7. (a) Swarm agent movement using LQR parameters generated by the genetic algorithm. (b) Agent swarm movements and tracking trajectories using LQR parameters generated by the genetic algorithm

Table 1.
Simulation results and performance

Description	LQR tracking (RMSE in m)	Flocking aggregation (RMSE in m)	Flocking speed at height (m)
Trial error	7.1220	0.2365	40
Genetic algorithm	4.5429	0.1959	45
Improvement	2.5791 (36.00 %)	0.0406 (17.17 %)	5 (12.50 %)

parameters obtained from GAs. The simulation results can be seen in Figure 7, this parameter produces a system RMSE of 0.1959. The swarm model performance improvement was 0.0406, or 17.17 %.

In Figure 7a, it can be seen that agent aggregation using LQR parameters is faster than the aggregation of agents before optimization (Figure 6a). Aggregation of agents in the simulation before optimization occurs around a height of 40 m, whereas agent aggregation in the simulation after optimization occurs around a height of 45 m. This faster aggregation is caused by the quadrotor's quick response to the tracking of the path produced by the swarm model. It can also be seen in Figure 7b that the movement of each quadrotor agent follows the tracking path provided by the swarm model better after the LQR is optimized with the GA, although this can have a large enough effect if the swarm target moves to produce a more complicated trajectory. The summary of the improved performance is given in Table 1.

IV. Conclusion

Tuning the weighting matrices of LQR is difficult by using the trial and error method because the nature of the system is random and complicated. Hence, it is difficult to find the best parameters as the amount of value sought is in a large range. This paper considered the tracking control optimization of the LQR weighting matrices in the swarm control system to produce quadrotor unmanned flocking vehicles with GA optimization. The weighting matrix optimization using the GA improved the performance of the swarm control system: in tracking performance, 4.5429 m (RMSE) improved by 2.5791 m (36.00 %); flocking aggregation 0.1959 m (RMSE) improved by 0.0406 m (17.17 %), and flocking speed at height 45 m means 5 m (12.50 %) faster.

Acknowledgement

This work was partially supported by ITB Research Program, Institut Teknologi Bandung. The authors would like to thank Institut Teknologi Bandung for providing research facilities to complete the research.

Declarations

Author contribution

All authors contributed correspondingly as the main contributor to this paper. All authors read and endorsed the final paper.

Funding statement

This research did not receive any particular grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] T. Sudiyanto, M. Muljowidodo, and A. Budiyono, "First principle approach to modeling of primitive quad rotor," *International Journal of Aeronautical and Space Sciences*, vol. 10(2), pp.148-160, 2009.

[2] P. Pounds, R. Mahony, J. Gresham, P. Corke, and J.M. Roberts, "Towards dynamically-favourable quad-rotor aerial robots," In *Proceedings of the 2004 Australasian Conference on Robotics & Automation. Australian Robotics & Automation Association*, 2004.

[3] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," MSc Theses, Lund University, 2008.
[4] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35(4), pp. 287-300, 2013.
[5] J. Leonard, Al Savvaris, and A. Tsourdos, "Towards a fully autonomous swarm of unmanned aerial vehicles," In *Proceedings of 2012 UKACC International Conference on Control*, pp. 286-291. IEEE, 2012.
[6] C.W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 25-34, August 1987.
[7] A.E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2(2-4), pp. 97-120, 2008.
[8] E. Ferrante, A.E. Turgut, C. Huepe, A. Pinciroli, and M. Dorigo, "Self-organized flocking with a mobile robot swarm: a novel motion control method," *Adaptive Behavior*, vol. 20(6), pp. 460-477, 2012.
[9] S. Ramroop, F. Arvin, S. Watson, J. Carrasco-Gomez, and B. Lennox, "A bio-inspired aggregation with robot swarm using real and simulated mobile robots," In *Annual conference towards autonomous robotic systems 2018*, Springer, Cham., pp. 317-329, 2018.
[10] E. Joelianto, A. Qurthobi, "Optimal Control Design for A Formation Tracking with Leader-Follower Concept of Multi-Agent Autonomous Helicopter Model," *Proceedings of International Conference on Intelligent Unmanned Systems*, vol. 7, 2011.
[11] E. Joelianto, and A. Sagala, "Swarm tracking control for flocking of a multi-agent system," *IEEE Conference on Control, Systems & Industrial Informatics*, pp. 75-80, September 2012.
[12] E. de Vries, and K. Subbarao, "Cooperative control of swarms of unmanned aerial vehicles," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, pp. 78, January 2011.
[13] Z. Hou, W. Wang, G. Zhang, and C. Han, "A survey on the formation control of multiple quadrotors," In *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 219-225. IEEE, 2017.
[14] Hönig, J.A. Preiss, T.S. Kumar, G.S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34(4), pp. 856-869, 2018.
[15] Miswanto, H. Pranoto, and D.M. Muhammad, "The Collective Behavior of Multi-Agents System for Tracking a Desired Path," *International Journal of Basic & Applied Sciences IJBAS-IJENS*, vol. 11, pp. 81-86, 2011.
[16] A. Trizuljak, F. Duchoň, J. Rodina, A. Babinec, M. Dekan, and R. Mykhailyshyn, "Control of a small quadrotor for swarm operation," *Journal of Electrical Engineering*, vol. 70(1), pp. 3-15, 2019.
[17] K. Choutri, M. Lagha, L. Dala, and M. Lipatov, "Quadrotors UAVs swarming control under Leader-Followers formation," In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 794-799. IEEE, 2018.
[18] I. M. Lazim, A.R. Husain, N.A.M. Subha, Z. Mohamed, and M.A. M. Basri, "Optimal formation control of multiple quadrotors based on particle swarm optimization," In *Asian Simulation Conference*, pp. 121-135. Springer, Singapore, 2017.
[19] D. Das, G. Gurrala, and U.J. Shenoy, "Linear quadratic regulator-based bumpless transfer in microgrids," *IEEE Transactions on Smart Grid*, vol. 9(1), pp. 416-425, 2016.
[20] Z. Ge, Y. Wang, and M. Lv, "Three-dimensional trajectory tracking guidance law based on linear quadratic regulator," *Journal of Physics: Conference Series*, vol. 1039(1), p. 012042. IOP Publishing, 2018.
[21] L. Cao, S. Tang, and D. Zhang, "Flight control for air-breathing hypersonic vehicles using linear quadratic regulator design based on stochastic robustness analysis," *Frontiers of Information Technology & Electronic Engineering*, vol. 18(7), pp. 882-897, 2017.
[22] M. İçen, A. Ateş, and C. Yeroğlu, "Optimization of LQR weight matrix to control three degree of freedom quadcopter," In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1-6. IEEE, 2017.
[23] E. Okyere, A. Bousbaine, G.T. Poyi, A.K. Joseph, and J.M. Andrade, "LQR controller design for quad-rotor helicopters," *The Journal of Engineering*, no. 17, pp. 4003-4007, 2019.
[24] B.D. Anderson. and J.B. Moore, "*Optimal control: linear quadratic methods*," Dover Publications, Inc., New York, 2007.
[25] E. Joelianto, "*Linear Quadratic Control: A State Space Approach*," ITB Press, 2017.
[26] J.M. Ahmed, "Optimal tuning linear quadratic regulator for gas turbine by genetic algorithm using integral time absolute error," *International Journal of Electrical & Computer Engineering*, vol. 10(2), pp. 1367-1375, 2020.
[27] M. Ali, S.T. Zahra, K. Jalal, A. Saddiqa, and M.F. Hayat, "Design of optimal linear quadratic gaussian (LQG) controller for load frequency control (LFC) using genetic algorithm (GA) in power system," *International Journal of Engineering Works*, vol. 5(3), pp. 40-49, 2018.
[28] H. Asadi, S. Mohamed, C.P. Lim, and S. Nahavandi, "Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47(2), pp. 238-254, 2016.
[29] G.B. Raharja, G.B. Kim, and K.J. Yoon, "Design of an autonomous hover control system for a small quadrotor," *International Journal of Aeronautical and Space Sciences*, vol. 11(4), pp. 338-344, 2010.
[30] V. Gazi, and K.M. Passino, "Stability analysis of social foraging swarms," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 34(1), pp. 539-557, 2004.
[31] V. Gazi, and K.M. Passino, "A class of attractions/repulsion functions for stable swarm aggregations," *International Journal of Control*, vol. 77(18), pp. 1567-1579, 2004.
[32] V. Gazi, and K.M. Passino, "Stability analysis of swarms," *IEEE transactions on automatic control*, vol. 48(4), pp. 692-697, 2003.
[33] K.M. Passino, "*Biomimicry for optimization, control, and automation*," Springer Science and Business Media, 2005.
[34] G. Lindfield, and J. Penny, "*Numerical methods: using MATLAB*," 4th Ed., Academic Press, 2018.
[35] R.C. Chakraborty, "*Fundamentals of genetic algorithms*," Reproduction, 22, p. 35, 2010.
[36] S. Mirjalili, "*Genetic algorithm. In Evolutionary algorithms and neural networks*," Springer, pp. 43-55, 2019.