AKADÉMIAI KIADÓ

UNIVERSITY of DEBRECEN

International Review of Applied Sciences and Engineering

14 (2023) 3, 374–382

DOI:
10.1556/1848.2022.00584

ORIGINAL RESEARCH PAPER

Evolutionary control system of asymmetric quadcopter

Hazim Albedran[1,2*] and Károly Jármai[2]

[1] University of Kufa, Najaf, Iraq

[2] University of Miskolc, H-3515 Miskolc, Egyetemváros, Hungary

Received: August 16, 2022 • Accepted: December 2, 2022
Published online: March 28, 2023

ABSTRACT

Drones, specifically quadcopters, have increased in importance during the last years due to their wide range of applications, from civil applications to military employment. One of the most important issues in quadcopters is the efficient control system. While many researchers have dealt with building control systems for symmetric quadcopters, this work presents an efficient control system for asymmetric quadcopters using evolutionary computations. The problem is well-defined throughout the paper, and the methodology is explained in detail in the respective sections. A genetic algorithm is used to tune the weighting matrix of the control system after formulating the control system as an optimization problem. The genetic algorithm was fast and active to increase the performance of the proposed system.

KEYWORDS

drone, quadcopter, control system, evolutionary computing, genetic algorithm

1. INTRODUCTION

The quadcopter is one of the autonomous robot types that do not need a supporting surface; it contains four symmetric rotors. The symmetry of the quadcopter is crucial because the control of the quadcopter depends on rotor speed and lift force variation [1, 2]. The symmetric structure of the quadcopter has many advantages for different applications like aerospace fields [3], military security systems [4], critical monitoring [5], and mineral exploration [6]. The key point in all quadcopter applications is the sensitive nature of the flight control of the quadcopter, which is expressed by many control techniques like Lyapunov-based control for indoor micro-quadcopter [7] and combinations of PID-based control [8]. Also, feedback linearization-based control is presented [9, 10], as well as model predictive control [11]. The control system looks like a brain that tells a dynamic system how to behave and react to responses. Adaptive backstepping control was designed [12] for the ball and beam system, considering parameter uncertainties. The backstepping controller [13] is a well-known technique in control systems that has been used in many applications. Optimal control was formulated for a quadcopter as a constrained optimization problem [14]. The PID controller is the most famous controller for the speed of motors [15]. It is worth mentioning that the PID controller is convenient for SISO systems because these systems have a single input and single output plant, which is easier to do using a PID controller. The quadcopter is an underactuated system, i.e. only longitudinal, lateral, and altitude axis and yaw angle can be directly controlled. Roll and pitch angles are systemically calculated, and that is why it is hard to control an asymmetric quadcopter.

In this paper, a unique control system based on evolutionary computing is developed for asymmetric quadcopters, which are hard to be controlled using traditional methods. The proposed methodology is simple and fast and employs the genetic algorithm for tuning the control parameters. The genetic algorithm used in this paper is one of the metaheuristic algorithms, i.e., other optimization techniques can be used for this controller like swarm

*Corresponding author.
E-mail: hazimn.bedran@uokufa.edu.iq

intelligence methods. The proposed structure in this work is an asymmetric quadcopter that can be used for aesthetic reasons or any other purposes. The unbalanced structure is a challenging issue for developing a control system.

2. CONTROLLING A QUADCOPTER

The control system of a quadcopter, which is shown in Fig. 1, can be modeled as a rotating mass in a frictionless environment; the dominating states vector is $[\theta, \dot{\theta}]$; angle, and angular rate. In this section, a full-state feedback controller using LQR is employed [16]. For this problem, a single actuation is assumed, representing the quadcopter's four thrusters working together to produce a single torque.

3. STATE SPACE REPRESENTATION

Ordinary differential equations can represent dynamic systems because the system's change is a function of its current state. If we zip through the derivation of a system, we end up with a differential equation. For instance, in the spring-mass system the way the system changes, which is acceleration, is a function of the current state which is position. If the dynamic system was initialized with some energy, it would continue to move on its own because of the relationship between the derivative of the state and the state itself. For any arbitrary dynamic system, we can make claims about relationships like stability if we consider how energy changes by analyzing the relationship between derivatives and their states. Equation (1) expresses the stability of a linear system which is a relation between a state and its derivative:

$$\dot{x} = f(x), \tag{1}$$

where x is the state and $\dot{x}$ is its derivative. If the energy is being dissipated over time, then the system is stable, and the faster the energy is dissipated, the more stable the system will be. If the energy increases over time, even for just a part of the time, the system is unstable. Thus, stability is the property that defines how states and their derivatives are connected. The pattern of the system movement in equation (1) can also be influenced by external energy u being added or moved over time:

$$\dot{x} = f(x, u). \tag{2}$$

In state space representation, we reform a high-order differential equation into a set of first-order differential equations. This reforming or repackaging is useful because it makes the system easy to analyze because we can notice the interconnected system's underlying behavior and how the system can be affected by an external input or multiple inputs. There are many control techniques that are built based on state space, such as:

1. Kalman filter [17].
2. Linear–quadratic regulator (LQR) control [12].
3. Rubost control [18].
4. Model predictive control (MPC) [19].
5. PID controller [15].

3.1. Pole placement

Pole placement or full-state feedback is a method used to develop a feedback controller from a model that is expressed by state space equations [20]. This method is not used extensively in the control industry like LQR; for pole placement control, the feedback occurs on the state vector instead of the output, as shown in Fig. 2. By finding the proper value of the gain K and subtracting the result from the multiplication of the reference signal and scaling term Kr we can get a new input value that can be fed to the plant. The Kr on the reference is a scalable value and is used to affect the steady-state error by controlling the input signal.

The state equation is defined as follows:

$$\dot{x} = Ax + Bu, \tag{3}$$

where A and B are matrices while $x, \dot{x}$, and u are the state, its derivative, and control action, respectively. The dynamics of a linear system are captured by the first term of equation (3) and express how the energy is stored and moved in the system. The second term of equation (3) shows how the system responds to the inputs. Any feedback controller has to modify the A matrix to change the system's dynamics. This is true when considering the system's stability, where the eigenvalues of the A matrix are the system's poles, and their locations on the root locus graph express the system's

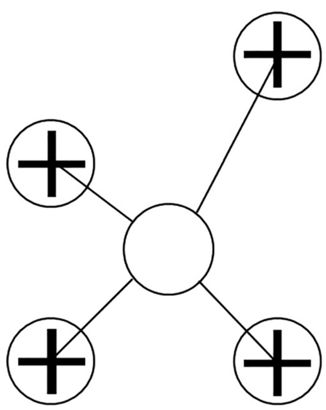

Fig. 1. Quadcopter structure

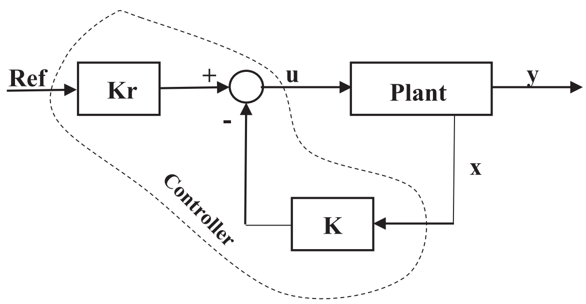

Fig. 2. Typical feedback control system

stability. Thus, we can generate closed-loop stability by moving poles or eigenvalues of the A matrix. We can set appropriate poles for simple problems of two state variables and find the gain vector values using the characteristic equation. However, the problem becomes complex or infeasible for problems with a large number of state variables. One of the disadvantages of the pole placement method is that it assumes feedback for any state in the system, which is not possible in many cases due to measurement deficiency.

3.2. Linear quadratic regulator

Linear quadratic regulator or LQR control is one of the optimal control types based on state space representation [16]. The structure of the LQR control is the same as the pole placement control, but both use different ways to choose K values. In LQR control, we search for the K matrix based on an objective function considering the system's performance and effort on the system itself. The K matrix can be found by minimizing the objective or cost function f as follows [14]:

$$f = \int_0^{\infty} \left(x^T Qx + u^T Ru\right) dt, \tag{4}$$

where Q and R are penalties on the performance and effort, respectively, and the following equation can estimate the gain matrix for a given dynamics of the system:

$$u = -kx. \tag{5}$$

The best performance has the minimum area under the curve in the state-time space considering squaring the values on the curve to avoid negative data. This means that the cost function will be turned into a quadratic function that has a definite minimum value that can be estimated using a metaheuristic algorithm, for example. R and Q are both positive definite matrices where R acts on the input vector while Q acts on the state vector. The cost function can be written as follows:

$$f = \left[x^T \ u^T\right] \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}. \tag{6}$$

By solving the LQR problem, we can get the gain matrix that produces a low cost for a given dynamic system. The best cost or best performance can be represented by the first term of equation (4) as the area under the curve in the state-time space, as shown in Fig. 3. In other words, Fig. 3 shows the first part of the cost function that has to be minimized to the lowest value as much as possible.

The second part of equation (4) expresses the actuation of the system, which also has to be minimized to save power, money, materials, or other costing efforts. The cross product of u and x in equation (6) is often penalized by adding penalty factors to the off-diagonal of the weighting matrix as follows:

$$f = \left[x^T \ u^T\right] \begin{bmatrix} Q & N \\ N^T & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}. \tag{7}$$

In other words, N^T and N has scalar values that can be used to punish or penalize wrong solutions. Thus, equation (4) is rewritten, including the penalty factors:

$$f = \int_0^{\infty} \left(x^T Qx + u^T Ru + 2x^T Nu\right) dt. \tag{8}$$

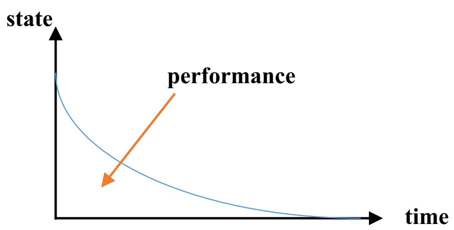

Fig. 3. Performance of the dynamic system

4. EVOLUTIONARY COMPUTATION

4.1. Evolutionary family

Evolutionary computation is focused on using natural evolution mechanisms to modify a population of individuals representing different points in the solutions space associated with a given problem to reach an optimal situation using a "survival of the fittest" procedure. The evolutionary computation family of algorithms classification divides the family into four main groups [21], as revealed in Fig. 4. Most of these categories share a common basic algorithm. Variations come from the way some operators are applied to the algorithm process and what is the basic representation of the individuals in the population.

4.2. Genetic algorithm

Genetic algorithm GA [22] is a heuristic optimization algorithm where the evolution and natural genetics form the algorithm's base. In the 1970s, Holland [23] first presented the concept of genetic algorithms. The inspiration came from the principle of survival of the fittest after the principle had been introduced by Charles Robert Darwin [24]. Most of the search spaces have complex shapes and multimodality containing multi-local optimum solutions. However, the probabilistic mechanism of the GA leads a specific solution optimally in a search space toward a point of the minimum cost or maximum benefits. In real life, individuals of species can survive and reproduce if they can adapt to the change in

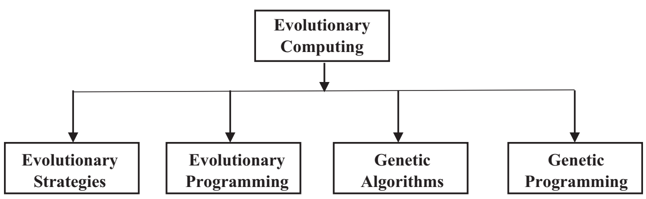

Fig. 4. Family of the evolutionary computation

environment and best fit the competition in the foraging process. Each individual has its own characteristics that make it unique.

On the other hand, the genetic content of the individual determines these characteristics. Each feature is represented by a chromosome consisting of a set of units called genes. Thus, the order or form of the set of genes is responsible for a specific feature in the individual. Hence, genes are the key points behind the survival of the individual. Evolution is successive changes in the characteristic of an individual during successive generations over time. However, we can define evolution as successive changes in the genetic content for an individual during successive generations. During reproduction, half of the genetic material comes from a male, and the other half comes from a female of a specific species. Thus, the offspring is a recombination of the genetic pool, and the role of the natural selection is to decide whether that offspring fits or does not fit to the competitive environment. This mechanism enables the individuals with best features to dominate at the expense of the individuals with weak features. Only the fittest will survive and couple to reproduce the next generation, which is another recombination of the genetic material. So, we can say that the best set of genes will survive in the existence battle. Specifically, the recombination of the genetic pool from parents to reproduce offspring is called crossover. For some reason, the arrangement of the genes in a chromosome may be changed in a probabilistic process called mutation, which leads to a change in some features of the individual. Repeated crossover and mutation may lead to the right possible individual that can be more fit for the competitive environment. In a genetic algorithm, sequences of generations are created using selection, crossover, and mutation; the crossover and mutation represent a search mechanism within the search space. Selection provides the process of choosing the best solution to survive, and many selection mechanisms were discussed in the literature. Each solution should be associated with a fitness value; the best solution is the one with the best fitness value.

5. TUNING WEIGHTING MATRIX

In this section, the effect of the weighting matrix on the performance of a control system is presented and studied with a numerical example. By tuning the weighting matrix, the system performance can be increased by decreasing the system's time response. Both matrices Q and R can be formulated as follows:

$$Q = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix}, \tag{9}$$

$$R = [p_3], \tag{10}$$

where p_1, p_2, and p_3 are the optimization variables that are desired to know their optimal values using the GA. As a numerical example, system dynamics can be described as follows:

$$A = \begin{bmatrix} 0 & 1 \\ 0.01 & 0 \end{bmatrix}, \tag{11}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{12}$$

$$C = [1 \quad 0], \tag{13}$$

$$D = 0. \tag{14}$$

Table 1 illustrates how tuning weighting parameters can affect the performance of the dynamic system, which is described by equations (11) to (14). The performance was recorded as the area under the curve of the response to initial conditions, as described in Fig. 3. The performance of the system can be considered as the inverse value of the area under the state-time curve. In other words, less area under a curve means the best performance. Thus, we search for a combination of p_1, p_2, and p_3 that provide maximum performance value.

The numerical example in Table 1 shows that tuning Q and R matrices is a minimization optimization problem that has a variable size of 3, global optimal is zero, and constraints on the variable can be set:

Minimize:

$$\text{cost}(p_1 \quad p_2 \quad p_3) = \text{Area under the curvein state} \\ - \text{time space}$$

Subject to:

$$0<(p_1 \quad p_2 \quad p_3)<10 \tag{15}$$

The minimization problem was solved using genetic algorithm, as shown in the convergence curve in Fig. 5, where the reduction of the cost function represents an amplification of the efficiency of the control system. The better efficiency can be described by the minimum time required to settle a signal to the steady state, as shown in Figs 6 and 7.

The advantages and disadvantages of employing an optimization algorithm for a control system are competitive depending on the nature of the algorithm itself [25, 26]. However, the population size used for the statistical results in Table 1 is 50, with one independent run with three variable size of integer values. The cost function search space is convex, similar to Fig. 8, which decreases monotonically towards the global point, which is zero.

Table 1. Effect of the weighting matrix on the performance of the system

No.	Parameter p_1	p_2	p_3	Area under state-time curve	Performance
1	1	1	1	17.43	0.05737
2	7	4	3	13.851	0.0721
3	2	10	5	28.595	0.0349
4	10	5	2	11.868	0.0842
5	8	8	8	17.434	0.0573

Figure 5 shows that the genetic algorithm finds the optimal values of $(p_1 \quad p_2 \quad p_3)$ after only 15 trials; it needs approximately 0.2 s. This short time of calculation is suitable for real-time applications, specifically suitable for drone control. The shape of the tuning parameters search space, which is shown in Fig. 8, is convex. In other words, all possible candidate solutions descend toward the optimal value which is located at the bottom of the surface. This type of search spaces is not recommended to be solved by metaheuristics which contain large values of randomness in their search engine. The reason is that randomness can shift the solution upward in the search space by a random value. For the search space shown in Fig. 8, the hill climbing optimization algorithm can be a better choice than using a genetic algorithm. However, this needs a separate comparative study that can be a topic for future work.

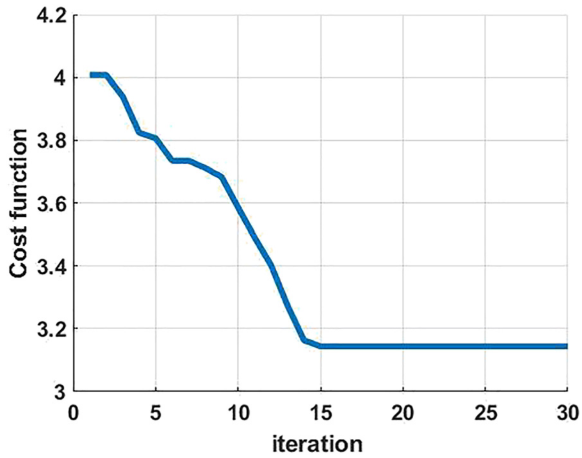

Fig. 5. Performance of the control system

6. QUADCOPTER FLIGHT DYNAMICS

Figure 9 shows the body frame O_b located at the center of mass of the quadcopter, which is related to the inertial frame O_i at the control station on the ground. The vehicle frame on the quadcopter is parallel to the inertial frame while the body frame transforms about the vehicle frame by Euler angles, as shown in Fig. 10.

The state vector is defined as:

$$X = [x, y, z, \phi, \theta, \psi . \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}], \tag{16}$$

where x, y, and z are the position vector of the body frame with respect to the inertial frame while $\phi, \theta,$ and ψ are Euler angles [27]. The transformation matrix from vehicle to body frame R can be expressed as follows:

$$R = \begin{bmatrix} C_\phi C_\theta - S_\phi S_\psi S_\theta & -C_\phi S_\psi & C_\phi S_\theta + C_\theta S_\phi S_\psi \\ C_\theta S_\psi + C_\phi S_\psi S_\theta & C_\phi C_\psi & S_\psi S_\theta - C_\psi C_\theta S_\phi \\ -C_\phi S_\theta & S_\phi & C_\phi C_\theta \end{bmatrix}, \tag{17}$$

where C and S denote cosine and sine functions. According to Newton's second law, the acceleration in the inertial frame will be:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R^T \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \tag{18}$$

where T is the whole thrust force generated by all the propellers, m is the mass of the quadcopter, and g is the gravitational constant. The angular velocities vector ω in the body frame can be calculated using the rates of Euler angles

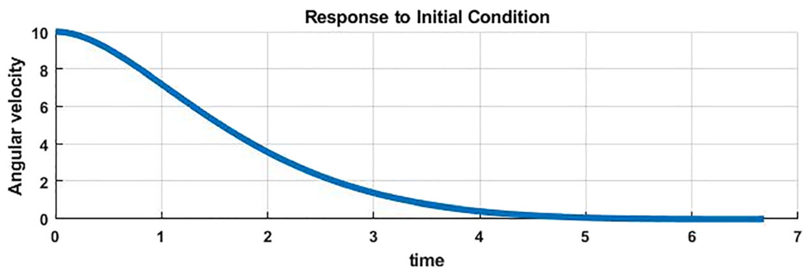

Fig. 6. Time response

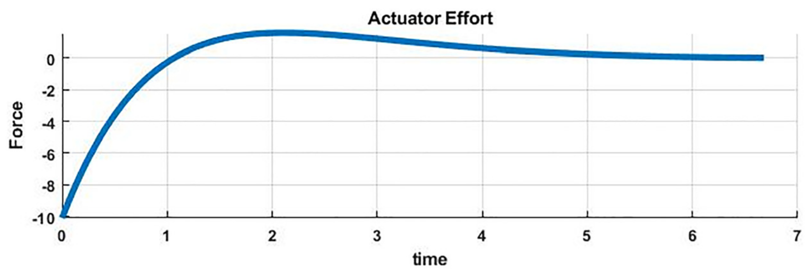

Fig. 7. Input development of the control system

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} C_\theta & 0 & -S_\theta C_\phi \\ 0 & 1 & S_\phi \\ S_\theta & 0 & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (19)$$

By differentiating both sides of equation (19), we get the angular acceleration vector of the quadcopter:

$$\dot{\omega} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = R_\omega \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + A, \quad (20)$$

where

$$R_\omega = \begin{bmatrix} C_\theta & 0 & -S_\theta C_\phi \\ 0 & 1 & S_\phi \\ S_\theta & 0 & C_\phi C_\theta \end{bmatrix}, \quad (21)$$

$$A = \begin{bmatrix} S_\phi S_{\theta\dot{\phi}\dot{\psi}} - C_\theta C_{\phi\dot{\theta}\dot{\psi}} - S_{\theta\dot{\theta}\dot{\phi}} \\ C_{\phi\dot{\phi}\dot{\psi}} \\ -S_\phi C_{\theta\dot{\phi}\dot{\psi}} - S_\theta C_{\phi\dot{\theta}\dot{\psi}} + C_{\theta\dot{\theta}\dot{\phi}} \end{bmatrix}. \quad (22)$$

To simplify equation (20), many research ignores matrix A for both linearized and nonlinear systems [28, 29]. By taking into consideration the Coriolis effect, we can get the angular acceleration by Newton's second law [10]:

$$J\dot{\omega} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - \omega \times J\omega, \quad (23)$$

where τ_ϕ, τ_θ, and τ_ψ are moments along $\widehat{i}^b$, $\widehat{j}^b$, and $\widehat{k}^b$ axis, respectively, which are shown in Fig. 10, while J is the inertia matrix which is a a diagonal:

$$J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (24)$$

where J_{xx}, J_{yy}, and J_{zz} are moments of inertia about $\widehat{i}^b$, $\widehat{j}^b$, and $\widehat{k}^b$, respectively. The system dynamics can be formulated using equations (18), (20), and (23) as follows:

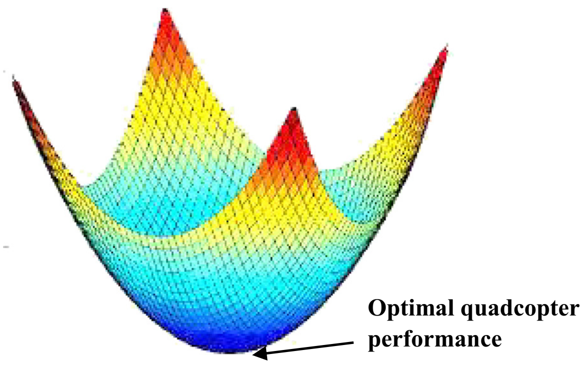

Fig. 8. Tuning parameters search space

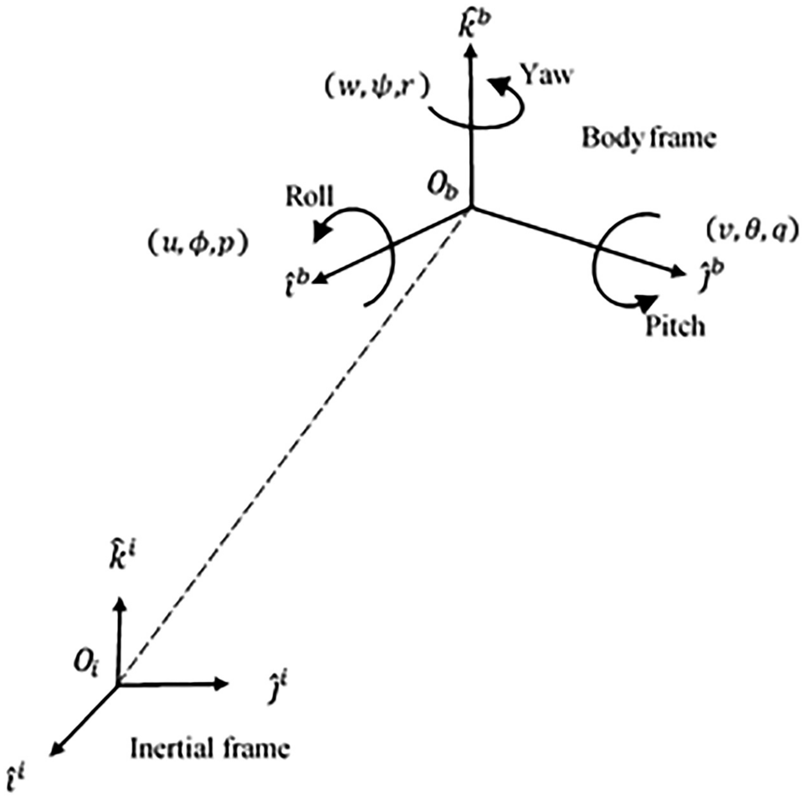

Fig. 9. Quadcopter frames assignment

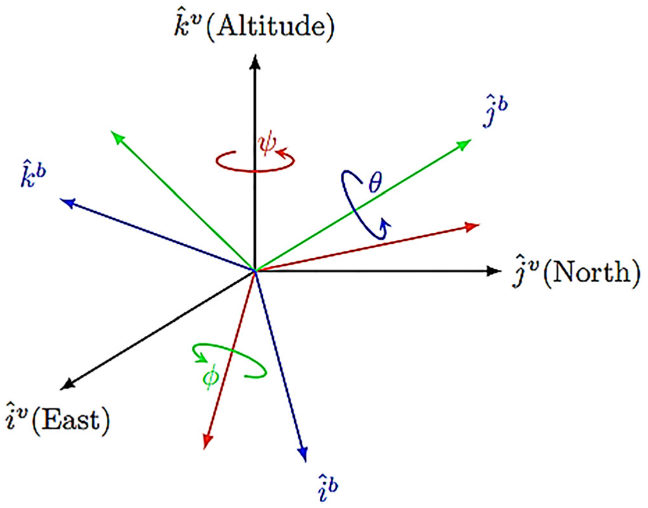

Fig. 10. Transformation between the vehicle frame and body frame

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R^T \begin{bmatrix} 0 \\ 0 \\ \frac{T}{m} \end{bmatrix} \quad (25)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = R_\omega^{-1} J^{-1} \left(\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - \omega \times J\omega \right) - R_\omega^{-1} A \quad (26)$$

7. OPTIMIZATION PROBLEM FORMULATION

The quadcopter can be controlled efficiently only by four inputs which are position $[x \quad y \quad z]$ and yaw angle ψ, while the other twelve states can be determined due to these four given inputs. The reduction of inputs relative to the total number of states is the reason why a quadcopter is an under-actuated system. The goal is to set the Cartesian position and yaw angle of the quadcopter as inputs and get the corresponding angular velocities of the actuators, as shown in Fig. 11.

We can arrange the inputs in the dynamic equations (25) and (26) by assuming:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}, \quad (27)$$

and

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \end{bmatrix}, \quad (28)$$

where u_x, u_y, u_z, u_ϕ, u_θ, and u_ψ are inputs of the control system that we can tune effectively. Now, equation (25) can be rewritten as follows:

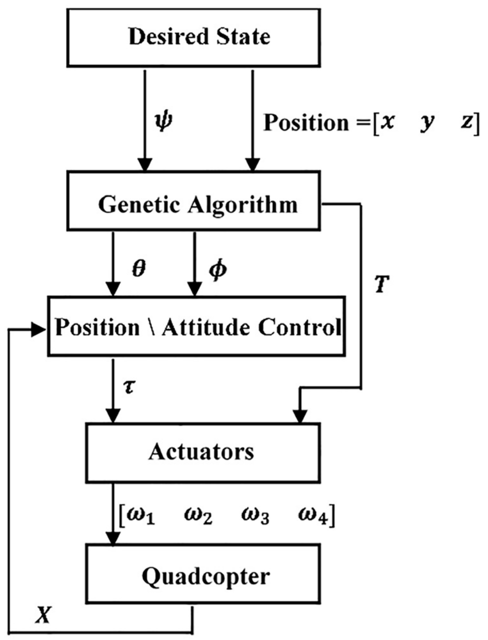

Fig. 11. Schematic diagram of a quadcopter control system

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R^T \begin{bmatrix} 0 \\ 0 \\ \frac{T}{m} \end{bmatrix}. \quad (29)$$

In feedback linearization-based control, equation (29) can be solved for ϕ, θ, and T by arranging equation (29) to be:

$$\begin{bmatrix} C_\phi C_\theta - S_\phi S_\psi S_\theta & -C_\phi S_\psi & C_\psi S_\theta + C_\theta S_\phi S_\psi \\ C_\theta S_\psi + C_\phi S_\psi S_\theta & C_\phi C_\psi & S_\psi S_\theta - C_\psi C_\theta S_\phi \\ -C_\phi S_\theta & S_\phi & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z + g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{T}{m} \end{bmatrix} \quad (30)$$

The zero values in the right-hand side of equation (30) come from the assumption that the quadcopter structure is symmetric, which is necessary to solve equation (30) analytically. The problem is getting worse when the quadcopter has an asymmetric structure. In this case, the right-hand side of equation (30) components will be all nonzero values, and that makes it hard to solve for ϕ, θ, and T. This paper proposes an evolutionary solution using a genetic algorithm for asymmetric quadcopter when the zero values in equation (30) become nonzeros as v_1, and v_2. The objective function that has to be minimized can be written as follows:

Minimize:

$$f = (C_\phi C_\theta - S_\phi S_\psi S_\theta)u_x - (C_\phi S_\psi)u_y + (C_\phi S_\theta + C_\theta S_\phi S_\psi)(u_z + g) - v_1 \quad (31)$$

subjected to

$$(C_\theta S_\psi + C_\phi S_\psi S_\theta)u_x + (C_\phi C_\psi)u_y + (S_\psi S_\theta - C_\psi C_\theta S_\phi)(u_z + g) - v_2 \leq 0, -180 \leq \phi, \theta \leq 180, \quad (32)$$

After finding the solution for ϕ, and θ, it will be easy to find the total thrust force directly:

$$T = m\left[(-C_\phi S_\theta)u_x + (S_\phi)u_y + (C_\phi C_\theta)(u_z + g)\right]. \quad (33)$$

This optimization problem can be minimized by many methods like swarm intelligence [30], biologically inspired algorithms [31], and so on. In other words, this research proposal opens the door for many possible future works that can employ metaheuristic algorithms [32] for control systems.

8. CONCLUSION

The evolutionary control system of a quadcopter introduces an efficient way to control quadcopters that have unsymmetrical structures, which are hard to solve their equations. Thus, the proposed methodology presents a direct way to control both symmetric and asymmetric structures. Building a control system with evolutionary computing is a kind of optimal control of systems. The analytical solution of the dynamic equations of an asymmetric quadcopter cannot be found during feedback linearization-based control. Thus, the evolutionary control system has the advantage of estimating the solutions for the control system. The genetic algorithm is used in this paper to find the optimal values for the weighting matrix. This algorithm has proven that tuning the weighting matrix of a control system is not a time expensive optimization problem. During a fraction of a second, the algorithm can find the optimal values of the weighting matrix that can drive the system to the best performance.

ACKNOWLEDGMENTS

The research was supported by the Hungarian National Research, Development and Innovation Office – NKFIH under project number K 134358.

REFERENCES

[1] J. O. de A. Limaverde Filho, T. S. Lourenço, E. Fortaleza, A. Murilo, and R. V Lopes, "Trajectory tracking for a quadrotor system: a flatness-based nonlinear predictive control approach," in 2016 IEEE Conference on Control Applications (CCA), 2016, pp. 1380–5.

[2] A. L. Salih, M. Moghavvemi, H. A. F. Mohamed, and K. S. Gaeid, "Modelling and PID controller design for a quadrotor unmanned air vehicle," in 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2010, vol. 1, pp. 1–5.

[3] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: a review," *Prog. Aerosp. Sci.*, vol. 91, pp. 99–131, 2017.

[4] R. Schneiderman, "Unmanned drones are flying high in the military/aerospace sector [special reports]," *IEEE Signal Process. Mag.*, vol. 29, no. 1, pp. 8–11, 2011.

[5] H. Fesenko, V. Kharchenko, A. Sachenko, R. Hiromoto, and V. Kochan, "An Internet of Drone-based multi-version post-severe accident monitoring system: structures and reliability," in *Dependable IoT for Human and Industry*. River Publishers, 2022, pp. 197–217.

[6] B. Heincke, et al., "Developing multi-sensor drones for geological mapping and mineral exploration: setup and first results from the MULSEDRO project," *GEUS Bull.*, vol. 43, 2019.

[7] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5, 2004, pp. 4393–8.

[8] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," in 2013 International Conference on Informatics, Electronics and Vision (ICIEV), 2013, pp. 1–6.

[9] S. Formentin and M. Lovera, "Flatness-based control of a quadrotor helicopter via feedforward linearization," in 2011 50th IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 6171–6.

[10] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 2520–5.

[11] J. B. Rawlings and D. Q. Mayne, "Model Predictive Control Theory and Design. 2009", *Madison, WI Nob Hill Publ. LLC.*, 2009.

[12] A. Q. Al-Dujaili, A. J. Humaidi, D. A. Pereira, and I. K. Ibraheem, "Adaptive backstepping control design for ball and beam system," *Int. Rev. Appl. Sci. Eng.*, vol. 12, no. 3, pp. 211–21, 2021.

[13] M. F. Badr, E. H. Karam, and N. M. Mjeed, "Control design of damper mass spring system based on backstepping controller scheme," *Int. Rev. Appl. Sci. Eng.*, 2020.

[14] H. M. Khalid et al., "Optimal control of a quadcopter: a constrained optimization problem," in 2019 8th International Conference on Industrial Technology and Management (ICITM), 2019, pp. 201–5.

[15] J. M. Kiss, P. T. Szemes, and P. Aradi, "Sliding mode control of a servo system in LabVIEW: comparing different control methods," *Int. Rev. Appl. Sci. Eng.*, vol. 12, no. 2, pp. 201–10, 2021.

[16] Y. Şahin and G. Husi, "Design of a Modified linear quadratic regulator for vibration control of suspension systems of military and civil vehicle," *Int. Rev. Appl. Sci. Eng.*, vol. 1, nos 1–2, pp. 55–60, 2010.

[17] F. Hajdu, P. Szalai, P. Mika, and R. Kuti, "Parameter identification of a fire truck suspension for vibration analysis," *Pollack Period.*, vol. 14, no. 3, pp. 51–62, 2019.

[18] E. H. Karam, N. S. Abdul-Jaleel, and B. J. Salah, "Design of hybrid Neuro-Robust deadbeat controller for higher order linear systems based on optimized mixed reduction method," *Int. Rev. Appl. Sci. Eng.*, vol. 11, no. 3, pp. 251–60, 2020.

[19] Y. Şahin and A. Göçer, "Control of air flow temperature and pressure in the pipelines with PID," *Int. Rev. Appl. Sci. Eng.*, 2020.

[20] Z. S. Hashim and I. K. Ibraheem, "A relative degree one modified active disturbance rejection control for four-tank level control system," *Int. Rev. Appl. Sci. Eng.*, 2021.

[21] D. Dumitrescu, B. Lazzerini, L. C. Jain, and A. Dumitrescu, *Evolutionary Computation*. CRC press, 2000.

[22] M. Srinivas and L. M. Patnaik, “Genetic algorithms: a survey,” *Computer (Long. Beach. Calif).*, vol. 27, no. 6, pp. 17–26, 1994.

[23] J. H. Holland, *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press, 1992.

[24] C. Darwin, *On the Origin of Species, 1859*. Routledge, 2004.

[25] H. N. Ghafil, S. Alsamia, and K. Jármai, “Fertilization optimization algorithm on CEC2015 and large scale problems,” *Pollack Period. Int. J. Eng. Inf. Sci.*, vol. 17, no. 1, pp. 24–9, 2022.

[26] S. Alsamia, H. Albedran, and K. Jármai, “Comparative study of different metaheuristics on CEC 2020 benchmarks,” in *Vehicle and Automotive Engineering*, 2023, pp. 709–19.

[27] M. Khan, “Quadcopter flight dynamics,” *Int. J. Sci. Technol. Res.*, vol. 3, no. 8, pp. 130–5, 2014.

[28] A. Joukhadar, M. AlChehabi, C. Stöger, and A. Müller, “Trajectory tracking control of a quadcopter UAV using nonlinear control,” in *Mechanism, Machine, Robotics and Mechatronics Sciences*. Springer, 2019, pp. 271–85.

[29] M. Navabi and H. Mirzaei, “θ-D based nonlinear tracking control of quadcopter,” in 2016 4th International Conference on Robotics and Mechatronics (ICROM), 2016, pp. 331–6.

[30] H. N. Ghafil and K. Jármai, “Dynamic differential annealed optimization: new metaheuristic optimization algorithm for engineering applications,” *Appl. Soft Comput.*, 2020, Paper no. 106392.

[31] H. Ghafil and K. Jármai, “Comparative study of particle swarm optimization and artificial bee colony algorithms,” in *Multiscience XXXII. MicroCAD International Multidisciplinary Scientific Conference*. Hungary, Miskolc-Egyetemváros, 2018, pp. 1–6. [Online]. Available: http://real.mtak.hu/84332/1/D1_Hazim_Nasir_Ghafil.pdf.

[32] H. N. Ghafil and K. Jármai, *Optimization for Robot Modelling with MATLAB*. Springer Nature, 2020.