Linear Quadratic Integral Differential Game applied to the Real-time Control of a 3DoF Experimental setup of a Quadrotor

Hadi Nobahari
Department of Aerospace Engineering
Sharif University of Technology
Tehran, Iran
nobahari@sharif.edu

Ali BaniAsad
Department of Aerospace Engineering
Sharif University of Technology
Tehran, Iran
alibaniasad1999@yahoo.com

***Abstract*—The accurate attitude control of a quadrotor is necessary, especially when facing disturbance. In this study, a linear quadratic with integral action based on the differential game theory is implemented on a three-degree-of-freedom experimental setup of a quadrotor. For this purpose, first, a continuous state-space model of the quadrotor is derived based on the linearization of the nonlinear equations of motion, and the parameters of the state-space model structure are identified with the experimental results. Then, the attitude control commands of the quadrotor are derived based on two players; one finds the best attitude control command, and the other creates the disturbance by minimizing a quadratic criterion, defined as the sum of outputs plus the weighted control effort. The performance of the proposed controller is evaluated in level flight and compared to the linear quadratic regulator controller. Results demonstrate that the proposed approach has an excellent performance in dissipating the disturbances created by the modeling error.**

***Index Terms*—Linear Quadratic Differential Game, Quadrotor, Real-time, 3DoF Experimental setup, Optimal Control, Robust Control**

I. Introduction

A quadcopter is a type of helicopter with four rotors. Quadcopters have extensive applications due to their excellent maneuverability and the possibility of hover flight with high balance. In recent years, companies, universities, and research centers have attracted more to this type of UAV. In this way, the facilities and the flight of these UAVs are continuously improving. Quadcopters are widely used in research, military, imaging, recreation, and agriculture. Mathematical models are used in game theory to examine how rational, intelligent beings cooperate or compete. Game theory can be applied to pursuit and evasion as one of its broad applications. There can be two [1] or more players [2] involved in the pursuit-evasion. Pursuit-evasion can occur indoors as well [3]. In some cases, machine learning and differential games pursuit-evade [4]. Players may play different roles in differential games, such as protecting some targets [5]. The differential game's ability to examine the actions of two or more players makes it powerful. Player cooperation can be used through swarm platooning [6]. Multi-agent [7] and self-driving automobiles [8] motion planning are two other applications of player cooperation.

Due to the widespread use of quadrotors, their control has become an important issue. In order to control quadrotors, neural networks [9] and machine learning [10] methods have been used. Two uses for quadrotor control include swarm flying [12] and motion planning [11]. In [13], Kyuman Lee, Daegyun Choi, and Donghoon Kim worked on Motion Planning for Quadcopters in Three-Dimensional Dynamic Environments with Potential Fields-Aided. To avoid collisions with obstacles, the controller should control the quadrotor to prevent collisions [14].

In partnership with BaniAsad Industries

II. Problem Statement

In this section, a nonlinear dynamic is presented for an experimental setup of a quadrotor, as shown in Fig.1. The quadrotor is free to rotate about its roll, pitch, and yaw axes. The Euler angle angles and angular velocities along three orthogonal axes are measured simultaneously using Attitude and Heading Reference Systems (AHRS). LQDG utilizes these noisy measurements for real-time control of the Euler angles. The block diagram of the control purpose is shown in Fig.2.

Fig. 1. 3DoF experimental setup of a quadrotor

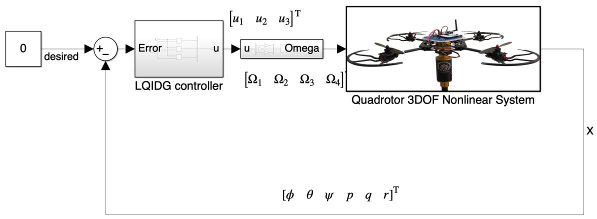


Fig. 2. block diagram of the control purpose

III. Modeling of an Experimental Setup of a Quadrotor

A. Configuration of the Quadrotor

The schematic of a quadrotor is given in Fig.3. Each rotor is considered a rigid disk is rotating about the axis Z_B in the body fixed frame with an angular velocity Ω_i. Rotors 1 and 3 rotate in the same direction, i.e., counterclockwise, while rotors 2 and 4 rotate in the opposite direction, i.e., clockwise, to cancel yawing moment of the quadrotor.

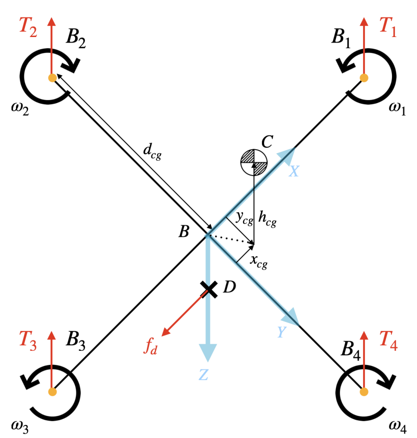


Fig. 3. Configuration of the quadrotor

B. Dynamic Model

The dynamic model of the quadrotor, obtained from the Newton-Euler method, is stated as follows [15], [16]:

$$\begin{aligned}
\dot{p} &= \frac{I_{yy}-I_{zz}}{I_{xx}}qr + q\frac{I_{\text{rotor}}}{I_{xx}}\Omega_r + \frac{u_{\text{roll}}}{I_{xx}} + \frac{d_{\text{roll}}}{I_{xx}} \\
\dot{q} &= \frac{I_{zz}-I_{zz}}{I_{yy}}rp + p\frac{I_{\text{rotor}}}{I_{xx}}\Omega_r + \frac{u_{\text{pitch}}}{I_{yy}} + \frac{d_{\text{pitch}}}{I_{yy}} \\
\dot{r} &= \frac{I_{xx}-I_{yy}}{I_{zz}}pq + \frac{u_{\text{yaw}}}{I_{zz}} + \frac{d_{\text{yaw}}}{I_{zz}}
\end{aligned} \tag{1}$$

where d_{roll}, d_{pitch}, and d_{yaw} are roll, pitch, and yaw moments, generated by the disturbance and (p, q, r) are the angular velocities, and (ϕ, θ, ψ) are roll, pitch, and yaw angles. The relation between Euler angles rates and the angular body rates are obtained as follows:

$$\begin{aligned}
\dot{\phi} &= p + q\sin(\phi)\cos(\theta) + r\cos(\phi)\tan(\theta) \\
\dot{\theta} &= q\cos(\phi) - r\sin(\phi) \\
\dot{\psi} &= (q\sin(phi) + r\cos(\phi))\sec(\theta)
\end{aligned} \tag{2}$$

where I_{xx}, I_{yy}, and I_{zz} are the principal moment of inertia and I_{rotor} is the inertia of a rotor about its axis. Moreover, Ω_r, called the overall residual propeller angular speed, is computed as:

$$\Omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4 \tag{3}$$

The control inputs u_{roll}, u_{pitch}, and u_{yaw} are roll, pitch, and yaw moments, generated by the propellers, defined as:

$$\begin{aligned}
u_{\text{roll}} &= bd_{cg}(\Omega_2^2 - \Omega_4^2) \\
u_{\text{pitch}} &= bd_{cg}(\Omega_1^2 - \Omega_3^2) \\
u_{\text{yaw}} &= d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)
\end{aligned} \tag{4}$$

Also, b and d are thrust and drag coefficients, respectively, and d_{cg} is the horizontal distance of each rotor from the center of gravity, as shown in Fig.3. Therefore, the angular velocity commands are obtained as:

$$\begin{aligned}
\Omega_{c,1}^2 &= \Omega_{\text{mean}}^2 + \frac{1}{2bd_{cg}}u_{\text{pitch}} + \frac{1}{4d}u_{\text{yaw}} \\
\Omega_{c,2}^2 &= \Omega_{\text{mean}}^2 + \frac{1}{2bd_{cg}}u_{\text{roll}} - \frac{1}{4d}u_{\text{yaw}} \\
\Omega_{c,3}^2 &= \Omega_{\text{mean}}^2 - \frac{1}{2bd_{cg}}u_{\text{pitch}} + \frac{1}{4d}u_{\text{yaw}} \\
\Omega_{c,4}^2 &= \Omega_{\text{mean}}^2 - \frac{1}{2bd_{cg}}u_{\text{roll}} - \frac{1}{4d}u_{\text{yaw}}
\end{aligned} \tag{5}$$

where Ω_{mean} is the average angular velocity of the rotors.

C. State-Space Form

Here, the state-space model of the experimental setup of the quadrotor is presented for the control purpose. by defining $x_1 = p$, $x_2 = q$, $x_3 = r$, $x_4 = \phi$, $x_5 = \theta$, and $x_6 = \psi$; the model of the experimental setup in state-space form are expressed as:

$$\begin{aligned}
\dot{x}_1 &= \frac{I_{yy}-I_{zz}}{I_{xx}}x_2x_3 + x_2\frac{I_{\text{rotor}}}{I_{xx}}\Omega_r + \frac{u_{\text{roll}}}{I_{xx}} + \frac{d_{\text{roll}}}{I_{xx}} \\
\dot{x}_2 &= \frac{I_{zz}-I_{zz}}{I_{yy}}x_1x_3 - x_1\frac{I_{\text{rotor}}}{I_{xx}}\Omega_r + \frac{u_{\text{pitch}}}{I_{yy}} + \frac{d_{\text{pitch}}}{I_{yy}} \\
\dot{x}_3 &= \frac{I_{xx}-I_{yy}}{I_{zz}}x_1x_2 + \frac{u_{\text{yaw}}}{I_{zz}} + \frac{d_{\text{yaw}}}{I_{zz}} \\
\dot{x}_4 &= x_1 + x_2\sin(x_4)\cos(x_5) + x_3\cos(x_4)\tan(x_5) \\
\dot{x}_5 &= x_2\cos(x_4) - x_3\sin(x_4) \\
\dot{x}_6 &= (x_2\sin(x_4) + x_3\cos(x_4))\sec(x_5)
\end{aligned} \tag{6}$$

The measurement model is written as:

$$\begin{aligned}
z &= \begin{bmatrix} p_m & q_m & r_m & \phi_m & \theta_m & \psi_m \end{bmatrix}^{\text{T}} \\
&= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^{\text{T}}
\end{aligned} \tag{7}$$

D. Linear Model

The continuous-time linear model is utilized to drive the control commands on the quadrotor. The linear state-space model is denoted as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{B_d}\mathbf{d}(t) \tag{8}$$

where d is the unknown input. $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{B_d}$ are the system input and unknown input matrices, respectively. Moreover, the measurements equation is stated as:

$$\mathbf{z}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{D_d}\mathbf{d}(t) \tag{9}$$

where $\mathbf{C}$ is the output matrix. Also, $\mathbf{D}$ and $\mathbf{D_d}$ are the feedforward matrices due to known and unknown inputs, respectively. According to eq ? - ?, the linear dynamic model around the equilibrium points ($\mathbf{x}_e = 0$ and $\mathbf{u} = 0$) of the quadrotor setup is denoted as:

$$\begin{aligned}\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{x}}_{\text{roll}}\\ \dot{\mathbf{x}}_{\text{pitch}}\\ \dot{\mathbf{x}}_{\text{yaw}}\end{bmatrix} &= \begin{bmatrix}\mathbf{A}_{\text{roll}} & 0 & 0\\ 0 & \mathbf{A}_{\text{pitch}} & 0\\ 0 & 0 & \mathbf{A}_{\text{yaw}}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{\text{roll}}\\ \mathbf{x}_{\text{pitch}}\\ \mathbf{x}_{\text{yaw}}\end{bmatrix}\\ &+ \begin{bmatrix}\mathbf{B}_{\text{roll}} & 0 & 0\\ 0 & \mathbf{B}_{\text{pitch}} & 0\\ 0 & 0 & \mathbf{B}_{\text{yaw}}\end{bmatrix}\begin{bmatrix}\mathbf{u}_{\text{roll}}\\ \mathbf{u}_{\text{pitch}}\\ \mathbf{u}_{\text{yaw}}\end{bmatrix}\\ &+ \begin{bmatrix}\mathbf{B}_{\text{roll}} & 0 & 0\\ 0 & \mathbf{B}_{\text{pitch}} & 0\\ 0 & 0 & \mathbf{B}_{\text{yaw}}\end{bmatrix}\begin{bmatrix}\mathbf{d}_{\text{roll}}\\ \mathbf{d}_{\text{pitch}}\\ \mathbf{d}_{\text{yaw}}\end{bmatrix}\end{aligned} \tag{10}$$

where $x_{\text{roll}} = \begin{bmatrix}p & \phi\end{bmatrix}^{\text{T}}$, $x_{\text{pitch}} = \begin{bmatrix}q & \theta\end{bmatrix}^{\text{T}}$, and $x_{\text{yaw}} = \begin{bmatrix}r & \psi\end{bmatrix}^{\text{T}}$. Also $d = \begin{bmatrix}x\end{bmatrix}^{\text{T}}$, is the .... Moreover, the state and input matrices are derived as:

$$\mathbf{A}_{\text{roll}} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix};\ \mathbf{A}_{\text{pitch}} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix};\ \mathbf{A}_{\text{yaw}} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} \tag{11}$$

$$\mathbf{B}_{\text{roll}} = \begin{bmatrix}0\\ \dfrac{1}{I_{xx}}\end{bmatrix};\ \mathbf{B}_{\text{pitch}} = \begin{bmatrix}0\\ \dfrac{1}{I_{yy}}\end{bmatrix};\ \mathbf{B}_{\text{yaw}} = \begin{bmatrix}0\\ \dfrac{1}{I_{zz}}\end{bmatrix} \tag{12}$$

E. Parameter Estimation

This section modifies the quadrotor stand parameters using the Simulink environment's simulation of different quadrotor channels and the stand's output data. The quadrotor stand parameters have been modified using the Parameter Estimator toolbox available in the Simulink environment. In order to perform the test, the quadrotor stand was released from various initial conditions and inputs, and data was collected using the output from the sensor. Then, Parameter Estimator takes the model and the recorded data of the sensor (stand states). Here is a comparison between the states of the quadrotor in simulation and reality after modifying various parameters.

TABLE I
PARAMETER ESTIMATION RESULTS

Parameter	*Initial Value*	*Value After Estimation*
A_1	7.312	4.152
A_3	1.1×10^{-4}	5.47×10^{-5}
B_1	4.53	4.36
B_3	1.1×10^{-4}	7.13×10^{-5}
C_2	5.45×10^{-5}	1.3×10^{-5}

IV. DIFFERENTIAL GAME

Differential games are a series of problems that arise while examining and simulating dynamic systems in game theory. Differential equations simulate how a state variable or set of state variables changes over time.

A. Differential Game Usage in a Quadrotor Control Loop

This paper describes the state of two players in different loop control of a quadrotor. Three groups of players are identified: two players for roll loop control, two players for pitch loop control, and two players for yaw loop control. The space state of roll, pitch, and yaw are defined below.

$$\begin{aligned}&\dot{\mathbf{x}}_{\mathbf{i}}(t) = \mathbf{A_i}\mathbf{x_i}(t) + \mathbf{B_i}\mathbf{u_i}(t) + \mathbf{B_{i_d}}\mathbf{u_{i_d}}(t)\\ &\mathbf{y_i}(t) = \mathbf{C_i}\mathbf{x_i}(t) + \mathbf{D_i}\mathbf{u_i}(t) + \mathbf{D_{i_d}}\mathbf{u_{i_d}}(t)\\ &i = 1,2,3\end{aligned} \tag{13}$$

Where $\mathbf{x}$ is the vector of the state variables, $\dot{\mathbf{x}}$ is the time derivative of the state vector, $\mathbf{u}$ is the controller input vector, $\mathbf{u_d}$ is the disturbance input vector, $\mathbf{y}$ is the output vector, $\mathbf{A}$ is the state matrix, $\mathbf{B}$ is the controller input matrix, $\mathbf{B_d}$ is the disturbance input matrix, $\mathbf{C}$ is the output matrix, $\mathbf{D}$ is controller the output matrix and $\mathbf{D_d}$ is disturbance the output matrix. Equation (13) demonstrates how both participants have an impact on the quadrotor's dynamics. The second player may progress toward the goal as a result of the first player's exertion, or vice versa. This paper considers the case that players do not cooperate in order to realize their goals. In this case, every player knows at time $t \in [0, T]$ just the initial state $\mathbf{x_0}$ and the model structure. For the game between two players in each loop control, the set of Nash equilibria is used. Formal Nash equilibrium is defined as follows. An admissible set of actions $(\boldsymbol{u_1}^*, \boldsymbol{u_{i_d}}^*)$ is a Nash equilibrium for the game between two player in each loop control; if for all admissible $(\boldsymbol{u_i}, \boldsymbol{u_{i_d}})$, the following inequalities hold:

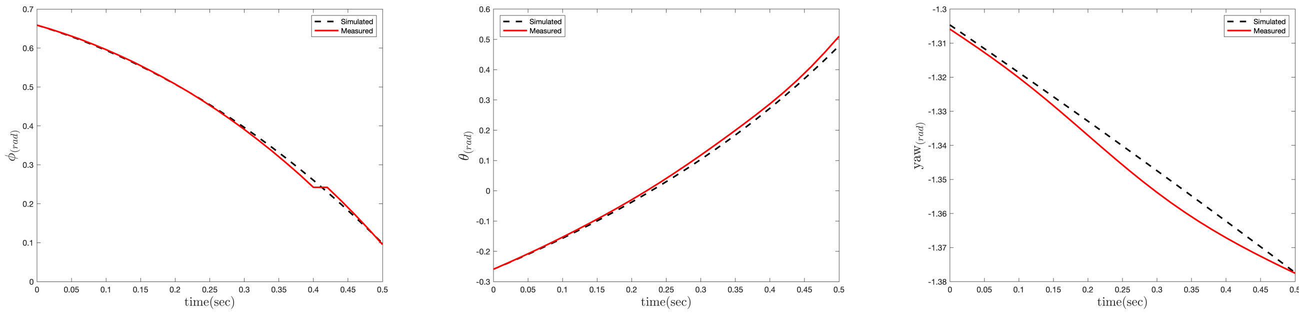
$$J_1(\mathbf{u_i}^*, \mathbf{u_{i_d}}^*) \le J_1(\mathbf{u_1}, \mathbf{u_{i_d}}^*)\ ,\ J_2(\mathbf{u_i}^*, \mathbf{u_{i_d}}^*) \le J_2(\mathbf{u_i}^*, \mathbf{u_{i_d}}) \tag{14}$$

B. LQDG controller

For the each control loop described in equation (13), LQDG optimum control effort calculates from equation (15).

$$\mathbf{u_i}(t) = -\mathbf{R_i}^{-1}\mathbf{B_i}^{\text{T}}\mathbf{P_i}(t)\mathbf{x}(t) = -\mathbf{K_i}(t)\mathbf{x}(t), \quad i = 1,2,3 \tag{15}$$

In equation (15), $\mathbf{K_i}$ is the optimal feedback gain. Assuming that the other players will make their worst move, this gain



(a) Roll Channel (b) Pitch Channel (c) Yaw Channel

Fig. 4. Comparison of quadrotor states in simulation and reality.

is calculated to minimize the quadratic cost function equation (16) of controller player for each control loop of quadrotor.

$$J_i(\mathbf{u_i}, \mathbf{u_{i_d}}) = \int_0^T \Big(\mathbf{x_i}^{\mathrm{T}}(t)\mathbf{Q_i}\mathbf{x_i}(t) + \mathbf{u_i}^{\mathrm{T}}(t)\mathbf{R_i}\mathbf{u_i}(t) + \mathbf{u_{i_d}}^{\mathrm{T}}(t)\mathbf{R_{i_d}}\mathbf{u_{i_d}}(t)\Big)\mathrm{d}t, \quad i = 1, 2, 3 \tag{16}$$

Here the matrices $\mathbf{Q_i}$ and $\mathbf{R_i}$ are assumed to be symmetric and $\mathbf{R_i}$ positive definite. $\mathbf{P_i}$ is found by solving the continuous time couple Riccati differential equation:

$$\begin{aligned}\dot{\mathbf{P}}_{\boldsymbol{i}}(t) &= -\mathbf{A_i}^{\mathrm{T}}\mathbf{P_i}(t) - \mathbf{P_i}(t)\mathbf{A_i} - \mathbf{Q_i} + \mathbf{P_i}(t)\mathbf{S_i}(t)\mathbf{P_i}(t) + \\ &\quad \mathbf{P_i}(t)\mathbf{S_{i_d}}(t)\mathbf{P_{i_d}}(t) \\ \dot{\mathbf{P}}_{\boldsymbol{i_d}}(t) &= -\mathbf{A_i}^{\mathrm{T}}\mathbf{P_{i_d}}(t) - \mathbf{P_{i_d}}(t)\mathbf{A_i} - \mathbf{Q_{i_d}} + \\ &\quad \mathbf{P_{i_d}}(t)\mathbf{S_{i_d}}(t)\mathbf{P_{i_d}}(t) + \mathbf{P_{i_d}}(t)\mathbf{S_i}(t)\mathbf{P_i}(t)\end{aligned} \tag{17}$$

Using the shorthand notation $\boldsymbol{S_i} := \boldsymbol{B_i}\boldsymbol{R_i}^{-1}\boldsymbol{B_i}^{\mathrm{T}}$.

C. LQIDG controller

The absence of an integrator in the LQDG controller may result in steady-state errors due to disturbances or modeling errors. The LQIDG controller is based on the LQDG controller to eliminate this error.

The LQIDG controller adds the integral of the difference between the system output and the desired value to the state vector. Therefore, The augmented space states of a continuous linear system are shown below.

$$\mathbf{x_a} = \begin{bmatrix} \mathbf{x_d} - \mathbf{x} \\ \int (\mathbf{y_d} - \mathbf{y}) \end{bmatrix} \tag{18}$$

Where $\mathbf{x_a}$ is the vector of augmented state variables, $\mathbf{x_d}$ is the vector of the desired state variables, and $\mathbf{y_d}$ is the desired output vector. As a result, the state vector and the output vector are equal.

$$\mathbf{y} = \mathbf{x} \tag{19}$$

The following represents the system dynamics in the augmented state space.

$$\dot{\mathbf{x}}_{\boldsymbol{a}}(t) = \mathbf{A_a}\mathbf{x_a}(t) + \mathbf{B_{a_1}}\mathbf{u_{a_1}}(t) + \mathbf{B_{a_2}}\mathbf{u_{a_2}}(t) \tag{20}$$

Where matrices $\mathbf{A_a}$ and $\mathbf{B_a}$ are defined as follows:

$$\mathbf{A_a} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix}, \quad \mathbf{B_a} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \tag{21}$$

By introducing a new space state for the system, the remaining design phases of the LQIDG controller are comparable to those of the LQDG controller. LQIDG optimum control effort calculates from equation (22).

$$\begin{aligned}\mathbf{u_i}(t) &= -\mathbf{R_{ii}}^{-1}\mathbf{B_{a_i}}^{\mathrm{T}}\mathbf{P_{a_i}}(t)\mathbf{x_a}(t) \\ \mathbf{u_i}(t) &= -\mathbf{K_{a_i}}(t)\mathbf{x_a}(t), \ i = 1, 2, 3\end{aligned} \tag{22}$$

In equation (22), $\mathbf{K_{a_i}}$ is the optimal feedback gain. Assuming that the other players will make their worst move, this gain is calculated to minimize the quadratic cost function, equation (23), of player number i.

$$J_i(\mathbf{u_i}, \mathbf{u_{i_d}}) = \int_0^T \Big(\mathbf{x_a}^{\mathrm{T}}(t)\mathbf{Q_i}\mathbf{x_a}(t) + \mathbf{u_i}^{\mathrm{T}}(t)\mathbf{R_i}\mathbf{u_i}(t) + \mathbf{u_{i_d}}^{\mathrm{T}}(t)\mathbf{R_{i_d}}\mathbf{u_{i_d}}(t)\Big)\mathrm{d}t \tag{23}$$

$\dot{\mathbf{P}}_{\boldsymbol{a_i}}$ is found by solving the continuous time couple Riccati differential equation:

$$\begin{aligned}\dot{\mathbf{P}}_{\boldsymbol{a_i}}(t) &= -\mathbf{A_a}^{\mathrm{T}}\mathbf{P_{a_i}}(t) - \mathbf{P_{a_i}}(t)\mathbf{A_a} - \mathbf{Q_i} + \\ &\quad \mathbf{P_{a_i}}(t)\mathbf{S_{a_i}}(t)\mathbf{P_{a_i}}(t) + \mathbf{P_{a_i}}(t)\mathbf{S_{a_{i_d}}}(t)\mathbf{P_{a_{i_d}}}(t) \\ \dot{\mathbf{P}}_{\boldsymbol{a_{i_d}}}(t) &= -\mathbf{A_a}^{\mathrm{T}}\mathbf{P_{a_{i_d}}}(t) - \mathbf{P_{a_{i_d}}}(t)\mathbf{A_a} - \mathbf{Q_{i_d}} + \\ &\quad \mathbf{P_{a_{i_d}}}(t)\mathbf{S_{a_{i_d}}}(t)\mathbf{P_{a_{i_d}}}(t) + \mathbf{P_{a_{i_d}}}(t)\mathbf{S_{a_i}}(t)\mathbf{P_{a_i}}(t)\end{aligned} \tag{24}$$

Using the shorthand notation $\boldsymbol{S_{a_i}} := \boldsymbol{B_{a_i}}\boldsymbol{R_i}^{-1}\boldsymbol{B_{a_i}}^{\mathrm{T}}$.

V. Simulation

In this section, the quadrotor roll loop control is simulated in the presence of LQR, LQDG, and LQIDG controllers. Then, the simulation of two and three degrees of freedom was done in the presence of the LQIDG controller.

A. Roll Loop Control

LQR weighting matrices are optimized using the TCACS optimization method in the simulation. ITSE is considered for the TCACS input cost function. Here are the weighting matrices for the optimized output.

$$\mathbf{Q}_{\text{LQR}} = \begin{bmatrix} 0.5215 & 0 \\ 0 & 0.0745 \end{bmatrix}, \quad R_{\text{LQR}} = 0.0001 \tag{25}$$

The weighting matrices used in the LQDG portion are chosen like that of the LOR.

$$\mathbf{Q}_{\text{LQDG}} = \begin{bmatrix} 100 & 0 \\ 0 & 0.078 \end{bmatrix}, \quad R_{\text{LQDG}} = 1, \quad R_{d_{\text{LQDG}}} = 99.96 \tag{26}$$

$$\mathbf{K_1} = \begin{bmatrix} 39.1188 & 8.8510 \end{bmatrix} \tag{27}$$

LQIDG weighting matrices are chosen like the method used in the LQR and LQDG sections.

$$\mathbf{Q}_{\text{LQIDG}} = \begin{bmatrix} 0.1707 & 0 & 0 & 0 \\ 0 & 0.12 & 0 & 0 \\ 0 & 0 & 837.8606 & 0 \\ 0 & 0 & 0 & 756.1341 \end{bmatrix} \tag{28}$$
$$R_{\text{LQIDG}} = 1, \quad R_{d_{\text{LQIDG}}} = 7.7422$$

$$\mathbf{K_{a_1}} = \begin{bmatrix} 28.1410 & 8.4017 & 27.2223 & 11.6894 \end{bmatrix} \tag{29}$$

B. Roll-Pitch Loop Control

$$\mathbf{Q}_{\text{LQIDG}_{\text{roll}}} = \begin{bmatrix} 585.9 & 0 & 0 & 0 \\ 0 & 31.1 & 0 & 0 \\ 0 & 0 & 83.8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathbf{Q}_{\text{LQIDG}_{\text{pitch}}} = \begin{bmatrix} 546.5 & 0 & 0 & 0 \\ 0 & 311.4 & 0 & 0 \\ 0 & 0 & 2.22 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{30}$$
$$R_{\text{LQIDG}} = 1, \quad R_{d_{\text{LQIDG}}} = 7.7422$$

C. Roll-Pitch-Yaw Loop Control

$$\mathbf{Q}_{\text{LQIDG}_{\text{roll}}} = \begin{bmatrix} 631.85 & 0 & 0 & 0 \\ 0 & 214.28 & 0 & 0 \\ 0 & 0 & 7.91 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$
$$\mathbf{Q}_{\text{LQIDG}_{\text{pitch}}} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 873.93 & 0 & 0 \\ 0 & 0 & 9853.09 & 0 \\ 0 & 0 & 0 & 0.12 \end{bmatrix} \tag{31}$$
$$\mathbf{Q}_{\text{LQIDG}_{\text{yaw}}} = \begin{bmatrix} 0.03 & 0 & 0 & 0 \\ 0 & 0.17 & 0 & 0 \\ 0 & 0 & 1.81 & 0 \\ 0 & 0 & 0 & 0.45 \end{bmatrix} \times 10^{-4}$$
$$R_{\text{LQIDG}} = 1, \quad R_{d_{\text{LQIDG}}} = 1.2577$$

VI. Result and Discussion

Here, the results of the LQIR-DG controller method are devoted to the control loops of the roll, pitch, and yaw of the experimental setup of the quadrotor. First, the controller parameters are tuned using the results of numerical simulations. Moreover, the performance of the LQIR-DG controller is compared to an LQR control strategy. The LQR controller parameters are presented in Table II.

TABLE II
PARAMETERS OF THE LQR CONTROLLER

Control Loop	*Initial Value*	*Value After Estimation*
A_1	7.312	4.152
A_3	1.1×10^{-4}	5.47×10^{-5}
B_1	4.53	4.36
B_3	1.1×10^{-4}	7.13×10^{-5}
C_2	5.45×10^{-5}	1.3×10^{-5}

A. Performance of the LQIR-DG Controller

The performance of the LQIR-DG controller is evaluated in Figure 8. The desired and actual outputs, including the roll, pitch, and yaw angles, are compared in Figures 8 (a) and (b). The desired scenario of the simulator is considered a level flight. These figures show that the attitude outputs of the quadrotor converge to the desired values in less than three seconds. Moreover, Figures 8 (c) and (d) show the angular velocity command of the quadrotor, respectively. These results illustrate that the LQIR-DG approach appropriately controls the attitude of the experimental setup of the quadrotor.

B. Comparison with Well-known Control Strategies

Here, the LQIR-DG controller performance is compared with famous control strategies such as the LQR Figure 10. Figure 10 (a) compares the quadrotor's desired and actual pitch angle in the presence of these controllers. These results indicate that the LQIR-DG controller can provide high tracking performance, such as good transient response and high rapid convergence relative to other controllers for pitch angle control of the experimental setup of the quadrotor.

VII. Conclusion

In this study, a linear quadratic with integral action based on the differential game theory, called LQIR-DG, was implemented for level attitude control in an experimental setup of a quadrotor. To implement the proposed controller structure, first, an accurate dynamic model of the quadrotor was linearized in the state-space form, and then the model parameters were estimated. Next, two players were considered for each of the quadrotor's roll, pitch, and yaw channels. The first player found the best control command for each channel of the setup of a quadrotor based on the minimization of a quadratic criterion; when the second player produced the disturbances. Finally, the performance of the proposed controller was evaluated in level flight and compared to the LQR controller. The implementation results verify the successful performance of the LQIR-DG method in the level of attitude control of the experimental setup of the quadrotor.

References

[1] Weintraub, I. E., Pachter, M., Garcia, E. (2020). An Introduction to Pursuit-evasion Differential Games. 2020 American Control Conference (ACC), 1049–1066. https://doi.org/10.23919/ACC45564.2020.9147205

[2] Garcia, E., Casbeer, D. W., Pachter, M. (2020). Optimal Strategies for a Class of Multi-Player Reach-Avoid Differential Games in 3D Space. IEEE Robotics and Automation Letters, 5(3), 4257–4264. https://doi.org/10.1109/LRA.2020.2994023

[3] [1]H. Lai, W. Liang, R. Yan, Z. Shi, and Y. Zhong, "LiDAR-Inertial based Localization and Perception for Indoor Pursuit-Evasion Differential Games," in 2021 40th Chinese Control Conference (CCC), 2021, pp. 7468–7473. doi: 10.23919/CCC52363.2021.9549330.

[4] F. Jiang, X. Guo, X. Zhang, Z. Zhang, and D. Dong, "Approximate Soft Policy Iteration Based Reinforcement Learning for Differential Games with Two Pursuers versus One Evader," in 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM), 2020, pp. 471–476. doi: 10.1109/ICARM49381.2020.9195328.

[5] M. He and X. Wang, "Nonlinear Differential Game Guidance Law for Guarding a Target," in 2020 6th International Conference on Control, Automation and Robotics (ICCAR), 2020, pp. 713–721. doi: 10.1109/ICCAR49639.2020.9108001.

[6] A. Yildiz and H. B. Jond, "Vehicle Swarm Platooning as Differential Game," in 2021 20th International Conference on Advanced Robotics (ICAR), 2021, pp. 885–890. doi: 10.1109/ICAR53236.2021.9659431.

[7] D. Fridovich-Keil, V. Rubies-Royo, and C. J. Tomlin, "An Iterative Quadratic Method for General-Sum Differential Games with Feedback Linearizable Dynamics," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 2216–2222. doi: 10.1109/ICRA40945.2020.9196517.

[8] T. Kessler, K. Esterle, and A. Knoll, "Linear Differential Games for Cooperative Behavior Planning of Autonomous Vehicles Using Mixed-Integer Programming," in 2020 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 4060–4066. doi: 10.1109/CDC42340.2020.9304495.

[9] S. Edhah, S. Mohamed, A. Rehan, M. AlDhaheri, A. AlKhaja, and Y. Zweiri, "Deep Learning Based Neural Network Controller for Quad Copter: Application to Hovering Mode," in 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019, pp. 1–5. doi: 10.1109/ICECTA48151.2019.8959776.

[10] R. G. do Nascimento, K. Fricke, and F. Viana, "Quadcopter Control Optimization through Machine Learning," in AIAA Scitech 2020 Forum, doi: 10.2514/6.2020-1148.

[11] V. Sumathy‘, R. Warier, and D. Ghose, "Design, Reachability Analysis, and Constrained Motion Planning for a Quadcopter Manipulator System," in AIAA SCITECH 2022 Forum, doi: 10.2514/6.2022-0269.

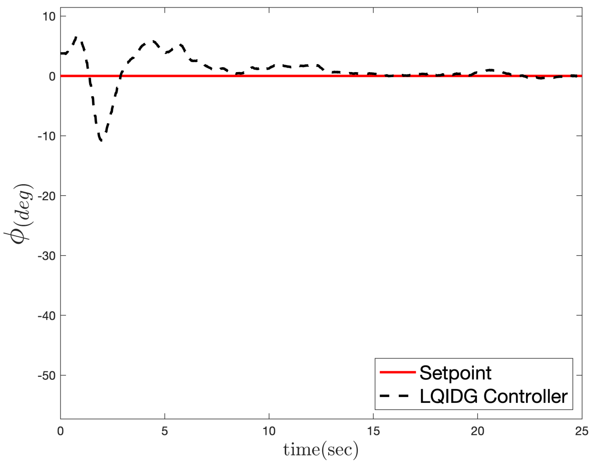
[12] Z. Liang, H. Rastgoftar, and E. M. Atkins, "Multi-Quadcopter Team Leader Path Planning Using Particle Swarm Optimization," in AIAA Aviation 2019 Forum, doi: 10.2514/6.2019-3258.

[13] K. Lee, D. Choi, and D. Kim, "Potential Fields-Aided Motion Planning for Quadcopters in Three-Dimensional Dynamic Environments," in AIAA Scitech 2021 Forum, doi: 10.2514/6.2021-1410.

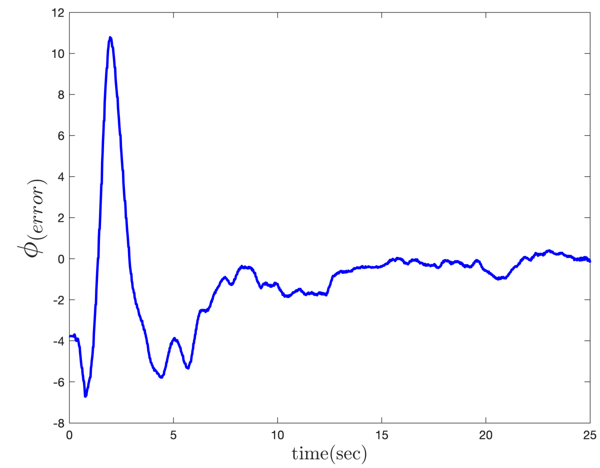
[14] J. P. Wilhelm and G. Clem, "Vector Field UAV Guidance for Path Following and Obstacle Avoidance with Minimal Deviation," Journal of Guidance, Control, and Dynamics, vol. 42, no. 8, pp. 1848–1856, 2019, doi: 10.2514/1.G004053.

[15] Bouabdallah, S., 2007. Design and Control of Quadrotors with Application to Autonomous Flying (Ph.D. thesis), University of Pennsylvania, Philadelphia.

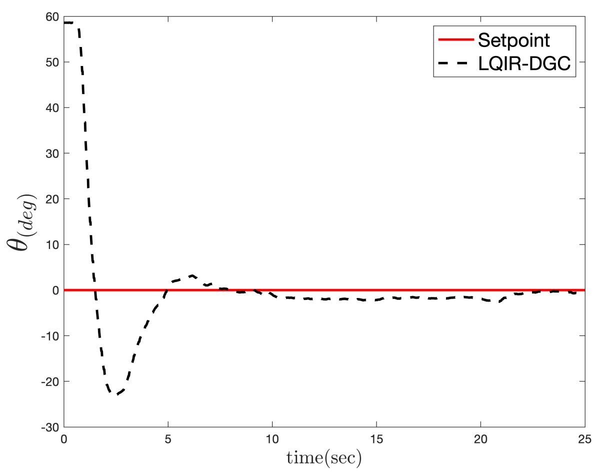
[16] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 153-158, doi: 10.1109/IROS.2007.4399042.



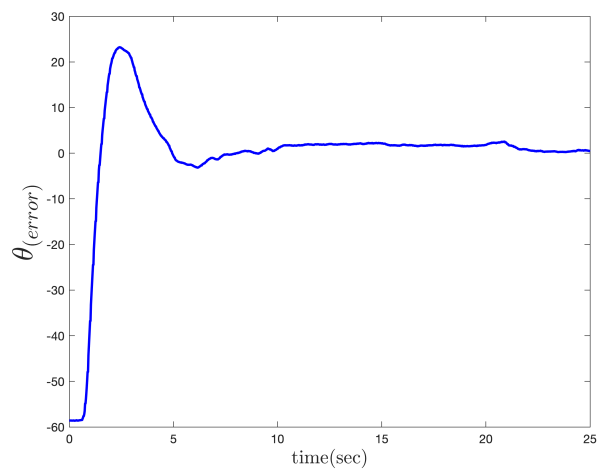
(a)



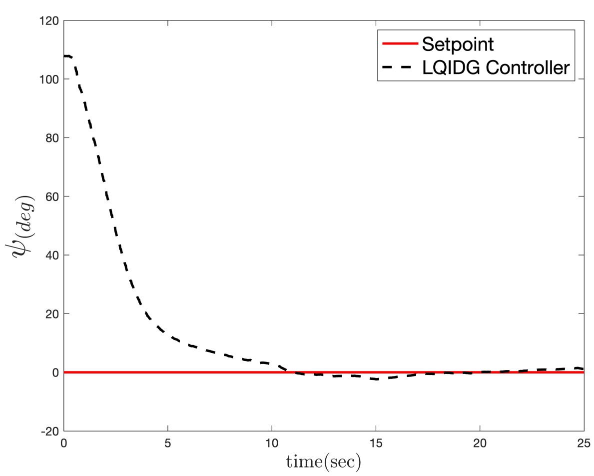
(b)



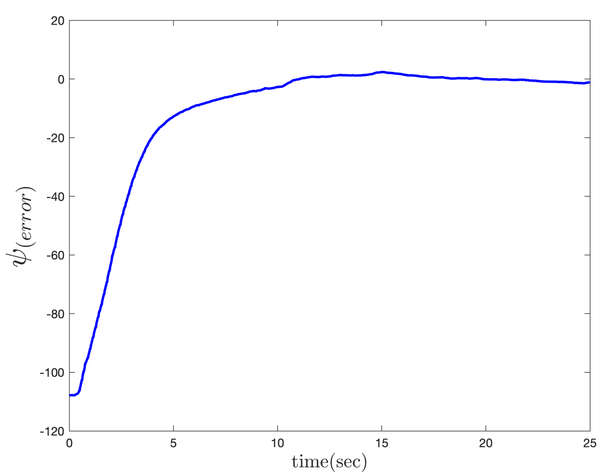
(c)



(d)



(e)



(f)

Fig. 5. implementation result

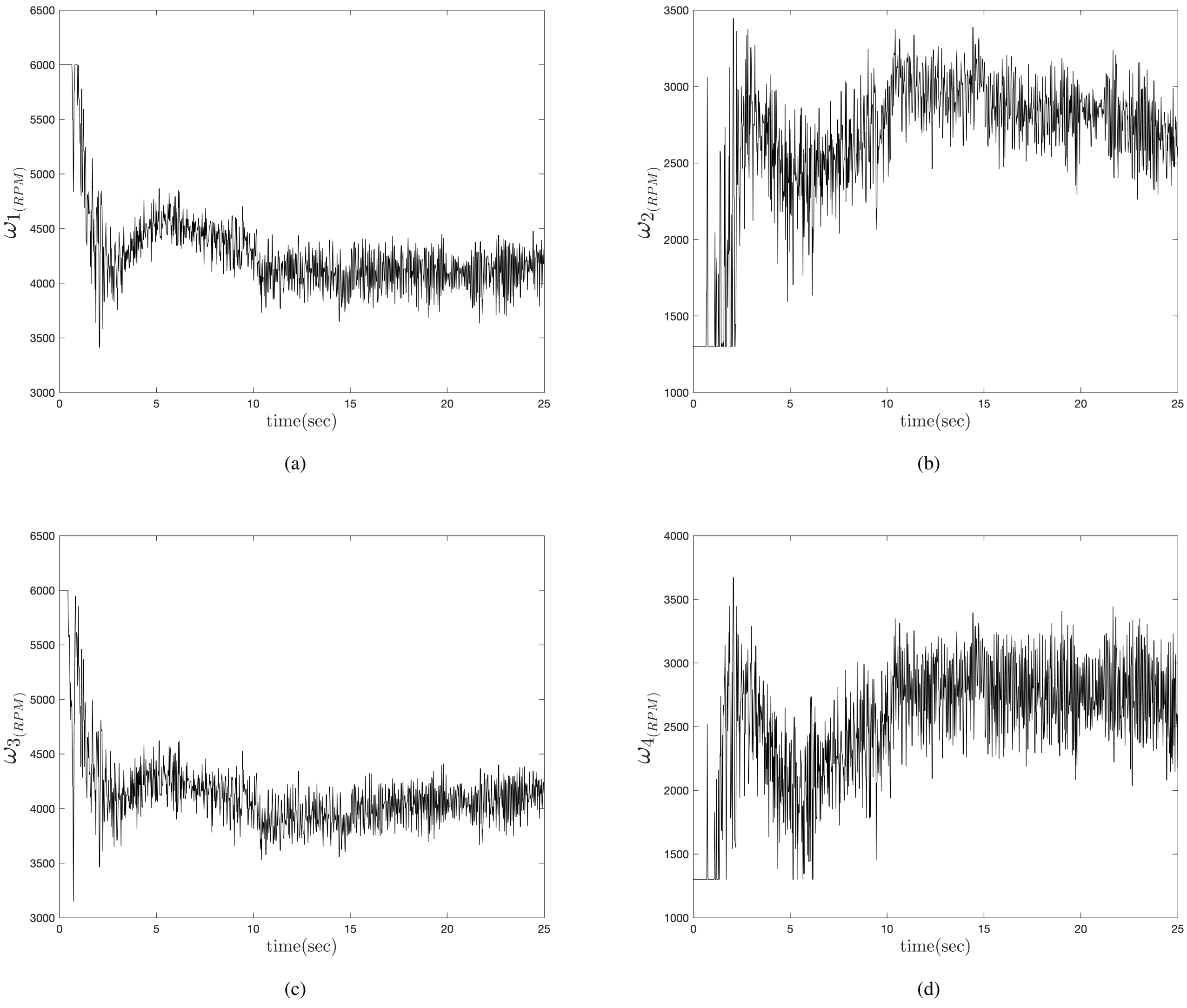


Fig. 6. implementation result omega