Control of a Three Degree of Freedom Quadcopter Stand Using Linear Quadratic Integral Based on the Differential Game Theory

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*—In this paper, a quadcopter stand with three degrees of freedom was controlled using game theory-based control. The first player tracks the desired input, and the second player creates a disturbance in the tracking of the first player to cause an error in the tracking. The control effort is chosen using the Nash equilibrium, which presupposes that the other player made the worst move. In addition to being resistant to input interruptions, this method may also be resilient to modeling system uncertainty. This method evaluated the performance through simulation in the Simulink environment and implementation on a three-degree-of-freedom stand.**

***Index Terms*—Quadcopter, Differential Game, Game Theory, Three Degree of Freedom Stand, Linear Quadratic**

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. Mathematical modeling

First, this section presents a nonlinear dynamic model of the three degrees of freedom quadrotor, Fig.1, for control purposes.

Fig. 1. Three degrees of freedom quadrotor in laboratory

Then, the system linearized for three SISO systems. The vehicle is assumed to be rigid in the dynamic modeling of the quadrotor. The space state of a three-degree-of-freedom quadcopter is defined as:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} \tag{1}$$

For simplicity, the system's inputs have been changed from rotational speed to influential forces in roll, pitch, and yaw modes. This change makes the problem from multi-input and

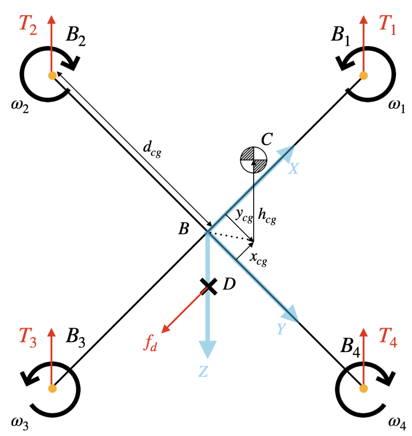


Fig. 2. Configuration of the quadrotor and the conventions

multi-output to three single-input problems. Then, the input vector is defined as

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^{\mathrm{T}} \tag{2}$$

where

$$u_1 = \omega_2^2 - \omega_4^2,\ u_2 = \omega_1^2 - \omega_3^2,\ u_3 = \omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 \tag{3}$$

The dynamic model of the three-degree-of-freedom quadcopter stand can be described as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{4}$$

where elements of $\mathbf{f}$ are nonlinear functions of the state vector $\mathbf{x}$ and the input vector $\mathbf{u}$.

$$\mathbf{f} = \begin{bmatrix} x_4 + x_5 \sin(x_1)\tan(x_2) + x_6 \cos(x_1)\tan(x_2) \\ x_5 \cos(x_1) - x_6 \sin(x_1) \\ (x_5 \sin(x_1) + x_6 \cos(x_1)) \sec(x_2) \\ A_1 \cos(x_2) \sin(x_1) + A_2 x_5 x_6 + A_3 u_1 \\ B_1 \sin(x_2) + B_2 x_4 x_6 + B_3 u_2 \\ C_1 x_4 x_5 + C_2 u_3 \end{bmatrix} \tag{5}$$

A. Linearization

Following are the steps of linearization of the system.

$$\delta\dot{\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{u} \tag{6}$$

Where $(*)$ term represents incremental variations around this point.

$$\begin{aligned} \mathbf{x}^* &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \\ \mathbf{u}^* &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{7}$$

Three single-input systems for each mode are presented here. The Space state matrices for the roll channel are shown below.

$$\begin{aligned} \mathbf{A}_{\text{roll}} &= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ A_1 & 0 \end{bmatrix} \\ \mathbf{B}_{\text{roll}} &= \begin{bmatrix} \frac{\partial f_1}{\partial u_1} \\ \frac{\partial f_4}{\partial u_1} \end{bmatrix} = \begin{bmatrix} 0 \\ A_3 \end{bmatrix} \end{aligned} \tag{8}$$

The Space state matrices for the pitch channel are shown below.

$$\begin{aligned} \mathbf{A}_{\text{pitch}} &= \begin{bmatrix} \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_5} \\ \frac{\partial f_5}{\partial x_2} & \frac{\partial f_5}{\partial x_5} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ B_1 & 0 \end{bmatrix} \\ \mathbf{B}_{\text{pitch}} &= \begin{bmatrix} \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_5}{\partial u_2} \end{bmatrix} = \begin{bmatrix} 0 \\ B_3 \end{bmatrix} \end{aligned} \tag{9}$$

The Space state matrices for the yaw channel are shown below.

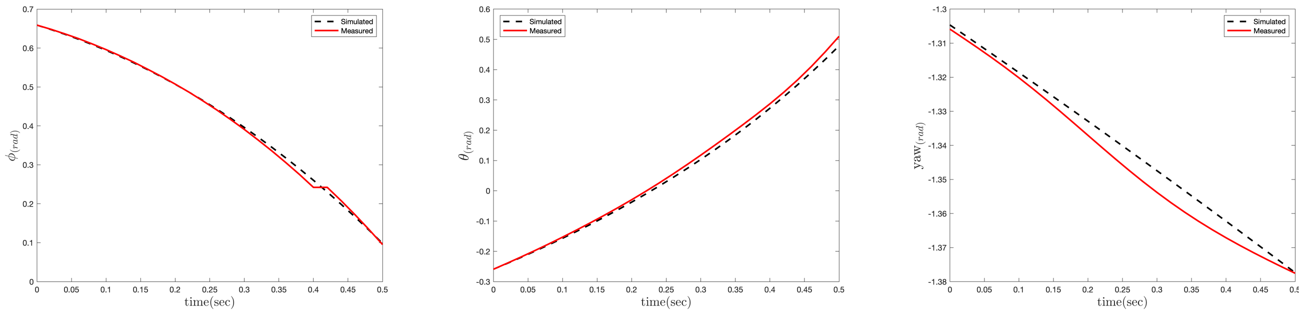
$$\begin{aligned} \mathbf{A}_{\text{yaw}} &= \begin{bmatrix} \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_6} \\ \frac{\partial f_6}{\partial x_3} & \frac{\partial f_6}{\partial x_6} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ \mathbf{B}_{\text{yaw}} &= \begin{bmatrix} \frac{\partial f_3}{\partial u_3} \\ \frac{\partial f_6}{\partial u_3} \end{bmatrix} = \begin{bmatrix} 0 \\ C_2 \end{bmatrix} \end{aligned} \tag{10}$$

B. 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.

III. 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) Roll Channel (b) Pitch Channel (c) Yaw Channel

Fig. 3. Comparison of quadrotor states in simulation and reality.

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}

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.

$$\begin{aligned} &\dot{\mathbf{x}}_\mathbf{i}(t) = \mathbf{A_i}\mathbf{x_i}(t) + \mathbf{B_i}\mathbf{u_i}(t) + \mathbf{B_{i_d}}\mathbf{u_{i_d}}(t) \\ &\mathbf{y_i}(t) = \mathbf{C_i}\mathbf{x_i}(t) + \mathbf{D_i}\mathbf{u_i}(t) + \mathbf{D_{i_d}}\mathbf{u_{i_d}}(t) \\ &i = 1,2,3 \end{aligned} \tag{11}$$

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 (11) 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:

$$J_1(\mathbf{u_i}^*, \mathbf{u_{i_d}}^*) \leq J_1(\mathbf{u_1}, \mathbf{u_{i_d}}^*) \ , \ J_2(\mathbf{u_i}^*, \mathbf{u_{i_d}}^*) \leq J_2(\mathbf{u_i}^*, \mathbf{u_{i_d}}) \tag{12}$$

B. LQDG controller

For the each control loop described in equation (11), LQDG optimum control effort calculates from equation (13).

$$\mathbf{u_i}(t) = -\mathbf{R_i}^{-1}\mathbf{B_i}^{\mathrm{T}}\mathbf{P_i}(t)\mathbf{x}(t) = -\mathbf{K_i}(t)\mathbf{x}(t), \quad i = 1,2,3 \tag{13}$$

In equation (13), $\mathbf{K_i}$ 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 (14) of controller player for each control loop of quadrotor.

$$\begin{aligned} J_i(\mathbf{u_i}, \mathbf{u_{i_d}}) = \int_0^T \Big(&\mathbf{x_i}^{\mathrm{T}}(t)\mathbf{Q_i}\mathbf{x_i}(t) + \mathbf{u_i}^{\mathrm{T}}(t)\mathbf{R_i}\mathbf{u_i}(t) + \\ &\mathbf{u_{i_d}}^{\mathrm{T}}(t)\mathbf{R_{i_d}}\mathbf{u_{i_d}}(t)\Big)\mathrm{d}t, \quad i = 1,2,3 \end{aligned} \tag{14}$$

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:

$$\begin{aligned} \dot{\mathbf{P}}_{\boldsymbol{i}}(t) = &-\mathbf{A_i}^{\mathrm{T}}\mathbf{P_i}(t) - \mathbf{P_i}(t)\mathbf{A_i} - \mathbf{Q_i} + \mathbf{P_i}(t)\mathbf{S_i}(t)\mathbf{P_i}(t) + \\ &\mathbf{P_i}(t)\mathbf{S_{i_d}}(t)\mathbf{P_{i_d}}(t) \\ \dot{\mathbf{P}}_{\boldsymbol{i_d}}(t) = &-\mathbf{A_i}^{\mathrm{T}}\mathbf{P_{i_d}}(t) - \mathbf{P_{i_d}}(t)\mathbf{A_i} - \mathbf{Q_{i_d}} + \\ &\mathbf{P_{i_d}}(t)\mathbf{S_{i_d}}(t)\mathbf{P_{i_d}}(t) + \mathbf{P_{i_d}}(t)\mathbf{S_i}(t)\mathbf{P_i}(t) \end{aligned} \tag{15}$$

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.

$$\mathbf{x_a} = \begin{bmatrix} \mathbf{x_d} - \mathbf{x} \\ \int (\mathbf{y_d} - \mathbf{y}) \end{bmatrix} \tag{16}$$

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.

$$\mathbf{y} = \mathbf{x} \tag{17}$$

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

$$\dot{\mathbf{x}}_{\boldsymbol{a}}(t) = \mathbf{A_a}\mathbf{x_a}(t) + \mathbf{B_{a_1}}\mathbf{u_{a_1}}(t) + \mathbf{B_{a_2}}\mathbf{u_{a_2}}(t) \tag{18}$$

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

$$\mathbf{A_a} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix}, \quad \mathbf{B_a} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \tag{19}$$

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 (20).

$$\begin{aligned} \mathbf{u_i}(t) &= -\mathbf{R_{ii}}^{-1}\mathbf{B_{a_i}}^{\mathrm{T}}\mathbf{P_{a_i}}(t)\mathbf{x_a}(t) \\ \mathbf{u_i}(t) &= -\mathbf{K_{a_i}}(t)\mathbf{x_a}(t), \ i = 1, 2, 3 \end{aligned} \tag{20}$$

In equation (20), $\mathbf{K_{a_i}}$ 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 (21), of player number i.

$$J_i(\mathbf{u_i}, \mathbf{u_{i_d}}) = \int_0^T \Big(\mathbf{x_a}^{\mathrm{T}}(t)\mathbf{Q_i}\mathbf{x_a}(t) + \mathbf{u_i}^{\mathrm{T}}(t)\mathbf{R_i}\mathbf{u_i}(t) + \mathbf{u_{i_d}}^{\mathrm{T}}(t)\mathbf{R_{i_d}}\mathbf{u_{i_d}}(t) \Big) \mathrm{d}t \tag{21}$$

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

$$\begin{aligned} \dot{\mathbf{P}}_{\boldsymbol{a_i}}(t) &= -\mathbf{A_a}^{\mathrm{T}}\mathbf{P_{a_i}}(t) - \mathbf{P_{a_i}}(t)\mathbf{A_a} - \mathbf{Q_i} + \\ &\quad \mathbf{P_{a_i}}(t)\mathbf{S_{a_i}}(t)\mathbf{P_{a_i}}(t) + \mathbf{P_{a_i}}(t)\mathbf{S_{a_{i_d}}}(t)\mathbf{P_{a_{i_d}}}(t) \\ \dot{\mathbf{P}}_{\boldsymbol{a_{i_d}}}(t) &= -\mathbf{A_a}^{\mathrm{T}}\mathbf{P_{a_{i_d}}}(t) - \mathbf{P_{a_{i_d}}}(t)\mathbf{A_a} - \mathbf{Q_{i_d}} + \\ &\quad \mathbf{P_{a_{i_d}}}(t)\mathbf{S_{a_{i_d}}}(t)\mathbf{P_{a_{i_d}}}(t) + \mathbf{P_{a_{i_d}}}(t)\mathbf{S_{a_i}}(t)\mathbf{P_{a_i}}(t) \end{aligned} \tag{22}$$

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

IV. 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.

$$\mathbf{Q}_{\mathrm{LQR}} = \begin{bmatrix} 0.5215 & 0 \\ 0 & 0.0745 \end{bmatrix}, \quad R_{\mathrm{LQR}} = 0.0001 \tag{23}$$

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

$$\mathbf{Q}_{\mathrm{LQDG}} = \begin{bmatrix} 100 & 0 \\ 0 & 0.078 \end{bmatrix}, \quad R_{\mathrm{LQDG}} = 1, \quad R_{d_{\mathrm{LQDG}}} = 99.96 \tag{24}$$

$$\mathbf{K_1} = \begin{bmatrix} 39.1188 & 8.8510 \end{bmatrix} \tag{25}$$

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

$$\mathbf{Q}_{\mathrm{LQIDG}} = \begin{bmatrix} 0.1707 & 0 & 0 & 0 \\ 0 & 0.12 & 0 & 0 \\ 0 & 0 & 837.8606 & 0 \\ 0 & 0 & 0 & 756.1341 \end{bmatrix} \tag{26}$$

$$R_{\mathrm{LQIDG}} = 1, \quad R_{d_{\mathrm{LQIDG}}} = 7.7422$$

$$\mathbf{K_{a_1}} = \begin{bmatrix} 28.1410 & 8.4017 & 27.2223 & 11.6894 \end{bmatrix} \tag{27}$$

B. Roll-Pitch Loop Control

$$\mathbf{Q}_{\mathrm{LQIDG_{roll}}} = \begin{bmatrix} 585.9 & 0 & 0 & 0 \\ 0 & 31.1 & 0 & 0 \\ 0 & 0 & 83.8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{Q}_{\mathrm{LQIDG_{pitch}}} = \begin{bmatrix} 546.5 & 0 & 0 & 0 \\ 0 & 311.4 & 0 & 0 \\ 0 & 0 & 2.22 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{28}$$

$$R_{\mathrm{LQIDG}} = 1, \quad R_{d_{\mathrm{LQIDG}}} = 7.7422$$

C. Roll-Pitch-Yaw Loop Control

$$
\mathbf{Q}_{\text{LQIDG}_{\text{roll}}} = \begin{bmatrix} 631.85 & 0 & 0 & 0 \\ 0 & 214.28 & 0 & 0 \\ 0 & 0 & 7.91 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}
$$

$$
\mathbf{Q}_{\text{LQIDG}_{\text{pitch}}} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 873.93 & 0 & 0 \\ 0 & 0 & 9853.09 & 0 \\ 0 & 0 & 0 & 0.12 \end{bmatrix} \tag{29}
$$

$$
\mathbf{Q}_{\text{LQIDG}_{\text{yaw}}} = \begin{bmatrix} 0.03 & 0 & 0 & 0 \\ 0 & 0.17 & 0 & 0 \\ 0 & 0 & 1.81 & 0 \\ 0 & 0 & 0 & 0.45 \end{bmatrix} \times 10^{-4}
$$

$$
R_{\text{LQIDG}} = 1, \quad R_{d_{\text{LQIDG}}} = 1.2577
$$

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.