# AUTONOMOUS AERIAL VEHICLES DISTRIBUTED CONTROL AND INTERACTIVE GAMES

by

YUSUF KARTAL

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy at
The University of Texas at Arlington
May, 2022

Arlington, Texas

Supervising Committee:
Frank Lewis, Supervising Professor
Atilla Dogan, Supervising Professor
Kamesh Subbarao
Animesh Chakravarthy
Shuo Linda Wang

*I dedicate this dissertation to my wife Tuğçe, my daughter, Gökçe, and my parents, Kamuran and Hami, for their constant support and enduring love.*

ACKNOWLEDGEMENTS

Starting from my academic advisors Dr. Frank Lewis and Dr. Atilla Dogan, I would like to thank people who had an important role in my academic career. First of all, I would like to thank my supervisor, teacher, and friend, Dr. Frank Lewis, for inspiring my interest in the development of innovative technologies and encouraging me on carrying out on the edge research. Thanks to his expertise and our fruitful discussions, I have collected unforgettable memories in these years. Without his continuous support, it would have been very difficult for me to maintain my motivation through this dissertation. I would also like to thank Dr. Atilla Dogan for providing me opportunity to obtain Ph.D. in UTA.

I also want to thank the members of my dissertation committee, Dr. Kamesh Subbarao, Dr. Animesh Chakravarthy and Dr. Shuo Linda Wang, for their insightful comments that help to improve my research. Dr. Kamesh Subbarao and Dr. Animesh Chakravarthy deserve special thanks for teaching me important concepts in their classes that contribute towards finalizing this dissertation.

With my whole heart I thank all of my friends and collaborators at UTA. I would also like to extend my appreciation to Turkish Aerospace for providing me financial support. I am extremely grateful to Dr. Ugur Zengin for his unconditional support. I wish to thank the University of Texas at Arlington Research Institute family for being very helpful and kind to me.

Finally, I would like to express my deepest gratitude to my friends in Turkey, and my family. I am extremely fortunate to have them in my life. This work was supported in part by Army Research Office under Grant W911NF-20-1-0132, Office of Naval Research under Grant N00014-18-1-2221, and National Science Foundation under Grant 1839804, EAGER.

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

AUTONOMOUS AERIAL VEHICLES DISTRIBUTED CONTROL AND INTERACTIVE GAMES

YUSUF KARTAL, Ph.D.

The University of Texas at Arlington, May, 2022

Supervising Professors: Frank Lewis, Atilla Dogan

As the number of quadrotors and other Unmanned Aerial Vehicles (UAVs) increases in industrial and urban areas, the development of reliable engineering methods to control their behavior as they interact with each other becomes of central interest in control research. With the increase in demand for UAVs to work together, several real-life challenges need to be addressed that include the implementation, test, and validation of the control algorithms in flight test experiments. On the one hand, the interests of UAVs may be in harmony to perform certain tasks such as transportation, surveillance and reconnaissance. On the other hand, the interests of UAVs may be directly opposite such as pursuing an evader UAV, and vice versa. This scenario is analyzed under category of the pursuit-evasion games in literature but constraints on the players' actions are not well considered. One of the primary requirements of autonomy is to be robust against the external disturbances, which can be achieved by designing a controller that guarantees $L_2$ gain boundedness by a prescribed attenuation level. Unfortunately, the standard approaches to the such control problems result in the sub-optimal gain solutions for guaranteed stability. In addition, when multi-

agent leader-follower control is considered, the mutual interests among the followers can be addressed within a well-established game theoretic framework. In particular, this can achieved via solving an output containment problem by introducing selfish followers where each follower only considers its own utility. However, standard approaches result in coupled Riccati equations that are hard to solve. Motivated by the desire to solve these problems, this dissertation has been written.

This dissertation first proposes a backstepping-based, distributed formation control method that is stable independent of time delays in communication among multiple UAVs. The proposed non-standard backstepping technique enables designer to develop an outer position & velocity control loop that interfaces seamlessly with the inner attitude controller of the cascaded control system for UAVs. Next, by using the nonstandard backstepping structure, we present a rigorous formulation for the pursuit-evasion (PE) game when velocity constraints are imposed on agents of the game or players. The game is formulated as an infinite-horizon problem using a non-quadratic functional, then sufficient conditions are derived to prove capture in a finite-time. Then, a new formulation for the $H_\infty$ static output-feedback (OPFB) control problem that guarantees both stability and $L_2$ gain boundedness of a Linear Time Invariant (LTI) system is given. This formulation allows us to extend multi-agent distributed formation control to multi-agent leader-follower (MLF) output containment game where the agents of game are UAVs. Lastly, the MLF output containment game is analyzed by introducing a novel cost functional whose solution provides both Nash and distributed robust control strategies in the sense that each follower uses the state information of its own and neighbors.

TABLE OF CONTENTS

# CHAPTER 1

## Introduction

Inspired by the naturally occurring biological groups such as herds and flocks where each member acts only under the influence of its neighbors [1], formation flight of quadrotors and other unmanned aerial vehicles (UAVs) has drawn great attention in recent years, due to their capability to perform certain tasks such as transportation [2], surveillance and reconnaissance, [3] and target search and detection [4]. With the increase in demand for UAVs to work together to accomplish these tasks, several real-life challenges need be addressed that include the implementation, test, and validation of the control algorithms in flight test experiments. In particular, we examine the challenge of designing a cooperative controller for UAVs to provide capabilities such as, performance despite communication time-delays in a leader-follower formation, employment of constrained input strategies, being robust against external disturbances, and adopting optimal actions to optimize mutual cost of the group.

In the second chapter of this dissertation, we propose a backstepping-based, distributed formation control method that is stable independent of time delays in communication among multiple UAVs. Instead of directly controlling the thrust generated by the propellers, we partition the mathematical model of the UAV into two subsystems, a linear attitude control loop and a nonlinear position control loop [5]. Centralized formation control of UAVs requires each agent to maintain a separation distance from other agents, which burdens the communication network of the UAVs. To overcome this problem, we consider a distributed control scheme wherein each agent updates its attitude and position based on the state information gathered

through its neighbors. Instead of directly controlling the thrust generated by the propellers, we partition the mathematical model of the UAV into two subsystems, a linear attitude control loop and a nonlinear position control loop. A backstepping-based outer position controller is then designed that interfaces seamlessly with the inner attitude controller of the cascaded control system. The closed-loop stability is established using a rigorous Lyapunov-Krasovskii analysis [6] under the influence of distributed network time delays. Using the directed graph topology and a distributed backstepping structure, it is shown that the stability criterion is delay-independent. The proposed control algorithms are verified in simulation and then implemented in hardware, and actual flight test experiments prove the validity of these algorithms.

This dissertation next copes with constrained input strategies for the pursuit-evasion (PE) game. A novel tracking Hamilton–Jacobi–Isaacs (HJI) equation associated with the non-quadratic value function is employed, which is solved for Nash equilibrium [7] velocity policies for each agent with arbitrary nonlinear dynamics. In contrast to the existing remedies for proof of capture in PE game, the proposed method does not assume players are moving with their maximum velocities and considers the velocity constraints a priori. Attaining the optimal actions requires the solution of HJI equations online and in real-time. We overcome this problem by presenting the on-policy iteration of Integral Reinforcement Learning (IRL) technique [8], [9]. The persistence of excitation for IRL to work is satisfied inherently until capture occurs, at which time the game ends. Furthermore, a nonlinear backstepping control method is proposed to track desired optimal velocity trajectories for players with generalized Newtonian dynamics. Simulation results are provided to show the validity of the proposed methods.

Then, a new formulation for the $H_\infty$ [10]-[15] static output-feedback (OPFB) control problem that guarantees both stability and $L_2$ gain boundedness of a Linear

Time Invariant (LTI) system is given. The problem is treated as a zero-sum differential game by introducing a quadratic performance index, and then a novel augmented Hamiltonian functional is proposed to solve for the Nash equilibrium point consisting of minimizing extrema (input) & maximizing extrema (disturbance) for the game of this kind. Unfortunately, the standard approaches to the $H_\infty$ control problem with static OPFB result in the sub-optimal gain solutions for guaranteed stability [10]-[11], [16]. In this chapter, we provide necessary and sufficient conditions of the optimal gain solutions that inherently stabilize the system dynamics while also guaranteeing Nash equilibrium. To obtain the optimal gain solution, two off-line iterative solution algorithms are given. The first algorithm is based on Lyapunov iterations requires an initial stabilizing gain. A second algorithm based on Riccati iterations obviates the initial stabilizing gain requirement. Then, based on the Lyapunov iterations, an off-policy Integral Reinforcement Learning (IRL) algorithm [17]-[18] is developed to learn the optimal gain solution online without requiring any knowledge of system state, control, and disturbance matrices. Simulation results are provided to show the validity of the proposed methods.

Finally, multiple leader and follower graphical games that constitute challenging problems for aerospace and robotics applications are considered. One of the challenges is to address the mutual interests among the followers with an optimal control point of view. In particular, the traditional approaches [19]-[20], treat the output containment problem by introducing selfish followers where each follower only considers its own utility. In this chapter, we propose a differential output containment game over directed graphs where the mutual interests among the followers are addressed with an objective functional that also considers the neighboring agents. The obtained output containment error system results in a formulation where outputs of all followers are proved to converge to the convex hull spanned by the outputs of leaders [21] in

a game optimal manner. The output containment problem is solved using the $\mathcal{H}_\infty$ output feedback method where the new necessary and sufficient conditions are presented. Another challenge is to design distributed Nash equilibrium control strategies for such games [22]-[26], which cannot be achieved with the traditional quadratic cost functional formulation. Furthermore, an $\mathcal{L}_2$ gain bound of the output containment error system that experiences worst-case disturbances with respect to the $\mathcal{H}_\infty$ criterion is investigated. The proposed methods are validated by means of multi-agent quadrotor Unmanned Aerial Vehicles (UAVs) output containment game simulations.

The resulting publications are listed below:

1. Yusuf Kartal , Kamesh Subbarao, Nicholas R. Gans, Atilla Dogan, Frank Lewis, Distributed backstepping based control of multiple UAV formation flight subject to time delays. *Published in IET Control Theory & Applications.* Volume 14, Issue 12. 2020.

   `https://doi.org/10.1049/iet-cta.2019.1151`

2. Yusuf Kartal, Kamesh Subbarao, Atilla Dogan, Frank Lewis, Optimal game theoretic solution of the pursuit-evasion intercept problem using on-policy reinforcement learning. *Published in International Journal of Robust and Nonlinear Control.* Volume 31, Issue 16. 2021.

   `https://doi.org/10.1002/rnc.5719`

3. Yusuf Kartal, Wenqian Xue, Atilla Dogan, Frank Lewis, New Solution for H-infinity Static Output-Feedback Control Using Integral Reinforcement Learning. 2021. *Under Review in IEEE Transactions on Cybernetics.*

4. Yusuf Kartal, Ahmet Taha Koru, Frank Lewis, Yan Wan, Atilla Dogan, Adversarial Multi-agent Output Containment Graphical Game with Local and Global Objectives for UAVs. 2021. *Under Review in IEEE Transactions on Control of Network Systems.*

The co-authors of above named work(s) acknowledge Yusuf Kartal as the primary author of the work(s) listed above. They authorize Yusuf Kartal to use the listed work(s) in this dissertation. They further agree that he may use this work to comply with requirements for graduation.

Distributed backstepping based control of multiple UAV formation flight subject to time delays [27]

Y. Kartal, K. Subbarao, N. Gans, A. Dogan and F. Lewis, *Published in IET Control Theory & Applications.* Volume 14, Issue 12. 2020.

https://doi.org/10.1049/iet-cta.2019.1151

# CHAPTER 2

## Distributed Backstepping Based Control of Multiple UAV Formation Flight Subject to Time Delays

Formation control is a type of multi-agent architecture that relies on relative motion of agents [28]. There exist different approaches to ensure formation control for multi-agent systems in the control community. Three recognized categories are behavior-based approaches [29], virtual structure-based approaches [30] and leader-follower approaches [31]. In the behavior-based approaches, each agent of the formation acts according to predefined behavior. This approach is behaviorally inflexible since motion is predefined. Alternately, virtual structure-based approach introduces a virtual vehicle for each vehicle in the formation and transforms the formation problem into a trajectory tracking problem. However, since the virtual vehicles are not exposed to any type of disturbance in the environment, there is a high chance that the followers break formation in the event of unexpected environmental disturbances. On the other hand, the leader-follower approach is easy to implement, and all agents react to any environmental change, but the network delay must be examined well to maintain formation.

Several works deal with linear dynamics of multi-agent systems [32, 33, 34, 35, 36]. Particularly, [33] reveals some of the necessary and sufficient conditions to achieve predefined time-varying formations with switching interaction topologies based on the algebraic Riccati equation. The authors of [36] propose a distributed adaptive control technique that uses adaptive gain scheduling to tune the coupling weights between the individuals of the multi-agent system. Whereas, [37] uses the same technique to satisfy

prescribed H-$\infty$ like performance and to manage the side effects of uncertainties in the system dynamics. However, since most physical systems are intrinsically nonlinear, these linear cooperative control methods cannot be applied directly [38]. Therefore, considerable number of works studied nonlinear dynamics of the multi-agent systems [38, 39, 40, 41, 42]. Neural-network based adaptive control and distributed impulsive control methods are examined to achieve leader-follower consensus with the class of nonlinear multi-agent systems in [38] and [39] respectively. The authors prove the stability of consensus error dynamics with well-known Lyapunov stability analysis techniques.

However, the problem of designing leader-follower formation strategies in which agents experience distributed network delays with pinning gain control requires more attention. One significant challenge associated with this relates to the construction of an appropriate compact nonlinear mathematical model of the multi-UAV system. In [6], authors study the effect of commensurate time delays on the closed-loop stability using the Retarded Functional Differential Equation (RFDE) form. By modeling the dynamics of the UAV as a double integrator, [43] and [44] treat each agent of the multi-UAV system as a point-mass system to apply time-varying consensus-based approaches on the multi-UAV system. This is a gross oversimplification of the UAV dynamics, especially for the quadrotor platforms considered in this paper. The inner/outer control loop partitioning allows us to deal with the delays occurring in the communication network of the UAVs, and to show that the stability of the outer control loop of the cascaded system is independent of delay. It is then shown that the closed-loop error dynamics of the whole system is also stable, independent of time delay.

Moreover, authors of [45, 46, 47, 48, 49] use nonlinear backstepping control method to deal with recursive design structure. This approach enables designer to

solve the stabilization problem partially for each submodule of the system of interest. Particularly, [45] proposes the adaptive backstepping control scheme to possess stronger stability properties while dealing with parametric uncertainties. [46] extends the adaptive backstepping control scheme in [45], to achieve both convergence to the path and predetermined dynamic behavior along the path simultaneously. In [47], integrator backstepping and quaternion feedback is adopted to stabilize the attitude of a micro satellite. Authors of [49] work on improving transient response of the closed-loop system by presenting a generalized backstepping process, based on the solvability of virtual controllers.

There are three major contributions of this work. Firstly, we use the second-order nonlinear dynamics of UAVs and synthesize a novel rigorous distributed backstepping control technique that has a form that easily extends to multiple UAV distributed control. Secondly, we partition the mathematical model of UAVs into two subsystems, an inner attitude control loop which is built into the quadrotor, and outer position controllers that consider relative motion of neighbors in formation flight. This allows us to rigorously analyze the delays occurring in the communication network of the UAVs. We prove that the stability of the outer control loop of the cascaded system is independent of delay, which implies that the closed-loop error dynamics of the whole system is also stable independent of delay. Lastly, we employ the directed graph topology to design formation control of the multi-UAV system, which reduces the work burden for the UAV communication network. It is rigorously shown that the stability criterion is delay-independent when each agent of the formation experiences distributed delays while communicating with its neighbors. The actual flight tests that show the validity and robustness of the developed control algorithms.

The rest of the paper is organized as follows. In Section 2.1, we provide the preliminaries of the mathematical model of the quadrotor and graph theory to under-

stand the basics of the distributed control approach. Section 2.2 brings an analysis of the control structures proposed, which involves the inner attitude controller and backstepping-based position controller for the task of trajectory tracking. We first explain the attitude controller design procedure for the quadrotors. Then we show the stability analysis of the backstepping control method. Section 2.3 illustrates how to extend the backstepping control algorithm to control multiple agents using the distributed backstepping tracker, which has a stable delay-independent system structure under the influence of non-constant distributed delays. Section 2.4 and 2.5 show the flight tests on a real UAV, where we illustrate the trajectories followed by AR.Drone 2.0 quadrotor with a full nonlinear backstepping tracker and by multiple AR.Drone 2.0's with a distributed backstepping trackers.

## 2.1 Preliminaries

The goal of this paper is to design a distributed controller for multi-UAV systems. Flying in a formation requires the agents to maintain separate distance from each other, which burdens the communication network and induces communication delays. In this section, we give preliminaries of a mathematical model of the quadrotor and graph theory to clarify the idea of the backstepping-based, distributed formation control method. In the next section, we present backstepping control for a single UAV. Then in Section 2.3, we present the formation controller with the network delays.

### 2.1.1 Mathematical model

This section introduces the standard nonlinear model of the quadrotor dynamics. To localize the quadrotor position, we use the Earth fixed frame. The origin of the three-dimensional (3D) axis system of the Body frame is assumed to be at

the center of mass of the quadrotor. The kinematics of the Euler angle rates can be expressed as

$$\boldsymbol{w_B} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & c\theta s\varphi \\ 0 & -s\varphi & c\theta c\varphi \end{bmatrix} \dot{\boldsymbol{\eta}} \tag{2.1}$$

where $c$ and $s$ refers to cosine and sine respectively and $\boldsymbol{w_B} \in \mathbb{R}^3$ is the angular velocity in the Body frame components. Particularly, $p$ is the roll rate, $q$ is the pitch rate, and $r$ is the yaw rate defined in the Body frame. Moreover, $\boldsymbol{\eta} \in \mathbb{R}^3$ is the Euler angle vector (roll, pitch, and yaw) i.e., $\boldsymbol{\eta} = [\varphi\ \theta\ \psi]^T$. Note that positive directions of Euler angles determined by right-hand rule, which are shown in Fig. 1.

The rotational dynamics are given by

$$\boldsymbol{I_B}\dot{\boldsymbol{w}}_{\boldsymbol{B}} = \boldsymbol{S}\left(\boldsymbol{w_B}\right)\boldsymbol{I_B}\boldsymbol{w_B} + \boldsymbol{\tau_B} \tag{2.2}$$

where $\boldsymbol{S}\left(\boldsymbol{w_B}\right) \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix [44], $\boldsymbol{\tau_B} = [\tau_\varphi\ \tau_\theta\ \tau_\psi]^T$is the torque vector and $\boldsymbol{I_B} \in \mathbb{R}^{3\times3}$ is the inertia matrix defined in Body frame.

The translational dynamics of the quadrotor, ignoring any aerodynamic effects, is expressed in the Body frame is obtained to be

$$m\dot{\boldsymbol{U}} = \begin{bmatrix} 0 \\ 0 \\ \mu \end{bmatrix} + \boldsymbol{RF_g} \tag{2.3}$$

where $\boldsymbol{U} = [u\ v\ w]^T$ is the velocity vector defined in the Body frame, $\mu$ is the total thrust produced by rotors in the Body frame $z_B$-axis. Note that $m$ is mass of the

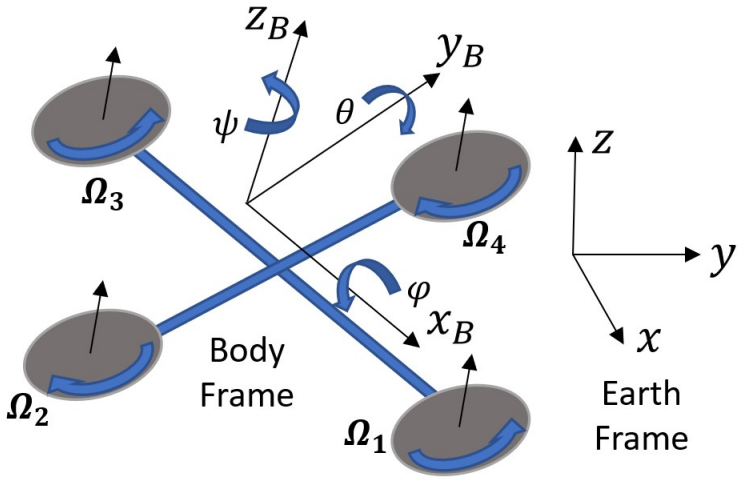


Figure 1: Coordinate systems of the quadrotor.

rigid body, $\boldsymbol{F_g} = [0\ 0\ -mg]^T$ is the gravitational force vector and $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix from the Earth frame to the Body frame. Moreover $u$, $v$ and $w$ stands for the velocities of the quadrotor in Body-axis coordinate system given in Fig.1. We obtain this rotation matrix using the yaw-pitch-roll (3-2-1) sequence. It is given by

$$\boldsymbol{R} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\varphi s\psi + s\varphi s\theta c\psi & c\varphi c\psi + s\varphi s\theta s\psi & s\varphi c\theta \\ s\varphi s\psi + c\varphi s\theta c\psi & -s\varphi c\psi + c\varphi s\theta s\psi & c\varphi c\theta \end{bmatrix} \tag{2.4}$$

Note that $\boldsymbol{R}$ belongs to the special orthogonal group and is of rank 3, or $SO(3)$, whose determinant is equal to 1.

The translational dynamics of the quadrotor in the Earth frame is then formulated as

$$m\ddot{\boldsymbol{\xi}} = m\dot{\boldsymbol{V}} = \boldsymbol{F} + \boldsymbol{F_g} \tag{2.5}$$

where $\boldsymbol{\xi} = [x\ y\ z]^T$ and $\boldsymbol{V} \in \mathbb{R}^3$ denotes the position and velocity vectors in the Earth frame respectively. And, $\boldsymbol{F} \in \mathbb{R}^3$ is the input force vector defined in the Earth frame. Then 2.3 and 2.5 gives the following relation

$$\boldsymbol{F} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \boldsymbol{R}^T \begin{bmatrix} 0 \\ 0 \\ \mu \end{bmatrix} = \begin{bmatrix} \mu(s\varphi s\psi + c\varphi s\theta c\psi) \\ \mu(-s\varphi c\psi + c\varphi s\theta s\psi) \\ \mu(c\varphi c\theta) \end{bmatrix} \tag{2.6}$$

### 2.1.2 Graph Theory

A Graph is constructed with a pair $\boldsymbol{G} = (V,\ E)$, where the set V $= \{\mathrm{v}_1, ..., \mathrm{v}_N\}$ defines the nodes or vertices, and E defines edges or arcs. The set E is composed of edge pairs $(\mathrm{v}_i, \mathrm{v}_j)$. If $(\mathrm{v}_i, \mathrm{v}_j)$ is equal to $(\mathrm{v}_j, \mathrm{v}_i)$ $\forall \mathrm{i}, \mathrm{j} \in [0, \mathrm{N}]$, i$\neq$j, then graph is said to be bidirectional. Each edge $(\mathrm{v}_j, \mathrm{v}_i) \in$ E, has a weight $a_{ij} > 0$ if and only if there exists a connection from node $j$ to $i$. The graph is called undirected if $a_{ij} = a_{ji}$, $\forall \mathrm{i}, \mathrm{j}$.The undirected graph is said to be weight balanced, which leads to symmetric adjacency matrix $\boldsymbol{A}$.

The diagonal matrix $\boldsymbol{D}$ is the i$^{th}$ row sum of $\boldsymbol{A}$ or weighted in-degree. Then, the Laplacian matrix is defined as

$$\mathcal{L} = \boldsymbol{D} - \boldsymbol{A}. \tag{2.7}$$

In this paper, the edge weights represent the trust between quadrotors, which are nodes of the formation graph. We create a graph topology based on adjacency or connectivity matrix $\mathbf{A} = [a_{ij}]$, realizing that $a_{ii} = 0$. The Laplacian matrices of all undirected graphs are real symmetric matrices. On the other hand, this is not valid for the digraphs. One of the contributions of this paper is proving consensus of the UAVs by adopting directed graph topology.

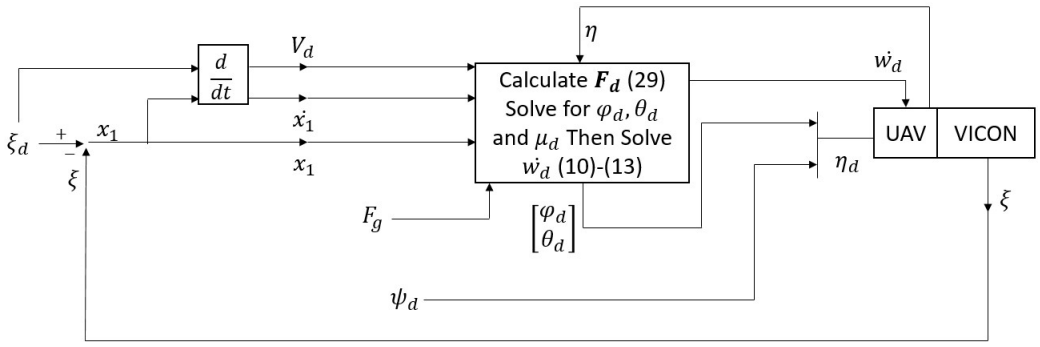


Figure 2: Desired input states calculation for the attitude controller.

## 2.2 Backstepping control

This section explains the full nonlinear backstepping control design for the quadrotor. The backstepping control structure derived here is shown in Fig.2. This is a nonstandard backstepping controller, has a novel form that allows direct extension to multiple interacting UAV control. Specifically, we derive the UAV error dynamics (2.33), which has a special form that is easily extended to multiple UAV formation control in Section 2.3. To apply the backstepping control method to the system defined in (2.5), we begin by adding and removing $\boldsymbol{F_d}$, an ideal virtual force input, and obtain the Newtonian model in terms of the desired forces

$$m\ddot{\boldsymbol{\xi}} = m\dot{\boldsymbol{V}} = \boldsymbol{F_d} + \boldsymbol{F_g} + \ \widetilde{\boldsymbol{F}}_{\boldsymbol{d}} \tag{2.8}$$

where $\widetilde{\boldsymbol{F}}_{\boldsymbol{d}} = \boldsymbol{F} - \boldsymbol{F_d}$. In Section 2.2.1, we show how to obtain desired Euler angle vector $\boldsymbol{\eta_d} = [\varphi_d \ \theta_d \ \psi_d]^T$ to generate $\boldsymbol{F_d}$, and the time rate of change of desired vertical speed, $\dot{w}_d$. Then in Section 2.2.2, $\boldsymbol{\tau_B}$ in (3.2) is designed using $\boldsymbol{\eta_d} = [\varphi_d \ \theta_d \ \psi_d]^T$ and $\dot{w}_d$ to get $\widetilde{\boldsymbol{F}}_{\boldsymbol{d}} \to \boldsymbol{0}$. Lastly in Section 2.2.3, $\boldsymbol{F_d}$ is selected to get $\boldsymbol{\xi} \to \boldsymbol{\xi_d}$, where $\boldsymbol{\xi_d} = [x_d \ y_d \ z_d]^T$ is the given desired position vector in the Earth frame. Proof of stability and tracking error convergence is given in Section 2.2.3.

### 2.2.1 Desired Euler angles

In Section 2.2.3, we show how to compute desired force vector $\boldsymbol{F_d}$ to obtain position and velocity tracking. Herein we show how to compute $\boldsymbol{\eta_d}$ and $\dot{w}_d$ from the desired force data $\boldsymbol{F_d}$ by using the inverse kinematics approach. Suppose we are given desired force $\boldsymbol{F_d}$. Note that from (2.6)

$$\boldsymbol{F_d} = \begin{bmatrix} f_{x_d} \\ f_{y_d} \\ f_{z_d} \end{bmatrix} = \begin{bmatrix} \mu_d(s\varphi_d s\psi_d + c\varphi_d s\theta_d c\psi_d) \\ \mu_d(-s\varphi_d c\psi_d + c\varphi_d s\theta_d s\psi_d) \\ \mu_d(c\varphi_d c\theta_d) \end{bmatrix} \tag{2.9}$$

where $\mu_d$ is the desired thrust in the Body frame. (3.47) can be solved for the desired Euler angles

$$\begin{aligned} \tan\theta_d &= \frac{f_{x_d}\cos\psi_d + f_{y_d}\sin\psi_d}{f_{z_d}}, \\ \theta_d &= \tan^{-1}\left(\frac{f_{x_d}\cos\psi_d + f_{y_d}\sin\psi_d}{f_{z_d}}\right), \end{aligned} \tag{2.10}$$

$$\begin{aligned} \tan\varphi_d &= \frac{\cos\theta_d\ (f_{x_d}\sin\psi_d - f_{y_d}\cos\psi_d)}{f_{z_d}}, \\ \varphi_d &= \tan^{-1}\left(\frac{\cos\theta_d\ (f_{x_d}\sin\psi_d - f_{y_d}\cos\psi_d)}{f_{z_d}}\right), \end{aligned} \tag{2.11}$$

$$\mu_d = \frac{f_{z_d}}{\cos\varphi_d\ \cos\theta_d}. \tag{2.12}$$

and $f_{z_d} \neq 0$. Although it should be mentioned that $f_{z_d} = 0$ only if $\theta_d$ and/or $\varphi_d = \pm\frac{\pi}{2}$ or $\mu_d = 0$. The condition $\theta_d = \pm\frac{\pi}{2}$ or $\varphi_d = \pm\frac{\pi}{2}$ correspond to singular orientations of the quadrotor and our domain of operation for $\theta$ and $\varphi$ is $\left(-\frac{\pi}{2} \quad \frac{\pi}{2}\right)$. Further $\mu_d = 0$ would correspond to zero total thrust. Furthermore, notice that $\psi_d$ can be arbitrarily prescribed, and only the variables $\theta_d$, $\varphi_d$ and $\mu_d$ must be found. The inner loop control design of the backstepping method requires the time rate of change of the

desired vertical speed in the Body frame, $\dot{w}_d$, which is calculated by using (2.3) such that

$$\dot{w}_d = \frac{\mu_d}{m} - g\cos\varphi_d \cos\theta_d \ . \tag{2.13}$$

Note that the information of time rate of change of the desired vertical speed or simply desired vertical acceleration acts as an input of the attitude controller loop will be given in Section 2.2.2. Derivation of this data is essential to control the height of the UAV, while accomplishing the path tracking objective accurately in 3D space.

### 2.2.2 Inner attitude control loop

In this section, we explain the inner attitude control of the backstepping method for the quadrotor. The attitude controller is generally built-in to the UAV and cannot be modified. This implies that the built-in attitude controller is assumed to track the quantities $\boldsymbol{\eta_d}$ and $\dot{w}_d$.

We begin with deriving the desired Euler rates $\boldsymbol{w_{Bd}} = [p_d \ q_d \ r_d]^T$ by using (2.1) and $\boldsymbol{\eta_d}$ such that

$$\begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta_d \\ 0 & c\varphi_d & c\theta_d s\varphi_d \\ 0 & -s\varphi_d & c\theta_d c\varphi_d \end{bmatrix} \begin{bmatrix} \dot{\varphi}_d \\ \dot{\theta}_d \\ \dot{\psi}_d \end{bmatrix} . \tag{2.14}$$

Then the following PID controller is designed to generate changes in angular velocity of the propellers

$$\begin{bmatrix} \Delta\Omega_{\varphi} \\ \Delta\Omega_{\theta} \\ \Delta\Omega_{\psi} \end{bmatrix} = \begin{bmatrix} P_{\varphi}\left(\varphi_d - \varphi\right) + D_{\varphi}\left(p_d - p\right) + I_{\varphi}\int\left(\varphi_d - \varphi\right) \\ P_{\theta}\left(\theta_d - \theta\right) + D_{\theta}\left(q_d - q\right) + I_{\theta}\int\left(\theta_d - \theta\right) \\ P_{\psi}\left(\psi_d - \psi\right) + D_{\psi}\left(r_d - r\right) + I_{\psi}\int\left(\psi_d - \psi\right) \end{bmatrix}. \tag{2.15}$$

Furthermore, (3.59) is used to obtain the desired angular velocity of each rotor [2] such that

$$\begin{bmatrix} \Omega_{1d} \\ \Omega_{2d} \\ \Omega_{3d} \\ \Omega_{4d} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \Omega_h + \Delta\Omega_{net} \\ \Delta\Omega_{\varphi} \\ \Delta\Omega_{\theta} \\ \Delta\Omega_{\psi} \end{bmatrix} \tag{2.16}$$

where $\Omega_{\mathrm{i}d}$, $i = 1, 2, 3, 4$, corresponds to the desired angular velocities of the rotors and $k$ is the thrust factor. And, $\Omega_h$ is the rotor speed required to hover such that

$$\Omega_h = \sqrt{\frac{mg}{4k}} \tag{2.17}$$

$\Delta\Omega_{net}$ is the outcome of desired vertical acceleration in the Body frame, $\dot{w}_d$ (2.13), in the form of

$$\Delta\Omega_{net} = \frac{m}{8\mathrm{k}\Omega_h}\dot{w}_d. \tag{2.18}$$

Notice that while producing $\Omega_h$ keeps the quadrotor at nominal condition (hover), $\Delta\Omega_{net}$ moves the UAV along $z_B$-axis. In addition, producing $\Delta\Omega_{\varphi}$, $\Delta\Omega_{\theta}$ and $\Delta\Omega_{\psi}$ deviates quadrotor from hover by resulting in roll, pitch and yaw respectively. Finally,

(3.60) yields the following torque vector expression, which stabilizes the rotational dynamics (3.2)

$$\boldsymbol{\tau_B}=\begin{bmatrix}\tau_\varphi\\ \tau_\theta\\ \tau_\psi\end{bmatrix}=\begin{bmatrix}lk({\Omega_{4\mathrm{d}}}^2-{\Omega_{2\mathrm{d}}}^2)\\ lk({\Omega_{3\mathrm{d}}}^2-{\Omega_{1\mathrm{d}}}^2)\\ d({\Omega_{1\mathrm{d}}}^2+{\Omega_{3\mathrm{d}}}^2-{\Omega_{2\mathrm{d}}}^2-{\Omega_{4\mathrm{d}}}^2)\end{bmatrix} \tag{2.19}$$

where $l$ is the lever length, and $d$ is the drag factor. The direction of angular velocities for each rotor is given in Fig.1, while first and third rotor turn anti-clockwise, the other two turn clockwise to cancel the yawing moments generated when the quadrotor is at nominal condition. The total thrust, $\mu_d$, is equal to the sum of thrusts generated by each rotor, that is

$$\mu_d=k({\Omega_{1\mathrm{d}}}^2+{\Omega_{2\mathrm{d}}}^2+{\Omega_{3\mathrm{d}}}^2+{\Omega_{4\mathrm{d}}}^2). \tag{2.20}$$

### 2.2.3 Outer position control loop

This section explains the outer position control loop of full nonlinear backstepping design for the quadrotor. We derive an expression for $\boldsymbol{F_d}$ that guarantees the dynamics of (2.8) are stable. A main result is the form (2.33) for the error dynamics, which has a special structure that is directly extended to multiple quadrotors in Section 2.3.

Begin with defining the position and velocity errors in the Earth frame as

$$\begin{aligned}\boldsymbol{x_1}&=\boldsymbol{e_p}=\boldsymbol{\xi_d}-\boldsymbol{\xi},\\ \boldsymbol{x_2}&=\boldsymbol{e_v}=\boldsymbol{V_d}-\boldsymbol{V}\end{aligned} \tag{2.21}$$

where $\boldsymbol{V_d}=\dot{\boldsymbol{\xi}}_{\boldsymbol{d}}\in\mathbb{R}^3$ is the desired velocity vector in the Earth frame.

The objective of the paper is to demonstrate the performance of the cooperative controller for any Commercial off the Shelf quadrotor platform with an inbuilt attitude controller. In such a situation, the assumption is reasonable that the inbuilt attitude controllers (typically PID) will accomplish this task. Thus the inner loop attitude controller is not analyzed further. Then to prove the convergence of error dynamics (3.45), which is the second step of backstepping method given in Section 2.2, we make following assumption

**Assumption 1.** *The inner attitude controller (3.43), tracks the Euler angles (3.48)-(3.49) and vertical acceleration (2.13). Hence the equilibrium of inner attitude control loop is stable.*

The next main theorem shows how to compute $\boldsymbol{F_d}$ to guarantee stable position and velocity tracking of (2.8).

**Theorem 1.** *Under Assumption 1, the following control law, applied to the system governed by (2.8) ensures that the position and velocity tracking errors in (3.45)* $\rightarrow 0$ *as* $t \rightarrow \infty$.

$$\begin{aligned} \boldsymbol{F_d} = m\dot{\boldsymbol{V}}_{\boldsymbol{d}} - \boldsymbol{F_g} + (m\boldsymbol{K_1} + m\boldsymbol{K_2})\,\dot{\boldsymbol{x}}_{\boldsymbol{1}} \\ + (m\boldsymbol{I_{3\times 3}} + m\boldsymbol{K_1}\boldsymbol{K_2})\,\boldsymbol{x_1}. \end{aligned} \tag{2.22}$$

*Proof.* The error dynamics are derived as

$$\dot{\boldsymbol{x}}_{\boldsymbol{1}} - \boldsymbol{x_2} + \boldsymbol{x}_{\boldsymbol{2}}^{\boldsymbol{v}} - \boldsymbol{x}_{\boldsymbol{2}}^{\boldsymbol{v}} = \boldsymbol{0} \tag{2.23}$$

where $\boldsymbol{x_2^v}$ is a virtual control signal. Moreover, by using (2.8) the error dynamics become

$$\dot{\boldsymbol{x}}_{\boldsymbol{2}} - \dot{\boldsymbol{V}}_{\boldsymbol{d}} + \frac{\boldsymbol{F_d}}{m} + \frac{\boldsymbol{F_g}}{m} + \frac{\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}}{m} = \boldsymbol{0}. \tag{2.24}$$

Then, define the following velocity error mismatch variable

$$\widetilde{\boldsymbol{x_2}} = \ \boldsymbol{x_2^v} - \boldsymbol{x_2}. \tag{2.25}$$

Substituting (2.25) in (2.23)

$$\dot{\boldsymbol{x}}_{\boldsymbol{1}} - \boldsymbol{x_2^v} = -\widetilde{\boldsymbol{x_2}}. \tag{2.26}$$

Now we pick $\boldsymbol{x_2^v} = -\boldsymbol{K_1 x_1}$ where $\boldsymbol{K_1} \in \mathbb{R}^{3\times 3}$ is a diagonal positive definite matrix. Then (2.26) becomes

$$\dot{\boldsymbol{x}}_{\boldsymbol{1}} + \boldsymbol{K_1 x_1} = -\widetilde{\boldsymbol{x_2}}. \tag{2.27}$$

To examine the stability of (2.27), we pick the Lyapunov function candidate as follows

$$\boldsymbol{\mathcal{V}} = \frac{1}{2}\boldsymbol{x_1}^{\boldsymbol{T}}\boldsymbol{x_1} + \frac{1}{2}\widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}}\widetilde{\boldsymbol{x_2}}. \tag{2.28}$$

Then the derivative of Lyapunov function candidate is derived using (2.24) and (2.26) as

$$\begin{aligned}\dot{\boldsymbol{\mathcal{V}}} &= \boldsymbol{x_1}^{\boldsymbol{T}}\dot{\boldsymbol{x}}_{\boldsymbol{1}} + \widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}}\dot{\widetilde{\boldsymbol{x}}}_{\boldsymbol{2}} \\ &= \boldsymbol{x_1}^{\boldsymbol{T}}\left(-\boldsymbol{K_1 x_1} - \widetilde{\boldsymbol{x_2}}\right)\end{aligned}$$

$$+ \widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}} \left( \dot{\boldsymbol{x}}_{\boldsymbol{2}}^{\boldsymbol{v}} \; - \dot{\boldsymbol{V}}_{\boldsymbol{d}} + \frac{\boldsymbol{F_d}}{m} + \frac{\boldsymbol{F_g}}{m} + \frac{\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}}{m} \right). \tag{2.29}$$

To have strictly negative definite Lyapunov function derivative, we set $\boldsymbol{F_d}$ as

$$\begin{aligned} \boldsymbol{F_d} &= m\dot{\boldsymbol{V}}_{\boldsymbol{d}} - \boldsymbol{F_g} - m\dot{\boldsymbol{x}}_{\boldsymbol{2}}^{\boldsymbol{v}} + m\boldsymbol{x_1} - m\boldsymbol{K_2}\widetilde{\boldsymbol{x_2}} \\ &= m\dot{\boldsymbol{V}}_{\boldsymbol{d}} - \boldsymbol{F_g} + (m\boldsymbol{K_1} + m\boldsymbol{K_2})\,\dot{\boldsymbol{x}}_{\boldsymbol{1}} \\ &\quad + (m\boldsymbol{I_{3\times 3}} + m\boldsymbol{K_1}\boldsymbol{K_2})\,\boldsymbol{x_1} \end{aligned} \tag{2.30}$$

where $\boldsymbol{K_2} \in \mathbb{R}^{3\times 3}$ is a diagonal positive definite matrix. Then, (2.29) becomes

$$\begin{aligned} \dot{\mathcal{V}} &= -\boldsymbol{x_1}^{\boldsymbol{T}}\boldsymbol{K_1}\boldsymbol{x_1} - \widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}}\boldsymbol{K_2}\widetilde{\boldsymbol{x_2}} + \widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}}\frac{\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}}{m}, \\ &\leq -\lambda_{min}(\boldsymbol{K_1}, \boldsymbol{K_2})\,\|\tilde{\boldsymbol{x}}\|^2 + \widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}}\frac{\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}(0)}{m}, \\ &\leq -\lambda_{min}(\boldsymbol{K_1}, \boldsymbol{K_2})\,\|\tilde{\boldsymbol{x}}\|^2 + \|\widetilde{\boldsymbol{x_2}}\|\,\frac{\left\|\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}(0)\right\|}{m} \end{aligned} \tag{2.31}$$

where $\tilde{\boldsymbol{x}} = \left[\boldsymbol{x}_{\boldsymbol{1}}^T \; \widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}}\right]^T$ and $\lambda_{min}(\boldsymbol{K_1}, \boldsymbol{K_2})$ stands for min eigenvalue of $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$. Note that in the worst case scenario, $\lambda_{min}(\boldsymbol{K_1}, \boldsymbol{K_2})$ must be bigger than $\frac{\left\|\widetilde{\boldsymbol{F}}_{\boldsymbol{d}}(0)\right\|}{m}$, which is a sufficient condition for asymptotic stability of origin. Moreover, from Assumption 1, the inner attitude control loop ensures that the quadrotor tracks the desired attitude angles $\varphi_d$, $\theta_d$, and the desired thrust $\mu_d$, i.e $\varphi \to \varphi_d$, $\theta \to \theta_d$, and $\mu \to \mu_d$. From the description of $\boldsymbol{F}$ and $\boldsymbol{F}_d$ in (2.6) and (3.47) respectively, we conclude that $\boldsymbol{F} \to \boldsymbol{F_d}$ and hence, $\widetilde{\boldsymbol{F}}_{\boldsymbol{d}} \to \boldsymbol{0}$. Then (2.31) becomes

$$\dot{\mathcal{V}} = -\boldsymbol{x_1}^{\boldsymbol{T}}\boldsymbol{K_1}\boldsymbol{x_1} - \widetilde{\boldsymbol{x_2}}^{\boldsymbol{T}}\boldsymbol{K_2}\widetilde{\boldsymbol{x_2}}, \tag{2.32}$$

which is strictly negative definite since $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ are positive definite matrices. Note that $\widetilde{\boldsymbol{x_2}} \to \boldsymbol{0}$ implies $\boldsymbol{x_2} \to \boldsymbol{x_2^v}$. Moreover, $\boldsymbol{x_2^v} \to \boldsymbol{0}$ as $\boldsymbol{x_2^v} = -\boldsymbol{K_1}\boldsymbol{x_1}$. Hence, $\boldsymbol{x_2} \to \boldsymbol{0}$ and the origin $(\boldsymbol{0}, \boldsymbol{0})$, which is the equilibrium of (3.45), is globally asymptotically stable. □

Using the control laws derived previously, from (2.30) the tracking error dynamics can be written in the state-space form as

$$\dot{\boldsymbol{x}} = \underbrace{\begin{bmatrix} \boldsymbol{0_{3\times 3}} & \boldsymbol{I_{3\times 3}} \\ -(\boldsymbol{K_1}\boldsymbol{K_2} + \boldsymbol{I_{3\times 3}}) & -(\boldsymbol{K_1} + \boldsymbol{K_2}) \end{bmatrix}}_{\boldsymbol{J}} \boldsymbol{x} \tag{2.33}$$

where $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_1}^{\boldsymbol{T}} & \boldsymbol{x_2}^{\boldsymbol{T}} \end{bmatrix}^T$. Note that $\boldsymbol{J}$ is Hurwitz. This form is instrumental in designing formation controllers for multiple UAV in the next section.

## 2.3 Distributed backstepping position control loop of multiple UAV with network delays

This section provides the connection of outer position control loop of backstepping method defined in Section 2.2.3 to the distributed multi-agent case. The error dynamics (2.33) are in a novel form which is easily extended in this section to multiple UAV formation.

We first treat the distributed system as delay-less. Then we perform rigorous stability analysis when agents experience both constant and distributed delays. Define dynamics (2.5) for each agent as

$$m\ddot{\boldsymbol{\xi}}_{\boldsymbol{i}} = m\dot{\boldsymbol{V}}_{\boldsymbol{i}} = \boldsymbol{F_i} + \boldsymbol{F_g}, \forall i = 1, ..., N. \tag{2.34}$$

### 2.3.1 No communication delay

In this section, we first extend the error dynamics (2.33) to multiple quadrotors. If there is no communication delay, define the position-based consensus error

$$\boldsymbol{e_{pi}} = \sum_{j \epsilon N_i} a_{ij} \left( \boldsymbol{\xi_j} - \boldsymbol{\Delta_j} - \boldsymbol{\xi_i} + \boldsymbol{\Delta_i} \right) + g_i \left( \boldsymbol{\xi_0} - \boldsymbol{\xi_i} + \boldsymbol{\Delta_i} \right) \tag{2.35}$$

$\forall i = 1, ..., N$ where $g_i$ is the pinning gain, $\boldsymbol{\Delta_i}$ (and $\boldsymbol{\Delta_j}$) is the $n$-dim constant tracking offset vector of the i$^{th}$ (and j$^{th}$) UAV with respect to the $n$-dim position of the leader, $\boldsymbol{\xi_0} \in \mathbb{R}^3$, of the formation. Lastly, $N$ is the number of the UAVs in the formation. Note that $g_i$ only takes values different than zero, if the node $i$ is directly connected to the leader node. For the sake of simplicity, we use following vector notations

$$\begin{aligned}
\boldsymbol{e_p^c} &= [\boldsymbol{e_{p_1}^T}\ \boldsymbol{e_{p_2}^T}\ ...\ \boldsymbol{e_{p_N}^T}]^T, \boldsymbol{e_p^c} \in \mathbb{R}^{Nn} \\
\underline{\boldsymbol{\Delta}} &= \left[\boldsymbol{\Delta_1^T}\ \boldsymbol{\Delta_2^T}\ ...\ \boldsymbol{\Delta_N^T}\right]^T, \underline{\boldsymbol{\Delta}} \in \mathbb{R}^{Nn} \\
\boldsymbol{\xi^c} &= \left[\boldsymbol{\xi_1^T \xi_2^T ... \xi_N^T}\right]^T, \boldsymbol{\xi^c} \in \mathbb{R}^{Nn} \\
\boldsymbol{V^c} &= \left[\boldsymbol{V_1^T V_2^T ... V_N^T}\right]^T, \boldsymbol{V^c} \in \mathbb{R}^{Nn} \\
\boldsymbol{F^c} &= \left[\boldsymbol{F_1^T F_2^T ... F_N^T}\right]^T, \boldsymbol{F^c} \in \mathbb{R}^{Nn}.
\end{aligned} \tag{2.36}$$

By using (2.5), the global system dynamics for followers can be written as

$$m\dot{\boldsymbol{V}}^{\boldsymbol{c}} = \boldsymbol{F^c} + \boldsymbol{1_N} \otimes \boldsymbol{F_g}. \tag{2.37}$$

Then, by using (2.7) and noting the fact $\mathcal{L}\boldsymbol{1_N} = \boldsymbol{0_N}$ since the row sum of $\mathcal{L}$ is zero, re-write (2.35) as

$$\boldsymbol{e_p}^{\boldsymbol{c}} = -((\mathcal{L} + \boldsymbol{G}) \otimes \boldsymbol{I_n}) \left( \boldsymbol{\xi^c} - \underline{\boldsymbol{\Delta}} \right) + (\boldsymbol{G} \otimes \boldsymbol{I_n})(\boldsymbol{1_N} \otimes \boldsymbol{\xi_0})$$

$$= -\left((\mathcal{L}+\boldsymbol{G})\otimes \boldsymbol{I_n}\right)\left(\boldsymbol{\xi^c}-\underline{\boldsymbol{\Delta}}-\boldsymbol{1_N}\otimes\boldsymbol{\xi_0}\right) \tag{2.38}$$

where $\boldsymbol{1_N}$ is $N$-dim vector whose all elements are ones and $\boldsymbol{G}\in\mathbb{R}^{N\times N}$ is a diagonal pinning gain matrix with the diagonal elements of $g_i\ \forall i=1,\ldots,N$. In addition, $\otimes$ stands for the Kronecker product. Define $\underline{\boldsymbol{\xi}}_{\boldsymbol{0}}=\boldsymbol{1_N}\otimes\boldsymbol{\xi_0}$. Then (2.38) becomes

$$\boldsymbol{e_p}^{\boldsymbol{c}}=\left((\mathcal{L}+\boldsymbol{G})\otimes\boldsymbol{I_n}\right)\left(\underline{\boldsymbol{\xi}}_{\boldsymbol{0}}+\underline{\boldsymbol{\Delta}}-\boldsymbol{\xi^c}\right). \tag{2.39}$$

The velocity-based consensus error is

$$\boldsymbol{e_v}^{\boldsymbol{c}}=\left((\mathcal{L}+\boldsymbol{G})\otimes\boldsymbol{I_n}\right)\left(\underline{\boldsymbol{V}}_{\boldsymbol{0}}-\boldsymbol{V^c}\right) \tag{2.40}$$

where $\underline{\boldsymbol{V}}_{\boldsymbol{0}}=\dot{\underline{\boldsymbol{\xi}}}_{\boldsymbol{0}}\in\mathbb{R}^{Nn}$, then the error dynamics are derived as

$$\begin{aligned}\dot{\boldsymbol{e}}_{\boldsymbol{p}}^{\boldsymbol{c}}&=\boldsymbol{e_v}^{\boldsymbol{c}},\\ m\dot{\boldsymbol{e}}_{\boldsymbol{v}}^{\boldsymbol{c}}&=\left((\mathcal{L}+\boldsymbol{G})\otimes\boldsymbol{I_n}\right)\left(m\dot{\underline{\boldsymbol{V}}}_{\boldsymbol{0}}-m\dot{\boldsymbol{V}}^{\boldsymbol{c}}\right).\end{aligned} \tag{2.41}$$

Substituting (2.37) in (2.41) results in

$$m\dot{\boldsymbol{e}}_{\boldsymbol{v}}^{\boldsymbol{c}}=\left((\mathcal{L}+\boldsymbol{G})\otimes\boldsymbol{I_n}\right)\left(m\dot{\underline{\boldsymbol{V}}}_{\boldsymbol{0}}-(\boldsymbol{F^c}+\boldsymbol{1_N}\otimes\boldsymbol{F_g})\right). \tag{2.42}$$

Now, we set the global desired force vector that contains desired force information for each agent of the formation (2.36) by using mixed-product property of Kronecker product, $(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D})=(\boldsymbol{AC})\otimes(\boldsymbol{BD})$ such that

$$\begin{aligned}\boldsymbol{F^c}=&\,m\dot{\underline{\boldsymbol{V}}}_{\boldsymbol{0}}-\boldsymbol{1_N}\otimes\boldsymbol{F_g}+m(\boldsymbol{I_N}\otimes(\boldsymbol{K_1}+\boldsymbol{K_2}))\boldsymbol{e_v}^{\boldsymbol{c}}\\ &+m(\boldsymbol{I_N}\otimes(\boldsymbol{I_{3\times3}}+\boldsymbol{K_1K_2}))\boldsymbol{e_p}^{\boldsymbol{c}}.\end{aligned} \tag{2.43}$$

Note that $\boldsymbol{F^c}$ is the global form of (2.30). Moreover, $\underline{\boldsymbol{\xi}}_{\mathbf{0}} + \underline{\boldsymbol{\Delta}}$ and $\underline{\boldsymbol{V}}_{\mathbf{0}}$ are the global form of $\boldsymbol{\xi_d}$ and $\boldsymbol{V_d}$ respectively. Then, we end up with the following second-order error dynamics such that

$$\begin{aligned}
\dot{\boldsymbol{e}}_{\boldsymbol{p}}^{\boldsymbol{c}} &= \boldsymbol{e_v}^{\boldsymbol{c}} \\
\dot{\boldsymbol{e}}_{\boldsymbol{v}}^{\boldsymbol{c}} &= -\left((\mathcal{L}+\boldsymbol{G}) \otimes (\boldsymbol{K_1 K_2} + \boldsymbol{I_{3\times 3}})\right) \boldsymbol{e_p}^{\boldsymbol{c}} \\
&\quad - \left((\mathcal{L}+\boldsymbol{G}) \otimes (\boldsymbol{K_1} + \boldsymbol{K_2})\right) \boldsymbol{e_v}^{\boldsymbol{c}}.
\end{aligned} \tag{2.44}$$

The state-space form of (2.44) is

$$\dot{\boldsymbol{x}}^{\boldsymbol{c}}(t) = \boldsymbol{J^c x^c}(t) \tag{2.45}$$

where $\boldsymbol{J^c} \in \mathbb{R}^{\boldsymbol{2Nn\times 2Nn}}$ is the global system matrix such that

$$\boldsymbol{J^c} = \begin{bmatrix} \mathbf{0}_{\boldsymbol{Nn\times Nn}} & \boldsymbol{I_{Nn\times Nn}} \\ -\left((\mathcal{L}+\boldsymbol{G}) \otimes \left(\boldsymbol{K_1 K_2} + \boldsymbol{I_{3\times 3}}\right)\right) & -\left((\mathcal{L}+\boldsymbol{G}) \otimes (\boldsymbol{K_1} + \boldsymbol{K_2})\right) \end{bmatrix} \tag{2.46}$$

and $\boldsymbol{x^c}(t) = \begin{bmatrix} \boldsymbol{e_p}^{\boldsymbol{cT}} & \boldsymbol{e_v}^{\boldsymbol{cT}} \end{bmatrix}^T \in \mathbb{R}^{\boldsymbol{2Nn\times 1}}$ is the global state vector. The global dynamics (2.45) are the combination of the single-agent dynamics (2.33) for the entire formation. Before we do the stability analysis for the closed-loop error dynamics given in (2.45), we make following assumption:

**Assumption 2.** *The graph topology of the multi-agent system contains a spanning tree with the root node being the leader node. This means that there is a directed path (not necessarily unique) from the leader node to every follower node.*

The next theorem extends the single-agent result in Theorem 1 to the multi-agent case by using M-matrix properties of the digraphs [50].

**Theorem 2.** *Given the Assumption 2, $\mathcal{L}+\boldsymbol{G}$ is an irreducible M-matrix and has all eigenvalues strictly in the open right-half plane [50]. Then, the equilibrium of closed-*

*loop error dynamics given in (2.45) is globally asymptotically stable point meaning that* $\boldsymbol{J^c}$ *is Hurwitz.*

*Proof.* Use the fact that Kronecker product of a positive diagonal matrix and an M-matrix has all eigenvalues strictly in the open right-half plane [51, 52]. Then, re-write (2.45) in form of the second-order differential equation such that

$$\begin{aligned}\ddot{\boldsymbol{e}}_{\boldsymbol{p}}^{\boldsymbol{c}} + ((\mathcal{L}+\boldsymbol{G})\otimes(\boldsymbol{K_1}+\boldsymbol{K_2}))\,\dot{\boldsymbol{e}}_{\boldsymbol{p}}^{\boldsymbol{c}} \\ + ((\mathcal{L}+\boldsymbol{G})\otimes(\boldsymbol{K_1}\boldsymbol{K_2}+\boldsymbol{I_{3\times 3}}))\,\boldsymbol{e_p}^{\boldsymbol{c}} = \boldsymbol{0}.\end{aligned} \tag{2.47}$$

Notice that all coefficient matrices of the characteristic polynomial of (2.47) have eigenvalues at open right half plane, hence the origin is globally asymptotically stable equilibrium by Routh-Hurwitz test. Note that if the graph topology was undirected, algorithms proposed in this paper would still work because $\mathcal{L}+\boldsymbol{G}$ would be positive definite symmetric matrix with the assumption of there exists a path from the leader node to every follower node. □

### 2.3.2 Communication delays

In this section, first we consider the system with constant communication delay, which occurs while the local positioning system shares the position of each agent to their neighbors. This delay may be created by processing time of the positioning system, data header analysis, and storage at routers, etc. Note that this delay is upper bounded by the practical limitations. Then (2.45) is written in the form of Retarded Functional Differential Equation [6] such that

$$\dot{\boldsymbol{x}}^{\boldsymbol{c}}(t) = \boldsymbol{J^c}\boldsymbol{x^c}\,(t) + \beta\boldsymbol{J^c}\boldsymbol{x^c}(t-\gamma) \tag{2.48}$$

where $\gamma$ is the network delay, $\beta$ is the gain of delayed term and $\boldsymbol{J^c}$ is the system matrix (2.46).

**Theorem 3.** *For the system in (2.48), the origin is stable equilibrium for $\beta \in (-1, 1]$ as the system matrix $\boldsymbol{J^c}$ is Hurwitz by Theorem 2.*

*Proof.* As $\gamma \to 0$, $\beta$ must be greater than -1 so that the overall system is stable, which is the lower bound of $\beta$.

To find the upper bound, use the fact that $\rho\left((jw\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) < 1\ \forall w > 0$ as given in [53] where $\rho(.)$ denotes the spectral radius of a matrix and $w$ denotes the frequency.

First assume that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) = 1,\ \forall w_t > 0$, which implies $e^{j\sigma_t}$ is the eigenvalue of the matrix $(jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}$ for $\sigma_t \in [0,\ 2\pi]$. Then, $\det\left(\boldsymbol{I} - (jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}e^{j\gamma_t w_t}\right) = 0$ for $\gamma_t = \frac{\sigma_t}{w_t}$, or equivalently by using matrix determinant lemma

$$\det\left(jw_t\boldsymbol{I} - \boldsymbol{J^c} - \boldsymbol{J^c}e^{j\gamma_t w_t}\right) = 0. \tag{2.49}$$

Hence (2.48) is not stable independent of delay with this assumption.

Next, assume that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) > 1,\ \forall w_t > 0$. Since $\rho(.)$ is continuous function of $w_t$ and

$$\lim_{w_t \to \infty} \rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) = 0. \tag{2.50}$$

Then $\exists w_t \in (w, \infty)$ such that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) = 1$, which makes (2.48) not stable independent of delay as this ends up with (2.49). Consequently, we show that $\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) < 1$, $\forall w_t > 0$. By using Gelfand Corollaries, this results in

$$\begin{aligned}\rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right) &\leq \rho\left((jw_t\boldsymbol{I} - \boldsymbol{J^c})^{-1})\rho(\boldsymbol{J^c})\right) \\ &\leq \frac{||\boldsymbol{J^c}||_\infty}{\sqrt{||\boldsymbol{J^c}||_\infty^{\boldsymbol{2}} + {w_t}^2}}.\end{aligned} \tag{2.51}$$

Then the upper bound of $\beta$ must be 1 by (2.51) to have $\rho\left(\beta(jw_t\boldsymbol{I}-\boldsymbol{J^c})^{-1}\boldsymbol{J^c}\right)<1$. To this end, we proved that as $\beta\in(-1,1]$, the system represented in (2.48) is stable independent of delay. □

Now, if we consider the system with non-constant distributed delays, (2.45) can be written as

$$\dot{\boldsymbol{x}}^{\boldsymbol{c}}(t)=\boldsymbol{J^c}\boldsymbol{x^c}(t)+\beta\int_{-\gamma}^{0}\boldsymbol{J^c}\boldsymbol{x^c}(t+s)ds \tag{2.52}$$

where $s\in[-\gamma,\ 0]$, $\beta$ is the gain of delayed term and $\gamma$ is the maximum delay.

**Theorem 4.** *The origin is an asymptotically stable equilibrium of (2.52) when there exist distributed delays in the communication network with gain $\beta\in[0,1]$, which is a sufficient condition for asymptotic stability.*

*Proof.* As shown in Theorem 3, the origin is stable equilibrium for $\beta\in(-1,1]$ since $\boldsymbol{J^c}$ is proven to be Hurwitz in Theorem 2. With this in mind, pick Lyapunov-Krasovskii functional as

$$\begin{aligned}\boldsymbol{V}\left(\boldsymbol{x_t^c}\right)&=\boldsymbol{x^{cT}}\left(t\right)\boldsymbol{P}\boldsymbol{x^c}\left(t\right)\\&\quad+\beta\int_{t-\gamma}^{t}\left[\int_{s}^{0}\boldsymbol{x^{cT}}\left(\ell\right)\boldsymbol{S}\boldsymbol{x^c}\left(\ell\right)d\ell\right]ds\end{aligned} \tag{2.53}$$

where $\boldsymbol{P}\in\mathbb{R}^{\boldsymbol{2Nn\times 2Nn}}$ and $\boldsymbol{S}\in\mathbb{R}^{\boldsymbol{2Nn\times 2Nn}}$ are positive definite, symmetric matrices. To have strictly positive definite Lyapunov-Krasovskii functional (2.53), the sufficient condition is $\beta>0$. Before taking the derivative of Lyapunov-Krasovskii functional and developing stability analysis, we use change of variable $\boldsymbol{f}\left(T\right)=\boldsymbol{x^c}\left(t+T\right)$ for arbitrary $T$, to simplify the stability analysis. Then, (2.52) and (2.53) become

$$\dot{\boldsymbol{f}}(0)=\boldsymbol{J^c}\boldsymbol{f}(0)+\beta\int_{-\gamma}^{0}\boldsymbol{J^c}\boldsymbol{f}(s)ds \tag{2.54}$$

$$\boldsymbol{V}(\boldsymbol{f}) = \boldsymbol{f}^{\boldsymbol{T}}(0)\,\boldsymbol{P}\boldsymbol{f}(0) + \beta\int_{-\gamma}^{0}\left[\int_{s}^{0}\boldsymbol{f}^{\boldsymbol{T}}(\ell)\,\boldsymbol{S}\boldsymbol{f}(\ell)\,d\ell\right]ds. \tag{2.55}$$

Furthermore, by using the Leibniz Integral Rule and (2.54), the derivative of Lyapunov-Krasovskii functional (2.55) becomes

$$\begin{aligned}\dot{\boldsymbol{V}}(\boldsymbol{f}) = {} & \boldsymbol{f}^{\boldsymbol{T}}(0)\left[\boldsymbol{P}\boldsymbol{J}^{\boldsymbol{c}} + \boldsymbol{J}^{\boldsymbol{c}\boldsymbol{T}}\boldsymbol{P} + \gamma\beta\boldsymbol{S}\right]\boldsymbol{f}(0) \\ & + 2\boldsymbol{f}^{\boldsymbol{T}}(0)\int_{-\gamma}^{0}\boldsymbol{P}\boldsymbol{J}^{\boldsymbol{c}}\boldsymbol{f}(s)\,ds \\ & - \beta\int_{-\gamma}^{0}\boldsymbol{f}^{\boldsymbol{T}}(s)\,\boldsymbol{S}\boldsymbol{f}(s)\,ds.\end{aligned} \tag{2.56}$$

To facilitate further development, (2.56) is written as

$$\begin{aligned}\dot{\boldsymbol{V}}(\boldsymbol{f}) = {} & \boldsymbol{f}^{\boldsymbol{T}}(0)\left[\boldsymbol{P}\boldsymbol{J}^{\boldsymbol{c}} + \boldsymbol{J}^{\boldsymbol{c}\boldsymbol{T}}\boldsymbol{P}\right]\boldsymbol{f}(0) \\ & + \int_{-\gamma}^{0}\boldsymbol{v}^{\boldsymbol{T}}\begin{bmatrix}\beta\boldsymbol{S} & \boldsymbol{P}\boldsymbol{J}^{\boldsymbol{c}} \\ \boldsymbol{J}^{\boldsymbol{c}\boldsymbol{T}}\boldsymbol{P} & -\beta\boldsymbol{S}\end{bmatrix}\boldsymbol{v}\;ds\end{aligned} \tag{2.57}$$

where $\boldsymbol{v} = \begin{bmatrix}\boldsymbol{f}(0)^{\boldsymbol{T}} & \boldsymbol{f}(s)^{\boldsymbol{T}}\end{bmatrix}^{\boldsymbol{T}}$. Notice that $\boldsymbol{V}(\boldsymbol{f}) \geq \varepsilon\left\| \boldsymbol{f}(0) \right\|^{\boldsymbol{2}}$ is satisfied for sufficiently small $\varepsilon > 0$. And, $\left[\boldsymbol{P}\boldsymbol{J}^{\boldsymbol{c}} + \boldsymbol{J}^{\boldsymbol{c}\boldsymbol{T}}\boldsymbol{P}\right] \leq -\varepsilon\boldsymbol{I}$ since $\boldsymbol{J}^{\boldsymbol{c}}$ is proved to be Hurwitz by Theorem 2. In addition, assuming $\exists\boldsymbol{P} = \boldsymbol{P}^{\boldsymbol{T}} > \boldsymbol{0}$ and using the linear matrix inequality [13], the negative definiteness of a matrix $\begin{bmatrix}\beta\boldsymbol{S} & \boldsymbol{P}\boldsymbol{J}^{\boldsymbol{c}} \\ \boldsymbol{J}^{\boldsymbol{c}\boldsymbol{T}}\boldsymbol{P} & -\beta\boldsymbol{S}\end{bmatrix}$ implies $\dot{\boldsymbol{V}}(\boldsymbol{f}) \leq \varepsilon\left\| \boldsymbol{f}(0) \right\|^{\boldsymbol{2}}$. Therefore, all conditions of the asymptotically stability by analyzing the derivative of Lyapunov-Krasovskii functional given in [23], have met meaning that the origin is asymptotically stable equilibrium. □

## 2.4 Experiment design and flight test details

This section addresses the crucial elements of our experiments, which are lab environment, flight controller design and simulations. An actual flight test is conducted in Section 2.5.

### 2.4.1 Lab environment

Equipment used are the Vicon, Parrot AR.Drone 2.0, and the master computer. Vicon is a motion capture system that provides the position of the UAVs.

The communication between master computer and Vicon is done via User Datagram Protocol (UDP). The frequency of the UDP Packets taken from the Vicon motion capture system is 100 Hz. The AR.Drone 2.0 has a built-in gyroscope and Inertial Measurement Unit (IMU) sensor suite. In practical applications, many quadrotors are designed with a built-in attitude controller and AR.Drone has its own attitude controller. This controller takes the desired values of $\varphi_d$, $\theta_d$ and $\psi_d$ as inputs. The communication between the master computer and the AR.Drone is done via UDP. The frequency of UDP packages is set to 500 Hz. MATLAB-Simulink is used to create UDP nodes that are communicating with AR.Drone and Vicon. The receiver and the sender UDP nodes are inserted to the Simulink model in the form of S-functions. The controller and the trajectory generation algorithms are implemented in the model. Simulink-Desktop Real Time Add-on is used to send the real time commands to the quadrotor. The UDP nodes tolerate up to 10% packet loss rate, which is necessary to handle communication channel noise created by the lab environment.

### 2.4.2 Flight controller design

The flight controller is a high-level decision-making mechanism that activates different modes of operation depending on the state of the UAV. We recognize three

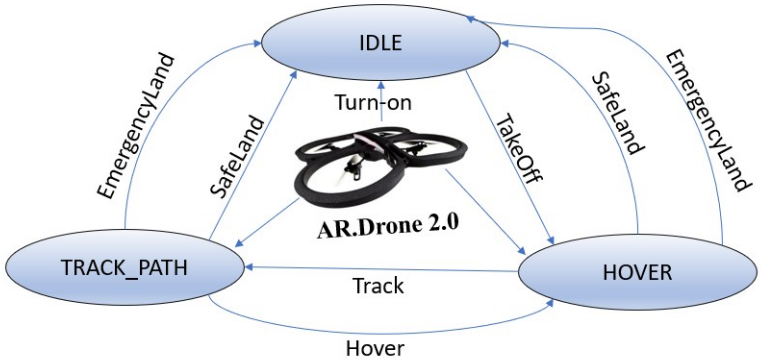


Figure 3: The flight controller FSM design.

modes of operation in our MATLAB implementation, which are IDLE, HOVER and TRACK_PATH as shown in Fig. 3. The quadrotor enters the IDLE mode, when either we turn-on the AR.Drone manually or it is landed by receiving the Land command. While in the IDLE, the UAV is actively receiving the data packets via UDP port communication and is ready to get the TakeOff command.

When the UAV reaches the desired height, $z_d$, the flight controller switches to the HOVER mode. In this mode the UAV is at the nominal condition, its attitude is parallel to ground and motionless in the air. If the Track command is received in the HOVER mode, the flight controller switches to the PATH_TRACK mode. This mode is triggered after 10 seconds passed from the transmission of TakeOff command. In this mode, AR.Drone begins to track the predetermined trajectory. The flight control algorithm, first reads the IMU sensor buffers and then Vicon buffer to construct the close loop error dynamics. To calculate desired Euler angles, (3.48)-(3.50) is used. Notice herein the desired yaw angle command is set to an arbitrary constant.

To land the quadrotor on the ground, we use either EmergencyLand or SafeLand commands. The difference of these two commands is the timing of stopping propeller movements. If we send the SafeLand command to the quadrotor, it reduces the

| Gains | $\Phi(Roll)$ | $\theta(Pitch)$ | $\psi(Yaw)$ |
|---|---|---|---|
| $\boldsymbol{P}$ | 6.42 | 6.42 | 4.82 |
| $\boldsymbol{D}$ | 5.54 | 5.54 | 7.89 |
| $\boldsymbol{I}$ | 1.85 | 1.85 | 0.11 |

Table 1: Inner attitude control loop PID parameters

propellers' speed till the height is in the range of 0-0.1 meter and shuts down the propellers. Else if we send the EmergencyLand to the quadrotor, it directly stops the propellers and lands on the ground. The appropriate structure for implementing the flight controller is the finite state machine (FSM) since the mode switching event is driven as shown in Fig. 3.

### 2.4.3 Simulations

The aim of this section is to verify control algorithms proposed in this paper by conducting different test scenarios. Before we implement the distributed backstepping control algorithm in the actual hardware, mathematical model in Section 2.1.1, Backstepping control method in Section 2.2 and distributed backstepping trackers in Section 2.3 are implemented in the Simulink. We first verified inner attitude control loop design in Section 2.2.2, the PID gains are given in Table 1.

To derive PID gains given in Table 1, thrust, drag factors, mass and arm length of the AR.Drone 2.0 must be measured. Mass and arm length of the quadrotor are measured as $m = 0.467kg$ and $l = 0.1785m$ respectively. To determine the thrust factor $k$, we first measure the angular velocity of a propeller with tachometer when the quadrotor is in hover. Then, we used (3.41) to derive $k$, which is found as $8 * 10^{-6} N * s^2/rad^2$. After that, using the thrust ratio analysis for small UAVs, the thrust factor $d$ is derived as $2 * 10^{-7} N * m * s^2/rad^2$. Moreover the diagonal elements of $\boldsymbol{K_1}$, $\boldsymbol{K_2}$ are tuned as $2, 2, 3$ and $1.5, 1.5, 3$ respectively.

To construct the desired circular trajectory for the formation leader, $x_d$ is set to $cos(\omega_t(t - t_{track}))$ and $y_d$ is set to $sin(\omega_t(t - t_{track}))$ where $t$ is simulation time, $t_{track}$ is the time at which the UAV begins to track circular trajectory, and $\omega_t$ stands for the frequency of the sinusoidal function. In our simulations, we pick $t_{track}$ as $15s$ and $\omega_t$ as $0.5rad/s$. Note that before formation leader begins to track circular trajectory, the $x_d$ value is linearly increased by $1m$ for $t \in [10, 15]$. Therefore, the leader UAV begins to track circular trajectory at $15s$. In Section 2.5.1, we double value of $\omega_t$ for $y_d$ setting to construct the eight figure trajectory.

#### 2.4.3.1 No delay, undirected graph

The aim of this section is to show validity of the proposed algorithms, when the undirected graph topology is adopted to design leader-follower formation control while multi-UAV system does not experience any delays. We share the control histories for the formation leader in Fig. 4 and Fig. 5. Particularly, in Fig. 4, we show attitude

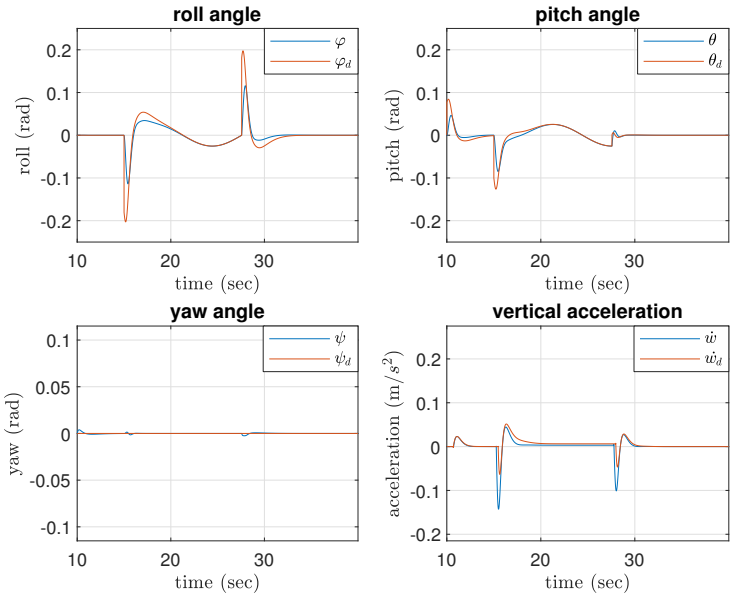


Figure 4: The attitude control inputs of leader UAV

control inputs (3.48)-(2.13) of the leader UAV. In Fig. 5, we show torque (3.43), and thrust (3.44) controls defined in the Body frame of the formation leader. For this test scenario, we pick Adjacency, pinning gain matrices and offset vector as follows

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
$$\underline{\boldsymbol{\Delta}}^T = \begin{bmatrix} 1.5 & 0 & 0 & 3 & 0 & 0 & 4.5 & 0 & 0 \end{bmatrix}. \tag{2.58}$$

Fig. 6 shows the leader and follower positions with the Adjacency and pining gain matrices given in (2.58), when there is no communication delay in the multi-UAV communication network.

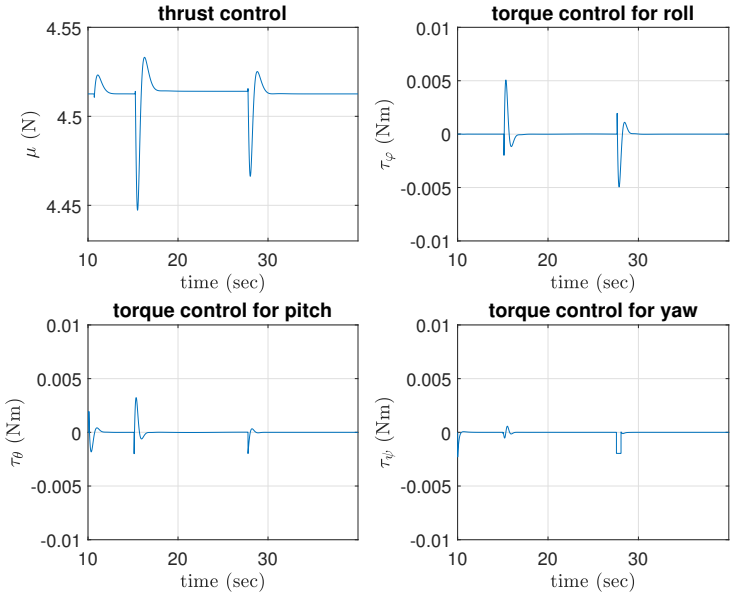


Figure 5: Torque and thrust controls of the leader UAV

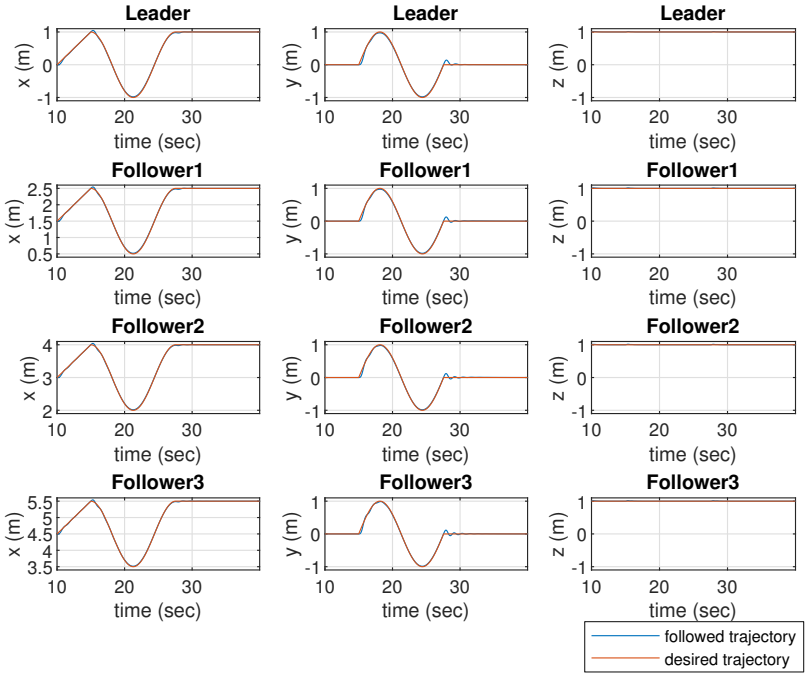


Figure 6: The path tracked by the UAVs when there exists no delay and undirected graph topology is used

#### 2.4.3.2 No delay, directed graph

For this test scenario, along with the offset vector given in (2.58) we pick Adjacency and pinning gain matrices that do no contradict Assumption 2, as follows

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.59}$$

Note that Fig. 7 is the same as Fig. 6 since the graph topology with the Adjacency and pinning gain matrices given in (5.56), contains a spanning tree.

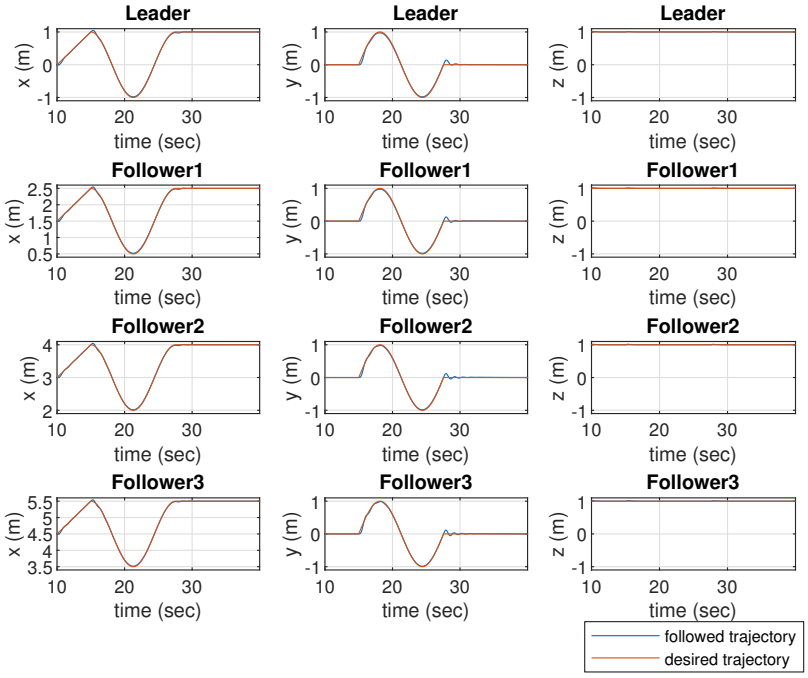


Figure 7: The path tracked by the UAVs when there exists no delay and directed graph topology is used

#### 2.4.3.3 With delay, directed graph

For the last test scenario of simulations, along with the offset vector given in (2.58), we pick Adjacency and pinning gain matrices given in (5.56). And, to test the stable independent of delay structure of the algorithms presented in Section 2.3, we added two seconds delay as a communication delay. By looking at Fig. 8, one can conclude that stable independent of delay property of the proposed algorithms, is verified.

## 2.5 Actual flight test results

This section reveals the flight test results obtained with single and multiple UAVs under the influence of time delays. We share the graphs of the desired and

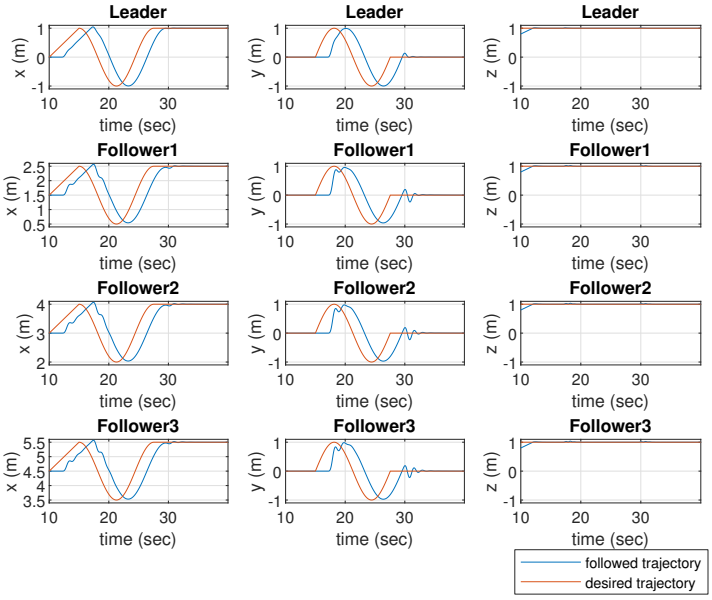


Figure 8: The path tracked by the UAVs when there exists two seconds delay and directed graph topology is used

followed trajectories when both the proposed methods in Section 2.2.3 and 2.3 is used.

### 2.5.1 Controller behavior with a single quadrotor

In this section, we present the performance of the backstepping control algorithm proposed in Section 3 by using both circular and figure-eight trajectories.

For the backstepping controller designed in Section 2.2.3, Fig. 9 and Fig. 10 shows the desired path of the UAV and the path traced by the UAV when both circular and eight-figure are desired trajectories. Notice that the tracking error is maintained inside the acceptable bounds, showing the performance of designed backstepping controller in terms of path following. Note that diagonal elements of $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$, are assigned respectively as $2, 2, 3$ and $1.5, 1.5, 3$.

To observe the network communication delay, we plot time vs desired and tracked positions in 3D space as shown in Fig. 11. Notice that communication delay

is about 2 seconds and tends to be commensurate through path following experiment. Moreover, when the quadrotor is following a nonlinear trajectory such as eight-figure and circular path, there exists time-delay between the desired position and followed position as shown in Fig. 11. This time delay is the summation of reaction time of UAV and the network delay caused by the local positioning system. That's why, the error seems to be bounded. However, after finishing the complete eight or circular figure trajectory, the error goes to zero since the desired position is constant at that time and quadrotor's position is the same as the desired position. This can be seen clearly from Fig. 9 and Fig. 10.

### 2.5.2 Controller behavior with multiple quadrotors

This section shows the formation control performance of the distributed backstepping control method given in Section 2.3. In the experiments of this section, the task of followers is to track the formation leader with a certain position offset. We show the leader and followers positions in the Fig. 12 and Fig. 13 while the leader

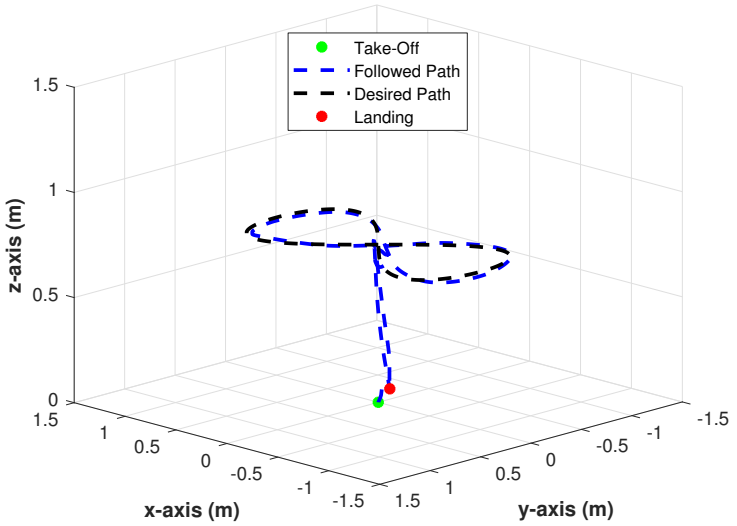


Figure 9: The path tracked by the UAV with the backstepping controller using 8-figure trajectory

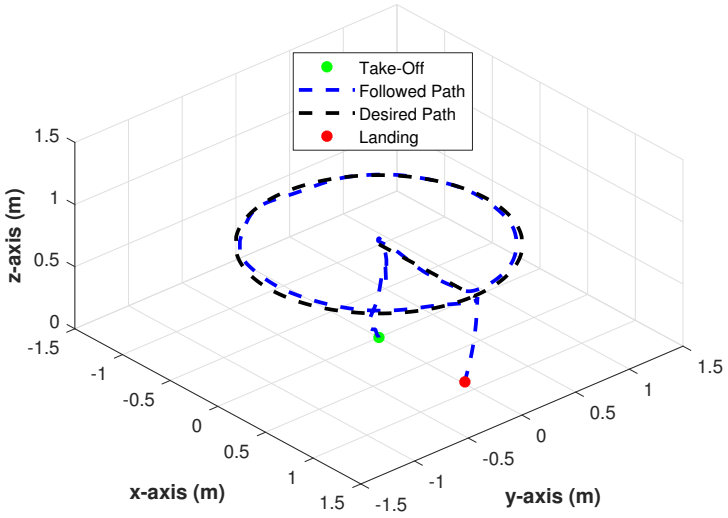


Figure 10: The path tracked by the UAV with the backstepping controller using a circular trajectory.

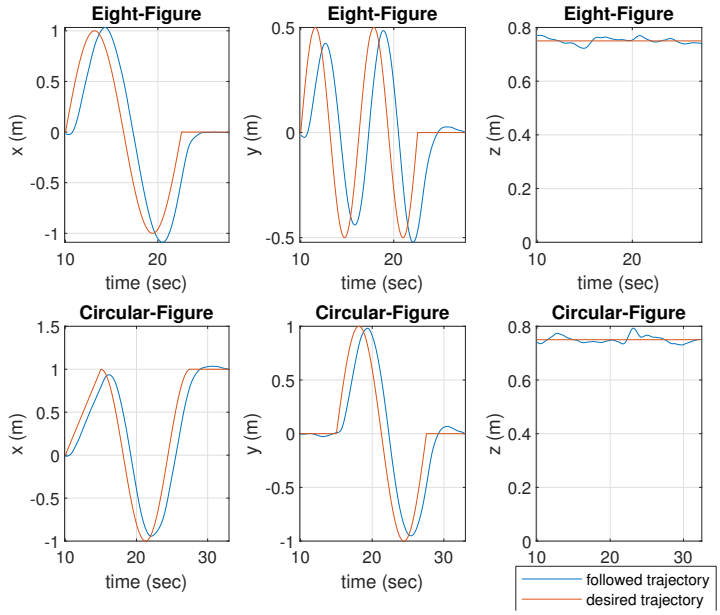


Figure 11: Controller behavior with time delay.

of the multi-agent system is following both eight-figure and circular trajectories re-

spectively. Adjacency and pinning gain matrices along with offset vector used in this actual hardware implementation are

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$
$$\underline{\boldsymbol{\Delta}}^T = \begin{bmatrix} 0 & -3 & 0 & 0 & -6 & 0 \end{bmatrix}. \tag{2.60}$$

Fig. 14 and Fig. 15 show the desired path of the UAVs and the path tracked by the UAVs when the leader is following both circular and eight-figure type trajectories. Note that the delay experienced by each agent of the formation is slightly different than each other. However, if the position offset of the quadrotors gets bigger, agents of the formation would experience more distributed delays.

We record a video of the experiments described in this section, the interested reader can use the link 'https://www.youtube.com/watch?v

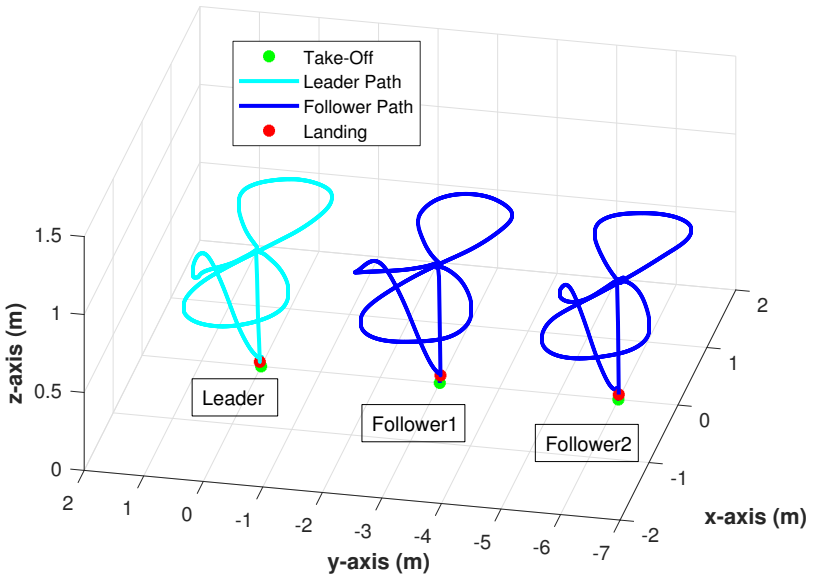


Figure 12: The path tracked by the UAVs with the distributed backstepping controller when the leader follows an eight-figure trajectory.

=rmY1LK42oPk' to have a visual understanding of the paper. Notice in the movie, formation control using the distributed backstepping method is influenced by the very strong wind effect that is produced by quadrotors themselves. This demonstrates the robustness of the proposed control method.

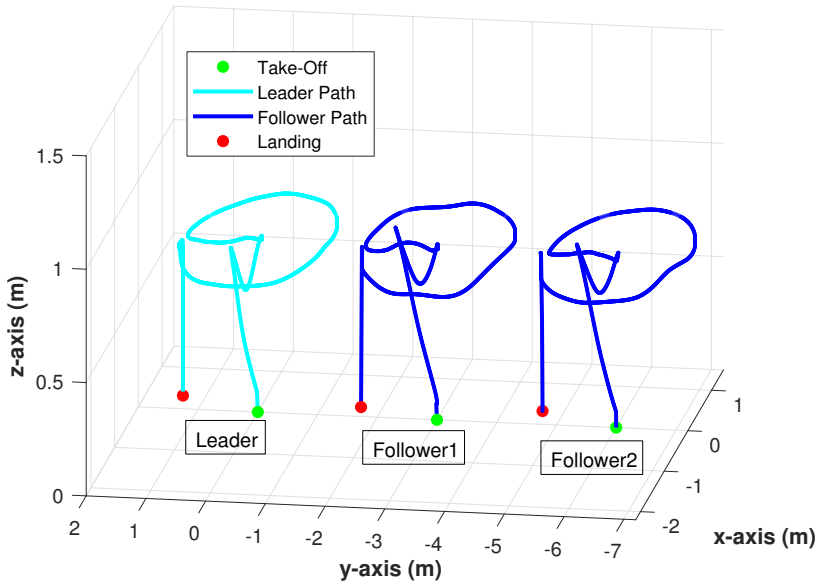


Figure 13: The path tracked by the UAVs with the distributed backstepping controller when the leader follows a circular trajectory.

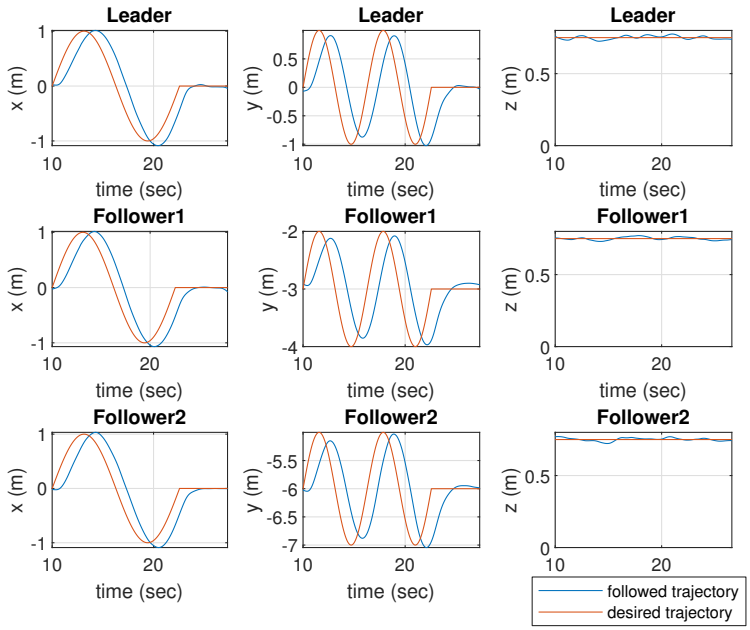


Figure 14: Observation of time delay graph when the leader follows an eight-figure trajectory.

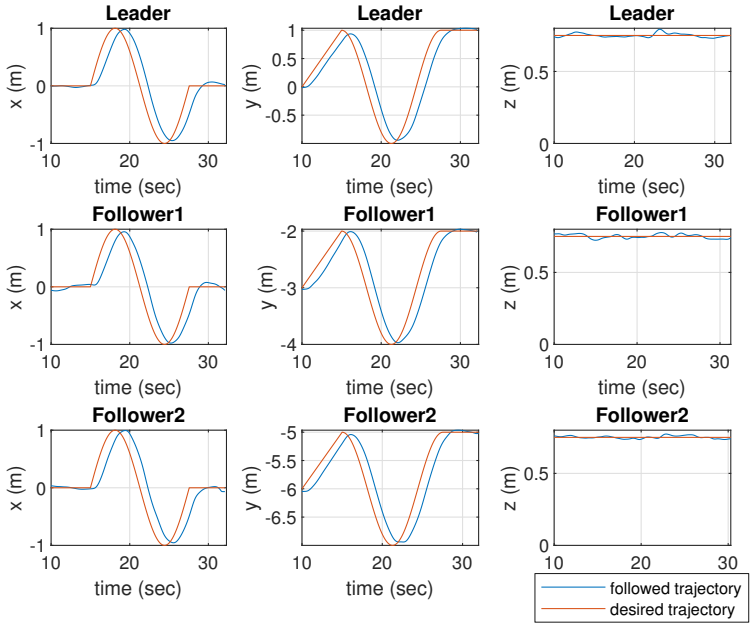


Figure 15: Observation of time delay graph when the leader follows a circular trajectory.

Optimal game theoretic solution of the pursuit-evasion intercept problem using on-policy reinforcement learning [54]

Y. Kartal, K.Subbarao, A. Dogan and F. Lewis, *Published in International Journal of Robust and Nonlinear Control.* Volume 31, Issue 16. 2021.

https://doi.org/10.1002/rnc.5719

# CHAPTER 3

## Optimal Game Theoretic Solution of the Pursuit-Evasion Intercept Problem Using On-Policy Reinforcement Learning

Inspired by the animal behaviors in hunting scenarios, the pursuit-evasion (PE) games have drawn great attention due to their applicability in areas such as missile guidance [55], collision avoidance systems [56] and controller designs [57]. The game of this kind is defined as a sub-category of differential game theory and provides the correct framework for the analysis of intercept problem and the choice of optimal policies for the agents involved in the two-player zero-sum (ZS) game.

Isaacs [7], founder of the differential game theory, initiated the development of strategic policies for both pursuer and evader in a PE problem. In Isaacs [58], the homicidal chauffeur game was analyzed in detail regarding players' speed and maneuverability capabilities. Bryson [59] introduced optimal feedback laws and demonstrated intercept strategies for players, by using the fixed final-time value function. Lewis et al. [60] made an extension of the Bellman equation, known as the Hamilton-Jacobi-Isaacs (HJI) equations to design $H_\infty$ control, by employing the ZS games solutions. Moreover, works [12], [61], and [13] deal with linear-quadratic ZS games, in which their objective is to minimize the maximum norm of inputs and states, where the maximum is taken over the unknowns, such as disturbances. Hayoun et al. [62] reveal a set-based computing method for solving a general class of ZS Stackelberg differential games where the authors come up with a novel class of differential inequalities to get convex outer approximations of backward and forward reachable sets. Bhattacharya et al. [63] worked on a visibility-based PE game when the envi-

ronment contains a circular obstacle. Furthermore, Li et al. [64] and Liu et al. [65] developed an reinforcement learning (RL) algorithm to learn the Nash equilibrium solution for designing model-free controller by solving the game algebraic Riccati equation forward in time.

Applications of PE games involve the proximate satellite interception guidance strategies. The work [66] studied the intercept problem of satellites where both of the interceptor and target satellite can perform orbital maneuvers with limited thrusts. In the work [67], the authors analyzed same problem by establishing a local moving coordinate frame and simplifying dynamics of each player to the linear Clohessy-Wiltshire equations. Using the terminal time as a cost function, Gong et al. [68] derived sufficient conditions for capture in the PE problem based on the players' hyper-reachable domain. Note that the common point of these papers is to utilize prescribed terminal time on the construction of the game-theoretical cost function. Jagat et al. [69] proposes quadratic infinite-horizon cost functional for both players but the finite-time capture is not proven mathematically. Instead, simulations are provided to show that capture occurs in a finite-time. Carr et al. [70] employ semi-direct collocation nonlinear programming method to solve optimal actions for agents of the pursuit–evasion game. Authors solve the minimax problem by considering co-state dynamics and boundary conditions simultaneously for the dynamical models.

Standard solution to constrained PE game is to impose external velocity or acceleration constraints. Unfortunately, this leads to discontinuous saturated solutions that are difficult to analyze [59],[60].

Recent works by Hayoun et al. [62], Shaferman et al. [71] and Weintraub et al. [72] focus on the missile-target engagement where the PE problem is formulated as a differential game with an objective of optimizing the linear quadratic cost functional. The work [62] propose bounded maneuverability of the evader to prove the capture

in ZS game whereas Weintraub et al. [72] consider an engagement scenario by introducing the defense of a non-maneuverable agent. Further, this work [72] reveals the inclusion of altitude and dynamics in 3-dimensions, which is more realistic for the modeling of aerial engagements.

We sum up the contributions of this paper into four categories as:

- Firstly, a backstepping based velocity tracker is developed for PE games where the pursuer and evader both have arbitrary nonlinear dynamics. Taking a priori velocity constraints into account, a novel non-quadratic scalar functional is solved to obtain the smooth optimal velocity policies for each player in contrast to the standard discontinuous solutions.
- Secondly, with the detailed Lyapunov analysis, sufficient conditions are given for the case where capture must be attained in finite-time.
- The on-policy Integral Reinforcement Learning method is employed to solve the corresponding HJI equation and achieve the game optimal velocity policies for both pursuer and evader.
- Finally, the full rotational dynamics are added to extend the results to full nonlinear dynamical PE systems.

Rest of the paper is organized as follows. Section 3.2 reviews the exponentially stabilizing nonlinear backstepping control method to track given velocity trajectories for a generalized Newtonian system dynamics. Section 3.3 obtains optimal actions for the players by making use of the Pontryagin's minimum principle and brings analysis of a Nash equilibrium in the PE game. Furthermore, having revealed the sufficient conditions for the asymptotic capture, we prove that PE game ends in a finite-time based on the derived sufficient conditions. Section 3.4 proposes an on-policy reinforcement learning algorithm for the solution of HJI equation and derives the proof of convergence to the optimal policies. Section 3.5 closes the backstepping

control loop by treating forces and/or moments as finalized inputs to the system and representing the attitude with unit quaternions to overcome the singularity problem of the Euler angles. Finally, the proposed control policies are illustrated via simulation results in Section 3.6.

## 3.1 Problem formulation and model description

We study the pursuit-evasion (PE) game for general Newtonian dynamics. A novel approach is given whereby we first design backstepping based velocity controllers for the pursuer and evader that guarantee a Nash solution to the PE game. We use a novel value function that ensures a solution under bounded velocities of the pursuer and the evader. This provides smooth solutions to the bounded velocity PE game in contrast to standard discontinuous solutions [59]. We conduct a Nash equilibrium analysis for a game of this kind. Further, we seek to obtain sufficient conditions for global exponential stability of the origin (equilibrium) using a rigorous Lyapunov analysis. Finally, we seek to derive conditions for a final time capture, and provide an upper bound on the time of capture.

The generalized translational and rotational dynamics for the pursuer and the evader can be modeled in their respective body frames of reference as

$$m^i \dot{\boldsymbol{v}}_{\boldsymbol{B}}^{\boldsymbol{i}} = m^i \boldsymbol{S}(\boldsymbol{w}_{\boldsymbol{B}}^{\boldsymbol{i}}) \boldsymbol{v}_{\boldsymbol{B}}^{\boldsymbol{i}} + \boldsymbol{N}^{\boldsymbol{i}} \boldsymbol{f}_{\boldsymbol{g}}^{\boldsymbol{i}} + \boldsymbol{f}_{\boldsymbol{B}}^{\boldsymbol{i}}, \tag{3.1}$$

$$\boldsymbol{I}_{\boldsymbol{B}}^{\boldsymbol{i}} \dot{\boldsymbol{w}}_{\boldsymbol{B}}^{\boldsymbol{i}} = \boldsymbol{S}(\boldsymbol{w}_{\boldsymbol{B}}^{\boldsymbol{i}}) \boldsymbol{I}_{\boldsymbol{B}}^{\boldsymbol{i}} \boldsymbol{w}_{\boldsymbol{B}}^{\boldsymbol{i}} + \boldsymbol{\tau}_{\boldsymbol{B}}^{\boldsymbol{i}} \tag{3.2}$$

where the superscript $\boldsymbol{i} \in \{\boldsymbol{p},\ \boldsymbol{e}\}$ with $\boldsymbol{p}$ denoting the pursuer and $\boldsymbol{e}$ denoting the evader respectively. Here, $\boldsymbol{v}_{\boldsymbol{B}}^{\boldsymbol{i}} \in \mathbb{R}^3$, $\boldsymbol{w}_{\boldsymbol{B}}^{\boldsymbol{i}} \in \mathbb{R}^3$ are the translational and angular velocities respectively, and $\boldsymbol{f}_{\boldsymbol{B}}^{\boldsymbol{i}} \in \mathbb{R}^3$, $\boldsymbol{\tau}_{\boldsymbol{B}}^{\boldsymbol{i}} \in \mathbb{R}^3$ are the control forces and moments respectively, in the body fixed reference frame. Further, $\boldsymbol{I}_{\boldsymbol{B}}^{\boldsymbol{i}} \in \mathbb{R}^{3\times 3}$ is the constant

nonsingular inertia matrix defined in the body frame and $m^i$ is a scalar quantity that denotes the mass of players' rigid bodies. In addition, $\boldsymbol{f_g^i} = [0\ 0\ m^i g]^T$ is the gravitational force vector whose components are written in the Inertial frame. $\boldsymbol{S}(\boldsymbol{w_B^i}) \in \mathbb{R}^{3\times3}$ represents a skew-symmetric matrix form of the vector $\boldsymbol{w_B^i}$. Moreover, $m^i\boldsymbol{S}(\boldsymbol{w_B^i})\boldsymbol{v_B^i}$ and $\boldsymbol{S}(\boldsymbol{w_B^i})\boldsymbol{I_B^i}\boldsymbol{w_B^i}$ are due to the derivative of the body referenced linear and angular momentum of the vehicles relative to the Inertial frame. $\boldsymbol{N^i} \in \mathbb{R}^{3\times3}$ is the rotation matrix from Inertial to body frame. Later in Section 3.5, we will call this Inertial frame as earth frame and give detailed explanation for the rotation matrix.

## 3.2 Developing velocity tracker using backstepping control method

In this section, we present an exponentially stabilizing backstepping control method to track given velocity trajectories. This velocity tracker is developed in this section, which uses only the translational dynamics (3.1). In Section 3.3, the velocity tracker is extended for PE games based on the translational dynamics (3.1) for both pursuer and evader. Then, in Section 3.5 we also consider rotational dynamics (3.2) to obtain general controllers for both velocity and attitude for pursuer and evader.

Note, we first derive the required velocity tracking control laws in the Inertial frame, and then subsequently Section 3.5 shows how they are realized using the dynamics in (3.1) and (3.2).

Thus, in the Inertial frame the translational dynamics is represented as,

$$m^i\dot{\boldsymbol{v}}^{\boldsymbol{i}} = \boldsymbol{f^i} + \boldsymbol{f_g^i} \tag{3.3}$$

where $\boldsymbol{v^i} \in \mathbb{R}^3$ is the velocity vector and $\boldsymbol{f^i} = \boldsymbol{N^{iT}} \boldsymbol{f_B^i}$ is the control force, in the Inertial frame $\boldsymbol{i} \in \{\boldsymbol{p},\ \boldsymbol{e}\}$. Introducing a desired virtual force $\boldsymbol{f_d^i}$, to the system dynamics (3.3) we obtain

$$m^i \dot{\boldsymbol{v}}^i = \boldsymbol{f_d^i} + \boldsymbol{f_g^i} + \widetilde{\boldsymbol{f}}^i \tag{3.4}$$

where $\widetilde{\boldsymbol{f}}^i = \boldsymbol{f^i} - \boldsymbol{f_d^i}$ is the difference of control and desired forces of the Newtonian system in 3-D.

Define velocity error as

$$\boldsymbol{\delta_v^i} = \boldsymbol{v_d^i} - \boldsymbol{v^i} \tag{3.5}$$

where $\boldsymbol{v_d^i} \in \mathbb{R}^3$ is the desired velocity designed for pursuer $\boldsymbol{v_d^p}$ and evader $\boldsymbol{v_d^e}$ in the next section. Take the derivative of (3.5) and substitute in (3.4) to obtain closed-loop velocity error dynamics as

$$m^i \dot{\boldsymbol{\delta}}_{\boldsymbol{v}}^{\boldsymbol{i}} = m^i \dot{\boldsymbol{v}}_{\boldsymbol{d}}^{\boldsymbol{i}} - \boldsymbol{f_g^i} - \widetilde{\boldsymbol{f}}^i - \boldsymbol{f_d^i}. \tag{3.6}$$

Then select ideal desired force as

$$\boldsymbol{f_d^i} = m^i \dot{\boldsymbol{v}}_{\boldsymbol{d}}^{\boldsymbol{i}} - \boldsymbol{f_g^i} + m^i \boldsymbol{K^i} \boldsymbol{\delta_v^i} \tag{3.7}$$

where $\boldsymbol{K^i} \in \mathbb{R}^{n \times n}$ is a positive-definite matrix. Substituting (3.7) in (3.6) yields

$$\dot{\boldsymbol{\delta}}_{\boldsymbol{v}}^{\boldsymbol{i}} = -\boldsymbol{K^i} \boldsymbol{\delta_v^i} - \frac{\widetilde{\boldsymbol{f}}^i}{m^i}. \tag{3.8}$$

This enables us to derive exponential stability of the origin, as long as an admissible $\boldsymbol{f_d^i}$ exists. In Section 3.5 we consider the rotational dynamics and show how to design the control force, $\boldsymbol{f^i}$ and hence $\boldsymbol{f_B^i}$ in (3.1) and (3.3) respectively, to make $\widetilde{\boldsymbol{f}}^i \to \boldsymbol{0}$ [27]. Then (3.8) shows that $\boldsymbol{\delta_v^i} \to \boldsymbol{0}$ exponentially.

**Remark 1.** *Tracking the vector quantity $\boldsymbol{f_d^i}$ in (3.7) not only guarantees exponential stability of the equilibrium of (3.6) but also gives the desired attitude of the Newtonian system (3.3) so that it is aligned with the direction of $\boldsymbol{f_d^i}$.*

The next section deals with the derivation of optimal velocity trajectories $\boldsymbol{v_d^i}$ for pursuer and evader, employed in (3.5). The design of the desired ideal forces $\boldsymbol{f_d^p}$, $\boldsymbol{f_d^e}$ is treated in Section 3.5.

## 3.3 Optimal game theoretic velocity generation for pursuit-evasion game

In this main section, we first propose a formulation of PE game and derive the optimal bounded desired velocity trajectories $\boldsymbol{v_d^p}$,$\boldsymbol{v_d^e}$ in (3.5) for the players. Secondly, we conduct a Nash equilibrium analysis for the game and derive sufficient conditions for global exponential stability of the origin by rigorous Lyapunov analysis. Finally, conditions for finite-time capture and its upper bound are given.

### 3.3.1 Pursuit-evasion game formulation

Assuming the players are governed by the velocity error dynamics (3.8), this section presents various definitions to develop the game-theoretically optimal solution of the PE game satisfying *velocity constraints* on the players. To simplify the notation, define desired velocity in (3.5) for the pursuer $\boldsymbol{v^p} = \boldsymbol{v_d^p}$ and the evader $\boldsymbol{v^e} = \boldsymbol{v_d^e}$.

The following kinematic expressions enable us to derive desired velocities and thereby the forces (3.7) for pursuer and evader

$$\begin{aligned} \boldsymbol{\dot{\xi}^p} &= \boldsymbol{v^p} \\ \boldsymbol{\dot{\xi}^e} &= \boldsymbol{v^e} \end{aligned} \tag{3.9}$$

where $\boldsymbol{\xi^p} \in \mathbb{R}^3$ and $\boldsymbol{\xi^e} \in \mathbb{R}^3$ denote the 3-dimensional position vectors $(x, y, z)$ of pursuer and evader respectively, which are defined with respect to Inertial frame. Hence $\boldsymbol{v^p} \in \mathbb{R}^3$ and $\boldsymbol{v^e} \in \mathbb{R}^3$ are desired velocity vectors of the pursuer and evader respectively. Note that (3.3) employs the translational velocity in the PE game. This allows analysis of ZS game for general nonlinear systems in Section 3.5.

Now, consider the following formulation for the zero-sum (ZS) PE game. Let the evader have an objective of maximizing the relative distance $\boldsymbol{\delta} \in \mathbb{R}^3$, defined as

$$\boldsymbol{\delta} = \boldsymbol{\xi^p} - \boldsymbol{\xi^e}, \tag{3.10}$$

whereas the pursuer tries to minimize (3.10). Moreover, let the velocities of both pursuer and evader be bounded by scalars $|v_j^p| \leq \lambda^p; |v_j^e| \leq \lambda^e \forall j = 1, ...n$. To satisfy these constraints, the value functional is defined as

$$V^{\pi^p, \pi^e}(\boldsymbol{\delta}) = \int_t^\infty \{\boldsymbol{\delta^T Q \delta} + U(\pi^p(\boldsymbol{\delta})) - U(\pi^e(\boldsymbol{\delta})\} d\tau \tag{3.11}$$

where $\boldsymbol{Q} \in \mathbb{R}^{nxn}$ is a positive-definite matrix, $\pi^p(.)$ and $\pi^e(.)$ stand for the policies of pursuer and evader respectively in ZS game such that

$$\begin{aligned} \pi^p(\boldsymbol{\delta}) &\triangleq \boldsymbol{v^p} \\ \pi^e(\boldsymbol{\delta}) &\triangleq \boldsymbol{v^e}. \end{aligned} \tag{3.12}$$

Moreover, $U(\boldsymbol{v^i})$ (for $i$ is either $p$ or $e$) is a generalized non-quadratic scalar functional [73], which ensures bounded velocities given by

$$U(\boldsymbol{v^i}) = 2 \int_{\mathbf{0}}^{\boldsymbol{v^i}} (\alpha^{-1}(\boldsymbol{u^i}/\lambda^i))^T \boldsymbol{R^i d u^i}, \tag{3.13}$$

$$\alpha^{-1}(\boldsymbol{u^i}/\lambda^i) = \begin{bmatrix} \alpha^{-1}(u_1^i/\lambda^i) & ... & \alpha^{-1}(u_n^i/\lambda^i) \end{bmatrix}^T,$$
$$\boldsymbol{u^i} = \begin{bmatrix} u_1^i & ... & u_n^i \end{bmatrix}^T, \boldsymbol{v^i} = \begin{bmatrix} v_1^i & ... & v_n^i \end{bmatrix}^T$$

where $\boldsymbol{R^i} \in \mathbb{R}^{nxn}$ is a symmetric positive-definite matrix and $\alpha(.)$ is a bounded one-to-one smooth function i.e it belongs to $C^\ell, \ell \geq 1$. This is a monotonic odd function with its first derivative bounded by a constant. An example of $\alpha(.)$ is tanh(.) and throughout this paper, we use tanh(.), which constrains the velocity to remain within predefined limits i.e $|v_j^i| \leq \lambda, \forall j = 1, ...n$ and $\forall i = p, e$. In ZS PE games, $\boldsymbol{R^i}$ plays a key role by restricting the rate of change of optimal velocities and hence constrains the accelerations of the each player.

The differential equivalent of (3.11) is the ZS game Bellman equation. Using (3.9), (3.10) and Leibniz's formula, the Bellman equation is obtained as

$$\begin{aligned} H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v^p}, \boldsymbol{v^e}) &\equiv \boldsymbol{\delta^T Q \delta} + U(\boldsymbol{v^p}) - U(\boldsymbol{v^e}) + \nabla V^T \dot{\boldsymbol{\delta}} \\ &\equiv \boldsymbol{\delta^T Q \delta} + U(\boldsymbol{v^p}) - U(\boldsymbol{v^e}) + \nabla V^T (\boldsymbol{v^p} - \boldsymbol{v^e}) = 0 \end{aligned} \tag{3.14}$$

where $\nabla V = \partial V^{\pi^p, \pi^e}/\partial \boldsymbol{\delta} \in \mathbb{R}^n$ is the gradient of value function (3.11), and $H(.)$ is the Hamiltonian.

To find the optimal policies ${\pi^i}_*(\boldsymbol{\delta}) = \boldsymbol{v_*^i}$ (for $i = p, e$) of players in the game, check stationarity conditions $\partial H/\partial \boldsymbol{v^p} = \boldsymbol{0}$ and $\partial H/\partial \boldsymbol{v^e} = \boldsymbol{0}$. For the pursuer, applying Pontryagin's minimum principle to (3.14) yields

$$\frac{\partial H}{\partial \boldsymbol{v^p}} \equiv \frac{\partial U(\boldsymbol{v^p})}{\partial \boldsymbol{v^p}} + \frac{\partial}{\partial \boldsymbol{v^p}} \{\nabla V^T (\boldsymbol{v^p} - \boldsymbol{v^e})\}. \tag{3.15}$$

Evaluating the derivatives at the right-hand-side of (3.15) using Leibniz's formula, and checking the stationarity condition $\partial H/\partial \boldsymbol{v^p} = \mathbf{0}$ yields

$$2\left(\tanh^{-1}(\frac{\boldsymbol{v^{p*}}}{\lambda^p})\right)^T \boldsymbol{R^p} = -\nabla V^{*T}. \tag{3.16}$$

Then, the optimal policy for the pursuer using the definition (3.12) is obtained as

$$\pi^{p*}(\boldsymbol{\delta}) \triangleq \boldsymbol{v^{p*}} = -\lambda^p \tanh\left(\frac{1}{2}(\boldsymbol{R^p})^{-1}\nabla V^*\right). \tag{3.17}$$

This velocity control bounded as required.

Likewise, one can follow the same steps to derive bounded optimal velocity policy for the evader as

$$\pi^{e*}(\boldsymbol{\delta}) \triangleq \boldsymbol{v^{e*}} = -\lambda^e \tanh\left(\frac{1}{2}(\boldsymbol{R^e})^{-1}\nabla V^*\right). \tag{3.18}$$

Let $V^*$ be the optimal value of (3.11) with the policies given in (3.17) and (3.18). Then Hamilton-Jacobi-Isaacs (HJI) equation is obtained as

$$H(\boldsymbol{\delta}, \nabla V^*, \boldsymbol{v_*^p}, \boldsymbol{v_*^e}) \equiv \boldsymbol{\delta^T Q \delta} + U(\boldsymbol{v_*^p}) - U(\boldsymbol{v_*^e}) + \nabla V^{*T}(\boldsymbol{v_*^p} - \boldsymbol{v_*^e}) = 0. \tag{3.19}$$

Note that the positive and negative definiteness of Hessians, $\partial^2 H/\partial \boldsymbol{v^{p2}} > 0$ and $\partial^2 H/\partial \boldsymbol{v^{e2}} < 0$, indeed show that pursuer's optimal policy aims to minimize the Hamiltonian (3.14) whereas evader's aims to maximize. Therefore, $(\boldsymbol{v_*^p}, \boldsymbol{v_*^e})$ is the game-theoretic saddle point. Furthermore, this is a Nash equilibrium since the game is of type ZS and (3.11) is separable [59]. Rigorous analysis of this is shown in Theorem 1.

### 3.3.2 Proof of Nash equilibrium

In this section, we derive the value of PE game at Nash equilibrium. The following lemmas and corollary are necessary steps to prove that the Nash equilibrium is reached with policies (3.17) and (3.18).

**Lemma 1.** *Let $V^{\pi^p,\pi^e}(\boldsymbol{\delta})$ be the corresponding solution of the Hamiltonian (3.14). Then following equality holds*

$$\begin{aligned} H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v^p}, \boldsymbol{v^e}) = H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v^p_*}, \boldsymbol{v^e_*}) + \nabla V^T((\boldsymbol{v^p} - \boldsymbol{v^p_*}) \\ + (\boldsymbol{v^e_*} - \boldsymbol{v^e})) + U(\boldsymbol{v^p}) - U(\boldsymbol{v^p_*}) + U(\boldsymbol{v^e_*}) - U(\boldsymbol{v^e}). \end{aligned} \tag{3.20}$$

*Proof.* Adding and subtracting the terms $U(\boldsymbol{v^p_*})$, $U(\boldsymbol{v^e_*})$, $\nabla V^T \boldsymbol{v^p_*}$ and $\nabla V^T \boldsymbol{v^e_*}$ to Hamiltonian (3.14) yields

$$\begin{aligned} H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v^p}, \boldsymbol{v^e}) = \boldsymbol{\delta^T Q \delta} + \nabla V^T(\boldsymbol{v^p_*} - \boldsymbol{v^e_*}) U(\boldsymbol{v^p_*}) \\ - U(\boldsymbol{v^e_*}) + \nabla V^T((\boldsymbol{v^p} - \boldsymbol{v^p_*}) + (\boldsymbol{v^e_*} - \boldsymbol{v^e})) \\ + U(\boldsymbol{v^p}) - U(\boldsymbol{v^p_*}) + U(\boldsymbol{v^e_*}) - U(\boldsymbol{v^e}), \end{aligned} \tag{3.21}$$

which completes the proof. □

**Lemma 2.** *Let $V^{\pi^p,\pi^e}(\boldsymbol{\delta})$ be the corresponding solution of the Hamiltonian (3.14) and define $V(\boldsymbol{\delta}(t_0))$ as the initial value of the game. Then following equality holds*

$$V^{\pi^p,\pi^e}(\boldsymbol{\delta}(t_0)) = \int_{t_0}^{\infty} H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v^p}, \boldsymbol{v^e}) d\tau + V(\boldsymbol{\delta}(t_0)). \tag{3.22}$$

*Proof.* Assume that capture occurs in the interval $t \in [t_0, \infty]$, which implies $\lim_{t\to\infty} V^{\pi^p,\pi^e}(\boldsymbol{\delta}(t)) = 0$. Then adding zero to (3.11) yields

$$\begin{aligned} V^{\pi^p,\pi^e}(\boldsymbol{\delta}(t_0)) &= \int_{t_0}^{\infty} \{\boldsymbol{\delta}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{\delta} + U(\pi^p(\boldsymbol{\delta})) - U(\pi^e(\boldsymbol{\delta}))\}d\tau + \int_{t_0}^{\infty} \dot{V}^{\pi^p,\pi^e} d\tau + V(\boldsymbol{\delta}(t_0)) \\ &= \int_{t_0}^{\infty} \{\boldsymbol{\delta}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{\delta} + U(\boldsymbol{v^p}) - U(\boldsymbol{v^e})\}d\tau + \int_{t_0}^{\infty} \nabla V^T(\boldsymbol{v^p} - \boldsymbol{v^e})d\tau + V(\boldsymbol{\delta}(t_0)) \\ &= \int_{t_0}^{\infty} H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v^p}, \boldsymbol{v^e})d\tau + V(\boldsymbol{\delta}(t_0)). \end{aligned} \tag{3.23}$$

This completes the proof. □

The next corollary extends the fact given in Lemma 1.

**Corollary 1.** *Suppose $V^*$ satisfies the HJI equation (3.19). Then $H(\boldsymbol{\delta}, \nabla V^*, \boldsymbol{v}_*^{\boldsymbol{p}}, \boldsymbol{v}_*^{\boldsymbol{e}}) = 0$ and (3.20) becomes*

$$H(\boldsymbol{\delta}, \nabla V^*, \boldsymbol{v^p}, \boldsymbol{v^e}) = \nabla V^{*T}((\boldsymbol{v^p} - \boldsymbol{v}_*^{\boldsymbol{p}}) + (\boldsymbol{v}_*^{\boldsymbol{e}} - \boldsymbol{v^e})) + U(\boldsymbol{v^p}) - U(\boldsymbol{v}_*^{\boldsymbol{p}}) + U(\boldsymbol{v}_*^{\boldsymbol{e}}) - U(\boldsymbol{v^e}). \tag{3.24}$$

The next theorem derives the optimal value of the ZS game and proves Nash equilibrium reached.

**Theorem 1.** *Consider kinematic expressions for the players (3.9) with the value function given in (3.11). Let $V^*$ be a positive definite smooth solution of HJI equation (3.19). Then, $(\boldsymbol{v}_*^{\boldsymbol{p}}, \boldsymbol{v}_*^{\boldsymbol{e}})$ given by (3.17),(3.18) is the Nash equilibrium and $V^*(\boldsymbol{\delta}(t_0))$ is the value of PE game.*

*Proof.* Using the facts given in Lemma 2 and Corollary 1, (3.23) becomes

$$\begin{aligned} V^{\pi^p,\pi^e}(\boldsymbol{\delta}(t)) = \int_{t}^{\infty} \{&\nabla V^{*T}((\boldsymbol{v^p} - \boldsymbol{v}_*^{\boldsymbol{p}}) + (\boldsymbol{v}_*^{\boldsymbol{e}} - \boldsymbol{v^e})) \\ &+ U(\boldsymbol{v^p}) - U(\boldsymbol{v}_*^{\boldsymbol{p}}) + U(\boldsymbol{v}_*^{\boldsymbol{e}}) - U(\boldsymbol{v^e})\}d\tau + V^*(\boldsymbol{\delta}(t_0)). \end{aligned} \tag{3.25}$$

To prove $(\boldsymbol{v}_*^{\boldsymbol{p}}, \boldsymbol{v}_*^{\boldsymbol{e}})$ is the Nash equilibrium of the game, we need to show that when the pursuer adopts policy given in (3.17), the best action for the evader to maximize the value function (3.11) is $\boldsymbol{v}^{\boldsymbol{e}*}$. Likewise, when the evader adopts policy given in (3.18), the best action for the pursuer to minimize the value function (3.11) is $\boldsymbol{v}^{\boldsymbol{p}*}$ i.e.

$$V^{\pi^{p*},\pi^e}(\boldsymbol{\delta}(t)) \leq V^{\pi^{p*},\pi^{e*}}(\boldsymbol{\delta}(t)) \leq V^{\pi^p,\pi^{e*}}(\boldsymbol{\delta}(t)). \tag{3.26}$$

Note that $V^{\pi^{p*},\pi^{e*}}(\boldsymbol{\delta}(t)) = V^*(\boldsymbol{\delta}(0))$ and call the integral term in (3.25) as $\beta(V^{\pi^p,\pi^e})$. Now we need to show $\beta(V^{\pi^{p*},\pi^e}) \leq 0$ and $\beta(V^{\pi^p,\pi^{e*}}) \geq 0$ so that (3.26) holds. Then using (3.13), (3.17),(3.18) and (3.25) we obtain

$$\begin{aligned}\beta(V^{\pi^{p*},\pi^e}) &= \int_t^\infty \{\nabla V^{*T}(\boldsymbol{v}_*^{\boldsymbol{e}} - \boldsymbol{v}^{\boldsymbol{e}}) + U(\boldsymbol{v}_*^{\boldsymbol{e}}) - U(\boldsymbol{v}^{\boldsymbol{e}})\}d\tau \\ &= \int_t^\infty \{-2(\tanh^{-1}(\boldsymbol{v}^{\boldsymbol{e}*}/\lambda^e))^T \boldsymbol{R}^{\boldsymbol{e}}(\boldsymbol{v}_*^{\boldsymbol{e}} - \boldsymbol{v}^{\boldsymbol{e}}) + 2\int_{\boldsymbol{v}^{\boldsymbol{e}}}^{\boldsymbol{v}_*^{\boldsymbol{e}}} (\tanh^{-1}(\boldsymbol{u}/\lambda^e))^T \boldsymbol{R}^{\boldsymbol{e}}\boldsymbol{du}\}d\tau.\end{aligned} \tag{3.27}$$

Now define $\phi^T(.) = \tanh^{-1}(.)$ and note that $\phi^T(.)$ is monotonically increasing function in the interval $[-\lambda^e, \lambda^e]$. To complete the proof, first assume that $\boldsymbol{v}^{\boldsymbol{e}*} \geq \boldsymbol{v}^{\boldsymbol{e}}$ and apply integral mean value theorem on (3.27)

$$\begin{aligned}\beta(V^{\pi^{p*},\pi^e}) &= \int_t^\infty \{-2\phi(\boldsymbol{v}^{\boldsymbol{e}*}/\lambda^e)\boldsymbol{R}^{\boldsymbol{e}}(\boldsymbol{v}_*^{\boldsymbol{e}} - \boldsymbol{v}^{\boldsymbol{e}}) + 2\int_{\boldsymbol{v}^{\boldsymbol{e}}}^{\boldsymbol{v}_*^{\boldsymbol{e}}} \phi(\boldsymbol{u}/\lambda^e)\boldsymbol{R}^{\boldsymbol{e}}\boldsymbol{du}\}d\tau \\ &\leq \int_t^\infty \{-2\phi(\boldsymbol{v}^{\boldsymbol{e}*}/\lambda^e)\boldsymbol{R}^{\boldsymbol{e}}(\boldsymbol{v}_*^{\boldsymbol{e}} - \boldsymbol{v}^{\boldsymbol{e}}) + 2\phi(\boldsymbol{v}^{\boldsymbol{e}*}/\lambda^e)\boldsymbol{R}^{\boldsymbol{e}}(\boldsymbol{v}_*^{\boldsymbol{e}} - \boldsymbol{v}^{\boldsymbol{e}})\}d\tau = 0.\end{aligned} \tag{3.28}$$

Then assume that $\boldsymbol{v}^{\boldsymbol{e}*} < \boldsymbol{v}^{\boldsymbol{e}}$ and again apply integral mean value theorem on (3.27)

$$\beta(V^{\pi^{p*},\pi^e}) = \int_t^\infty \{2\phi(\boldsymbol{v}^{\boldsymbol{e}*}/\lambda^e)\boldsymbol{R}^{\boldsymbol{e}}(\boldsymbol{v}^{\boldsymbol{e}} - \boldsymbol{v}^{\boldsymbol{e}*}) - 2\int_{\boldsymbol{v}_*^{\boldsymbol{e}}}^{\boldsymbol{v}^{\boldsymbol{e}}} \phi(\boldsymbol{u}/\lambda^e)\boldsymbol{R}^{\boldsymbol{e}}\boldsymbol{du}\}d\tau$$

$$\leq \int_t^{\infty} \{2\phi(\boldsymbol{v^{e*}}/\lambda^e)\boldsymbol{R^e}(\boldsymbol{v^e} - \boldsymbol{v_*^e}) - 2\phi(\boldsymbol{v^{e*}}/\lambda^e)\boldsymbol{R^e}(\boldsymbol{v^e} - \boldsymbol{v_*^e})\}d\tau = 0, \tag{3.29}$$

which shows that $\beta(V^{\pi^{p*},\pi^e}) \leq 0$. The same procedure can be performed to show $\beta(V^{\pi^p,\pi^{e*}}) \geq 0$. Then the inequality given in (3.26) is verified, which implies that $(\boldsymbol{v_*^p}, \boldsymbol{v_*^e})$ is the Nash equilibrium and $V^*(\boldsymbol{\delta}(t_0))$ is the value of the PE game. □

### 3.3.3 Stability and finite-time capture analysis

This section first reveals the sufficient conditions for the asymptotic capture of the evader by the pursuer. Then, by making use of these conditions, derives the globally exponential stability of the origin. Finally it is shown that under certain conditions, finite-time capture is ensured.

Before developing analysis for the asymptotic capture, let $\boldsymbol{R^i}$ in (3.17) and (3.18) be a diagonal matrix with elements of $r_j^i > 0, \forall j \in \{1, 2, 3\}$ and $\forall i = p, e$. This enables us to simplify the analysis that will be developed in the rest of the paper. Employing this assumption, the next theorem shows the sufficient conditions for the asymptotic capture in ZS PE games.

**Theorem 2.** *Consider kinematic expressions for the players (3.9) with the value function given in (3.11). Then the equilibrium of tracking error dynamics* $\dot{\boldsymbol{\delta}} = \boldsymbol{v^{p*}} - \boldsymbol{v^{e*}}$, *is asymptotically stable point with candidate Lyapunov function* $L(\boldsymbol{\delta}) = V^{\pi^{p*},\pi^{e*}}(\boldsymbol{\delta})$. *The sufficient conditions for asymptotic capture are* $\lambda^p > \lambda^e$ *and* $r_{e_i} \geq r_{p_i} \forall i = 1, ...n$.

*Proof.* Since $V^{\pi^{p*},\pi^{e*}}(\boldsymbol{\delta})$ does not depend on the time explicitly, equality $\dot{L}(\boldsymbol{\delta}) = \nabla L^T \dot{\boldsymbol{\delta}}$ holds. By (3.14), derivative of the Lyapunov function, $\dot{L}(\boldsymbol{\delta})$ is obtained as

$$\dot{L}(\boldsymbol{\delta}) = -\boldsymbol{\delta^T Q \delta} - U(\boldsymbol{v^{p*}}) + U(\boldsymbol{v^{e*}}). \tag{3.30}$$

Assumption of the equality $\boldsymbol{R^p} = \boldsymbol{R^e}$, implies the pursuer and evader are moving in the same direction by (3.17) and (3.18). For intercept, the position of the pursuer and evader must be equal. To meet this criteria, we propose $\lambda^p > \lambda^e$ so that asymptotic capture occurs as the row elements of optimal actions satisfy $|v^{p*}|_i > |v^{e*}|_i \; \forall i = 1, ..., n$ . Furthermore, taking (3.30) into account, the condition of $\boldsymbol{R^p} = \boldsymbol{R^e}$ is relaxed as $\boldsymbol{R^e} \geq \boldsymbol{R^p}$ since proposition $\lambda^p > \lambda^e$ implies $U(\boldsymbol{v^{p*}}) \geq U(\boldsymbol{v^{e*}})$, $\dot{L}(\boldsymbol{\delta})$ becomes strictly negative definite. Then sufficient conditions for the asymptotic capture is proved to be $\lambda^p > \lambda^e$ and $r_{e_i} \geq r_{p_i} \forall i = 1, ...n$. □

**Remark 2.** *Aysmptotic capture in Theorem 2 can be strengthened to finite-time capture with the assumption that players involved in the game satisfy the sufficient conditions derived in the proof of Theorem 2. See Lemma 3.*

Following theorem extends the Theorem 2 to exponential stability of the origin.

**Theorem 3.** *Consider sufficient conditions and Lyapunov function, $L(\boldsymbol{\delta})$ given in Theorem 2. Then, there exists positive scalars $c_1$, $c_2$ and $\epsilon$, which satisfies*

$$\begin{aligned} c_1 ||\boldsymbol{\delta}||_2^2 \leq L(\boldsymbol{\delta}) \leq c_2 ||\boldsymbol{\delta}||_2^2 \\ \dot{L}(\boldsymbol{\delta}) \leq -\epsilon L(\boldsymbol{\delta}), \end{aligned} \tag{3.31}$$

*which implies that the origin is an exponentially stable equilibrium. Furthermore, radially unboundedness of the $L(\boldsymbol{\delta})$ implies the globally exponentially stability of the origin [74], which is an essential result as the initial positional offset between the pursuer and evader should not be problem to prove the capture in PE game.*

*Proof.* The inequality $U(\boldsymbol{v^{p*}}) \geq U(\boldsymbol{v^{e*}})$ by Thoerem 2 and the strict convexity of $U(\boldsymbol{v}_i)$ (for $i = p, e$), imply the existence of positive scalars $c_1$ and $c_2$ [75]. Now, define

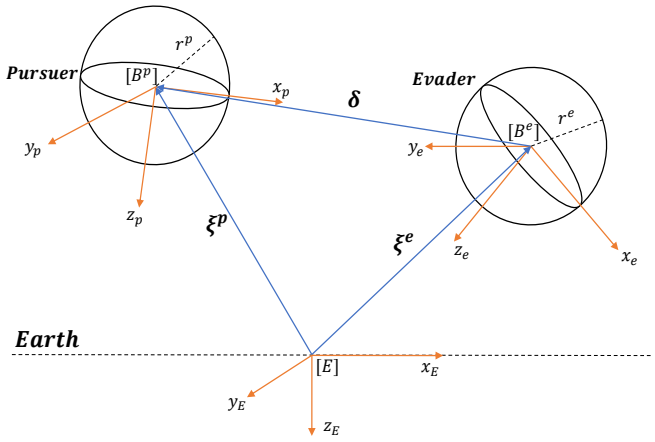


Figure 16: Sphere of collisions for players and their frames in 3-dimensions used for finite-time capture analysis.

convex function $U_s(\boldsymbol{\delta})$ that satisfies the inequality $U_s(\boldsymbol{\delta}) \leq U(\boldsymbol{v^{p*}}) - U(\boldsymbol{v^{e*}})$. Using this and (3.30), the following inequality is derived as

$$\dot{L}(\boldsymbol{\delta}) \leq -\boldsymbol{\delta^T Q \delta} - U_s(\boldsymbol{\delta}). \tag{3.32}$$

Substituting (3.32) in (3.11) with the optimal policies (3.17) and (3.18), results in

$$L(\boldsymbol{\delta}) \leq \int_t^\infty \{\boldsymbol{\delta^T Q \delta} + U_s(\boldsymbol{\delta})\} d\tau, \tag{3.33}$$

which stands for the proof of $\dot{L}(\boldsymbol{\delta}) \leq -\epsilon L(\boldsymbol{\delta})$ for sufficiently small $\epsilon$, which completes the proof. □

Notice that PE game given in Section 3.3 is formulated by treating the players as unit masses since the kinematic expressions (3.9) is employed in the value function (3.11). In Section 3.5, we consider full nonlinear dynamics (3.1), (3.2). Now, consider

the volume of pursuer and evader in 3-dimensional space and let the pursuer and evader have a sphere of collision with radius $r^p$ and $r^e$ respectively, as illustrated in Fig. 26. Then capture occurs when the distance between the center of masses of players is less than $r^p + r^e$. With this in mind, the next main lemma proves that the capture of evader by pursuer indeed occurs in finite-time in PE game.

**Lemma 3.** *There exists an upper-bound for the capture time in PE game when the conditions $\lambda^p > \lambda^e$ and $r_{e_i} \geq r_{p_i} \forall i = 1, ...n$ derived in Theorem 2 are satisfied. This also implies that the PE game ends in finite-time as required.*

*Proof.* The globally exponentially stability of the origin derived in Theorem 3 implies the equation of positional offset between the players is in the form of

$$||\boldsymbol{\delta}(t)||_2 \leq b_1 ||\boldsymbol{\delta}(t_0)||_2 e^{-b_2(t-t_0)} \quad \forall t > t_0 \tag{3.34}$$

where $b_i$ is a positive scalar $\forall i = 1, 2$. Then the upper bound for capture time $t_c$ is derived as

$$t_c \leq t_0 + \frac{1}{b_2} \log \left( \frac{b_1 ||\boldsymbol{\delta}(t_0)||_2}{r^e + r^p} \right) \tag{3.35}$$

where log(.) is a natural logarithm function and this completes the proof. □

**Remark 3.** *It is seen that for finite-time capture, the velocity bound $\lambda^p$ for the pursuer must be greater then the velocity bound $\lambda^e$ on the evader. Moreover, the sufficient condition on weights (3.13) is found as $r_{e_i} \geq r_{p_i} \forall i = 1, ...n$. Note that capture time is also studied for multi-agent systems in the work [76] by assuming the players are using their maximum efforts. In Lemma 3, we showed that capture time is upper bounded under certain conditions even the players are not using their maximum efforts.*

## 3.4 Online solution of HJI equation using integral reinforcement learning (IRL)

The PE game formulation in Section 3.3 requires the generation of velocity set-points online and in real-time for both agents of the game. With this in mind, we employ the following synchronous IRL algorithm [8] to solve the HJI equation (3.19) in real-time and hence, reach the Nash equilibrium velocity policies $(\boldsymbol{v}_*^{\boldsymbol{p}}, \boldsymbol{v}_*^{\boldsymbol{e}})$ online by observing measured data. In the work [28], it is emphasized that persistence of excitation condition must be satisfied so that the IRL algorithm convergences. This is achieved in most applications [8]-[9] by adding small probing noise. In our case, the persistence of excitation for IRL to work is satisfied inherently until capture occurs, at which time the game ends.

The tracking HJI equation (3.19) is nonlinear in the value function gradient $\nabla V^*$, and non-quadratic partial differential equation that is extremely difficult to solve. However, (3.36) can be solved for the value function and its gradient by collecting position data over some interval $[t, t+T]$. Therefore, finding the value of game optimal velocity policies by solving (3.36) is easier than solving (3.19). This is the motivation of introducing an iterative algorithm for approximating the tracking HJI solution, which is necessary to evaluate game optimal velocity policy for pursuer (3.17), and evader (3.18).

### 3.4.1 Policy iteration solution for PE game

In this section, we present a policy iteration algorithm that avoids solution of (3.19) and also does not require knowledge of the system dynamics. The following lemma enables us to recognize IRL form of the value function (3.11).

**Lemma 4.** *Let $V^{\pi^p,\pi^e}(\boldsymbol{\delta})$ be the corresponding solution of the Bellman equation (3.14). Then, the value function (3.11), can be written in the IRL form as*

$$V^{\pi^p,\pi^e}(\boldsymbol{\delta}(t)) = \int_t^{t+T} \{\boldsymbol{\delta}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{\delta} + U(\pi^p(\boldsymbol{\delta})) - U(\pi^e(\boldsymbol{\delta}))\}d\tau + V^{\pi^p,\pi^e}(\boldsymbol{\delta}(t+T)). \quad (3.36)$$

*Proof.* The equality $\dot{V}^{\pi^p,\pi^e} = -\boldsymbol{\delta}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{\delta} - U(\pi^p(\boldsymbol{\delta})) + U(\pi^e(\boldsymbol{\delta}))$ holds by the differentiation of ZS game Bellman equation (3.14). Then integrating both sides from $t$ to $t+T$, results in

$$\int_t^{t+T} \dot{V}d\tau = -\int_t^{t+T} \{\boldsymbol{\delta}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{\delta} + U(\pi^p(\boldsymbol{\delta})) - U(\pi^e(\boldsymbol{\delta}))\}d\tau, \quad (3.37)$$

which verifies (3.36). □

The online policy-iteration Algorithm 1 performs a sequence of four-step iterations to find the optimal control policies for players. Notice that these policies stand for the optimal desired velocities, which are employed in (3.5). Furthermore, they are also Nash equilibrium velocity policies by Theorem 1.

1. Select any policy $\pi_0^p$ and $\pi_0^e$ for the players
2. Policy evaluation

$$V^{\pi_j^p,\pi_j^e}(\boldsymbol{\delta}(t)) = \int_t^{t+T} \{\boldsymbol{\delta}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{\delta} + U(\pi_j^p(\boldsymbol{\delta})) - U(\pi_j^e(\boldsymbol{\delta})\}d\tau + V^{\pi_j^p,\pi_j^e}(\boldsymbol{\delta}(t+T)). \quad (3.38)$$

3. Policy improvement

$$\begin{aligned} \pi_{j+1}^p(\boldsymbol{\delta}) &= -\lambda^p \tanh\left(\frac{1}{2}(\boldsymbol{R}^{\boldsymbol{p}})^{-1}\nabla V^{\pi_j^p,\pi_j^e}\right), \\ \pi_{j+1}^e(\boldsymbol{\delta}) &= -\lambda^e \tanh\left(\frac{1}{2}(\boldsymbol{R}^{\boldsymbol{e}})^{-1}\nabla V^{\pi_j^p,\pi_j^e}\right). \end{aligned} \quad (3.39)$$

4. On convergence stop; else go to step 2. □

**Algorithm 1:** Online policy-iteration algorithm

Notice that the position data of each player is collected through each iteration over the period $T$. The proof of convergence of Algorithm 1 to the optimal policies is shown in the following theorem.

**Theorem 4.** *Using the temporal difference (TD) learning method, Algorithm 1 converges to the Nash value $V^*(\boldsymbol{\delta}(t_0))$ and Nash equilibrium policies $(\pi^{p*}, \pi^{e*})$, which optimizes velocity trajectories for the players in a game theoretic manner.*

*Proof.* First, evaluate the value function $V^{\pi^{pj},\pi^{ej}}(\boldsymbol{\delta}(t))$, which solves the (3.38) by TD method. Then by Theorem 1, Isaacs' condition is derived as

$$H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v_*^p}, \boldsymbol{v^e}) \leq H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v_*^p}, \boldsymbol{v_*^e}) \leq H(\boldsymbol{\delta}, \nabla V, \boldsymbol{v^p}, \boldsymbol{v_*^e}). \tag{3.40}$$

Noting $\beta(V^{\pi^{p*},\pi^e}) \leq 0$ and $\beta(V^{\pi^p,\pi^{e*}}) \geq 0$ is proved in the Theorem 1, the uniform convergence of Algorithm 1 immediately follows from Dini's theorem as reinforcement $H(\boldsymbol{\delta}, \nabla V^j, \boldsymbol{v^p}, \boldsymbol{v^e})$ converges to $H(\boldsymbol{\delta}, \nabla V^*, \boldsymbol{v_*^p}, \boldsymbol{v_*^e}) = 0$ by Corollary 1. Moreover, due to the uniqueness of the value function (3.11), it follows that $lim_{j\to\infty} V^{\pi_j^p,\pi_j^e}(\boldsymbol{\delta}(t)) = V^*(\boldsymbol{\delta}(t_0))$. □

### 3.4.2 Value function approximation to find game optimal pursuer and evader velocity policies

This section presents a critic neural network structure for policy-evaluation step in Algorithm 1.

**Remark 4.** *The IRL method given in Algorithm 1 requires the value function approximation (VFA), which can be achieved in a least-squares sense that is also known as single hidden layer critic Neural Network (NN). We employ this technique as in [8] that guarantees the successive least-squares iterations converge to the optimal value function of the HJI equation (3.19), and hence $\nabla V^*$.*

**Remark 5.** *Note that the pair* $(\boldsymbol{v}_*^{\boldsymbol{p}}, \boldsymbol{v}_*^{\boldsymbol{e}})$ *stands for the Nash equilibrium by Theorem 1, thereby the Algorithm 1 converges to optimal actions for both players. Unlike the works [9] and [77] that use the IRL technique to reach min or max point of the value functional, we employ this technique to converge game theoretic saddle point by using the Isaacs' condition derived in Theorem 4. In addition, the system dynamics (3.1) does not appear in the value functional, which implies that we do not need to implement actor NN [77] [78] and the solution of HJI (3.19) can be obtained by using only critic NN, see [79].*

By Remarks 4 and 10, we approximate the game optimal value functional in step 2 of Algorithm 1 using Weierstrass approximator such that

$$\begin{aligned} \hat{V}(\boldsymbol{\delta}) &= \hat{\boldsymbol{W}}^{\boldsymbol{T}} \Phi(\boldsymbol{\delta}), \\ \nabla \hat{V} &= \Phi(\boldsymbol{\delta})^T \hat{\boldsymbol{W}} \end{aligned} \tag{3.41}$$

where $\Phi(\boldsymbol{\delta}) \in \mathbb{R}^{nk}$ is the $k$-times concatenated basis function vector, $n = 3$ as $\boldsymbol{\delta} \in \mathbb{R}^3$, and $\hat{\boldsymbol{W}}$ is a critic NN weight vector to be determined. Using (3.41), the *policy evaluation* step of the IRL Algorithm 1 can be re-written as

$$e_b = \hat{\boldsymbol{W}}^{\boldsymbol{T}} \triangle \boldsymbol{\Phi}(\boldsymbol{\delta}) - \kappa(t) \tag{3.42}$$

where $e_b$ is the continuous-time counterpart residual error of the TD, $\triangle \boldsymbol{\Phi}(\boldsymbol{\delta}) = \Phi(\boldsymbol{\delta}(t)) - \Phi(\boldsymbol{\delta}(t+T))$, and reinforcement

$$\kappa(t) = \int_t^{t+T} \{\boldsymbol{\delta}^{\boldsymbol{T}} \boldsymbol{Q} \boldsymbol{\delta} + U(\pi^p(\boldsymbol{\delta})) - U(\pi^e(\boldsymbol{\delta}))\} d\tau. \tag{3.43}$$

Therefore, (3.42) implies that the problem of solving the HJI equation is converted to tuning the critic NN weights such that $e_b$ to be minimized. Now, to adjust these weights, the following objective function is employed

$$E_b = \frac{1}{2} e_b^2. \tag{3.44}$$

Then, the TD gradient descent algorithm [77] to minimize $e_b$ is obtained by using the chain rule

$$\dot{\hat{\boldsymbol{W}}} = -\frac{\alpha_L \triangle \boldsymbol{\Phi}(\boldsymbol{\delta})}{(1 + \triangle \boldsymbol{\Phi}(\boldsymbol{\delta})^T \triangle \boldsymbol{\Phi}(\boldsymbol{\delta}))^2} e_b \tag{3.45}$$

where $\alpha_L > 0$ is the learning rate. The proof of convergence of critic NN weights is shown in the Theorem 3 of Modares et al. [77].

## 3.5 Generalized rotational dynamics of the pursuer and evader

The analysis in the preceding sections has shown how to derive velocity tracker for the PE game given velocity dynamics (3.3). In this section, we analyze the general rotational dynamics (3.2) that are coupled to (3.1), and hence (3.3). We first derive the desired attitude of the system by using the $Z$-$Y$-$X$ Euler angle rotation matrix from [E] (earth frame) to [B$^i$] (body frames of pursuer or evader) as shown in Fig. 26, and desired force vector $\boldsymbol{f_d^i}$ in (3.7). Then, by the analysis developed on the desired Euler angles, we propose the desired attitude representation with unit quaternions to overcome the singularity problem of the Euler angles. Lastly, by treating forces and/or moments as final inputs to the Newtonian system, we close the backstepping control loop to track desired force vector $\boldsymbol{f_d^i}$ in (3.7). Note that in this section, $i$ represents either $p$ or $e$.

Assume that the gravity $g$ is constant and the Earth is flat in the 3-dimensional space as illustrated in the Fig. 26. Then, the vehicle carrier frame is aligned with the body frame [B$^i$]. Thereby the rotation matrix from [E] to [B$^i$] frames shown in Fig. 26, can be given in terms of the Euler angles as

$$\boldsymbol{N(\eta^i)} = \begin{bmatrix} c\theta^i c\psi^i & c\theta^i s\psi^i & -s\theta^i \\ -c\varphi^i s\psi^i + s\varphi^i s\theta^i c\psi^i & c\varphi^i c\psi^i + s\varphi^i s\theta^i s\psi^i & s\varphi^i c\theta^i \\ s\varphi^i s\psi^i + c\varphi^i s\theta^i c\psi^i & -s\varphi^i c\psi^i + c\varphi^i s\theta^i s\psi^i & c\varphi^i c\theta i \end{bmatrix} \tag{3.46}$$

where $c$ and $s$ refers to cosine and sine respectively, and $\boldsymbol{\eta^i} = [\psi^i \ \theta^i \ \varphi^i]^T$ is the Euler angle vector. Note that $\boldsymbol{N(\eta^i)}$ belongs to the special orthogonal group and is of rank 3, or $SO(3)$, whose determinant is equal to 1.

Assuming the direction of the thrust force to be along the nose of players' bodies or positive $x_i$-axis ($\forall i = p, e$). This enables us to write that the desired force vector is indeed in the form of $\boldsymbol{f^i_{B_d}} = [\mu^i_d \ 0 \ 0]^T$, whose components written in [B$^i$]. Using (3.7) and expressing the desired force $[\mu^i_d \ 0 \ 0]^T$ in [E] by $\boldsymbol{f^i_d} = \boldsymbol{N^T(\eta^i_d)} \boldsymbol{f^i_{B_d}}$, following relation is derived

$$\boldsymbol{f^i_d} = \begin{bmatrix} f^i_{xd} \\ f^i_{yd} \\ f^i_{zd} \end{bmatrix} = \boldsymbol{N^T(\eta^i_d)} \begin{bmatrix} \mu^i_d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mu^i_d(c\theta^i_d c\psi^i_d) \\ \mu^i_d(c\theta^i_d s\psi^i_d) \\ \mu^i_d(-s\theta^i_d) \end{bmatrix} \tag{3.47}$$

where $\boldsymbol{\eta^i_d} = [\psi^i_d \ \theta^i_d \ \varphi^i_d]^T$ is the desired Euler angle vector.

Then, (3.47) can be solved for desired attitude angles $\theta_d$, $\psi_d$ and $\mu_d$ as

$$\theta^i_d = -\tan^{-1}\left(\frac{f^i_{zd}}{f_{xd}\cos\psi^i_d + f^i_{yd}\sin\psi^i_d}\right), \tag{3.48}$$

$$\psi_d^i = \tan^{-1}\left(\frac{f_{yd}^i}{f_{xd}^i}\right), \tag{3.49}$$

$$\mu_d^i = \sqrt{{f_{xd}^i}^2 + {f_{yd}^i}^2 + {f_{zd}^i}^2}. \tag{3.50}$$

Note that $\varphi_d^i$ can be arbitrarily prescribed. However, (3.48)-(3.50) assumes that the equality conditions $f_{xd}^i = 0$, $f_{yd}^i = 0$ cannot occur simultaneously since (3.48) and (3.49) become indefinite. This singularity problem is also known as gimbal lock, which is associated with $\theta_d^i = \pi/2$.

To avoid gimbal lock, define the following unit quaternion representation

$$\boldsymbol{q^i} = \begin{bmatrix} q_0^i & q_1^i & q_2^i & q_3^i \end{bmatrix}^T = \begin{bmatrix} q_0^i; \boldsymbol{q_v^i} \end{bmatrix} \tag{3.51}$$

$$q_0^i = \cos\phi^i/2 \tag{3.52}$$

$$\boldsymbol{q_v^i} = \boldsymbol{k^i}\sin\phi^i/2 \tag{3.53}$$

where $\phi^i$ is the rotation about equivalent axis $\boldsymbol{k^i}$. Moreover, the kinematics equation for unit quaternion is

$$\dot{\boldsymbol{q}}^{\boldsymbol{i}} = \frac{1}{2}\boldsymbol{J^T}(\boldsymbol{q^i})\boldsymbol{w_B^i} \tag{3.54}$$

where $\boldsymbol{J}(\boldsymbol{q^i}) \in \mathbb{R}^{3\times 4}$ satisfies the equalities $\boldsymbol{J}(\boldsymbol{q^i})\boldsymbol{J^T}(\boldsymbol{q^i}) = \boldsymbol{I_{3\times 3}}$, $\boldsymbol{J}(\boldsymbol{q^i})\boldsymbol{q^i} = \boldsymbol{0}$, and can be expressed as

$$\boldsymbol{J}(\boldsymbol{q^i}) = [-\boldsymbol{q_v^i} \quad \boldsymbol{S}(\boldsymbol{q_v^i}) + q_0^i\boldsymbol{I_{3\times 3}}] \tag{3.55}$$

$$\text{where } \boldsymbol{S}(\boldsymbol{q_v^i}) = \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix}.$$

Then, the rotation matrix from [B$^i$] to [E] in terms of the unit quaternion (3.51) is given by

$$\boldsymbol{N^T(q^i)} = \boldsymbol{I_{3\times 3}} - 2q_0^i\boldsymbol{S(q_v^i)} + 2\boldsymbol{S^2(q_v^i)}, \tag{3.56}$$

which is also known as *Rodrigues* formula. The following set of equations can be obtained by substituting the rotation matrix with the argument $\boldsymbol{q_d^i}$ (3.56) into (3.47) along with selected $\varphi_d^i$

$$\begin{aligned} \begin{bmatrix} f_{x_d}^i \\ f_{y_d}^i \\ f_{z_d}^i \end{bmatrix} &= \mu_d^i \begin{bmatrix} 1 + 2(-{q_{2_d}^i}^2 - {q_{3_d}^i}^2) \\ 2q_{0_d}^i q_{3_d}^i + q_{1_d}^i q_{2_d}^i \\ -2q_{0_d}^i q_{2_d}^i + q_{1_d}^i q_{3_d}^i \end{bmatrix} \\ \varphi_d &= \tan^{-1}\Big(\frac{2(q_{0_d}^i q_{1_d}^i + q_{2_d}^i q_{3_d}^i)}{1 - 2({q_{1_d}^i}^2 + {q_{2_d}^i}^2)}\Big). \end{aligned} \tag{3.57}$$

Notice that $\boldsymbol{f_d^i} = \begin{bmatrix} f_{x_d}^i & f_{y_d}^i & f_{z_d}^i \end{bmatrix}^T$, $\varphi_d^i$ and $\mu_d^i$ are known by (3.7) and (3.48)-(3.50). Thence, (3.57) can be solved for the desired unit quaternion $\boldsymbol{q_d^i} = \begin{bmatrix} q_{0_d}^i & q_{1_d}^i & q_{2_d}^i & q_{3_d}^i \end{bmatrix}^T$ as (3.57) represents four equations with four unknowns, which are the elements of $\boldsymbol{q_d^i}$. Further substitute $\boldsymbol{q_d^i}$ into kinematics equation (3.54) to find the desired angular velocity $\boldsymbol{w_{B_d}^i}$ such that

$$\boldsymbol{w_{B_d}^i} = 2\boldsymbol{J(q_d^i)}\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}}. \tag{3.58}$$

**Remark 6.** *For any Newtonian system (3.1) or (3.3) and (3.2), we know that forces and moments are coupled to each other, which implies that $\boldsymbol{\tau_B^i}$ is required to be compatible with the selected desired force $\boldsymbol{f_d^i}$ in (3.7).*

Then applying the dynamic inversion technique, $\boldsymbol{\tau_B^i}$ is given using (3.58) as

$$\boldsymbol{\tau_B^i} = \boldsymbol{I_B^i}\dot{\boldsymbol{w}}_{\boldsymbol{B_d}}^{\boldsymbol{i}} - \boldsymbol{S(w_{B_d}^i)I_B^i w_{B_d}^i}. \tag{3.59}$$

Notice that we treat $\boldsymbol{\tau_B^i}$ as final input for the general rotational dynamics (3.2). Consequently, we will not develop further analysis by giving location of thrusters and actuators, which is a control allocation problem and out of scope of this paper. Interested reader can check our work [27] to examine how to generate $\boldsymbol{\tau_B^i}$ for the quadrotors.

## 3.6 Implementation on dynamic system

This section reveals the simulation results of ZS PE game with different scenarios. First, we consider when both the pursuer and evader follows their game optimal velocities given in (3.17) and (3.18) respectively. Then, we show the scenario in which the pursuer tracks its game optimal velocity (3.17) whereas the evader adopts a sub-optimal velocity policy.

In order to model the constrained optimal velocity trajectories, (3.13) is evaluated for pursuer and evader. Then, the resultant integral is found as

$$U(\boldsymbol{v^{i*}}) = \lambda^i (\nabla V^*)^T \tanh(\boldsymbol{v^{i*}}) - 2\lambda^i \underline{\boldsymbol{R}}^{\boldsymbol{i}} \log(\cosh(\boldsymbol{v^{i*}})) \quad \forall i = p, e \tag{3.60}$$

where $\log(.)$ is the natural logarithm, $\underline{\boldsymbol{R}}^{\boldsymbol{i}} \triangleq diag(\boldsymbol{R^i}) = [r_1^i \; r_2^i \; r_3^i]^T$, and $\boldsymbol{v^{i*}}$ stands for the optimal velocity policy given by (3.17) and (3.18) $\forall i = p, e$.

When the evader is moving with the sub-optimal velocity, we set $U(\pi^e(\boldsymbol{\delta}))$ term in (3.11) to zero and thereby we obtain Hamilton Jacobi Bellman (HJB) equation instead of HJI (3.19). Notice that HJB equation in this case stands for the single player game where the pursuer is the only player. Furthermore, the existence of unique Nash equilibrium by Theorem 1 implies that the value functional (3.11) is convex in $\boldsymbol{v^{p*}}$ for $|v_j^e| \leq \lambda^e \; \forall j \in \{1, 2, 3\}$ given in (3.13), and the functional (3.11) is concave in $\boldsymbol{v^{e*}}$ for $|v_j^p| \leq \lambda^p \; \forall j \in \{1, 2, 3\}$. Then, (3.11) is separable, and solution of

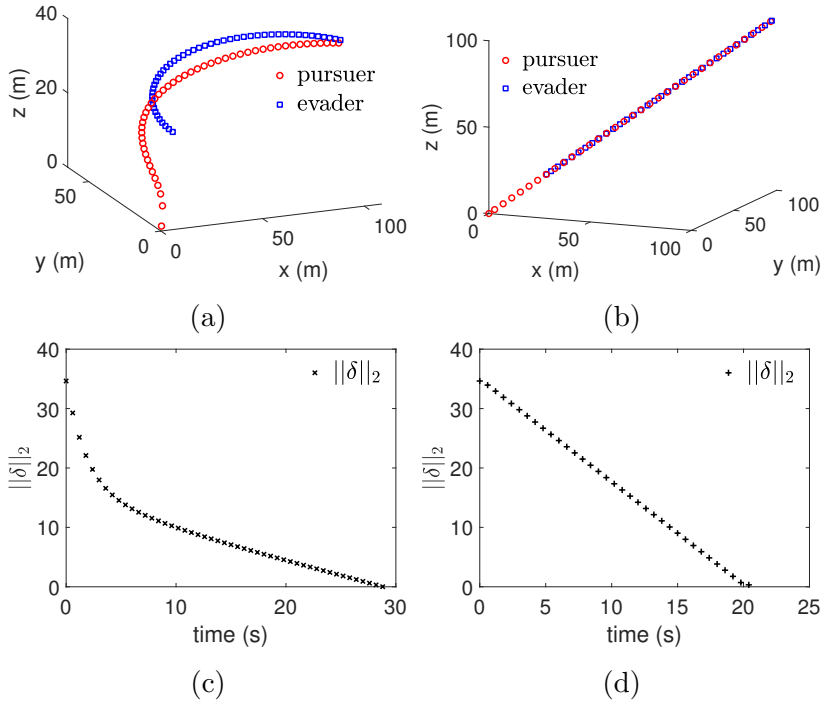


Figure 17: Position of the pursuer and evader: (a) $\pi^p(\boldsymbol{\delta}) \triangleq$ (3.17), $\pi^e(\boldsymbol{\delta}) \triangleq$ *suboptimal*, (b) $\pi^p(\boldsymbol{\delta}) \triangleq$ (3.17), $\pi^e(\boldsymbol{\delta}) \triangleq$ (3.18). $L_2$ norm of (3.10): (c) $\pi^p(\boldsymbol{\delta}) \triangleq$ (3.17), $\pi^e(\boldsymbol{\delta}) \triangleq$ *suboptimal*, (d) $\pi^p(\boldsymbol{\delta}) \triangleq$ (3.17), $\pi^e(\boldsymbol{\delta}) \triangleq$ (3.18).

the HJB in terms of the optimal velocity policy for the pursuer remains the same as (3.17).

We conducted two simulation scenarios to validate the proposed methods in this paper. We first consider ZS game with the value functional (3.11), and get the players track desired game optimal velocity trajectories (3.17), (3.18) by selecting ideal forces of players derived in (3.7) and corresponding moments (3.59). Then, we set $U(\pi^e(\boldsymbol{\delta}))$ term in (3.11) to zero, and by solving the corresponding HJB equation, we played single-player game where the pursuer is the only player. Fig. 27 shows the trajectories followed by the players for each of these scenarios.

In these simulations (Figs. 27 and 28), parameters of the system (3.1), (3.2) are selected as $m^i = 1kg$, $g = 9.81m/s^2$, $\boldsymbol{I_B^i} = \boldsymbol{I_{3\times 3}}$, where $\boldsymbol{I_{3\times 3}}$ is a 3x3 identity

matrix. The backstepping gain $\boldsymbol{K^i} = 5\boldsymbol{I_{3\times 3}}$. In addition, the bounds (3.13) are $\lambda^p = 5$,$\lambda^e = 4$, and value functional parameters (3.11) are $\boldsymbol{Q} = 3\boldsymbol{I_{3\times 3}}$, $\boldsymbol{R^p} = 0.1\boldsymbol{I_{3\times 3}}$, $\boldsymbol{R^e} = 0.125\boldsymbol{I_{3\times 3}}$. The position data of each player is collected through each iteration over the period $T = 0.01s$. Lastly $r^e + r^p$ (3.35) and shown in Fig. 26 is selected as $0.25m$.

Notice that Fig. 27 shows the trajectories of the players (Figs. 17a and 17b), and corresponding $L_2$ norm of the position offset (Figs. 17c and 17d). In addition, regarding the optimal velocity policies for the pursuer and evader, i.e. when $\pi^p(\boldsymbol{\delta}) \triangleq$ (3.17), $\pi^e(\boldsymbol{\delta}) \triangleq$ (3.18), Fig. 28 illustrate optimal velocities (3.17),(3.18), control forces (3.47), $L_2$ norm of velocity error (3.5), and Euler angles (3.48),(3.49).

Fig. 30 shows the simulation results of the PE game when the velocity bounds are $\lambda^p = 10$, $\lambda^e = 9$, and other simulation parameters remain the same as in the PE game illustrated in Figs. 27 and 28.

New Solution for H-infinity Static Output-Feedback Control Using Integral Reinforcement Learning.

Yusuf Kartal, Wenqian Xue, Atilla Dogan, Frank Lewis, 2021. *Under Review in IEEE Transactions on Cybernetics.*

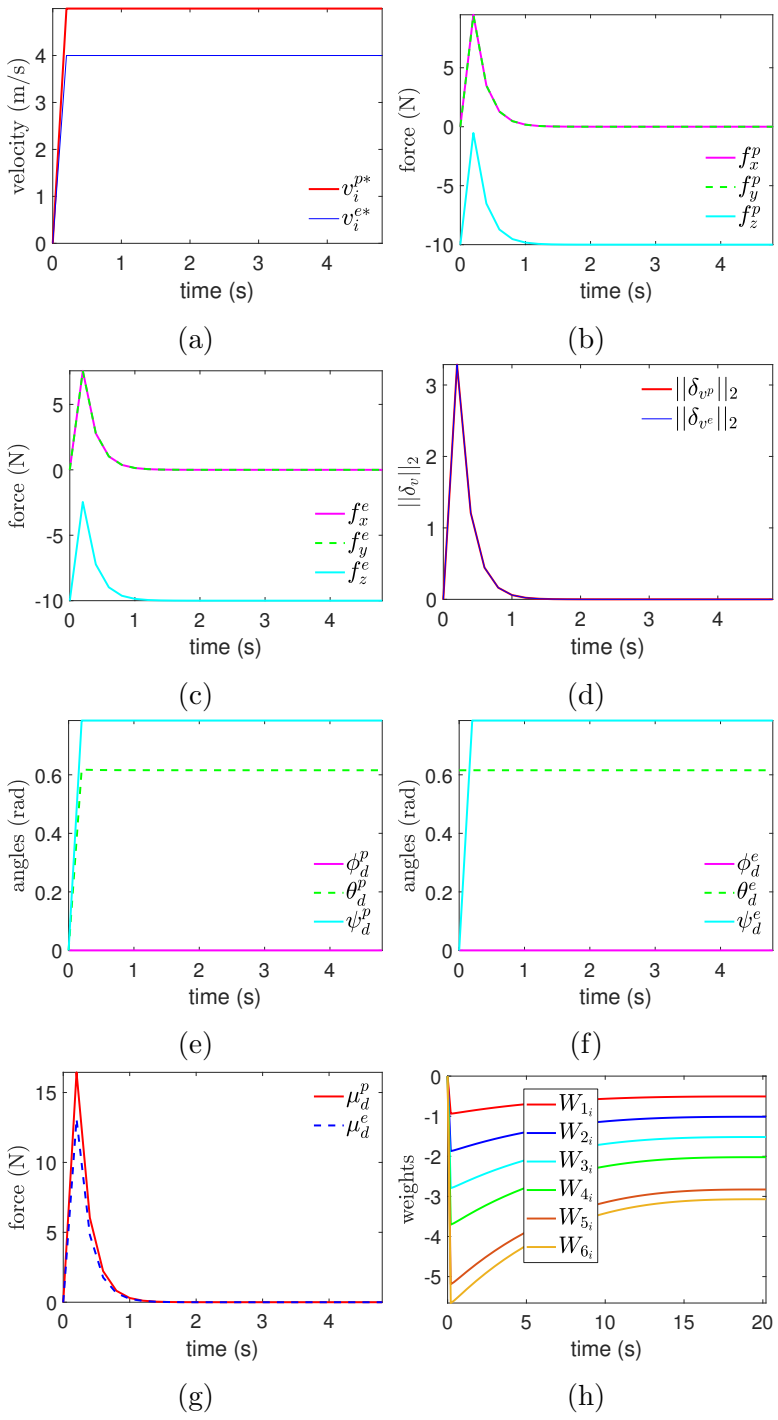


Figure 18: PE game when $\lambda^p = 5$, $\lambda^e = 4$: (a) game-optimal velocity of the pursuer and evader $\forall i \in \{x, y, z\}$, (b) control force of the pursuer (3.47), (c) control force of the evader (3.47), (d) $L_2$ norm of (3.5) for each player, (e) Euler angles of the pursuer by (3.47), (f) Euler angles of the evader by (3.47), (g) body evaluated control force of the players (3.50), (h) weights $\forall i \in \{x, y, z\}$ convergence for the critic NN (3.45).

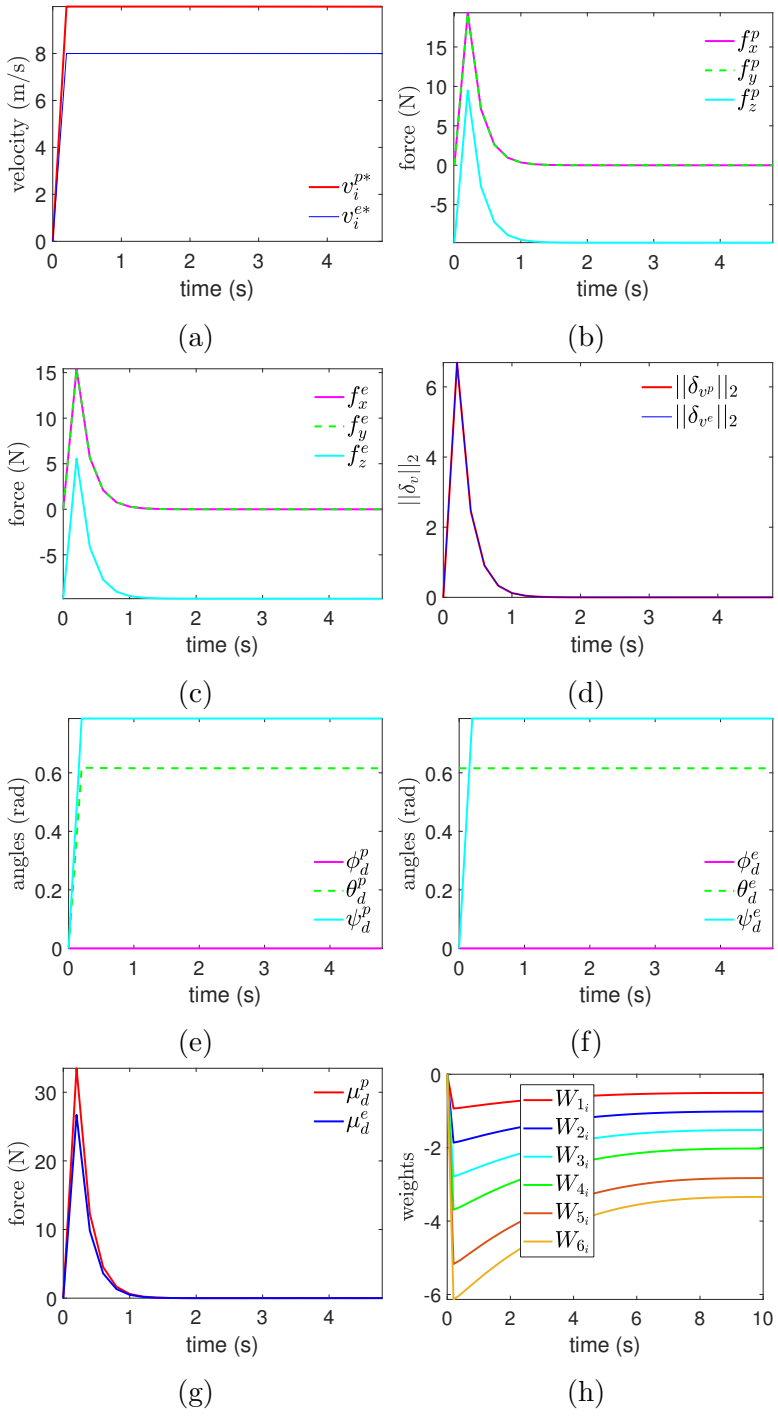


Figure 19: PE game when $\lambda^p = 10$, $\lambda^e = 8$: (a) game-optimal velocity of the pursuer and evader $\forall i \in \{x, y, z\}$, (b) control force of the pursuer (3.47), (c) control force of the evader (3.47), (d) $L_2$ norm of (3.5) for each player, (e) Euler angles of the pursuer by (3.47), (f) Euler angles of the evader by (3.47), (g) body evaluated control force of the players (3.50), (h) weights $\forall i \in \{x, y, z\}$ convergence for the critic NN (3.45).

# CHAPTER 4

## New Solution for H-infinity Static Output-Feedback Control Using Integral Reinforcement Learning

One of the primary objectives in control system design is often to seek a stabilizing controller to regulate the output of a system that experiences disturbances. However, stability is not the only requirement in control system design. An $L_2$ gain bound of the system, optimality of the control method and detectability of unknown system parameters are other common design specifications. Existing solutions of $H_\infty$ static output-feedback (OPFB) yield stability and bounded $L_2$ gain, but employee non-Nash equilibrium solutions. The new formulation of $H_\infty$ static OPFB control method developed in this paper, is a key to meet these requirements since it guarantees $L_2$ gain bound of the system by a prescribed attenuation level, asymptotic stability of equilibrium point, and also Nash equilibrium solutions. Then, the integral reinforcement learning technique based on $H_\infty$ static OPFB Lyapunov iterations is presented to deal with unknown system parameters.

$H_\infty$ control methods has been widely studied in the literature, [10], [11], [12], [13], [14], [15] due to their applicability in variety of engineering areas. Some of these methods guarantee stability and bounded $L_2$ gain, but an extra condition is required to yield Nash equilibrium. [10] uses this method to design a gain-scheduled normal acceleration control loop for an air-launched unmanned aerial vehicle. Authors of [11] apply this control method on an industrial-type mass spring damper system. The efficacy of control law and the disturbance accommodation properties are shown on a rotor-craft design example in [16]. Moreover, [80] develop an autopilot controller

for an F-16 aircraft by using the $H_\infty$ static OPFB control method on a linear discrete-time system.

During the last few years, reinforcement learning (RL) algorithms [81], [82] has been used extensively to replace model parameters with a collected system's data [83], [84], [18], [85], [86], [87]. Particularly, offline iterative RL algorithms were studied in [88], [89], and [90]. The work by [88] consider the two-player policy iterations to solve for the feedback strategies of a continuous-time zero-sum game [91] in a sub-optimal manner that requires complete knowledge of system parameters. [84] presented an online RL algorithm to solve the linear quadratic tracking (LQT) problem for partially-unknown continuous-time systems. In [92], authors prove the convergence of IRL algorithm to a sub-optimal OPFB solution without considering the disturbance term when the drift dynamics are unknown. The optimal average cost learning framework is introduced to solve output regulation problem for linear systems with unknown dynamics is studied in [93].

To design an efficient RL algorithm and achieve data-driven optimal control, the RL-based controller designs with neural networks (NNs) in an actor-critic structure [94], critic-only form [79] are proposed. In the off-policy RL algorithm, the system data, which are used to learn the solution of corresponding Hamiltonian, can be generated with arbitrary policies rather than the evaluating policy. This approach is suitably implemented using NNs in [17] and [18].

Standard existing solutions to the static OPFB regulator problem [10], [92], [95], [96], [97] require some additive gain matrix to prove the stability of equilibrium. Unfortunately, this results in the non-Nash equilibrium solutions. In this paper, we propose a novel augmented Hamiltonian, and develop a new iterative algorithms based on stationarity conditions of the augmented Hamiltonian to obtain Nash equilibrium solutions. The salient contributions of this paper summed up into four categories as:

- This paper presents a new solution of the $H_\infty$ static OPFB control problem. This solution guarantees not only stability and bounded $L_2$ gain but also Nash equilibrium solutions considering corresponding min-max game.
- Our new solution for static OPFB may have a solution when there is no static OPFB solution using existing techniques.
- Two off-line iterative solution algorithms are given. The first algorithm is based on the Kleinman's algorithm and updates the disturbance gain term. A second algorithm is developed to get the Kleinman's algorithm in the IRL applicable form that only updates the control gain.
- Two off-policy Integral Reinforcement Learning (IRL) algorithms are developed based on the stationarity conditions of an augmented Hamiltonian. This enables the designer to learn the Nash equilibrium solution online without requiring any knowledge of system dynamics' state, control, and disturbance matrices.

The rest of paper is organized as follows. In Section 4.1, preliminaries on control design requirements are introduced, and the formulation of $H_\infty$ static OPFB control presented. A new solution of optimal $H_\infty$ control problem and corresponding offline iterative solution algorithms are given in Section 4.2. In Section 4.3, an online off-policy IRL algorithm is developed based on stationarity conditions obtained in Section 4.2. Finally, we have shown the effectiveness of proposed algorithms by applying them to the linearized lateral dynamics of the F-16 aircraft at a particular flight condition in Section 5.

**Notations**. We use the following notations throughout this paper $\boldsymbol{I_n} \in \mathbb{R}^{n\times n}$ is the identity matrix. The condition $A > 0$ ($\geq 0$) denotes the positive (semi) definiteness of a matrix. The operator $tr()$ denotes trace of a matrix. $\boldsymbol{C^+ = C^T(CC^T)^{-1}}$ is the right-inverse of the full row-rank matrix $\boldsymbol{C}$ and the Kronecker product operator is denoted by $\otimes$. The determinant of a square matrix is denoted by $|.|$. $vec(A)$ stands

for the $mn$-vector formed by stacking the columns of $A \in \mathbb{R}^{n\times m}$ on top of one another, i.e., $vec(A) = [a_1^T \ ... \ a_m^T]^T$ where $a_i \in \mathbb{R}^n$ are the columns of A. Lastly, $diag(\zeta_i)$ represents a diagonal matrix with $\zeta_i \ \forall i \in 1, ..., N$ on its diagonal.

## 4.1 Preliminaries and problem formulation

In this section, preliminaries on Linear Time Invariant (LTI) system, and the corresponding controller design requirements are first introduced. Then, the problem description is presented.

### 4.1.1 System description and definitions

This section introduces system dynamics and performance specifications that are of interest. Consider the state-space representation of the continuous-time LTI system as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{D}\boldsymbol{d} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{4.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, $\boldsymbol{D} \in \mathbb{R}^{n\times p}$ are system-state, input, disturbance matrices, and $\boldsymbol{C} \in \mathbb{R}^{q\times n}$ is assumed to be a full row-rank output matrix to avoid redundant measurements. The corresponding vectors $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, $\boldsymbol{d}(t)$, and $\boldsymbol{y}(t)$ stand for the state, input, disturbance and output respectively.

**Assumption 3.** *The pair $(\boldsymbol{A}, \boldsymbol{B})$ is stabilizable and the pair $(\boldsymbol{A}, \boldsymbol{C})$ is detectable.*

**Assumption 4.** *The system (4.1) is OPFB stabilizable in the sense that the row-space of output matrix $\boldsymbol{C}$ contains the sub-space that is spanned by the right eigenvectors correspond to the unstable modes of $A$.*

**Assumption 5.** *The non-zero columns of the output matrix $\boldsymbol{C}$ are linearly independent.*

**Remark 7.** *The Assumption 7 can be interpreted such that all unstable modes are measured by the output matrix* $\boldsymbol{C}$ *that represents the sensors installed in the system (4.1). The Assumption 8 enables us to recover a state element* $x_i$ *precisely from the output vector* $\boldsymbol{y}$ *once it is left multiplied with* $C^+$.

Now, define the fictitious performance output $\boldsymbol{z}(t)$ that satisfies

$$\|\boldsymbol{z}\|_2^2 = \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{u} \tag{4.2}$$

where $\boldsymbol{Q} \geq 0$ and $\boldsymbol{R} > 0$ are symmetric design matrices with appropriate dimensions. We assume that $\boldsymbol{Q}$ is selected such that the pair $(\boldsymbol{A}, \sqrt{\boldsymbol{Q}})$ is *observable*, which is a standard assumption [60]. Using the property $\|\boldsymbol{d}\|_2^2 = \boldsymbol{d}^{\boldsymbol{T}}\boldsymbol{d}$, a realization of the following inequality $\forall \boldsymbol{d} \in [0, \infty)$ implies that the system $L_2$-gain is bounded by a prescribed disturbance attenuation level denoted by $\gamma$

$$\int_0^\infty \|\boldsymbol{z}\|_2^2 dt \leq \gamma^2 \int_0^\infty \|\boldsymbol{d}\|_2^2 dt + \beta \tag{4.3}$$

for any non-zero energy-bounded disturbance input d [98] where $\beta$ is a non-negative constant. The condition (5.12) is also called as non-nexpansivity constraint in [74]. Call $\gamma^*$ the minimum gain for which this occurs. In [99] and [100], an algorithm to find $\gamma^*$ is given, and a formulation for explicit $\gamma^*$ that depends on Riccati equation solution is derived for LTI systems under some assumptions. In this paper, we assume that the attenuation level is prescribed and satisfies $\gamma > \gamma^*$.

A static OPFB control to regulate the system (4.1) is

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{y} = -\boldsymbol{K}\boldsymbol{C}\boldsymbol{x} \tag{4.4}$$

where $\boldsymbol{K} \in \mathbb{R}^{m \times q}$ is the gain matrix. Note that main objective of $H_\infty$ control using OPFB is to find the stabilizing $\boldsymbol{K}$ in an optimal manner while satisfying the condition (5.12), which can be achieved by solving corresponding Hamilton-Jacobi-Isaacs (HJI) equation.

### 4.1.2 Problem formulation and existing solution of static OPFB Problem

In this section, we relate zero-sum differential game theory to the static OPFB regulation problem in a global optimal manner by revealing various definitions. To satisfy $L_2$-gain bound (5.12) with the stabilizing gain in (5.13), an objective functional defined as

$$J(\boldsymbol{u}, \boldsymbol{d}) = \int_0^\infty (\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{R} \boldsymbol{u} - \gamma^2 \boldsymbol{d}^{\boldsymbol{T}} \boldsymbol{d}) d\tau. \tag{4.5}$$

Now $H_\infty$ control problem can be represented as a two-player zero-sum differential game by treating $\boldsymbol{u}(t)$ as a minimizing player, whereas $\boldsymbol{d}(t)$ maximizing player of (4.5). Then, the game can be formulated as

$$V(\boldsymbol{x}(\boldsymbol{0})) = J(\boldsymbol{u}^*, \boldsymbol{d}^*) = \min_{\boldsymbol{u}} \max_{\boldsymbol{d}} J(\boldsymbol{u}, \boldsymbol{d}) \tag{4.6}$$

where $V(\boldsymbol{x})$ denotes the value functional corresponding to (4.5) such that

$$V(\boldsymbol{x}) = \int_t^\infty (\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{R} \boldsymbol{u} - \gamma^2 \boldsymbol{d}^{\boldsymbol{T}} \boldsymbol{d}) d\tau. \tag{4.7}$$

and the pair $(\boldsymbol{u}^*, \boldsymbol{d}^*)$ denotes the game theoretic saddle point. The game of this kind admits a unique solution pair $(\boldsymbol{u}^*, \boldsymbol{d}^*)$, if the following Nash condition holds

$$\min_{\boldsymbol{u}} \max_{\boldsymbol{d}} J(\boldsymbol{u}, \boldsymbol{d}) = \max_{\boldsymbol{d}} \min_{\boldsymbol{u}} J(\boldsymbol{u}, \boldsymbol{d}). \tag{4.8}$$

The next theorem recalls the necessary and sufficient conditions for the sub-optimal $H_\infty$ OPFB control method [10].

**Theorem 5.** *The system (4.1) is OPFB stable using the control* $\boldsymbol{u_o^e = -K_o^e y}$ *with* $L_2$ *gain bounded by* $\gamma > \gamma^*$ *if and only if*

1. $(\boldsymbol{A}, \boldsymbol{B})$ *is stabilizable and* $(\boldsymbol{A}, \boldsymbol{C})$ *is detectable.*
2. *There exists* $\boldsymbol{K_o^e}$ *and* $\boldsymbol{L}$ *such that*

$$\boldsymbol{K_o^e = R^{-1}(B^T P + L)C^+} \tag{4.9}$$

*where* $\boldsymbol{P = P^T > 0}$ *is the solution of Riccati equation*

$$\boldsymbol{PA + A^T P - PBR^{-1}B^T P + Q + \gamma^{-2} PDD^T P + L^T R^{-1} L = 0}. \tag{4.10}$$

**Proof.** See [10] and [98] for the same proof. □

**Remark 8.** *On the one hand, introducing the additive gain matrix* $\boldsymbol{L}$ *provides the extra design freedom. Note that if* $\boldsymbol{L = 0}$ *there may not be a stabilizing solution to (4.9),(5.22) (See Theorem 1 in [10]). On the other hand, this results in a sub-optimal solution for the gain* $\boldsymbol{K^s}$ *(4.9). The Nash equilibrium gain occurs if the Theorem 5 holds with* $\boldsymbol{L = 0}$.

The following lemma recalls the Nash equilibrium solution for the standard $H_\infty$ control problem.

**Lemma 1.** *The pair* $(u^*, d^*)$ *constitutes the Nash equilibrium of the game (4.8) such that*

$$\boldsymbol{u^* = -K^* x = -R^{-1} B^T P x}, \tag{4.11}$$

$$\boldsymbol{d^* = \gamma^{-2} D^T P x}. \tag{4.12}$$

**Proof.** Begin with deriving the Hamiltonian to solve the game theoretic saddle point or Nash equilibrium strategy of the game (4.8) as

$$H(\boldsymbol{V_x}, \boldsymbol{u}, \boldsymbol{d}) = \boldsymbol{x^T Q x} + \boldsymbol{u^T R u} - \gamma^2 \boldsymbol{d^T d} + \boldsymbol{V_x}(\boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{Dd}) \tag{4.13}$$

where $\boldsymbol{V_x} = \partial V / \partial \boldsymbol{x}$ is the co-state vector. Using the quadratic form $V(\boldsymbol{x}) = \boldsymbol{x^T P x}$, and applying the stationarity conditions $\partial H(\boldsymbol{V_x}, \boldsymbol{u}, \boldsymbol{d})/\partial \boldsymbol{u} = \boldsymbol{0}$ and $\partial H(\boldsymbol{V_x}, \boldsymbol{u}, \boldsymbol{d})/\partial \boldsymbol{d} = \boldsymbol{0}$ yields the optimal control and disturbance respectively as (5.20) and (5.21).

Notice that the sign of Hessians, $\partial^2 H(\boldsymbol{V_x}, \boldsymbol{u}, \boldsymbol{d})/\partial \boldsymbol{u^2} > 0$ and $\partial^2 H(\boldsymbol{V_x}, \boldsymbol{u}, \boldsymbol{d})/\partial \boldsymbol{d^2} < 0$, along with unboundedness of the limits $\lim_{d\to\infty} J(\boldsymbol{u^*}, \boldsymbol{d})$, $\lim_{u\to\infty} J(\boldsymbol{u}, \boldsymbol{d^*})$ indeed show that (5.20) and (5.21) are the global optimal minimizing and maximizing extrema respectively [61]. This indeed verifies that the pair $(\boldsymbol{u^*}, \boldsymbol{d^*})$ denotes the Nash equilibrium point, which completes the proof. □

**Remark 9.** *The HJI equation can be obtained as* $H(\boldsymbol{V_x}, \boldsymbol{u^*}, \boldsymbol{d^*}) = 0$ *with the boundary condition V(0)=0, which also verifies the sub-optimality of gain expression (4.9). Additionally, considering Nash equilibrium policies (5.20) and (5.21), the Game Algebraic Riccati Equation (GARE) is obtained as*

$$\boldsymbol{PA} + \boldsymbol{A^T P} - \boldsymbol{PBR^{-1}B^T P} + \boldsymbol{Q} + \gamma^{-2}\boldsymbol{PDD^T P} = \boldsymbol{0}. \tag{4.14}$$

If there is no $\boldsymbol{K^s}$ to satisfy (4.9) and (5.22), the OPFB $H_\infty$ control problem may not have even a sub-optimal solution. In the next section, we rigorously analyze this and reveal some novel results.

## 4.2 New solution of OPFB $H_\infty$ game

Notice that Theorem 5 provides necessary and sufficient conditions for static OPFB in a sub-optimal manner. Thence, this does not yield a Nash equilibrium solution unless $L = 0$. Moreover, there may not even exist a static OPFB solution to the equations in Theorem 5. In this main section, two methods are proposed to solve $H_\infty$ OPFB problem. The first method parameterizes the state feedback gains by using the Nash strategies, and applies them to the OPFB design. The second method derives the new optimal $H_\infty$ OPFB regulator formulation by introducing an augmented Hamiltonian. This method is introduced by [60], but is highly overlooked in the literature. However, it appears to be instrumental in $H_\infty$ regulator design.

### 4.2.1 Necessary and Sufficient Conditions for the Stabilizing Nash Gain

Herein, we first parameterize static state feedback gains, and then explain how to apply them to the $H_\infty$ OPFB design.

Note that the Assumptions 7 and 8 are missing in the papers [10], [11], [16], and [98]. However, they are indeed required when applying state feedback gains to the OPFB design.

**Theorem 6.** *Given the necessary conditions in the Assumption 3 and the sufficient condition* $\boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}$. *The system (4.1) is asymptotically stable using the control* $\boldsymbol{u^*} = -\boldsymbol{R^{-1}B^TPx}$ *(5.20) with* $d = 0$, *and* $L_2$ *gain bounded by* $\gamma$ $\forall \|\boldsymbol{d}\|_2 \in (0, \infty)$.

**Proof.** To prove $L_2$ gain bound condition (5.12), first re-write the Hamiltonian (4.13) by completing the squares as

$$\begin{aligned} H(\boldsymbol{V_x}, \boldsymbol{u}, \boldsymbol{d}) =& H(\boldsymbol{V_x}, \boldsymbol{u^*}, \boldsymbol{d^*}) + (u - u^*)^T R(u - u^*) \\ & - \gamma^{-2}(d - d^*)^T(d - d^*). \end{aligned} \tag{4.15}$$

Then, the objective functional can be re-expressed as

$$J(\boldsymbol{u},\boldsymbol{d}) = \int_0^\infty \Big( H(\boldsymbol{V_x},\boldsymbol{u^*},\boldsymbol{d^*}) + (u-u^*)^T R(u-u^*) \\ - \gamma^{-2}(d-d^*)^T(d-d^*)\Big)dt + V(x(0)). \tag{4.16}$$

Realize that the non-expansivity constraint (5.12) implies that $J(u,d) \leq \beta$. Select $\beta = V(x(0))$, $u = u^*$, and note that HJI equation $H(\boldsymbol{V_x},\boldsymbol{u^*},\boldsymbol{d^*}) = 0$ holds with the boundary condition V(0)=0. Then, (5.52) reduces to

$$J(\boldsymbol{u^*},\boldsymbol{d}) = -\int_0^\infty \gamma^{-2}(d-d^*)^T(d-d^*)dt + V(x(0)), \\ \leq V(x(0)), \quad \forall \left\|\boldsymbol{d}\right\|_2 \in (0,\infty). \tag{4.17}$$

This proves that the $L_2$ gain bound condition (5.12) holds with $\beta = V\boldsymbol{x}(0))$, $\boldsymbol{u} = \boldsymbol{u^*}$. Additionally, the value of game (4.8) with $u = u^*$ and $d = d^*$ is $V(x(0))$ by (5.52).

To verify asymptotic stability, consider Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x^T P x}$ where $\boldsymbol{P}$ is solution of the GARE (4.14). Note that for $\boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}$, the GARE (4.14) has a unique stabilizing solution $P = P^T$, which is indeed positive definite. This is illustrated in the following realization

$$\begin{aligned} \dot{V} &= \boldsymbol{\dot{x}^T P x} + \boldsymbol{x^T P \dot{x}} \\ &= \boldsymbol{x^T}\big( - \boldsymbol{PBR^{-1}B^T P} - \boldsymbol{Q} + \gamma^{-2}\boldsymbol{PDD^T P}\big)\boldsymbol{x} \\ &\leq -\boldsymbol{x^T Q x} \quad \Leftarrow \ \boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}. \end{aligned} \tag{4.18}$$

Then, the observability of $(\boldsymbol{A},\boldsymbol{Q})$ verifies that the undisturbed system (4.1), i.e., $\boldsymbol{d} = \boldsymbol{0}$, is asymptotically stable by LaSalle's invariance principle [74]. This completes the proof. □

**Corrolary 1.** *To verify asymptotic stability of the disturbed system, benefit from gain margin* $[c_{lower}, \infty)$ *with* $c_{lower} < \frac{1}{2}$ *property of the* $\mathcal{H}_\infty$ *control [74]. Note that the* $\mathcal{H}_\infty$ *has gain margin less than* $\frac{1}{2}$ *by Chapter 10 in [74] but the lower bound* $c_{lower}$ *is not precisely defined. Then, if the sufficient condition in Theorem 6 is strengthened as* $BR^{-1}B^T \geq 2\gamma^{-2}DD^T$*, the disturbed system (4.1) with* $d = d^*$*, becomes asymptotically stable. Thence, the closed-loop matrix* $\boldsymbol{A - BR^{-1}B^TP + \gamma^{-2}DD^TP}$ *becomes Hurwitz.*

Till now, we actually parameterize all the stabilizing static state-feedback gains since we used the Nash strategies (5.20) and (5.21) in the proof of Theorem 6. The following Remark explains how to apply them to the $H_\infty$ OPFB design.

**Remark 10.** *Instead of Nash strategies (5.20) and (5.21), assume that the control and disturbance are selected as*

$$\boldsymbol{u_o^*} = -(R^{-1}B^TPC^+)C(C^+y) = -\underbrace{(R^{-1}B^TPC^+)}_{K_o^*}y \tag{4.19}$$

$$\boldsymbol{d_o^*} = (\gamma^{-2}\boldsymbol{D^TP}C^+)C(C^+y). \tag{4.20}$$

*Note that if* $\boldsymbol{C}$ *is an invertible matrix, then all states would be regulated optimally by Theorem 6. On the other hand, if it is not square but full row-rank, then only the states spanned by row space of the output matrix* $\boldsymbol{C}$ *would be regulated optimally given the Assumption 8, and the fact that* $\boldsymbol{C^+C}$ *projects* $\mathbb{R}^n$ *onto the row space of* $\boldsymbol{C}$*. Additionally, the other states would converge to the origin given the Assumption 7. Realize that the Assumption 3 implies that there could be an unstable mode that is observable but does not belong to the row space of* $\boldsymbol{C}$*. Therefore, the Assumptions 7 and 8 are indeed required to apply static state feedback gains to the OPFB design. Lastly, the system (4.1) is stable against the worst-case disturbance (4.20), which*

*affects only the states that belong to the row space of* $\boldsymbol{C}$ *if* $\boldsymbol{BR^{-1}B^T} \geq 2\gamma^{-2}\boldsymbol{DD^T}$ *by Corollary 1.*

### 4.2.2 A Direct Method to Obtain OPFB Optimal Gain Solutions

In this section, we propose a new methodology to obtain stabilizing Nash solutions for the $H_\infty$ OPFB control. This method is direct in the sense that it reaches the same gain solutions as the Section 4.2.1 but do not require two step

Consider the optimal value obtained in the Theorem 6 that corresponds to the zero-effort for each player such that

$$J_0 = \boldsymbol{x^T}(0)\boldsymbol{P}\boldsymbol{x}(0) = tr(\boldsymbol{P}\boldsymbol{X_0}) \tag{4.21}$$

where $tr()$ stands for the trace of a matrix, and $\boldsymbol{X_0} = \boldsymbol{x}(0)\boldsymbol{x^T}(0)$.

The following Lemma is an essential step before introducing the augmented Hamiltonian.

**Lemma 2.** *Given the Assumption 3, let* $\boldsymbol{K^a}$ *be a gain that stabilizes the system (4.1), and the corresponding OPFB control is* $\boldsymbol{u_o^a} = -\boldsymbol{K_o^a}\boldsymbol{y}$*. Additionally, let the disturbance takes the form* $\boldsymbol{d_o^a} = \boldsymbol{N_o}\boldsymbol{y}$ *to guarantee it does not affect unobservable modes of the system (4.1). Then the corresponding Lyapunov equation can be derived as*

$$\boldsymbol{P}\boldsymbol{A_c} + \boldsymbol{A_c^T}\boldsymbol{P} + \boldsymbol{C^T}\boldsymbol{K_o^{aT}}\boldsymbol{R}\boldsymbol{K_o^a}\boldsymbol{C} - \gamma^2\boldsymbol{C^T}\boldsymbol{N_o^T}\boldsymbol{N_o}\boldsymbol{C} + \boldsymbol{Q} \equiv \boldsymbol{T} \tag{4.22}$$

*where* $\boldsymbol{T} = \boldsymbol{T^T} = \boldsymbol{0}$ *and* $\boldsymbol{A_c} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K_o^a}\boldsymbol{C} + \boldsymbol{D}\boldsymbol{N_o}\boldsymbol{C}$*.*

**Proof.** Consider the quadratic form of the value functional (4.5) as $V(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{P}\boldsymbol{x}$, and then substitute expressions $\boldsymbol{u} = -\boldsymbol{K}^{\boldsymbol{a}}\boldsymbol{y}$ and $\boldsymbol{d}_{\boldsymbol{o}}^{\boldsymbol{a}} = \boldsymbol{N}_{\boldsymbol{o}}\boldsymbol{y}$ into (4.5) to obtain

$$\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{P}\boldsymbol{x} = \int_t^{\infty} \boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}\boldsymbol{T}}\boldsymbol{R}\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}}\boldsymbol{C} + \boldsymbol{Q} - \gamma^2\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{o}}^{\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{o}}\boldsymbol{C})\boldsymbol{x}d\tau. \tag{4.23}$$

Now, take the derivative of left-side (4.23) and substitute (5.21), $\boldsymbol{u} = -\boldsymbol{K}^{\boldsymbol{a}}\boldsymbol{C}\boldsymbol{x}$ expressions. Lastly, take the derivative of integral in right-side (4.23) using Leibniz's rule, which yields

$$\begin{aligned}\boldsymbol{x}^{(}\boldsymbol{A}_{\boldsymbol{c}}^{\boldsymbol{T}}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_{\boldsymbol{c}})\boldsymbol{x} = \\ \boldsymbol{x}^{\boldsymbol{T}}(-\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}\boldsymbol{T}}\boldsymbol{R}\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}}\boldsymbol{C} - \boldsymbol{Q} + \gamma^2\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{o}}^{\boldsymbol{T}}\boldsymbol{N}_{\boldsymbol{o}}\boldsymbol{C})\boldsymbol{x}.\end{aligned} \tag{4.24}$$

Realize that the zero equivalent is nothing but the Lyapunov equation given in (4.22). This completes the proof. □

It is now clear that performing a min-max operation on (4.5), subject to dynamical constraint (4.1) is equivalent to the algebraic problem of finding the pair $(\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}}, \boldsymbol{N}_{\boldsymbol{o}})$ that performs min-max of (4.21) subject to the constraint (4.22). Define following augmented Hamiltonian to solve this modified problem as

$$H^a(\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}}, \boldsymbol{N}_{\boldsymbol{o}}, \boldsymbol{S}) = tr(\boldsymbol{P}\boldsymbol{X}_{\boldsymbol{0}}) + tr(\boldsymbol{T}\boldsymbol{S}) \tag{4.25}$$

where $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ is a symmetric matrix of Lagrange multipliers [60] that needs to be determined, and $\boldsymbol{T}$ is given in (4.22).

The next main theorem is a key to find $\boldsymbol{S}$ along with a Nash equilibrium control matrix $\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}}$ with respect to (4.25).

**Theorem 7.** *Given the Assumptions 3, 7, and 8, the system (4.1) is asymptotically stable using the control* $\boldsymbol{u_o^a = -K_o^a y}$ *with* $d_o^a = 0$ *and* $L_2$ *gain bounded by* $\gamma$ *if*

$$\frac{\partial H^a}{\partial \boldsymbol{S}} \equiv \boldsymbol{P A_c + A_c^T P + C^T K_o^{aT} R K_o^a C - \gamma^2 C^T N_o^T N_o C + Q = 0}, \tag{4.26}$$

$$\frac{\partial H^a}{\partial \boldsymbol{P}} \equiv \boldsymbol{S A_c^T + A_c S + X_0 = 0}, \tag{4.27}$$

$$\frac{\partial H^a}{\partial \boldsymbol{K_o^a}} \equiv \boldsymbol{2 R K_o^a C S C^T - 2 B^T P S C^T = 0}, \tag{4.28}$$

$$\frac{\partial H^a}{\partial \boldsymbol{N_o}} \equiv \boldsymbol{-2\gamma^2 N_o^a C S C^T + 2 D^T P S C^T = 0}. \tag{4.29}$$

*Furthermore, the following gain expressions solves the* $H_\infty$ *static OPFB problem in an optimal manner*

$$\boldsymbol{K_o^a = K_o^* = R^{-1} B^T P C^+} \tag{4.30}$$

$$\boldsymbol{N_o = \gamma^{-2} D^T P C^+} \tag{4.31}$$

**Proof.** Consider (4.25), which is a constant along the system trajectories since the system (4.1) is LTI and $\boldsymbol{z}(t)$ (4.2) is not explicit function of time. This implies that we can apply the constraint test and check the stationarity conditions on the augmented Hamiltonian (4.25) that yields the second-order Lyapunov equation (4.26) and standard Lyapunov equation (4.27) respectively.

Define the variable $X = xx^T$ that includes the system state information, and take the derivative using (4.1) to obtain

$$\boldsymbol{X A_c^T + A_c X = \dot{X}}. \tag{4.32}$$

Now, assume that $A_c$ is Hurwitz. Then, taking the integral of both sides (4.32) from 0 to $\infty$ yields (4.27) where $S = \int_0^\infty X dt$, thereby $\boldsymbol{K_o^a}$ should not depend on the solution

$S$. Therefore, the gain solutions (4.30) and (4.31) are immediate. Realize that the stability of an LTI system does not depend on the initial condition, i.e, local stability implies the global stability. Thence, the gain solutions (4.30) and (4.31) should not depend on $S$ that depends on the initial condition $X_0$. This also verifies our reason to select gain solutions in the given forms, which completes the proof. □

**Remark 11.** *The Theorem 11 gives the necessary conditions, i.e. the Assumption 3 and 7, and sufficient condition* $\boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}$ *to prove* $L_2$ *gain boundedness by a prescribed attenuation level* $\gamma$ *(5.12) and OPFB stability considering the worst-case disturbance (4.20). Realize that the condition* $\boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}$ *is only a sufficient condition, which implies that there may be an optimal gain solution which stabilizes (4.1) but does not satisfy* $\boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}$*. However, in that case, one may not achieve a positive definite solution* $\boldsymbol{P}$ *for the Riccati equation (5.22). Additionally, the positive definiteness of the Riccati equation solution plays a key role for the Kleinman's algorithm in Section 4.2.3, and the IRL algorithm in Section 4.3.*

**Remark 12.** *The Theorem 11 proves that the condition* $\boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}$ *is* $suf ficient$ *to obtain the Nash equilibrium solution. The gain* $\boldsymbol{K_o^e}$ *in Theorem 5 is a sub-optimal stabilizing gain solution with respect to the value functional (4.7). However, the gain solution* $\boldsymbol{K^a}$ *in Theorem 11 always gives a stabilizing Nash equilibrium gain solutions with respect to the game (4.8).*

**Remark 13.** *Note that the system (4.1) is stable in the presence of matched disturbances with the gains (4.30) and (4.31). The unmatched disturbances are only* $L_2$ *gain bounded by Theorem 6. Thence, the control (4.19) is robustly stabilizing the equilibrium origin even if the unmatched disturbances exist given Assumptions 7-8.*

The next section proposes an offline model-based algorithm to find the optimal gain solution $\boldsymbol{K_o^a}$ iteratively, that plays a key role to develop the IRL Algorithm that will be detailed later in Section 4.3. Note that given the necessary and sufficient

conditions in Theorem 11 and Remark 11, one does not need an iterative solution algorithm to find optimal stabilizing gain $\boldsymbol{K_o^a}$ (4.30). However, to develop a model-free algorithm, an iterative solution algorithm is required.

### 4.2.3 Offline Iterative Solution Algorithms for $H_\infty$ OPFB

This section presents two iterative solution algorithms to obtain minimizing gain (4.30) by using the conditions given in (4.26)-(4.28). In the first algorithm, we employ Kleinman's algorithm (4.26) to obtain Nash equilibrium gain solutions (4.30) and (4.31), whereas in the second algorithm, we use a corresponding Lyapunov equation to not deal with the disturbance gain term $N_o$.

The next algorithm performs a sequence of four-step iterations based on the Kleinman's Algorithm [101] to find the optimal control gains (4.30) and (4.31).

**Algorithm 1.** *(Offline iterative solution with Lyapunov equations. Kleinman's Algorithm.)*

1. *Initialize: Set* $k = 1$, $P_0 = 0$, $N_0 = 0$ *and given the Assumption 3 and the sufficient condition* $\boldsymbol{BR^{-1}B^T} \geq 2\gamma^{-2}\boldsymbol{DD^T}$, *select a gain* $\boldsymbol{F_0}$ *such that* $\boldsymbol{A}-\boldsymbol{BF_0}$ *is asymptotically stable.*
2. $k^{th}$ *iteration: Solve for* $\boldsymbol{P_k}$

$$\boldsymbol{0 = P_k A_k + A_k^T P_k + F_{k-1}^T R F_{k-1} + Q} \tag{4.33}$$

*where* $\boldsymbol{A_k = A - BF_k} + \frac{1}{2}DN_k$. *Finally, update the gains*

$$\boldsymbol{F_k = R^{-1}B^T P_k}, \tag{4.34}$$

$$\boldsymbol{N_k = \gamma^{-2} D^T P_k}. \tag{4.35}$$

*Set the cost* $J_k = tr(\boldsymbol{P_k X_0})$.

3. *Check: If* $\boldsymbol{F_{k-1}}$ *and* $\boldsymbol{F_k}$ *are close enough to each other, go to step* (4) *else go to step* (2).
4. *Terminate: Given the Assumptions 7 and 8, set the OPFB gains* $\boldsymbol{K_o^a} = \boldsymbol{F_k C^+}$, $\boldsymbol{N_o} = \boldsymbol{N_k C^+}$ *and the cost* $\boldsymbol{J} = \boldsymbol{J_k}$. □

Note that the closed-loop stability, and $L_2$ gain boundedness implies that (4.36) has a unique stabilizing optimal solution $\boldsymbol{P} > \boldsymbol{0}$ by Theorem 6. A comprehensive study for the solution of generalized Riccati equations can be found in [102]. The Algorithm 2 is based on the iterative solution algorithm presented in [103] whose convergence is proved by establishing the connection between Newton's method in [104]. Additionally, by considering the condition $\boldsymbol{BR^{-1}B^T} \geq \gamma^{-2}\boldsymbol{DD^T}$, the monotonic convergence, i.e, $P_k < P_{k-1}$, is straight-forward from the Theorem 13.5.8 in [104]. A related algorithm is also examined in Chapter 8 of the book [60].

Now, to develop an algorithm that accounts only the optimal gain $\boldsymbol{K_o^a}$, we manipulate the steps of the Algorithm 1. The resultant Algorithm 2 finds the the optimal control gain $\boldsymbol{K_o^a}$ (4.30) iteratively.

**Algorithm 2.** *(Offline iterative solution with Lyapunov equations. Modified Kleinman's Algorithm.)*

1. *Initialize: Set* $k = 1$, $\boldsymbol{P_0} = \boldsymbol{0}$, *and given the Assumption 3 and the sufficient condition* $\boldsymbol{BR^{-1}B^T} \geq 2\gamma^{-2}\boldsymbol{DD^T}$, *select a gain* $\boldsymbol{F_0}$ *such that* $\boldsymbol{A - BF_0}$ *is asymptotically stable.*
2. $k^{th}$ *iteration: Solve for* $\boldsymbol{P_k}$

$$\boldsymbol{0 = P_k A_k + A_k^T P_k + F_{k-1}^T R F_{k-1} + Q + \gamma^{-2} P_{k-1} D D^T P_{k-1}} \tag{4.36}$$

*where* $\boldsymbol{A_k} = \boldsymbol{A} - \boldsymbol{BF_k}$. *Finally, update the gain*

$$\boldsymbol{F_k} = \boldsymbol{R^{-1}B^T P_k}. \tag{4.37}$$

*Set the cost* $J_k = tr(\boldsymbol{P_k X_0})$.

3. *Check: If* $\boldsymbol{F_{k-1}}$ *and* $\boldsymbol{F_k}$ *are close enough to each other, go to step* (4) *else go to step* (2).
4. *Terminate: Given the Assumptions 7 and 8, set the OPFB gain* $\boldsymbol{K_o^a} = \boldsymbol{F_k C^+}$ *and the cost* $J = J_k$. □

The next section uses the Algorithm 2 to develop model-free algorithms considering different scenarios for the availability of system data.

## 4.3 Online integral reinforcement learning solution algorithm for $H_\infty$ OPFB

In this section, we first develop an online off-policy integral reinforcement learning (IRL) algorithm [105], which is a model-free version of the Algorithm 2. This algorithm assumes that the system state data is available to deal with unknown $A$, $B$ and $D$ matrices. Then, we develop a novel IRL algorithm that solves the optimal $H_\infty$ regulator problem by learning the Nash equilibrium gain solution (4.30) without requiring knowledge of the system state data.

### 4.3.1 Online off-policy IRL algorithms

This section introduces an off-policy IRL algorithm to deal with the unknown system matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{D}$. In this case, both of Algorithm 1 and 2 lose their applicability as they are model-based. However, we still benefit from the convergence

properties of Algorithm 2 while developing off-policy IRL method. To this end, we represent system dynamics (4.1) as

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u_k} + \boldsymbol{D}\boldsymbol{d} + \boldsymbol{B}(\boldsymbol{u} - \boldsymbol{u_k}) \\ &= \boldsymbol{A_k}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{d} + \boldsymbol{B}(\boldsymbol{u} - \boldsymbol{u_k}) \end{aligned} \tag{4.38}$$

where $\boldsymbol{u_k} = -\boldsymbol{F_k}\boldsymbol{x} \in \mathbb{R}^m$ is the control policy to be updated with $\boldsymbol{F_k}$ given in Algorithm 2.

Firstly, to obtain $\boldsymbol{P_k}$ without information of the system matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{D}$, take the derivative of value functional $V(\boldsymbol{x}(t)) = \boldsymbol{x^T}(t)\boldsymbol{P}\boldsymbol{x}(t)$ by using the new representation of the system dynamics (4.38)

$$\begin{aligned}\dot{V} &= \boldsymbol{x^T}\boldsymbol{A_k^T}\boldsymbol{P_k}\boldsymbol{x} + \boldsymbol{x^T}\boldsymbol{P_k}\boldsymbol{A_k}\boldsymbol{x} \\ &\quad + 2\boldsymbol{d^T}\boldsymbol{D^T}\boldsymbol{P_k}\boldsymbol{x} + 2(\boldsymbol{u} + \boldsymbol{F_k}\boldsymbol{x})^T\boldsymbol{B^T}\boldsymbol{P_k}\boldsymbol{x}. \end{aligned} \tag{4.39}$$

To employ the approach given in Algorithm [101], we define the following two new variables

$$\boldsymbol{G_{k+1}} = \gamma^{-2}\boldsymbol{D^T}\boldsymbol{P_k}, \quad \boldsymbol{F_{k+1}} = \boldsymbol{R^{-1}}\boldsymbol{B^T}\boldsymbol{P_k}. \tag{4.40}$$

Re-write (4.36) in terms of new variable (4.40) to get the Algorithm 2 in the Kleinman's form as

$$\bar{\boldsymbol{Q}} = \boldsymbol{P_k}\boldsymbol{A_k} + \boldsymbol{A_k^T}\boldsymbol{P_k} \tag{4.41}$$

where $\bar{\boldsymbol{Q}} = -\boldsymbol{F_k^T}\boldsymbol{R}\boldsymbol{F_k} - \boldsymbol{Q} - \gamma^2\boldsymbol{G_k^T}\boldsymbol{G_k}$. Additionally, express (4.39) in terms of the new variables introduced in (4.40) and (4.41) as

$$\dot{V} = 2(\gamma^2\boldsymbol{d^T}\boldsymbol{G_{k+1}} + (\boldsymbol{u} + \boldsymbol{F_k}\boldsymbol{x})^T\boldsymbol{R}\boldsymbol{F_{k+1}})\boldsymbol{x} + \boldsymbol{x^T}\bar{\boldsymbol{Q}}\boldsymbol{x}. \tag{4.42}$$

Then, integrate both sides from $t$ to $t+T$ to obtain

$$V(t+T) - V(t) = \int_t^{t+T} 2(\boldsymbol{u} + \boldsymbol{F_k x})^T \boldsymbol{R F_{k+1} x} d\tau + \int_t^{t+T} 2\gamma^2 \boldsymbol{d^T G_{k+1} x} d\tau + \int_t^{t+T} \boldsymbol{x^T \bar{Q} x} d\tau. \tag{4.43}$$

Based on these manipulations, the online off-policy IRL algorithm can be developed.

Note that this new IRL algorithm and the Algorithm 2 are equivalent. However, on the contrary offline Algorithm 2, the IRL Algorithm 3 does not require information of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{D}$ matrices. It only requires the sufficient amount of data that belongs $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, and $\boldsymbol{d}(t)$ vectors, which can be collected online.

**Algorithm 3.** *(Off-policy IRL algorithm assuming x is given.)*

1. *Initialize: Set* $k = 1$, $\boldsymbol{G_0} = \boldsymbol{0}$ *and given Assumption 3 determine a stabilizing gain* $\boldsymbol{F_0}$.
2. $k^{th}$ *iteration: Use (4.43) to update* $\boldsymbol{P_k}$, $\boldsymbol{G_{k+1}}$ *and* $\boldsymbol{F_{k+1}}$ *simultaneously*

$$\boldsymbol{x^T}(t+T)\boldsymbol{P_k x}(t+T) - \boldsymbol{x^T}(t)\boldsymbol{P_k x}(t) - \int_t^{t+T} 2\gamma^2 \boldsymbol{d^T G_{k+1} x} d\tau - \int_t^{t+T} 2(\boldsymbol{u} + \boldsymbol{F_k x})^T \boldsymbol{R F_{k+1} x} d\tau = \int_t^{t+T} \boldsymbol{x^T \bar{Q} x} d\tau. \tag{4.44}$$

   *Set the cost* $J_k = tr(\boldsymbol{P_k X_0})$.

3. *Check: If* $\boldsymbol{F_k}$ *and* $F_{k+1}$ *are close enough to each other, go to step* (4) *else go to step* (2).
4. *Terminate: Given the Assumptions 7 and 8, set* $\boldsymbol{K_o^a} = \boldsymbol{F_{k+1} C^+}$ *and* $\boldsymbol{J} = \boldsymbol{J_k}$.

□

**Remark 14.** *Note that right-side of the equation (4.44) in the Algorithm 3 consists of known terms, and hence it can be solved for* $\boldsymbol{P_k}$, $\boldsymbol{G_{k+1}}$ *and* $\boldsymbol{F_{k+1}}$ *matrices using well*

*established least-squares technique by converting them to the set of linear equations [17]. Since (4.44) does not use any system matrix information except from* $\boldsymbol{C}$*, the Algorithm 3 is said to be model-free. Realize that the output matrix* $\boldsymbol{C}$ *represents the sensors placed to the system (4.1), which is clearly known.*

Realize that the Algorithm 3 achieves the static OPFB gain assuming $x$ is available. However, we are only given the output data $\boldsymbol{y}$. Therefore, we need to come up with an another algorithm that does not require system state data. One approach is to make use of the observability matrix while developing a model free algorithm [106]. However, this approach indeed requires the pair $(\boldsymbol{A}, \boldsymbol{C})$ to be observable. Thence, from now on we assume that the detectability condition in the Assumption 3 is strengthened as observability condition.

**Theorem 8.** *For any given n-dimensional observable system, there exists a sufficiently small time interval* $[0\ T]$ *such that if* $N$ *sampling times satisfy* $0 \leq t - i\Delta t \leq T\ \forall i \in 1, ..., N$ *where* $\Delta t$ *is the delayed time and assumed fixed, then the system is* $N$*-sample observable.*

**Proof.**: See [107] for the same proof.

To make use of the observability matrix define

$$\boldsymbol{x}(t)^T \boldsymbol{P} \boldsymbol{x}(t) = \boldsymbol{Y^T}(t) \tilde{\boldsymbol{P}} \boldsymbol{Y}(t) \tag{4.45}$$

where $\boldsymbol{Y}(t) = [\boldsymbol{y^T}(t)\ \dot{\boldsymbol{y}}\boldsymbol{^T}(t)\ ...\ \boldsymbol{y^{n-1T}}(t)]^T = \mathcal{O}\boldsymbol{x}$, and hence $\tilde{\boldsymbol{P}} = \mathcal{O}_L^T \boldsymbol{P} \mathcal{O}_L$ with $\mathcal{O}_L \in \mathbb{R}^{n \times nq}$ is the left-inverse of the observability matrix $\mathcal{O} \in \mathbb{R}^{nq \times n}$. Note that each derivative of the output $\boldsymbol{y}$ can be obtained by making use of the Taylor Series expansion on the collected data $\boldsymbol{y}(t + i\Delta t)$ around $\boldsymbol{y}(t)$. An example is $\ddot{\boldsymbol{y}} = \frac{\boldsymbol{y}(t+\Delta t)+\boldsymbol{y}(t-\Delta t)-2y(t)}{\Delta t^2}$ where $\boldsymbol{y}(t + \Delta t) = \boldsymbol{y}(t) + \Delta t \dot{\boldsymbol{y}}(t) + 0.5\Delta^2 t \ddot{\boldsymbol{y}}(t)$ and $\boldsymbol{y}(t - \Delta t) = \boldsymbol{y}(t) - \Delta t \dot{\boldsymbol{y}}(t) + 0.5\Delta^2 t \ddot{\boldsymbol{y}}(t)$.

Now, select $Q = kC^TC$ with $k > 0$ is a scalar, and define the following variables

$$\tilde{\boldsymbol{F}}_{\boldsymbol{k}} = \boldsymbol{F}_{\boldsymbol{k}}\mathcal{O}_L,\ \tilde{\boldsymbol{G}}_{\boldsymbol{k}} = \boldsymbol{G}_{\boldsymbol{k}}\mathcal{O}_L, \tag{4.46}$$

and the known term $\tilde{\boldsymbol{Q}} = -\tilde{\boldsymbol{F}}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{R}\tilde{\boldsymbol{F}}_{\boldsymbol{k}} - \hat{\boldsymbol{Q}} - \gamma^2\tilde{\boldsymbol{G}}_{\boldsymbol{k}}^{\boldsymbol{T}}\tilde{\boldsymbol{G}}_{\boldsymbol{k}}$ where $\hat{\boldsymbol{Q}} = k[\boldsymbol{I}_{\boldsymbol{q}}\ 0\ ...\ 0]_{qn\times q}^{T}[\boldsymbol{I}_{\boldsymbol{q}}\ 0\ ...\ 0]_{q\times qn}$. Realize that $\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{y} = \boldsymbol{Y}^{\boldsymbol{T}}\hat{\boldsymbol{Q}}\boldsymbol{Y}$ holds when $\boldsymbol{Q} = k\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{C}$, and it enables us to treat $\tilde{\boldsymbol{Q}}$ as a known term. Further, since the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable $(\boldsymbol{A}, k\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{C})$ is also observable. Based on these manipulations, a new Algorithm 4 can be developed.

**Algorithm 4.** *(Off-policy IRL algorithm, $\boldsymbol{x}$ is not required but the pair $(\boldsymbol{A}, \boldsymbol{C})$ must be observable.)*

1. *Initialize: Set $k = 1$, $\tilde{\boldsymbol{G}}_{\boldsymbol{0}} = \boldsymbol{0}$ and given Assumption 3 determine a stabilizing gain $\tilde{\boldsymbol{F}}_{\boldsymbol{0}}$.*
2. *$k^{th}$ iteration: Update $\tilde{\boldsymbol{P}}_{\boldsymbol{k}}$, $\tilde{\boldsymbol{G}}_{\boldsymbol{k+1}}$ and $\tilde{\boldsymbol{F}}_{\boldsymbol{k+1}}$ simultaneously*

$$\begin{aligned}&\boldsymbol{Y}^{\boldsymbol{T}}(t+T)\tilde{\boldsymbol{P}}_{\boldsymbol{k}}\boldsymbol{Y}(t+T) - \boldsymbol{Y}^{\boldsymbol{T}}(t)\tilde{\boldsymbol{P}}_{\boldsymbol{k}}\boldsymbol{Y}(t) - \int_t^{t+T} 2\gamma^2\boldsymbol{d}^{\boldsymbol{T}}\tilde{\boldsymbol{G}}_{\boldsymbol{k+1}}\boldsymbol{Y}\,d\tau \\ &- \int_t^{t+T} 2(\boldsymbol{u} + \tilde{\boldsymbol{F}}_{\boldsymbol{k}}\boldsymbol{Y})^T\boldsymbol{R}\tilde{\boldsymbol{F}}_{\boldsymbol{k+1}}\boldsymbol{Y}\,d\tau = \int_t^{t+T} \boldsymbol{Y}^{\boldsymbol{T}}\tilde{\boldsymbol{Q}}\boldsymbol{Y}\,d\tau.\end{aligned} \tag{4.47}$$

   *Set the cost $J_k = tr(\tilde{\boldsymbol{P}}_{\boldsymbol{k}}\boldsymbol{X}_{\boldsymbol{0}})$.*
3. *Check: If $\tilde{\boldsymbol{F}}_{\boldsymbol{k}}$ and $\tilde{\boldsymbol{F}}_{\boldsymbol{k+1}}$ are close enough to each other, go to step (4) else go to step (2).*
4. *Terminate: Given the Assumptions 7 and 8, set $\boldsymbol{u} = -\tilde{\boldsymbol{F}}_{\boldsymbol{k+1}}\boldsymbol{Y}$ and $\boldsymbol{J} = \boldsymbol{J}_{\boldsymbol{k}}$.* □

Realize that the Algorithm 4 converges with the static state feedback expression since $\boldsymbol{u} = -\tilde{\boldsymbol{F}}_{\boldsymbol{k+1}}\boldsymbol{Y} = -\boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}_{\boldsymbol{k}}\mathcal{O}_L\mathcal{O}\boldsymbol{x} = -\boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}_{\boldsymbol{k}}\boldsymbol{x}$ (5.20). Thence, it regulates not only the state elements spanned in the row-space of $\boldsymbol{C}$ but all states. However, it creates additional complexity as it requires more data to be collected to converge. On the other hand, the Algorithm 4 does not require system state data

$\boldsymbol{x}$, and it directly achieves the Nash equilibrium control $\boldsymbol{u}$ by making use of the new variables $\tilde{\boldsymbol{F}}_{\boldsymbol{k+1}}$ and $\boldsymbol{Y}$ instead of calculating the OPFB gain $\boldsymbol{K}_{\boldsymbol{o}}^{\boldsymbol{a}}$. Additionally, since the Algorithm 4 is obtained with the change of variables in the Algorithm 3, it shares the similar convergence properties with the Algorithm 3, and hence the Algorithm 2. The next section shows a way of solving coupled linear equations.

### 4.3.2 Data-based implementation of the IRL Algorithm

This section introduces a least-squares method to solve step (2) in Algorithm 3. Although, value function approximation is a popular tool and can be employed to solve step 2 in Algorithm 3, it requires three Neural Networks (NNs), i.e., the actor NN to approximate the value functional (4.7), the critic NN to update control policy and the disturber NN to update disturbance policy [18]. This causes a complicated NN design procedure. Instead, we use a least-squares method to solve for $\boldsymbol{P}_{\boldsymbol{k}}$ in step (2) of Algorithm 3, however, we first need to find $\boldsymbol{P}_{\boldsymbol{k}}$.

Now, we use the Kronecker product property $\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{b} = (\boldsymbol{b}^{\boldsymbol{T}} \otimes \boldsymbol{a}^{\boldsymbol{T}})vec(\boldsymbol{W})$ to rewrite (4.44) as

$$
\begin{aligned}
&[\hat{\boldsymbol{x}}(t+T) - \hat{\boldsymbol{x}}(t)]^T \hat{\boldsymbol{P}}_{\boldsymbol{k}} - 2\gamma^2 (\int_t^{t+T} \boldsymbol{x}^{\boldsymbol{T}} \otimes \boldsymbol{d}^{\boldsymbol{T}} d\tau) vec(\boldsymbol{G}_{\boldsymbol{k+1}}) \\
&- 2(\int_t^{t+T} \boldsymbol{x}^{\boldsymbol{T}} \otimes [(\boldsymbol{u} + \boldsymbol{F}_{\boldsymbol{k}}\boldsymbol{x})^T \boldsymbol{R}] d\tau) vec(\boldsymbol{F}_{\boldsymbol{k+1}}) = \int_t^{t+T} \boldsymbol{x}^{\boldsymbol{T}} \bar{\boldsymbol{Q}} \boldsymbol{x} d\tau
\end{aligned} \tag{4.48}
$$

where the vectors $\hat{\boldsymbol{x}} \in \mathbb{R}^{\frac{n(n-1)}{2}}$ and $\hat{\boldsymbol{P}}_{\boldsymbol{k}} \in \mathbb{R}^{\frac{n(n-1)}{2}}$ are defined in the following forms

$$
\begin{aligned}
\hat{\boldsymbol{x}} &= [x_1^2, 2x_1x_2, \cdots, 2x_1x_n, x_2^2, \cdots, 2x_{n-1}x_n, x_n^2]^T \\
\hat{\boldsymbol{P}}_{\boldsymbol{k}} &= [P_{k(11)}, \cdots, P_{k(1n)}, P_{k(22)}, \cdots, P_{k(2n)}, \cdots, P_{k(nn)}]^T.
\end{aligned} \tag{4.49}
$$

To solve $\boldsymbol{P_k}$, $\boldsymbol{G_{k+1}}$ and $\boldsymbol{F_{k+1}}$ in (4.48), define

$$\boldsymbol{d_x} = [\hat{\boldsymbol{x}}(t+T) - \hat{\boldsymbol{x}}(t), \cdots, \tag{4.50}$$
$$\hat{\boldsymbol{x}}(t+s_1T) - \hat{\boldsymbol{x}}(t+(s_1-1)T)]^T \in \mathbb{R}^{s_1 \times \frac{n(n-1)}{2}};$$
$$\boldsymbol{I_{xd}} = [\int_t^{t+T} (\boldsymbol{x} \otimes \boldsymbol{d}) d\tau, \cdots, \tag{4.51}$$
$$\int_{t+(s_1-1)T}^{t+s_1T} (\boldsymbol{x} \otimes \boldsymbol{d}) d\tau]^T \in \mathbb{R}^{s_1 \times np};$$
$$\boldsymbol{I_{xu}} = [\int_t^{t+T} \boldsymbol{x} \otimes (\boldsymbol{u} + \boldsymbol{F_k x}) d\tau, \cdots, \tag{4.52}$$
$$\int_{t+(s_1-1)T}^{t+s_1T} \boldsymbol{x} \otimes (\boldsymbol{u} + \boldsymbol{F_k x}) d\tau]^T \in \mathbb{R}^{s_1 \times nm};$$
$$\boldsymbol{\Phi} = [\boldsymbol{d_x}, -\boldsymbol{2I_{xd}}, -\boldsymbol{2I_{xu}}(\boldsymbol{I_n} \otimes \boldsymbol{R})]; \tag{4.53}$$
$$\boldsymbol{\Psi} = -[\int_t^{t+T} \boldsymbol{x^T \bar{Q} x} d\tau, \cdots \int_{t+(s_1-1)T}^{t+s_1T} \boldsymbol{x^T \bar{Q} x} d\tau]^T, \tag{4.54}$$

where integer $s_1 > 0$ is sampling data group number. Then solution can be obtained by

$$\begin{bmatrix} \hat{\boldsymbol{P}}_{\boldsymbol{k}} \\ \mathrm{vec}(\boldsymbol{G_{k+1}}) \\ \mathrm{vec}(\boldsymbol{F_{k+1}}) \end{bmatrix} = (\boldsymbol{\Phi^T \Phi})^{-1} \boldsymbol{\Phi^T \Psi}. \tag{4.55}$$

**Remark 15.** *To ensure that (4.55) achieves the unique solution, the persistence of excitation condition needs to be satisfied. To this end, probing noise should be injected to control input u in (37). This is also called as exploration noise that does not affect the convergence [17]. In addition, the data group number $s_1$ should be no less than $\frac{n(n+1)}{2} + np + nm$, which is the number of unknown parameters to be calculated by (4.55).*

In the next section, the correct performance of the proposed methods are validated by applying them to the lateral control control of linearized F-16 dynamics.

## 4.4 Simulation results

In this section, an example is given to verify the correct performance of proposed algorithms that solves optimal $H_\infty$ static OPFB regulator problem. We used the F-16 linearized lateral dynamics at a particular flight condition given in example 5.3-1 of the book [108]. Parameters of the linearized F-16 lateral dynamics and the corresponding system vectors are

$$
\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} -0.32 & 0.064 & 0.036 & -0.992 & 0 & 0.001 \\ 0 & 0 & 1 & 0.004 & 0 & 0 \\ -30.65 & 0 & -3.68 & 0.665 & -0.733 & 0.132 \\ 8.54 & 0 & -0.025 & -0.476 & -0.032 & -0.062 \\ 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 \end{bmatrix}; \\
\boldsymbol{B} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 20.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20.2 \end{bmatrix}^T; \\
\boldsymbol{C} &= \begin{bmatrix} 0 & 0 & 0 & 57.2958 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 & 0 & 0 \\ 57.2958 & 0 & 0 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 & 0 \end{bmatrix}; \\
\boldsymbol{x} &= \begin{bmatrix} \beta & \phi & p & r & \delta_a & \delta_r \end{bmatrix}^T; \ \boldsymbol{u} = \begin{bmatrix} u_a & u_r \end{bmatrix}^T; \\
\boldsymbol{y} &= \begin{bmatrix} r & p & \beta & \phi \end{bmatrix}^T
\end{aligned}
\tag{4.56}
$$

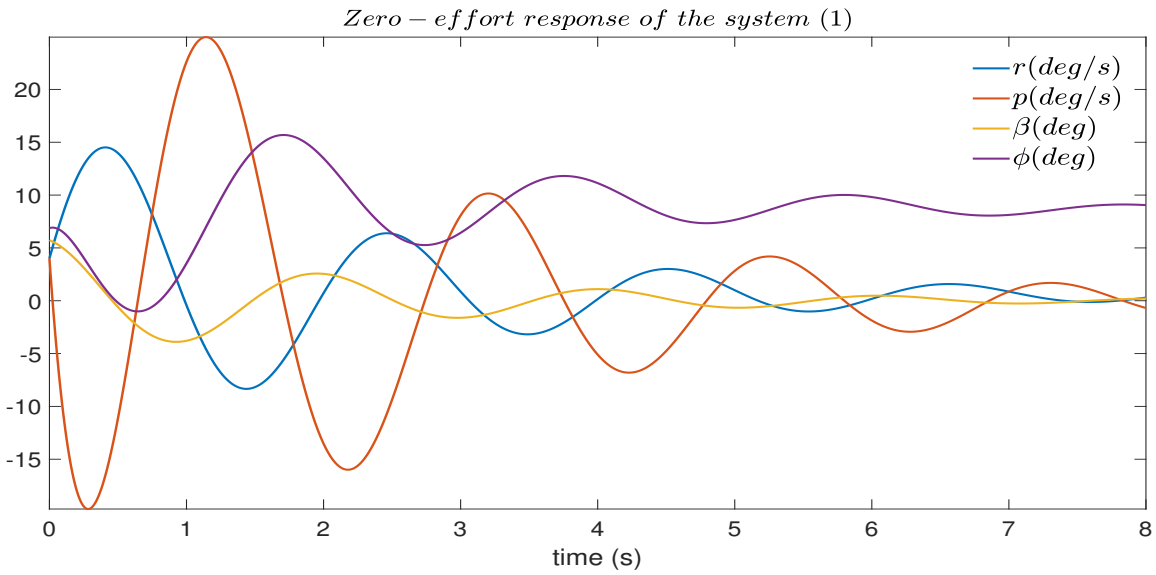


Figure 20: System response when no control is applied.

where $\beta$ denotes the side-slip, $\phi$ is the bank angle, $p$ is the roll rate, $r$ is the yaw rate, $\delta_a$ is the aileron actuator angle, $\delta_r$ is the rudder actuator angle, $u_a$ is the aileron servo input, $u_r$ is the rudder servo input. The factor of 57.2958 in the output-matrix $\boldsymbol{C}$ converts radians to degrees.

Additionally, the objective functional (4.5) parameters are selected as $\boldsymbol{Q} = diag([50\ 100\ 100\ 50\ 0\ 0])$, $\boldsymbol{R} = \rho \times diag([0.1\ 0.7])$ with $\rho = 0.3$ for computation of the OPFB gain. Also, select the disturbance matrix as

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}^T, \tag{4.57}$$

and set the attenuation level $\gamma = 2.5$. To examine the robustness, assume that the system (4.1) experiences the worst-case disturbance (4.20). Realize that all of the Assumptions 3-8, and the sufficient condition $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}} \geq 2\gamma^{-2}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}$ are satisfied with the given parameters in (4.56) and (4.57). Now, we first illustrate the

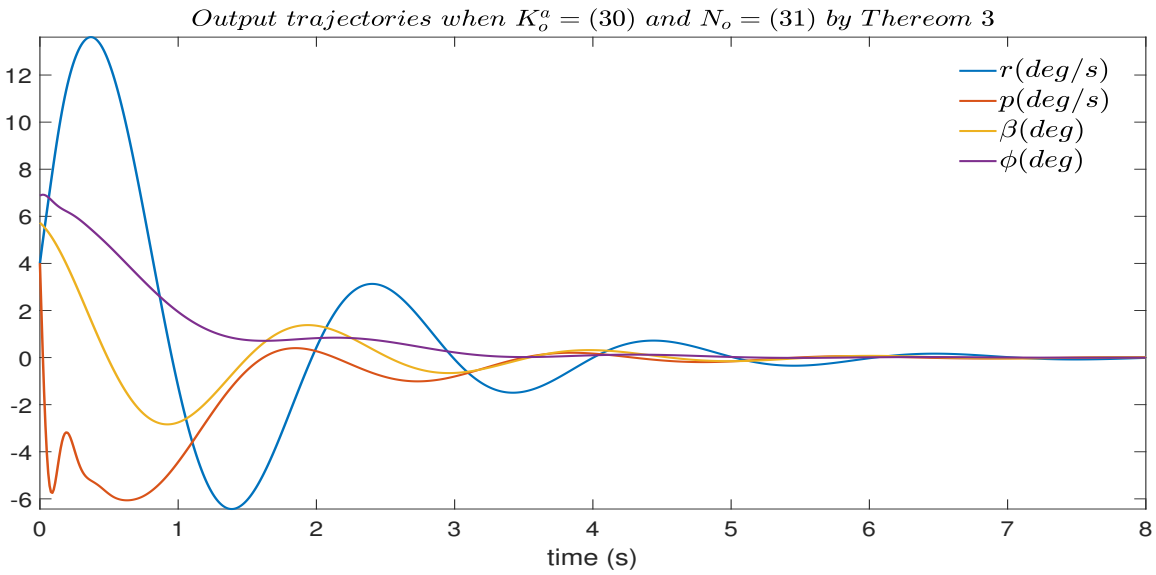


Figure 21: System response when the Nash gains (4.30) and (4.31) are employed.

performance of model-based Nash gain solutions (4.30) and (4.31), and then check whether the the Algorithms 1 and 2 converges the same game solutions as (4.30) and (4.31). After verifying the correctness of them, we illustrate the performance of the Nash solutions obtained in the Algorithms 3 and 4 that are model-free.

The Fig. 26 illustrates the zero control-effort response of the system (4.1). The Fig. 27 illustrates the optimal stabilizing gain (4.30) performance, which is derived in Theorem 11. The Nash OPFB gains found by (4.30) and (4.31) are

$$\boldsymbol{K_o^a} = \begin{bmatrix} -0.1395 & -0.8714 & 1.4785 & -1.0000 \\ -0.1410 & 0.0273 & -0.1070 & 0.0330 \end{bmatrix}, \tag{4.58}$$

$$\boldsymbol{N^o} = \begin{bmatrix} -0.0001 & -0.0006 & 0.0011 & -0.0007 \\ -0.0005 & 0.0001 & -0.0004 & 0.0001 \end{bmatrix}. \tag{4.59}$$

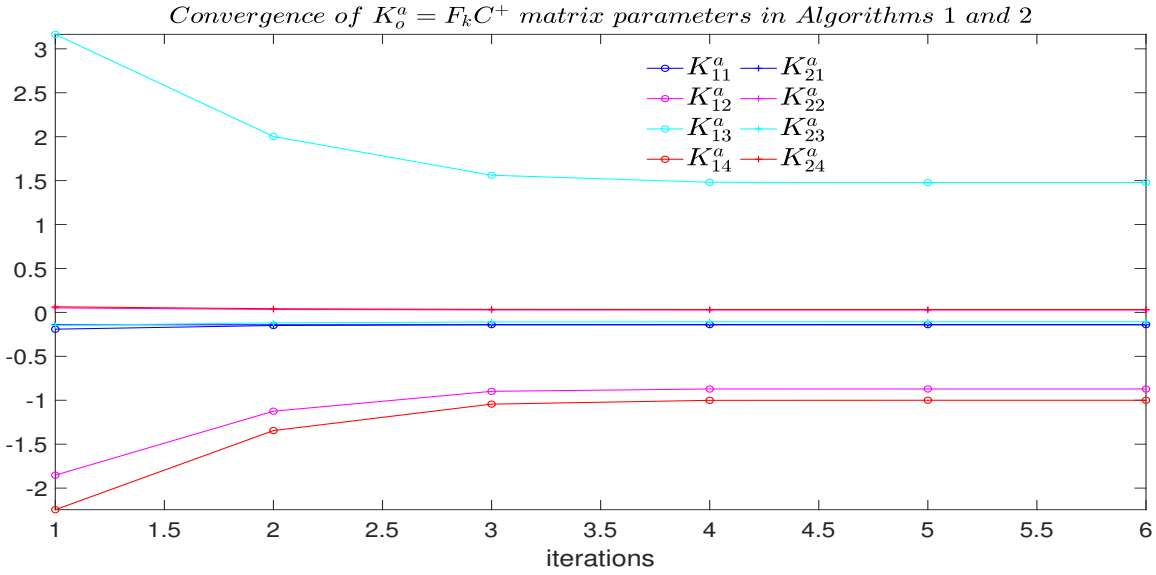


Figure 22: Convergence of the gain matrix $\boldsymbol{K_o^a}$ parameters by using the Algorithms 1 and 2.

Now, to examine the performance of the Algorithms 1 and 2, we set the initial sub-optimal stabilizing gain as

$$\boldsymbol{F_0} = \begin{bmatrix} -0.0888 & -0.1875 & 0.7076 & -0.2328 \\ -0.1382 & 0.0105 & -0.0884 & 0.0141 \end{bmatrix} C. \tag{4.60}$$

The convergence of optimal gain matrix parameters $\boldsymbol{K_o^a}$ can be observed from Fig. 28. Note that their convergence properties are exactly the same as each other and they converge to the same gain matrix (4.58) as expected. Therefore, the output trajectories figure with this resultant optimal stabilizing gain $\boldsymbol{K^a}$ is the same as Fig. 27. Now compare the system responses in Fig. 26 and Fig. 27 to observe the regulation performance. The convergence of disturbance gain matrix parameters $N_o$ is also illustrated in Fig. 30. Note that the converged parameters of $N_o$ are the same the OPFB disturbance gain given in (4.59) as illustrated.

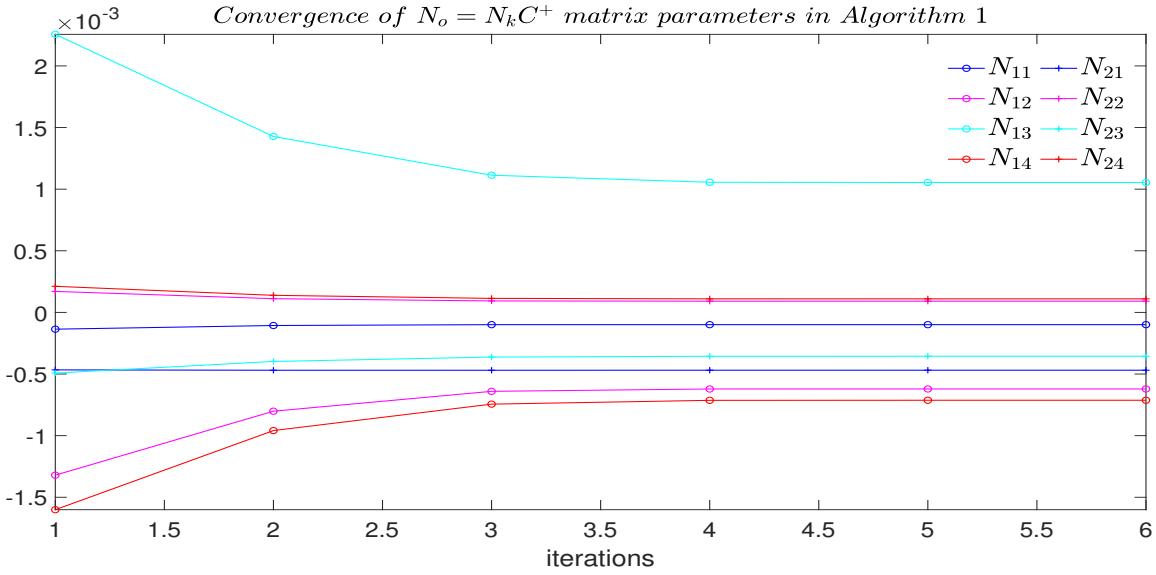


Figure 23: Convergence of the $\boldsymbol{N_o}$ matrix parameters by using the Algorithm 1.

Next, to show the correct performance of model free Algorithm 3, we set the initial stabilizing gain $\boldsymbol{F_0}$ as the same as (4.60). Then, based on the state information gathered from the system, the Algorithm 3 is employed to learn the Nash equilibrium gain solution $\boldsymbol{K_o^a}$ online. Once the IRL Algorithm converged, we applied the corresponding control $\boldsymbol{u} = -\boldsymbol{K_o^a y}$ using to the actual system (4.1). The Algorithm 3 is also converged to the same Nash gain $\boldsymbol{K_o^a}$ (4.58), thereby the resultant output vector states are the same as Fig. 27 as expected.

On the other hand, to check the correctness of the Algorithm 4, we select $\boldsymbol{Q} = k\boldsymbol{C^T C}$ with $k = 0.05$ as explained in the Section 4.3.1. The resultant output trajectories are shown in the Fig. 6. Additionally, we have calculated the the Nash state feedback gain expression $\boldsymbol{K^*}$ given in (5.20), and also the observability matrix $\mathcal{O}$ for the system (4.1). Then, we compared the $\boldsymbol{K^*}$ with the $\bar{\boldsymbol{F}}_{\boldsymbol{k+1}}\mathcal{O}$. It has been seen that the two matrices have almost the same elements and $L_2$ norm difference of them is calculated as 0.84, which verifies the correct performance of the Algorithm 4.

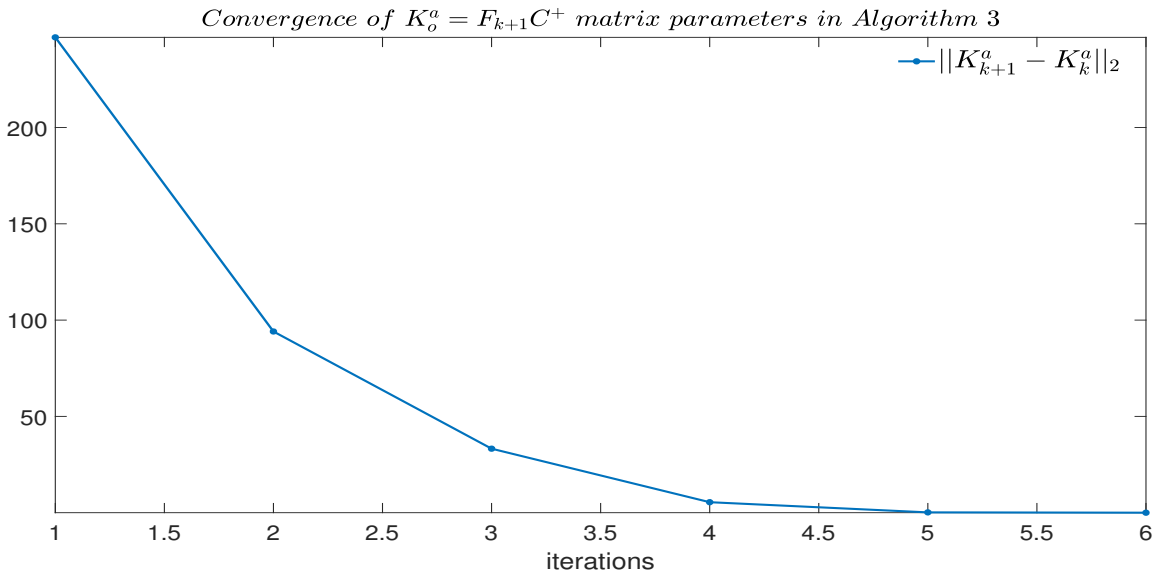


Figure 24: Convergence of the gain matrix $\boldsymbol{K_o^a}$ parameters by using the Algorithm 3.

Lastly, the critical attenuation level obtained in the Theorems 6 and 11 is $\gamma^* = 0.05$. Note that with this critical attenuation $\gamma^*$ level, the sufficient condition $\boldsymbol{BR^{-1}B^T} \geq 2(\gamma^*)^{-2}\boldsymbol{DD^T}$ is relaxed. However, this condition indeed required for the Kleinman based Algorithms 1-4, and the critical attenuation level is obtained as $\gamma^* = 0.49$, which is also compatible with the sufficient condition $\boldsymbol{BR^{-1}B^T} \geq 2(\gamma^*)^{-2}\boldsymbol{DD^T}$. Therefore, we conclude that the model free algorithms reduces the $L_2$ gain performance.

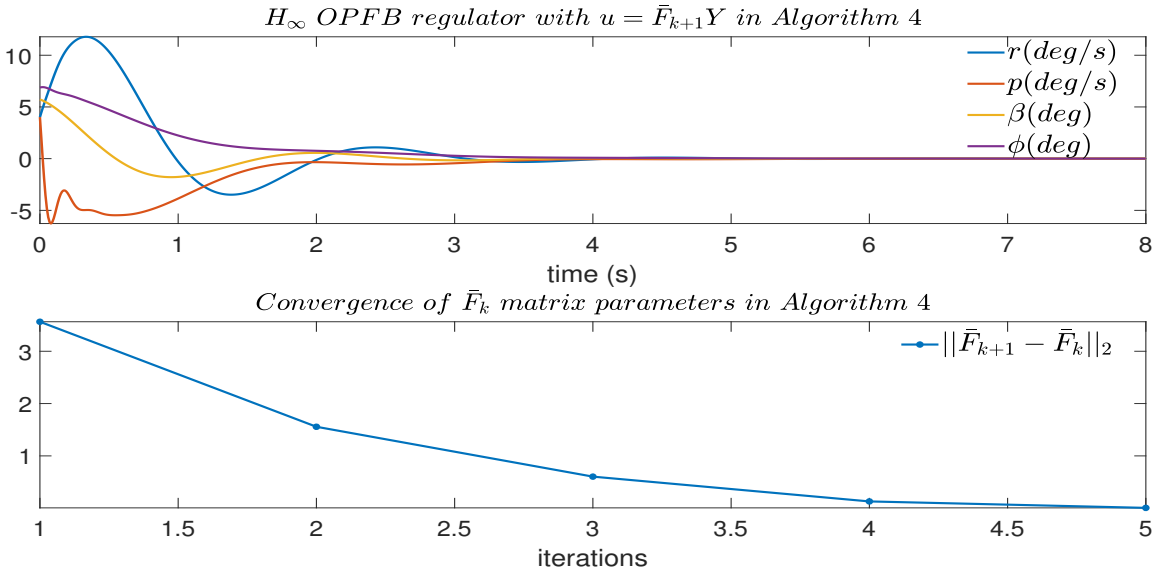


Figure 25: Performance of the Algorithm 4.

Adversarial Multi-agent Output Containment Graphical Game with Local and Global Objectives for UAVs.

Yusuf Kartal, Ahmet Taha Koru, Frank Lewis, Yan Wan, Atilla Dogan, 2021.

*Under Review in IEEE Transactions on Control of Network Systems.*

# CHAPTER 5

## Adversarial Multi-agent Output Containment Graphical Game with Local and Global Objectives for UAVs

Multi-agent systems (MAS) control is one of the most widely studied phenomena in recent years, due to the capability of MAS to perform certain tasks such as transportation [109], surveillance and reconnaissance [110], and target search and detection [111]. Three recognized categories of MAS control are single-leader & single-follower [112], single-leader & multi-follower [96], and multi-leader & multi-follower (MLF) [19]. For the MLF systems, one control objective is to guarantee the convergence of the output of each follower to the dynamic convex hull spanned by the outputs of leaders [19], [20], [21] which can be achieved by regulating the corresponding output containment error. These studies do not consider a practical case where the followers have mutual interests. Formulating the MLF system as a differential graphical game can address the mutual interests among the agents, and hence constitute a correct framework for the analysis of the MLF systems.

In the last decade, extensive efforts have been made to achieve Nash equilibrium strategies in differential graphical games [22],[23], [24], [25], [26]. The Nash equilibrium strategy in [22] is a centralized one since it requires to access the global state information of MAS. Min-max strategies in [76] guarantee a security-level performance when a Nash solution to graphical game does not exist. A modified objective functional in [23] guarantees the existence of both distributed and Nash solutions for the single-leader & multi-follower MAS.

In practice, besides Nash strategies, a MAS control design requirements may include $\mathcal{L}_2$ gain boundedness, robustness against external disturbances, and output-feedback stabilizability. These design specifications are indeed local objectives of the graphical MLF game. Therefore, a new concept so called local and global objectives arises in the literature [113]. The works [76], [114], [115] are indeed related to the this new concept. [76] analyzed the multi-agent pursuit-evasion game where pursuer group have local objective, staying together, and global objective, capturing the evaders. [113] employed local objectives for a subset of agents, which on the other hand, are tasks determined around the global objective by agent-specific exosystems. Most of these works do not yield local and graphical Nash solutions simultaneously.

The standard existing solutions to the output containment MLF games uses an additive gain matrices to prove stability [19], [20]. In [19], sufficient local conditions in terms of stabilizing the local followers' dynamics and satisfying a certain $\mathcal{H}_\infty$ criterion are investigated without considering mutual interests among the followers. The resulting strategies do not constitute a Nash equilibrium solution. In this paper, we propose graph theoretic and algebraic conditions to obtain Nash equilibrium strategies for the output containment of a MLF system. The salient contributions of this paper summed up into four categories as:

- The graphical output containment game is introduced where the mutual interests among the followers that experiences worst-case disturbances is addressed by introducing differential graphical games. For this graphical game, the output containment problem is the global objective of the follower group and disturbance attenuation is the local objective for some agents.
- A new solution to the $\mathcal{H}_\infty$ static output feedback (OPFB) control problem that yields Nash equilibrium strategies for the graphical game is presented. The Nash solutions are proved to guarantee stability and bounded $L_2$ gain considering

corresponding local min-max game worst-case disturbances with the developed novel necessary and sufficient conditions.

- The conditions for the distributed strategy based on available output data is presented to accomplish the local $\mathcal{H}_\infty$ objective and the global output containment objective for the MLF system.
- To verify correctness of the proposed methods, the linear quadrotor model is developed around a particular flight condition. Then, the strategies constituting Nash equilibrium solution to output containment game are tested by means of the MLF quadrotor UAV simulations.

The rest of the paper is organised as follows. Section 5.2.1 considers a local $\mathcal{H}_\infty$ game where the control inputs aim to minimize corresponding objective functional, whereas the disturbances aim to maximize. This corresponds to a well-known zero-sum game concept [61]. After playing the local game, we consider differential graphical game in Section 5.2.2 where followers' mutual interests are addressed. Furthermore, we give sufficient conditions on the objective functional's design parameters to satisfy both of the local and global objectives simultaneously. Section 5.3 reveals the $\mathcal{L}_2$ gain bound stability analysis of the games introduced. Lastly, the proposed methods are tested by means of the MLF quadrotor UAV simulations in Section 5.4.

**Notations**. We use the following notations throughout this paper. $\boldsymbol{I_n} \in \mathbb{R}^{n\times n}$ is the identity matrix, $\mathbf{1_N} \in \mathbb{R}^N$ is a vector whose elements are all ones, and similarly $\mathbf{0_N} \in \mathbb{R}^N$ is a vector whose elements are all zeros. $\mathbf{0_{n\times m}} \in \mathbb{R}^{n\times m}$ is a matrix whose elements are all zeros. $diag(\zeta_i)$ represents a diagonal matrix with $\zeta_i$ $\forall i \in 1, ..., N$ on its diagonal. The condition $A > 0$ ($\geq 0$) denotes the positive (semi) definiteness of a matrix. The operator $tr()$ denotes the trace of a matrix. $\boldsymbol{C^+ = C^T(CC^T)^{-1}}$ is the right-inverse of the full row-rank matrix $\boldsymbol{C}$ and the Kronecker product operator is denoted by $\otimes$. The determinant of a square matrix is denoted by $|.|$. Lastly, distance

from $\boldsymbol{x} \in \mathbb{R}^n$ to the set $\mathbb{C} \subseteq \mathbb{R}^n$ is defined via Euclidean norm as $dist(\boldsymbol{x}, \mathbb{C}) = \inf_{\boldsymbol{y} \in \mathbb{C}} ||\boldsymbol{x} - \boldsymbol{y}||_2$.

## 5.1 Preliminaries

This section presents various definitions on adversarial multi-agent leader-follower (MLF) games by revealing multi-agent system dynamics and communication graph topologies that are of interest. Till the simulation section, the MLF game with generalized Linear Time Invariant (LTI) system dynamics are analyzed. Then, the specific LTI system dynamics are introduced for quadrotor UAVs in the Section 5.4.

### 5.1.1 Graph Topologies

A communication graph for $N$ followers is denoted with a pair $\mathrm{G^f} = (V^f, E^f)$, where the set $V^f = \{\mathrm{v_1^f}, ..., \mathrm{v_N^f}\}$ denotes the nodes, and $E^f$ stands for the edges, which is composed of node pairs $(\mathrm{v_i^f}, \mathrm{v_k^f})$. Each edge $\left(\mathrm{v_k^f}, \mathrm{v_i^f}\right) \in E^f$, has a weight $a_{ik}^f = 1$ if node $k$ is connected to node $i$ and $a_{ik}^f = 0$ otherwise. The graph is called as undirected if $a_{ik}^f = a_{ki}^f$, $\forall \mathrm{i, k}$, otherwise it is termed a digraph. The in-degree matrix $\boldsymbol{D^f} = diag\{d_i^f\}$ where $d_i^f = \sum_{k=1}^{N} a_{ik}^f$. The matrix $\boldsymbol{\mathcal{A}^f} = [a_{ik}^f] \in \mathbb{R}^{N \times N}$ denotes the adjacency or connectivity matrix. Then, the graph Laplacian matrix for the follower group is defined as $\boldsymbol{\mathcal{L}^f} = \boldsymbol{D^f} - \boldsymbol{\mathcal{A}^f}$. In this paper, we assume that $\mathrm{G^f}$ is a digraph that does not contain self loops, i.e, $a_{ii}^f = 0$.

In addition, define a bipartite graph $\mathrm{G^b} = (V^f, V^l, E^b)$ that consists of follower nodes $V^f$, leader nodes $V^l$, and edges $E^b$, which captures the information exchange among the follower and leader groups. Let $g_{ji}^b$ denote the pinning of follower $i$ to leader $j$, with $g_{ji}^b = 1$ if follower $i$ is connected to leader $j$, and $g_{ji}^b = 0$ otherwise. Additionally, let $g_{ij}^b$ denote the pinning of leader $j$ to follower $i$, with $g_{ij}^b = 1$ if leader $j$ is pinned to follower $i$, and $g_{ij}^b = 0$ otherwise. Then pinning matrices of follower $i$

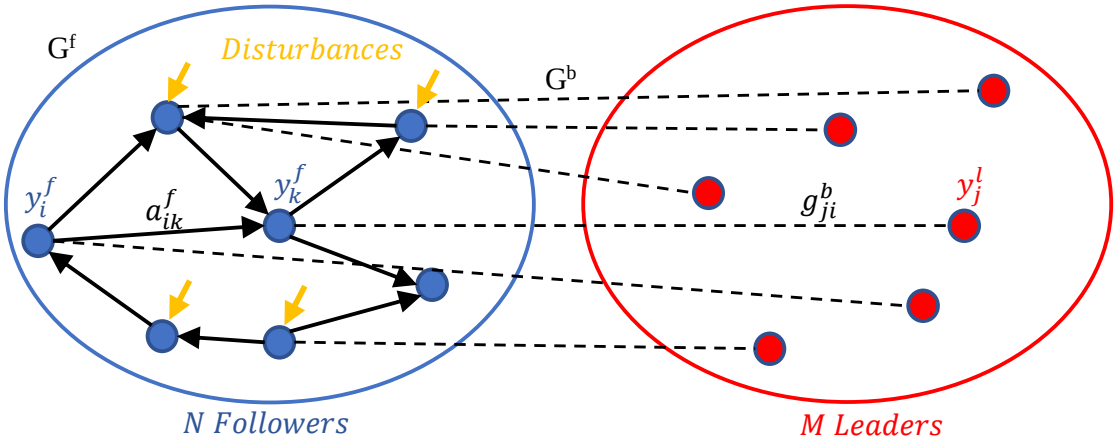


Figure 26: Layout of Adversarial Leader-Follower Graphical Game

and leader $j$ in $\mathrm{G}^{\mathrm{b}}$ are $\boldsymbol{G_j^b}$ and $\boldsymbol{G_i^b}$ respectively. One of the contributions of this paper is to analyze MLF games with directed communication graph topologies that lead to the limited measurement capabilities among the agents ($\mathrm{G}^{\mathrm{f}}$ nodes) of game. Note that the graph formulation in this paper is valid for the non-negative connectivity and pinning weights i.e., $a_{ik}^f \geq 0$ and $g_{ij}^f \geq 0$ but they are selected as ones and zeros so as not to increase gain of outer control loop for quadrotor UAVs in Section 5.4.

### 5.1.2 Multi-agent System Dynamics and Local Errors

Consider a follower group consisting of $N$ agents with the dynamics given by

$$\begin{aligned} \dot{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{f}} &= \boldsymbol{A}\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{f}} + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{i}} + \boldsymbol{D}\boldsymbol{w}_{\boldsymbol{i}} \\ \boldsymbol{y}_{\boldsymbol{i}}^{\boldsymbol{f}} &= \boldsymbol{C}\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{f}}, \end{aligned} \tag{5.1}$$

$\forall i \in \{1, ..., N\}$, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, $\boldsymbol{D} \in \mathbb{R}^{n\times p}$ are system-state, input, disturbance matrices, and $\boldsymbol{C} \in \mathbb{R}^{q\times n}$ is assumed to be a full row-rank output matrix to avoid redundant measurements. The corresponding vectors $\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{f}}(t)$, $\boldsymbol{u_i}(t)$, $\boldsymbol{w_i}(t)$, and $\boldsymbol{y}_{\boldsymbol{i}}^{\boldsymbol{f}}(t)$ stand for the state, input, disturbance and output of $i^{th}$ follower respectively.

Additionally, consider a leader group that consists of $M$ agents with the dynamics given by

$$\begin{aligned} \dot{\boldsymbol{x}}_{\boldsymbol{j}}^{\boldsymbol{l}} &= \boldsymbol{A}\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{l}} \\ \boldsymbol{y}_{\boldsymbol{j}}^{\boldsymbol{l}} &= \boldsymbol{C}\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{l}}, \end{aligned} \tag{5.2}$$

$\forall j \in \{1, ..., M\}$, where vectors $\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{l}}(t)$, and $\boldsymbol{y}_{\boldsymbol{j}}^{\boldsymbol{l}}(t)$ stand for the state, and output of $j^{th}$ leader respectively.

**Assumption 6.** *The pair $(\boldsymbol{A}, \boldsymbol{B})$ is stabilizable and the pair $(\boldsymbol{A}, \boldsymbol{C})$ is detectable.*

**Assumption 7.** *The system (5.1) is OPFB stabilizable in the sense that the row-space of output matrix $C$ contains the sub-space that is spanned by the right eigenvectors corresponding to the unstable modes of $A$.*

**Assumption 8.** *The non-zero columns of the output matrix $C$ are linearly independent.*

**Remark 16.** *The Assumption 7 can be interpreted such that all unstable modes are measured by the output matrix $C$ that represents the sensors installed in the systems (5.1) and (5.2). The Assumption 8 enables us to recover a state element $x_i$ that is spanned in the row space of $\boldsymbol{C}$ precisely from the output vector $y$ once it is left multiplied with $\boldsymbol{C}^{+}$.*

**Definition 1.** *A set $\mathbb{S}$ is convex if the line segment between any two points in $\mathbb{S}$ lies in $\mathbb{S}$, i.e., $(\theta s_1 + (1-\theta)s_2) \in \mathbb{S}$ holds for any $s_1, s_2 \in \mathbb{S}$ and any $\theta \in [0, 1]$ [116]. The convex hull of a set $\mathcal{X} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_M}$ is denoted by $conv(\mathcal{X}) = \{\sum_{j=1}^{M} \gamma_j \boldsymbol{x_j} | \boldsymbol{x_j} \in \mathcal{X}, \gamma_j \geq 0, \sum_{j=1}^{M} \gamma_j = 1\}$.*

*Let $Y_{G^l} = \{\boldsymbol{y_1^l}, ..., \boldsymbol{y_M^l}\}$ be a set of leaders' positions. Note that the centroid of leaders' positions $\sum_{j=1}^{M} \gamma_j \boldsymbol{y_j^l}$ with $\gamma_j = 1/M, \forall j \in \{1, ..., M\}$ is an element of $conv(Y_{G^l})$.*

**Definition 2.** *Consider the $i^{th}$ follower dynamics (5.1) and the $j^{th}$ leader dynamics (5.2). The outputs of all followers are said to converge to the convex hull spanned by the outputs of leaders if*

$$\lim_{t\to\infty} dist(\boldsymbol{y_i^f}, conv(Y_{G^l})) = 0, \forall i \in 1, ..., N. \tag{5.3}$$

Motivated by [19], the local relevant output containment vector for $i^{th}$ follower is defined as

$$\begin{aligned}\boldsymbol{\xi_i} &= \sum_{k=1}^{N} a_{ik}^f(\boldsymbol{y_k^f} - \boldsymbol{y_i^f}) + \sum_{j=1}^{M} g_{ij}^b(\boldsymbol{y_j^f} - \boldsymbol{y_i^l}) \\ &= \sum_{j=1}^{M}\left(\frac{1}{M}\sum_{k=1}^{N} a_{ik}^f(\boldsymbol{y_k^f} - \boldsymbol{y_i^f}) + g_{ij}^b(\boldsymbol{y_j^f} - \boldsymbol{y_i^l})\right).\end{aligned} \tag{5.4}$$

The global form of (5.4) is

$$\boldsymbol{\xi} = -\sum_{j=1}^{M}\Big((\boldsymbol{\psi_j}\otimes \boldsymbol{I_q})(\boldsymbol{y^f} - \underline{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{l}})\Big) \tag{5.5}$$

where $\boldsymbol{\xi} = [\boldsymbol{\xi_1^T}, ..., \boldsymbol{\xi_N^T}]^T$, $\boldsymbol{\psi_j} = \frac{1}{M}\boldsymbol{\mathcal{L}^f} + \boldsymbol{G_j^b}$, $\underline{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{l}} = \boldsymbol{1_N}\otimes \boldsymbol{y_j^l}$, and $\boldsymbol{y^f} = [\boldsymbol{y_1^f}', ..., \boldsymbol{y_N^f}']^T$. The following steps enable us to verify that the output containment control objective (5.3) is achieved if $\lim_{t\to\infty}\boldsymbol{\xi} = \boldsymbol{0}$.

**Definition 3.** *A matrix is called semi-simple if the geometric multiplicity of each eigenvalue is equal to its algebraic multiplicity.*

It is known that a matrix is diagonalizable over real set $\mathbb{R}$, if it is semi-simple and every eigenvalue is real [117].

**Definition 4.** *A follower is called a well-informed one if it is pinned to the all leaders and uninformed one if it is not pinned to any leader.*

**Assumption 9.** *The follower $i$ in $\mathrm{G^f}$ is either a well-informed one or an uninformed one. There exists a directed path from one of the well informed followers $k$ to an*

*uninformed follower $i$ with $k \neq i$ $\forall k = \{1, ..., N\}$ in $\mathrm{G^f}$. Additionally, the graph is assumed to be acyclic and there is no directed connection from any uninformed follower $i$ to well informed follower $k$.*

Note that the existence of directed path from well-informed follower $k$ to uninformed follower $i$ is standard in multi-leader output containment control [19],[20]. Apart from this, we also assume that the graph $\mathrm{G^f}$ is acyclic and there is no directed connection from any uninformed follower $i$ to well informed follower $k$, which allows us to modify the standard objective functionals in the MLF games. The next corollary is an essential step before we introduce further details on the MLF games.

**Remark 17.** *Given Assumption 9, $\boldsymbol{\psi_j}$ and $\sum_{j=1}^{M} \boldsymbol{\psi_j}$ are non-singular M matrices, and hence the real parts of their eigenvalues are positive [118]. Furthermore, since the graph $\mathrm{G^b}$ is assumed to be acyclic, the pinned Laplacian matrix $(\sum_{j=1}^{M} \boldsymbol{\psi_j})$ is indeed a simple matrix that has purely positive real eigenvalues, which is introduced in the Chapter 5.7.3 of [119].*

**Remark 18.** *Assumption 9 also implies that there exists a directed path from at least one of the leaders to any follower in a unified communication graph $\mathrm{G^f} \cup \mathrm{G^b}$. This allows us to verify non-singular M-matrix property of the matrices $\boldsymbol{\psi_j}$ and $\sum_{j=1}^{M} \boldsymbol{\psi_j}$. Therefore, their inverses, $(\boldsymbol{\psi_j})^{-1}$ and $(\sum_{j=1}^{M} \boldsymbol{\psi_j})^{-1}$, exist and are non-negative. A comprehensive fifty properties of M-matrices are detailed in [120].*

**Lemma 3.** *Given the Assumption 9, consider the follower and leader dynamics in (5.1) and (5.2). The condition (5.3) is achieved if $\lim_{t\to\infty} \boldsymbol{\xi} = \mathbf{0}$.*

*Proof*: Re-write the global form of relevant output containment vector (5.5) as

$$\boldsymbol{\xi} = -\left(\sum_{j=1}^{M} (\boldsymbol{\psi_j} \otimes \boldsymbol{I_q})\right) \boldsymbol{y^f} + \sum_{j=1}^{M} \left((\boldsymbol{\psi_j} \otimes \boldsymbol{I_q}) \underline{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{l}}\right) \tag{5.6}$$

$\lim_{t\to\infty} \boldsymbol{\xi} = \mathbf{0}$ implies that

$$\begin{aligned}
\boldsymbol{y^f} &\to \left(\sum_{j=1}^{M}(\boldsymbol{\psi_j}\otimes \boldsymbol{I_q})\right)^{-1}\sum_{j=1}^{M}\left((\boldsymbol{\psi_j}\otimes \boldsymbol{I_q})\underline{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{l}}\right)\\
&\to \sum_{j=1}^{M}\left[\left(\sum_{r=1}^{M}(\boldsymbol{\psi_r}\otimes \boldsymbol{I_q})\right)^{-1}(\boldsymbol{\psi_j}\otimes \boldsymbol{I_q})\underline{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{l}}\right]\\
&\to \sum_{j=1}^{M}\left[\left(\left(\sum_{r=1}^{M}\boldsymbol{\psi_r}\right)^{-1}\boldsymbol{\psi_j}\mathbf{1_N}\right)\otimes \boldsymbol{y}_{\boldsymbol{j}}^{\boldsymbol{l}}\right]
\end{aligned} \tag{5.7}$$

as $t\to\infty$. Realize that

$$\begin{aligned}
\sum_{j=1}^{M}\left[\left(\sum_{r=1}^{M}\boldsymbol{\psi_r}\right)^{-1}\boldsymbol{\psi_j}\mathbf{1_N}\right] &= \left(\sum_{r=1}^{M}\boldsymbol{\psi_r}\right)^{-1}\left(\sum_{j=1}^{M}\boldsymbol{\psi_j}\mathbf{1_N}\right)\\
&= \mathbf{1_N},
\end{aligned} \tag{5.8}$$

which implies that each row sum of the vectors $\left(\sum_{r=1}^{M}\boldsymbol{\psi_r}\right)^{-1}\boldsymbol{\psi_j}\mathbf{1_N}\ \forall j\in 1,...,M$ is equal to 1. Using Remarks 17 and 18, one can derive non-negative definiteness of each entry of the vectors $\left(\sum_{r=1}^{M}\boldsymbol{\psi_r}\right)^{-1}\boldsymbol{\psi_j}\mathbf{1_N}$ [118]. Therefore, $\lim_{t\to\infty}\boldsymbol{\xi}=\mathbf{0}$ indeed verifies that the objective (5.3) is achieved by Definition 1. This completes the proof. □

### 5.1.3 Multi-agent Error Dynamics and $\mathcal{L}_2$ Gain Bound

Based on (5.7) in Lemma 3, define the global output containment vector as

$$\boldsymbol{\delta_y} = \boldsymbol{y^f} - \left(\sum_{j=1}^{M}(\boldsymbol{\psi_j}\otimes \boldsymbol{I_q})\right)^{-1}\sum_{j=1}^{M}\left((\boldsymbol{\psi_j}\otimes \boldsymbol{I_q})\underline{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{l}}\right). \tag{5.9}$$

Note that $\lim_{t\to\infty} \boldsymbol{\delta_y} = \mathbf{0} \implies \lim_{t\to\infty} \boldsymbol{\xi} = \mathbf{0}$ and (5.3) holds. Then, the global state containment vector is

$$\boldsymbol{\delta} = \boldsymbol{x^f} - \left(\sum_{j=1}^{M}(\boldsymbol{\psi_j} \otimes \boldsymbol{I_n})\right)^{-1} \sum_{j=1}^{M}\left((\boldsymbol{\psi_j} \otimes \boldsymbol{I_n})\underline{\boldsymbol{x}}_{\boldsymbol{j}}^{\boldsymbol{l}}\right) \tag{5.10}$$

where $\boldsymbol{\delta} = [\boldsymbol{\delta_1^T}, ..., \boldsymbol{\delta_N^T}]^T$, $\boldsymbol{x^f} = [\boldsymbol{x_1^f}\prime, ..., \boldsymbol{x_N^f}\prime]^T$ and $\underline{\boldsymbol{x}}_{\boldsymbol{j}}^{\boldsymbol{l}} = \mathbf{1_N} \otimes \boldsymbol{x_j^l}$.

Consider the global containment vectors (5.9) and (5.10), and the follower dynamics in (5.1) and the leader dynamics in (5.2). Then, by using the fact that $\boldsymbol{\xi} = -\left(\sum_{j=1}^{M}(\boldsymbol{\psi_j} \otimes \boldsymbol{I_q})\right)\boldsymbol{\delta_y}$, the output containment error system can be written in the following global form

$$\begin{aligned} \dot{\boldsymbol{\delta}} &= (\boldsymbol{I_N} \otimes \boldsymbol{A})\boldsymbol{\delta} + (\boldsymbol{I_N} \otimes \boldsymbol{B})\boldsymbol{u} + (\boldsymbol{I_N} \otimes \boldsymbol{D})\boldsymbol{w} \\ \boldsymbol{\xi} &= -\left(\sum_{j=1}^{M}(\boldsymbol{\psi_j} \otimes \boldsymbol{C})\right)\boldsymbol{\delta}. \end{aligned} \tag{5.11}$$

where $\boldsymbol{u} = [\boldsymbol{u_1^T}, ..., \boldsymbol{u_N^T}]^T$, and $\boldsymbol{w} = [\boldsymbol{w_1^T}, ..., \boldsymbol{w_N^T}]^T$.

**Definition 5.** *Realization of the following inequality $\forall \boldsymbol{w} \in [0, \infty)$ implies that the $\mathcal{L}_2$ gain of system (5.11) is bounded by a prescribed disturbance attenuation level denoted by $\gamma$*

$$\int_0^\infty \frac{\boldsymbol{\xi^T\xi}}{\|\boldsymbol{C^TC}\|_2} dt \leq \gamma^2 \int_0^\infty \boldsymbol{w^Tw} dt + \beta \tag{5.12}$$

*where $\beta$ is a non-negative constant. The condition (5.12) is also called as nonexpansivity constraint in [74].*

The following Lemma is inspired by [96].

**Lemma 4.** *Given the Assumption 9 and the output containment error system (5.11). Assume that a stabilizing static output feedback (OPFB) control takes the form*

$$\boldsymbol{u_i} = -c_i \boldsymbol{K_i} \boldsymbol{\xi_i}. \tag{5.13}$$

*where c is a positive constant. Then, the output containment objective (5.3) and $\mathcal{L}_2$ gain bound condition (5.12) hold if the distributed N systems*

$$\begin{aligned} \dot{\hat{\boldsymbol{\delta}}}_{\boldsymbol{i}} &= \boldsymbol{A}\hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \boldsymbol{B}\hat{\boldsymbol{u}}_{\boldsymbol{i}} + \boldsymbol{D}\hat{\boldsymbol{w}}_{\boldsymbol{i}} \\ &= (\boldsymbol{A} - c_i\lambda_i \boldsymbol{B}\boldsymbol{K_i}\boldsymbol{C})\hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \boldsymbol{D}\hat{\boldsymbol{w}}_{\boldsymbol{i}} \\ \hat{\boldsymbol{\xi}}_{\boldsymbol{i}} &= -\lambda_i \boldsymbol{C}\hat{\boldsymbol{\delta}}_{\boldsymbol{i}} \end{aligned} \tag{5.14}$$

*are both asymptotically stable and $\mathcal{L}_2$ gain bounded by $\gamma > 0$, where $\lambda_i$ are eigenvalues of the matrix $\sum_{j=1}^{M} \boldsymbol{\psi_j}$.*

*Proof*: By Definition 3, the M-matrix $\sum_{j=1}^{M} \boldsymbol{\psi_j}$ is diagonalizable given the Assumption 9, thereby there exists an invertible transformation matrix $\boldsymbol{T}$ such that

$$\boldsymbol{T}(\sum_{j=1}^{M} \boldsymbol{\psi_j})\boldsymbol{T^{-1}} = \boldsymbol{\Lambda} = diag(\lambda_1, ..., \lambda_N). \tag{5.15}$$

Let $\hat{\boldsymbol{\delta}} = (\boldsymbol{T} \otimes \boldsymbol{I_n})\boldsymbol{\delta}$, $\boldsymbol{w} = (\boldsymbol{T^{-1}} \otimes \boldsymbol{I_p})\hat{\boldsymbol{w}}$, and $\hat{\boldsymbol{\xi}} = (\boldsymbol{T} \otimes \boldsymbol{I_q})\boldsymbol{\xi}$. Then (5.14) can be written in the global form as

$$\begin{aligned} \dot{\hat{\boldsymbol{\delta}}} &= (\boldsymbol{I_N} \otimes \boldsymbol{A} - c_i\boldsymbol{\Lambda} \otimes \boldsymbol{BKC})\hat{\boldsymbol{\delta}} + (\boldsymbol{I_N} \otimes \boldsymbol{D})\hat{\boldsymbol{w}} \\ \hat{\boldsymbol{\xi}} &= -(\boldsymbol{\Lambda} \otimes \boldsymbol{C})\hat{\boldsymbol{\delta}}. \end{aligned} \tag{5.16}$$

Realize that the $\mathcal{L}_2$ norm equivalence of the pairs $(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})$ and $(\boldsymbol{w}, \hat{\boldsymbol{w}})$ are straight forward as $\boldsymbol{T}$ is invertible, and hence (5.11) and (5.16) have the same $\mathcal{L}_2$ gains.

Additionally, since they have the same transfer function, stability of the equilibrium (origin) in both systems are equivalent, which completes the proof. □

## 5.2 Multi-agent leader-follower game formulation

In this section, two types of games are analyzed. The first game is a local game, which is played between the control and disturbance terms of the $i^{th}$ follower. On the other hand, the second game is a global graphical game, which is played between the $i^{th}$ and $j^{th}$ followers considering the worst case disturbance that is achieved in the corresponding local game.

**Remark 19.** *Herein, we employ state feedback control method to derive the Nash equilibrium strategies. Then, the achieved control strategies can be re-written in the OPFB form given the Assumptions 6-8. This will be illustrated later in the Section 5.3.*

### 5.2.1 Local $\mathcal{H}_\infty$ Game Solution

This section presents a local $\mathcal{H}_\infty$ game solution, which is formulated in the form of zero-sum game by introducing the following objective functional

$$\mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) = \int_0^\infty (\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{Q}}_{\boldsymbol{i}} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \frac{1}{c_i \lambda_i} \hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{R}_{\boldsymbol{i}} \hat{\boldsymbol{u}}_{\boldsymbol{i}} - \gamma^2 \hat{\boldsymbol{w}}_{\boldsymbol{i}}^{\boldsymbol{T}} \hat{\boldsymbol{w}}_{\boldsymbol{i}}) d\tau \tag{5.17}$$

where $\boldsymbol{\mathcal{Q}_i} \geq 0$ and $\boldsymbol{R_i} > 0$ are symmetric design matrices with appropriate dimensions. We assume that $\boldsymbol{S_i}$ is selected such that the pair $(\boldsymbol{A}, \sqrt{\boldsymbol{\mathcal{Q}_i}})$ is observable.

Now the $\mathcal{H}_\infty$ control problem can be solved by treating $\hat{\boldsymbol{u}}_{\boldsymbol{i}}$ as a minimizing player, whereas $\hat{\boldsymbol{w}}_{\boldsymbol{i}}$ as a maximizing player of the cost (5.17). Then, the game can be formulated as

$$\begin{aligned}\mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}) \triangleq \mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}^*, \hat{\boldsymbol{w}}_{\boldsymbol{i}}^*) &= \min_{\hat{\boldsymbol{u}}_{\boldsymbol{i}}} \max_{\hat{\boldsymbol{w}}_{\boldsymbol{i}}} \mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) \\ &= \max_{\hat{\boldsymbol{w}}_{\boldsymbol{i}}} \min_{\hat{\boldsymbol{u}}_{\boldsymbol{i}}} \mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}})\end{aligned} \tag{5.18}$$

where $\mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}})$ denotes the value functional such that

$$\mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}) = \int_t^\infty (\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{Q}}_{\boldsymbol{i}} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \frac{1}{c_i \lambda_i} \hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{R}_{\boldsymbol{i}} \hat{\boldsymbol{u}}_{\boldsymbol{i}} - \gamma^2 \hat{\boldsymbol{w}}_{\boldsymbol{i}}^{\boldsymbol{T}} \hat{\boldsymbol{w}}_{\boldsymbol{i}}) d\tau, \tag{5.19}$$

and the pair $(\hat{\boldsymbol{u}}_{\boldsymbol{i}}^*, \hat{\boldsymbol{w}}_{\boldsymbol{i}}^*)$ is the local game theoretic saddle point (local game Nash equilibrium).

**Lemma 5.** *The pair $(\hat{\boldsymbol{u}}_{\boldsymbol{i}}^*, \hat{\boldsymbol{w}}_{\boldsymbol{i}}^*)$ constitutes a Nash equilibrium of the game (5.18) if*

$$\hat{\boldsymbol{u}}_{\boldsymbol{i}}^* = -c_i \lambda_i \boldsymbol{R}_{\boldsymbol{i}}^{-\mathbf{1}} \boldsymbol{B}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}, \tag{5.20}$$

$$\hat{\boldsymbol{w}}_{\boldsymbol{i}}^* = \gamma^{-2} \boldsymbol{D}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}, \tag{5.21}$$

*where $\boldsymbol{\mathcal{P}}_{\boldsymbol{i}}$ is the solution of corresponding Game Algebraic Riccati Equation (GARE) such that*

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{i}} \boldsymbol{A} + \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} - c_i \lambda_i \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} \boldsymbol{B} \boldsymbol{R}_{\boldsymbol{i}}^{-\mathbf{1}} \boldsymbol{B}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} + \boldsymbol{\mathcal{Q}}_{\boldsymbol{i}} + \gamma^{-2} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} \boldsymbol{D} \boldsymbol{D}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} = \mathbf{0}. \tag{5.22}$$

*Proof*: Begin with deriving the Hamiltonian to solve the minimizing & maximizing extrema that satisfies the Nash condition (5.18) as

$$\begin{aligned}\mathcal{H}_i(\nabla \mathcal{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) = \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{Q}}_{\boldsymbol{i}} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \frac{1}{c_i \lambda_i} \hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{R}_{\boldsymbol{i}} \hat{\boldsymbol{u}}_{\boldsymbol{i}} - \gamma^2 \hat{\boldsymbol{w}}_{\boldsymbol{i}}^{\boldsymbol{T}} \hat{\boldsymbol{w}}_{\boldsymbol{i}} \\ + \nabla \mathcal{V}_i^T (\boldsymbol{A} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \boldsymbol{B} \hat{\boldsymbol{u}}_{\boldsymbol{i}} + \boldsymbol{D} \hat{\boldsymbol{w}}_{\boldsymbol{i}}) = 0\end{aligned} \tag{5.23}$$

where $\nabla \mathcal{V}_i = \partial \mathcal{V}_i / \partial \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}$ is the co-state vector and the boundary condition is $\mathcal{V}_i(0) = 0$. Using the quadratic form $\mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}) = \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}$ where $\boldsymbol{\mathcal{P}}_{\boldsymbol{i}} = \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}^{\boldsymbol{T}}$, and applying the stationarity conditions $\partial \mathcal{H}_i(\nabla \mathcal{V}_i^*, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) / \partial \hat{\boldsymbol{u}}_{\boldsymbol{i}} = \mathbf{0}$ and $\partial \mathcal{H}_i(\nabla \mathcal{V}_i^*, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) / \partial \hat{\boldsymbol{w}}_{\boldsymbol{i}} = \mathbf{0}$ yields the optimal control and disturbance respectively as (5.20) and (5.21). Additionally, substituting (5.20) and (5.21) into (5.23) and equating resultant HJI equation to zero, gives the Riccati equation (5.22).

Note that the sign of Hessians, $\partial^2 \mathcal{H}_i(\nabla \mathcal{V}_i^*, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) / \partial \hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\mathbf{2}} > 0$ and $\partial^2 \mathcal{H}_i(\nabla \mathcal{V}_i^*, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) / \partial \hat{\boldsymbol{w}}_{\boldsymbol{i}}^{\mathbf{2}} < 0$, along with the unboundedness of limits $\lim_{d \to \infty} \mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{*}, \hat{\boldsymbol{w}}_{\boldsymbol{i}})$, $\lim_{\hat{u}_i \to \infty} \mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}^{*})$ indeed show that (5.20) and (5.21) are the global optimal minimizing and maximizing extrema respectively. This indeed verifies that the pair $(\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{*}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}^{*})$ denotes the Nash equilibrium, which completes the proof. □

**Remark 20.** *The main purpose of introducing the local game (5.18 is to find the worst case disturbance for the $i^{th}$ follower (5.1). Since $i^{th}$ player is responsible for acting according to not only the worst case disturbance but also strategy of the $j^{th}$ player, $\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{*}$ (5.20) is not the finalized control strategy yet.*

**Corrolary 2.** *Let the distributed $N$ systems (5.14) experience the worst-case disturbance derived in (5.21), i.e., $\hat{\boldsymbol{w}}_{\boldsymbol{i}} = \hat{\boldsymbol{w}}_{\boldsymbol{i}}^{*}$, and introduce a new matrix $\boldsymbol{F} = \boldsymbol{A} + \gamma^{-2} \boldsymbol{D} \boldsymbol{D}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}$ to facilitate the analysis. Then, the transformed local output containment error dynamics become*

$$\begin{aligned} \dot{\hat{\boldsymbol{\delta}}}_{\boldsymbol{i}} &= (\boldsymbol{A} - c_i \lambda_i \boldsymbol{B} \boldsymbol{K} \boldsymbol{C} + \gamma^{-2} \boldsymbol{D} \boldsymbol{D}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}) \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} \\ &= \boldsymbol{F} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \boldsymbol{B} \hat{\boldsymbol{u}}_{\boldsymbol{i}} \\ \hat{\boldsymbol{\xi}}_{\boldsymbol{i}} &= -\lambda_i \boldsymbol{C} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} \end{aligned} \tag{5.24}$$

*as following is the natural outcome of using transformations* $\hat{\boldsymbol{\delta}} = (\boldsymbol{T} \otimes \boldsymbol{I_n})\boldsymbol{\delta}$ *and* $\boldsymbol{w^*} = (\boldsymbol{T^{-1}} \otimes \boldsymbol{I_p})\hat{\boldsymbol{w}}^*$ *on (5.21) such that*

$$\hat{\boldsymbol{w}}^* = \gamma^{-2}(\boldsymbol{I_N} \otimes \boldsymbol{D^T \mathcal{P}_i})\hat{\boldsymbol{\delta}}. \tag{5.25}$$

### 5.2.2 Global Graphical Game Solution

This section proposes a modified cost functional for the graphical game whose players are the nodes of $\mathrm{G^f}$. The main challenge in graphical games, is to design distributed Nash equilibrium control strategies, which cannot be achieved with the traditional quadratic cost functional formulation [23]. Therefore, a modified cost functional that provides both Nash and distributed control strategies in the sense that each follower uses the state information of its own and neighbors can be defined such that

$$J_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}) = \int_0^\infty \sum_{k=1}^{N} (\hat{\boldsymbol{\delta}}_{\boldsymbol{ik}}^{\boldsymbol{T}} \boldsymbol{Q_{ik}} \hat{\boldsymbol{\delta}}_{\boldsymbol{ik}} + \frac{1}{c_i \lambda_i} \hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{R_i} \hat{\boldsymbol{u}}_{\boldsymbol{i}} - \frac{a_{ik}^f}{(c_i \lambda_i)^2} \hat{\boldsymbol{u}}_{\boldsymbol{k}}^{\boldsymbol{T}} \boldsymbol{R_{ik}} \hat{\boldsymbol{u}}_{\boldsymbol{k}}) d\tau. \tag{5.26}$$

where $\hat{\boldsymbol{\delta}}_{\boldsymbol{ik}} = [\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}}, \hat{\boldsymbol{\delta}}_{\boldsymbol{k}}^{\boldsymbol{T}}]^T$, $\boldsymbol{Q_{ik}} = [\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}, \boldsymbol{0_{n\times n}}; \boldsymbol{0_{n\times n}}, \hat{\boldsymbol{Q}}_{\boldsymbol{ik}}]$, and $\hat{\boldsymbol{u}}_{\boldsymbol{-i}}$ is the set of control strategies of the $k^{th}$ player in $\mathrm{G^f}$ where $k \neq i$. Then, the corresponding game is defined as

$$V_i^g(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}, \hat{\boldsymbol{\delta}}_{\boldsymbol{-i}}) \triangleq J_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{g}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}^{\boldsymbol{g}}) \leq J_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}^{\boldsymbol{g}}) \tag{5.27}$$

where the N-tuple $\{V_1^g(\hat{\boldsymbol{\delta}}_{\boldsymbol{1}}, \hat{\boldsymbol{\delta}}_{\boldsymbol{-1}}), ..., V_N^g(\hat{\boldsymbol{\delta}}_{\boldsymbol{N}}, \hat{\boldsymbol{\delta}}_{\boldsymbol{-N}})\}$ is called the global Nash equilibrium outcome and the N-tuple $\{\hat{\boldsymbol{u}}_{\boldsymbol{1}}^{\boldsymbol{g}}, ..., \hat{\boldsymbol{u}}_{\boldsymbol{N}}^{\boldsymbol{g}}\}$ denotes the Nash equilibrium strategies of the game (5.27).

**Remark 21.** *A distributed Nash equilibrium strategy can be achieved if the functional $V_i^g(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}, \hat{\boldsymbol{\delta}}_{\boldsymbol{-i}})$ depends on the its argument $\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}$ solely, i.e., $V_i^g(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}, \hat{\boldsymbol{\delta}}_{\boldsymbol{-i}}) = V_i^g(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}})$. Otherwise, the resultant game optimal strategy would not be a distributed strategy.*

Define a quadratic function that will play a key role in the corresponding Hamiltonian derivation such that

$$V_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}, \hat{\boldsymbol{\delta}}_{\boldsymbol{-i}}) \triangleq V_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}). \tag{5.28}$$

By the Remark 21, the following holds

$$\begin{aligned} \frac{\partial \tilde{V}_i^T}{\partial \hat{\boldsymbol{\delta}}_{\boldsymbol{ik}}} \dot{\hat{\boldsymbol{\delta}}}_{\boldsymbol{ik}} &= \begin{bmatrix} \frac{\partial \tilde{V}_i^T}{\partial \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}} & \frac{\partial \tilde{V}_i^T}{\partial \hat{\boldsymbol{\delta}}_{\boldsymbol{k}}} \end{bmatrix}^T \begin{bmatrix} \dot{\hat{\boldsymbol{\delta}}}_{\boldsymbol{i}} \\ \dot{\hat{\boldsymbol{\delta}}}_{\boldsymbol{k}} \end{bmatrix} \\ &= \frac{\partial \tilde{V}_i^T}{\partial \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}} \left( \boldsymbol{F} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \boldsymbol{B} \hat{\boldsymbol{u}}_{\boldsymbol{i}} \right). \end{aligned} \tag{5.29}$$

**Theorem 9.** *Given the cost functional (5.26), local output containment error dynamics and quadratic form (5.28). Assume that the element of matrix $\boldsymbol{Q}_{\boldsymbol{ik}}$ satisfies*

$$\hat{\boldsymbol{Q}}_{\boldsymbol{ik}} = a_{ik}^f \tilde{\boldsymbol{P}}_{\boldsymbol{k}} \boldsymbol{B} \boldsymbol{R}_{\boldsymbol{k}}^{-\mathbf{1}} \boldsymbol{R}_{\boldsymbol{ik}} \boldsymbol{R}_{\boldsymbol{k}}^{-\mathbf{1}} \boldsymbol{B}^{\boldsymbol{T}} \tilde{\boldsymbol{P}}_{\boldsymbol{k}} \tag{5.30}$$

*where $\tilde{\boldsymbol{P}}_{\boldsymbol{i}}$ and $\tilde{\boldsymbol{P}}_{\boldsymbol{k}}$ solve the corresponding algebraic Riccati equations such that*

$$\boldsymbol{F}^{\boldsymbol{T}} \tilde{\boldsymbol{P}}_{\boldsymbol{i}} + \tilde{\boldsymbol{P}}_{\boldsymbol{i}} \boldsymbol{F} + \tilde{\boldsymbol{Q}}_{\boldsymbol{i}} - c_i \lambda_i \tilde{\boldsymbol{P}}_{\boldsymbol{i}} \boldsymbol{B} \boldsymbol{R}_{\boldsymbol{i}}^{-\mathbf{1}} \boldsymbol{B}^{\boldsymbol{T}} \tilde{\boldsymbol{P}}_{\boldsymbol{i}} = \mathbf{0}, \tag{5.31}$$

$$\boldsymbol{F}^{\boldsymbol{T}} \tilde{\boldsymbol{P}}_{\boldsymbol{k}} + \tilde{\boldsymbol{P}}_{\boldsymbol{k}} \boldsymbol{F} + \tilde{\boldsymbol{Q}}_{\boldsymbol{k}} - c_k \lambda_k \tilde{\boldsymbol{P}}_{\boldsymbol{k}} \boldsymbol{B} \boldsymbol{R}_{\boldsymbol{k}}^{-\mathbf{1}} \boldsymbol{B}^{\boldsymbol{T}} \tilde{\boldsymbol{P}}_{\boldsymbol{k}} = \mathbf{0}. \tag{5.32}$$

*Then, the Nash equilibrium strategy for $i^{th}$ follower in the graphical game (5.27) takes the following form*

$$\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{g}} = -c_i \lambda_i \boldsymbol{R}_{\boldsymbol{i}}^{-\mathbf{1}} \boldsymbol{B}^{\boldsymbol{T}} \tilde{\boldsymbol{P}}_{\boldsymbol{i}} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}. \tag{5.33}$$

*Proof*: This proof consists of two parts. The first part deals with the selection of $\boldsymbol{Q_{ik}}$ to satisfy (5.28) and the second part rigorously analyzes the Nash equilibrium property of minimizing controls $\hat{\boldsymbol{u}}_{\boldsymbol{i}}$ derived in the first part.

*Selecting* $\boldsymbol{Q_{ik}}$: By (5.29), the Hamiltonian associated with the cost functional (5.26) is

$$\begin{aligned} H_i(\nabla\tilde{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}) = \sum_{k=1}^{N} \Big(\nabla\tilde{V}_i^T(\boldsymbol{F}\hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \boldsymbol{B}\hat{\boldsymbol{u}}_{\boldsymbol{i}})\Big) \\ + \sum_{k=1}^{N}(\hat{\boldsymbol{\delta}}_{\boldsymbol{ik}}^{\boldsymbol{T}}\boldsymbol{Q_{ik}}\hat{\boldsymbol{\delta}}_{\boldsymbol{ik}} + \frac{1}{c_i\lambda_i}\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{T}}\boldsymbol{R_i}\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \frac{a_{ik}^f}{(c_i\lambda_i)^2}\hat{\boldsymbol{u}}_{\boldsymbol{k}}^{\boldsymbol{T}}\boldsymbol{R_{ik}}\hat{\boldsymbol{u}}_{\boldsymbol{k}}) = 0 \end{aligned} \tag{5.34}$$

where $\nabla\tilde{V}_i = \partial\tilde{V}_i/\partial\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}$ with the boundary condition $\tilde{V}_i(\boldsymbol{0}) = \boldsymbol{0}$. To find the best responses, check the stationarity condition $\partial H_i(\nabla\tilde{V}_i^g, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}})/\partial\hat{\boldsymbol{u}}_{\boldsymbol{i}} = \boldsymbol{0}$, which yields

$$\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{g}} = -\frac{c_i\lambda_i}{2}\boldsymbol{R_i^{-1}}\boldsymbol{B^T}\nabla\tilde{V}_i^g \tag{5.35}$$

Substituting (5.35) into (5.34) yields the Hamilton-Jacobi (HJ) equation, i.e., $H_i(\nabla\tilde{V}_i^g, \hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{g}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}^{\boldsymbol{g}}) = 0$. Additionally, using the quadratic form of $\tilde{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}})$ (5.28) in the HJ equation, one obtains

$$\begin{aligned} &H_i(\nabla\tilde{V}_i^g, \hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{g}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}^{\boldsymbol{g}}) = \\ &\textstyle\sum_{k=1}^{N}\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}}(\boldsymbol{F^T}\tilde{\boldsymbol{P}}_{\boldsymbol{i}} + \tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{F} + \tilde{\boldsymbol{Q}}_{\boldsymbol{i}} - c_i\lambda_i\tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{B}\boldsymbol{R_i^{-1}}\boldsymbol{B^T}\tilde{\boldsymbol{P}}_{\boldsymbol{i}})\hat{\boldsymbol{\delta}}_{\boldsymbol{k}} \\ &+ \textstyle\sum_{k=1}^{N}\hat{\boldsymbol{\delta}}_{\boldsymbol{k}}^{\boldsymbol{T}}(\hat{\boldsymbol{Q}}_{\boldsymbol{ik}} - a_{ik}^f\tilde{\boldsymbol{P}}_{\boldsymbol{k}}\boldsymbol{B}\boldsymbol{R_k^{-1}}\boldsymbol{R_{ik}}\boldsymbol{R_k^{-1}}\boldsymbol{B^T}\tilde{\boldsymbol{P}}_{\boldsymbol{k}})\hat{\boldsymbol{\delta}}_{\boldsymbol{k}} = 0. \end{aligned} \tag{5.36}$$

This gives the element of matrix $\boldsymbol{Q_{ik}}$ (5.30), and the Riccati equations (5.31)-(5.32), which completes the first part of the proof.

*Graphical Nash Equilibrium*: In this part, we prove that the N-tuple $\{\hat{\boldsymbol{u}}_{\boldsymbol{1}}^{\boldsymbol{g}}, ..., \hat{\boldsymbol{u}}_{\boldsymbol{N}}^{\boldsymbol{g}}\}$ indeed constitutes the Nash equilibrium strategies of the game (5.27). Realize that substituting the quadratic form (5.28) into (5.35) yields (5.33).

Now re-write the Hamiltonian (5.34) as

$$H_i(\nabla\tilde{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}) = \sum_{k=1}^{N} H_{ik}(\nabla\tilde{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}) \tag{5.37}$$

where $H_{ik}(\nabla\tilde{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}})$ can be expressed by completing the squares as

$$\begin{aligned} &H_{ik}(\nabla\tilde{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}) = H_{ik}(\nabla\tilde{V}_i, \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{-i}}) \\ &+\frac{1}{c_i\lambda_i}(\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}})^T \boldsymbol{R_i}(\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}}) - \frac{a^f_{ik}}{(c\lambda_k)^2}(\hat{\boldsymbol{u}}_{\boldsymbol{k}} - \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{k}})^T \boldsymbol{R_{ik}}(\hat{\boldsymbol{u}}_{\boldsymbol{k}} - \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{k}}) \\ &-\frac{2a^f_{ik}}{(c\lambda_k)^2}\hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{k}\prime}\boldsymbol{R_{ik}}(\hat{\boldsymbol{u}}_{\boldsymbol{k}} - \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{k}}) \end{aligned} \tag{5.38}$$

since the equality $\nabla\tilde{V}_i^T \boldsymbol{B}\boldsymbol{u}^{\boldsymbol{g}}_{\boldsymbol{i}} = -\frac{2}{c}(\hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}})^T \boldsymbol{R_i}\hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}}$ holds by (5.35). Having the fact that the Hamiltonian (5.34) is a differential equivalent of the cost functional (5.26), we can write

$$J_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}}) = \int_{t_0}^{\infty} \sum_{k=1}^{N} H_{ik}(\nabla\tilde{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{\boldsymbol{-i}})dt + V_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(t_0)). \tag{5.39}$$

According to rules of the game (5.27), select $\hat{\boldsymbol{u}}_{\boldsymbol{k}} = \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{k}}$ and use the Hamiltonian form (5.38) in (5.39) to obtain

$$J_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{-i}}) = \int_{t_0}^{\infty} \sum_{k=1}^{N} \frac{1}{c_i\lambda_i}(\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}})^T \boldsymbol{R_i}(\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}})dt + V_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(t_0)), \tag{5.40}$$

which clearly satisfies the game condition $J_i(\hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{-i}}) \leq J_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{-i}})$, and hence the N-tuple $\{\hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{1}}, ..., \hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{N}}\}$ is indeed the Nash equilibrium of the game (5.27). This completes the second part of the proof. □

**Remark 22.** *Realize that the local $\mathcal{H}_\infty$ Nash control strategy $\hat{\boldsymbol{u}}^*_{\boldsymbol{i}}$ (5.20) does not have the same form as the global graphical Nash control $\hat{\boldsymbol{u}}^{\boldsymbol{g}}_{\boldsymbol{i}}$ (5.33). Therefore, we have not obtained a control strategy that belongs to Nash equilibrium for both games (5.18) and*

*(5.27). The next Theorem 10 gives the necessary conditions to provide a dual Nash control strategy by selecting appropriate* $\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}$ *and* $\boldsymbol{R}_{\boldsymbol{ik}}$ *design matrices.*

**Theorem 10.** *The Nash equilibrium control strategy* $\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{*}$ *(5.20) for the local* $\mathcal{H}_\infty$ *game (5.18) stands for the same Nash equilibrium control strategy* $\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{g}}$ *(5.33) for the global graphical game (5.27) if the design matrices* $\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}$ *and* $\boldsymbol{R}_{\boldsymbol{ik}}$ *are selected as*

$$\tilde{\boldsymbol{Q}}_{\boldsymbol{i}} = -\gamma^{-2}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}} + \boldsymbol{\mathcal{Q}}_{\boldsymbol{i}} \tag{5.41}$$

$$\boldsymbol{R}_{\boldsymbol{ik}} = N^2\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}}\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}}\boldsymbol{B} + \tilde{\boldsymbol{R}}_{\boldsymbol{ik}} \tag{5.42}$$

*where* $\tilde{\boldsymbol{R}}_{\boldsymbol{ik}} > 0$.

*Proof*: First we work on selection of the design matrix $\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}$. Begin with expanding the Riccati equation (5.31) using $\boldsymbol{F} = \boldsymbol{A} + \gamma^{-2}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}}$, which yields

$$\begin{aligned}\tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{A} + \boldsymbol{A}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}} - c_i\lambda_i\tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{B}\boldsymbol{R}_{\boldsymbol{i}}^{-\boldsymbol{1}}\boldsymbol{B}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}} + \tilde{\boldsymbol{Q}}_{\boldsymbol{i}} \\ + \gamma^{-2}(\tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}} + \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}}) = \boldsymbol{0}.\end{aligned} \tag{5.43}$$

Select $\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}$ as in (5.41), and assume that the positive definite matrix $\boldsymbol{\mathcal{P}}_{\boldsymbol{i}}$ solves the local $\mathcal{H}_\infty$ GARE (5.22). Then, to verify that it also solves the graphical game Riccati equation (5.43), it is required to obtain the equality $\tilde{\boldsymbol{P}}_{\boldsymbol{i}} = \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}$ where $\tilde{\boldsymbol{P}}_{\boldsymbol{i}}$ is the solution of Graphical Riccati equation (5.43), illustrated in the following realization

$$\begin{aligned}(\tilde{\boldsymbol{P}}_{\boldsymbol{i}} - \boldsymbol{\mathcal{P}}_{\boldsymbol{i}})\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}(\tilde{\boldsymbol{P}}_{\boldsymbol{i}} - \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}) = \tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}} + \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}} \\ - \tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{\mathcal{P}}_{\boldsymbol{i}} - \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}}.\end{aligned} \tag{5.44}$$

Thereby (5.43) can be expressed in terms of $\tilde{\boldsymbol{P}}_{\boldsymbol{i}}$ with selected $\tilde{\boldsymbol{Q}}_i$ (5.41 as

$$\tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{A} + \boldsymbol{A}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}} - c_i\lambda_i\tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{B}\boldsymbol{R}_{\boldsymbol{i}}^{-\boldsymbol{1}}\boldsymbol{B}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}} + \boldsymbol{\mathcal{Q}}_{\boldsymbol{i}} + \gamma^{-2}\tilde{\boldsymbol{P}}_{\boldsymbol{i}}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}\tilde{\boldsymbol{P}}_{\boldsymbol{i}} = \boldsymbol{0}. \tag{5.45}$$

Realize that (5.45) has a dual form of (5.22). Therefore, the Nash equilibrium control strategies $\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{*}$ (5.20), and $\hat{\boldsymbol{u}}_{\boldsymbol{i}}^{\boldsymbol{g}}$ (5.33) are the same as each other since the Riccati equation (5.22) has a unique positive definite solution under some conditions that will be detailed in Section 5.3. Additionally, setting the disturbance matrix $\boldsymbol{D} = \boldsymbol{0}$ guarantees the equality $\tilde{\boldsymbol{P}}_{\boldsymbol{i}} = \boldsymbol{\mathcal{P}}_{\boldsymbol{i}}$ as $\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}$ becomes $\boldsymbol{\mathcal{Q}}_{\boldsymbol{i}}$ and (5.43) reduces to (5.45).

To complete the proof, the weighting matrix $\boldsymbol{Q}_{\boldsymbol{ik}}$ in (5.26 should be at least positive semi-definite with the selected design matrices (5.30) and (5.41)-(5.42). This can be achieved if both $\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}$ and $\hat{\boldsymbol{Q}}_{\boldsymbol{ik}}$ are at least positive semi-definite matrices by applying the Schur complement method. Since $\hat{\boldsymbol{Q}}_{\boldsymbol{ik}}$ is clearly positive semi-definite, we conclude that $\boldsymbol{\mathcal{Q}}_{\boldsymbol{i}}$ must be selected large enough to make $\tilde{\boldsymbol{Q}}_{\boldsymbol{i}}$ positive semi-definite as well. This completes the proof. □

**Remark 23.** *Theorem 10 implies that $i^{th}$ player can minimize its global cost (5.26) by only playing the local $\mathcal{H}_\infty$ (5.18) game. Hence, if the $i^{th}$ player stabilizes the local error dynamics (5.14) with the Nash equilibrium control strategy (5.20), then both of the local and global objectives can be satisfied simultaneously.*

## 5.3 Stability and $\mathcal{L}_2$ gain bound analysis with output feedback

This section re-formulates the control strategy (local and global) derived in the Section 5.2 using the static output feedback method, and establishes corresponding $\mathcal{L}_2$ gain bound by a prescribed attenuation level.

**Remark 24.** *Given the design parameters (5.30) and (5.41)-(5.42), the local $\mathcal{H}_\infty$ game (5.18) solution and global graphical game (5.27) are indeed the same as each other by Theorem 10. Therefore, the stability and $\mathcal{L}_2$ gain bound analysis conducted in this section, is valid for both games.*

Now, realize that the OPFB of $i^{th}$ follower's control (5.13) can be re-written in the transformed coordinates as

$$\hat{\boldsymbol{u}}_{\boldsymbol{i}} = -c_i \lambda_i \boldsymbol{K_i C} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}. \tag{5.46}$$

The next main theorem is a key to find the OPFB matrix $\boldsymbol{K_i^*}$.

**Theorem 11.** *Given Assumptions 6-8, the undisturbed system (5.14), i.e.,* $\hat{\boldsymbol{w}}_{\boldsymbol{i}} = \boldsymbol{0}$*, is asymptotically stable if*

$$c_i = \frac{1}{\lambda_i}; \quad \boldsymbol{B R_i^{-1} B^T} \geq \gamma^{-2} \boldsymbol{D D^T} \tag{5.47}$$

*where* $\gamma^*$ *is the critical attenuation level [102]. Furthermore, the disturbed system (5.14) is* $L_2$ *gain bounded by* $\gamma$ *and asymptotically stable using the control* $\hat{\boldsymbol{u}}_{\boldsymbol{i}} = c \boldsymbol{K_i^*} \hat{\boldsymbol{\xi}}_{\boldsymbol{i}}$ *(5.46) and the disturbance* $\hat{\boldsymbol{w}}_{\boldsymbol{i}} = -\boldsymbol{N_i^*} \hat{\boldsymbol{\xi}}_{\boldsymbol{i}}$ *where*

$$\boldsymbol{K_i^*} = \boldsymbol{R_i^{-1} B^T \mathcal{P}_i C^+} \tag{5.48}$$

$$\boldsymbol{N_i^*} = \gamma^{-2} \boldsymbol{D^T \mathcal{P}_i C^+} \tag{5.49}$$

*if the following sufficient condition is satisfied*

$$c_i = \frac{1}{\lambda_i}; \quad \boldsymbol{B R_i^{-1} B^T} \geq 2\gamma^{-2} \boldsymbol{D D^T}. \tag{5.50}$$

*Proof*: Herein, we first derive full state feedback gains and then explain how to apply them to the OPFB design. The proof consists of two parts, we first prove the $L_2$ gain bound condition (5.12) and then asymptotic stability of the equilibrium origin considering the system (5.14) without and with the worst-case disturbance.

*$L_2$ gain bound condition*: Select $c_i$ as given in (5.50) and re-write the Hamiltonian (5.23) by completing the squares as

$$\begin{aligned}\mathcal{H}_i(\nabla\mathcal{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{d}}_{\boldsymbol{i}}) =& \mathcal{H}_i(\nabla\mathcal{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*, \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*) + (\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*)^T R_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*) \\ & - \gamma^{-2}(\hat{\boldsymbol{d}}_{\boldsymbol{i}} - \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*)^T(\hat{\boldsymbol{d}}_{\boldsymbol{i}} - \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*). \end{aligned} \tag{5.51}$$

Note that the HJI equation $\mathcal{H}_i(\nabla\mathcal{V}_i, \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*, \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*) = 0$ holds with the boundary condition $\mathcal{V}_i(\mathbf{0}) = 0$. Then, the objective functional (5.17) can be re-expressed as

$$\begin{aligned}\mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) = \int_0^\infty \Big( & (\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*)^T \boldsymbol{R}_{\boldsymbol{i}}(\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*) \\ & - \gamma^{-2}(\hat{\boldsymbol{d}}_{\boldsymbol{i}} - \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*)^T(\hat{\boldsymbol{d}}_{\boldsymbol{i}} - \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*)\Big) dt + \mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(0)). \end{aligned} \tag{5.52}$$

Realize that the condition $\mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) \leq \beta$ implies that the non-expansivity constraint (5.12) holds. Select $\beta = \mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(0))$, $\hat{\boldsymbol{u}}_{\boldsymbol{i}} = \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*$. Then, (5.52) reduces to

$$\begin{aligned}\mathcal{J}_i(\hat{\boldsymbol{u}}_{\boldsymbol{i}}^*, \hat{\boldsymbol{w}}_{\boldsymbol{i}}) &= -\int_0^\infty \gamma^{-2}(\hat{\boldsymbol{d}}_{\boldsymbol{i}} - \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*)^T(\hat{\boldsymbol{d}}_{\boldsymbol{i}} - \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*) dt + \mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(0)) \\ &\leq \mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(0)), \quad \forall \hat{\boldsymbol{d}}_{\boldsymbol{i}} \in (0, \infty). \end{aligned} \tag{5.53}$$

This proves that the $L_2$ gain bound condition (5.12) holds with $\beta = \mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(0))$, $\hat{\boldsymbol{u}}_{\boldsymbol{i}} = \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*$. Additionally, the value of game (5.18) with $\hat{\boldsymbol{u}}_{\boldsymbol{i}} = \hat{\boldsymbol{u}}_{\boldsymbol{i}}^*$ and $\hat{\boldsymbol{d}}_{\boldsymbol{i}} = \hat{\boldsymbol{d}}_{\boldsymbol{i}}^*$ is $\mathcal{V}_i(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}(0))$ by (5.52).

*Asymptotic stability (5.14)*: To verify asymptotic stability of the undisturbed system (5.14), we consider the algebraic Riccati equation (5.45) and benefit from gain margin $[c_{lower}, \infty)$ with $c_{lower} < \frac{1}{2}$ property of the $\mathcal{H}_\infty$ control[74]. Note that the $\mathcal{H}_\infty$ has gain margin less than $\frac{1}{2}$ by Chapter 10 in [74] but the lower bound $c_{lower}$ is not precisely defined, and hence the conditions given in (5.47)-(5.50) are only sufficient.

With this in mind, if $\boldsymbol{K_i^* C} = \boldsymbol{R_i^{-1} B^T \mathcal{P}_i}$ and $c_i = \frac{1}{\lambda_i}$, the control (5.46) becomes $\hat{\boldsymbol{u}}_{\boldsymbol{i}} = -\boldsymbol{R_i^{-1} B^T \mathcal{P}_i \hat{\delta}_i}$.

Now, consider the Lyapunov function candidate $\mathcal{V}(\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}) = \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}_i} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}$. Realize that $\boldsymbol{\mathcal{P}_i}$ is the unique positive definite solution of GARE (5.22) if the condition $BR^{-1}B^T \geq \gamma^{-2} DD^T$ is satisfied. This is illustrated in the following realization

$$\begin{aligned}
\dot{\mathcal{V}} &= \dot{\hat{\boldsymbol{\delta}}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}_i} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} + \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{P}_i} \dot{\hat{\boldsymbol{\delta}}}_{\boldsymbol{i}} \\
&= \hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \big( - \boldsymbol{\mathcal{P}_i B R_i^{-1} B^T \mathcal{P}_i} - \boldsymbol{\mathcal{Q}_i} + \gamma^{-2} \boldsymbol{\mathcal{P}_i D D^T \mathcal{P}_i} \big) \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} \\
&\leq -\hat{\boldsymbol{\delta}}_{\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{\mathcal{Q}_i} \hat{\boldsymbol{\delta}}_{\boldsymbol{i}} \quad \Leftarrow \; \boldsymbol{B R_i^{-1} B^T \geq \gamma^{-2} D D^T}.
\end{aligned} \tag{5.54}$$

Then, the observability of $(\boldsymbol{A}, \sqrt{\boldsymbol{\mathcal{Q}_i}})$ verifies that the undisturbed system (5.14) is asymptotically stable. Notice that the condition $\boldsymbol{B R_i^{-1} B^T \geq \gamma^{-2} D D^T}$ only enables designer to prove asymptotic stability of the closed-loop matrix $(\boldsymbol{A} - c_i \lambda_i \boldsymbol{R_i^{-1} B^T \mathcal{P}_i})$ by solving GARE (5.22) [102]. To prove stability of the closed-loop matrix $(\boldsymbol{A} - c_i \lambda_i \boldsymbol{R_i^{-1} B^T \mathcal{P}_i} + \gamma^{-2} \boldsymbol{D^T \mathcal{P}_i \hat{\delta}_i})$ that considers $\hat{\boldsymbol{w}}_{\boldsymbol{i}} = \hat{\boldsymbol{w}}_{\boldsymbol{i}}^*$ (5.21), use the gain margin approach. Herein the disturbance gain $\boldsymbol{N_i^* C} = \gamma^{-2} \boldsymbol{D^T \mathcal{P}_i}$ is employed. According to the gain margin approach, the sufficient condition (5.50) is immediate. Futhermore, since the upper bound for gain margin is infinite, the strict equality conditions in (5.47)-(5.50) can be relaxed as inequaility conditions such that $c_i \geq \frac{1}{\lambda_i}$.

Lastly, we explain how to apply state feedback gains to the OPFB design. Note that if $\boldsymbol{C}$ is an invertible matrix, then all local errors (5.14) would be regulated optimally as $\boldsymbol{C}^+$ becomes $\boldsymbol{C^{-1}}$, and $\boldsymbol{K_i^* C} = \boldsymbol{R_i^{-1} B^T \mathcal{P}_i C^{-1} C}$ with $\boldsymbol{K_i^*}$ in (5.48). On the other hand, if it is not square but full row-rank, then only the states spanned by row space of the output matrix $\boldsymbol{C}$ would be regulated optimally given Assumptions 7-8, and the fact that $\boldsymbol{C^+ C}$ projects $\mathbb{R}^n$ onto the row space of $\boldsymbol{C}$. Additionally, the other states would converge to the origin given Assumption 7. Realize that Assumption 6

implies that there could be an unstable mode that is observable but does not belong to the row space of $\boldsymbol{C}$. Therefore, Assumptions 7 and 8 are indeed required to apply static state feedback gains to the OPFB design. Lastly, the system (5.14) is stable against the worst-case disturbance $\hat{\boldsymbol{w}}_{\boldsymbol{i}} = -\boldsymbol{N}_{\boldsymbol{i}}^{*}\hat{\boldsymbol{\xi}}_{\boldsymbol{i}}$ with $\boldsymbol{N}_{\boldsymbol{i}}^{*}$ in (5.49), which affects only the states that belong to the row space of $\boldsymbol{C}$ given the condition (5.50). This completes the proof. □

**Remark 25.** *Note that the condition* $\boldsymbol{B}\boldsymbol{R}_{\boldsymbol{i}}^{-\mathbf{1}}\boldsymbol{B}^{\boldsymbol{T}} \geq 2\gamma^{-2}\boldsymbol{D}\boldsymbol{D}^{\boldsymbol{T}}$ *is only sufficient condition for the stability of the disturbed system i.e., there may be a Nash gain solution which stabilizes (5.14) but does not satisfy* $BR^{-1}B^{T} \geq 2\gamma^{-2}DD^{T}$. *This is because of the critical attenuation level* $\gamma^{*}$ *that is introduced in [102]. According to the Chapter 2 of [102], the undisturbed system (5.14) is asymptotically stable when* $\gamma > \gamma^{*}$. *However, in that case, one may not achieve a positive definite solution* $P$ *for the GARE (5.22), which should be avoided by Theorem 10. On the other hand, Assumptions 6-9 are indeed necessary for the MLF game to have a stabilizing Nash solution.*

**Remark 26.** *Note that the proposed methods in this paper aim to regulate the output containment error vector* $\boldsymbol{\xi}$ *(5.5) instead of global state vector* $\boldsymbol{\delta}$ *(5.11). The reasoning behind this approach is not to deal with inner loop of the system dynamics of the agents for the graphical game (5.27) since they can be treated separately from the outer loop. An example for the inner loop is the attitude control loop of the Unmanned Aerial Vehicles (UAVs) [5]. Interested reader can examine our work [27] to observe how to design a controller to deal with the attitude and position control loops of the multi-quadrotor systems seperately. Herein, we only aim to control the position of the UAVs that correspond to the outer position loop, and the attitude controller is assumed be stable by Assumption 7.*

**Remark 27.** *If the graph* $\mathrm{G}^{\mathrm{f}}$ *is a tree with the root node being a well informed follower* $k$ *and each follower's connectivity weight in* $\boldsymbol{\mathcal{A}}^{\boldsymbol{f}}$ *is multiplied with* $M$ *and*

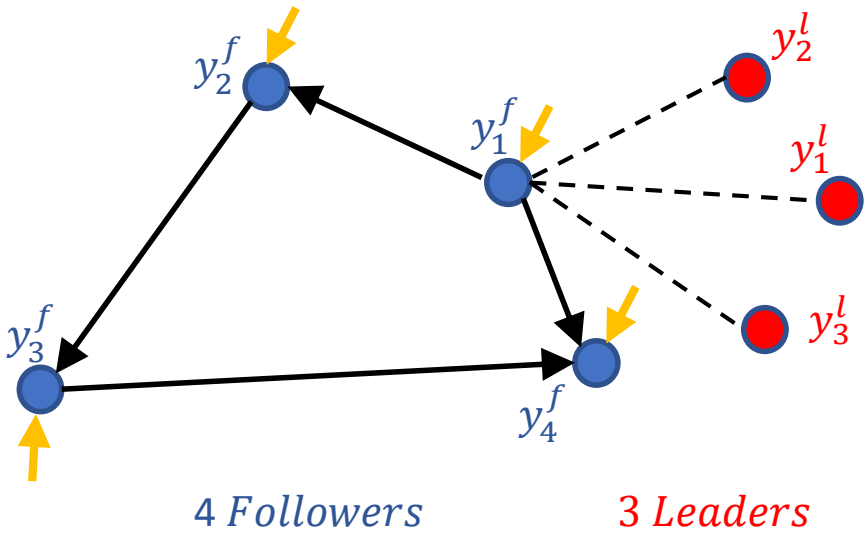


Figure 27: The communication graph of the four followers and three leaders MLF game.

*all pinning gains are multiplied with* $1/M$*, the pinned Laplacian matrix* $(\sum_{j=1}^{M} \boldsymbol{\psi_j})$ *would have all eigenvalues ones since the Laplacian matrix of the graph* $\mathrm{G^f}$*, i.e.,* $\boldsymbol{\mathcal{L}^f}$ *is nilpotent matrix [121] in this case. Then, the term* $\lambda_i$ *can be removed from the objective functionals (5.17) and (5.26), and hence from the corresponding Nash strategies (5.20) and (5.33). Therefore, the condition (5.47) would be modified as* $c = 1$.

## 5.4 Simulation results

In this section, we first introduce the multi-agent quadrotor Unmanned Aerial Vehicle (UAV) dynamics by using the backstepping control phenomena from our previous works [5] and [27]. Then, we apply the proposed methods to the MLF UAV game to verify correct performances of them.

It is well-known that the quadrotor linear model can be obtained around a flight condition that is called as hover where quadrotor's Body frame is aligned with the Earth (Inertial) frame (See Fig. 1 in [27] for frame illustrations and North-West-Up coordinate axis system convention). Then, the system matrices for linear model of the quadrotor's outer position & velocity control loop can be obtained as

$$\dot{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{f}} = \begin{bmatrix} \mathbf{0_{3\times 3}} & \boldsymbol{I_3} \\ \mathbf{0_{3\times 3}} & \mathbf{0_{3\times 3}} \end{bmatrix} \boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{f}} + \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ g & 0 & 0 & 0 \\ 0 & -g & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m} \end{bmatrix} \boldsymbol{u}_{\boldsymbol{i}}^{\boldsymbol{f}} \tag{5.55}$$

where $\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{f}} = [x_i \ y_i \ z_i \ u_i \ v_i \ w_i]^T$ is a vector of stacked position and velocity vectors, $\boldsymbol{u}_{\boldsymbol{i}}^{\boldsymbol{f}} = [\theta_i \ \phi_i \ \psi_i \ \mu_i]^T$ is a vector of stacked Euler angles and total thrust produced by the quadrotor, $g = 9.81 m/sec^2$ is the gravitational acceleration and $m = 0.467kg$ is the mass of the quadrotor UAV. To test the robustness, the disturbance matrix is selected as $\boldsymbol{D} = [0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$. In this case, the output matrix is an identity matrix, which means that the follower group is required to measure only position and velocity information from the leader group, and hence there is no need for the attitude information of any agent in the MLF UAV game. Notice that Assumptions 7 and 8 are naturally satisfied with this well partitioned UAV model.

The nonstandard backstepping control method in [27] aims to find the desired Euler angles that needs to be tracked by the attitude controller. Herein, we use the same convention but assume that the $\theta$ pitch angle and $\phi$ roll angle remains in the interval $[-\pi/6, \pi/6]$ to make sure that the quadrotor UAV shows a linear behaviour as small angle approximation is valid in this interval [5]. Furthermore, since the linear model of quadrotor is obtained around the hover flight condition, the input

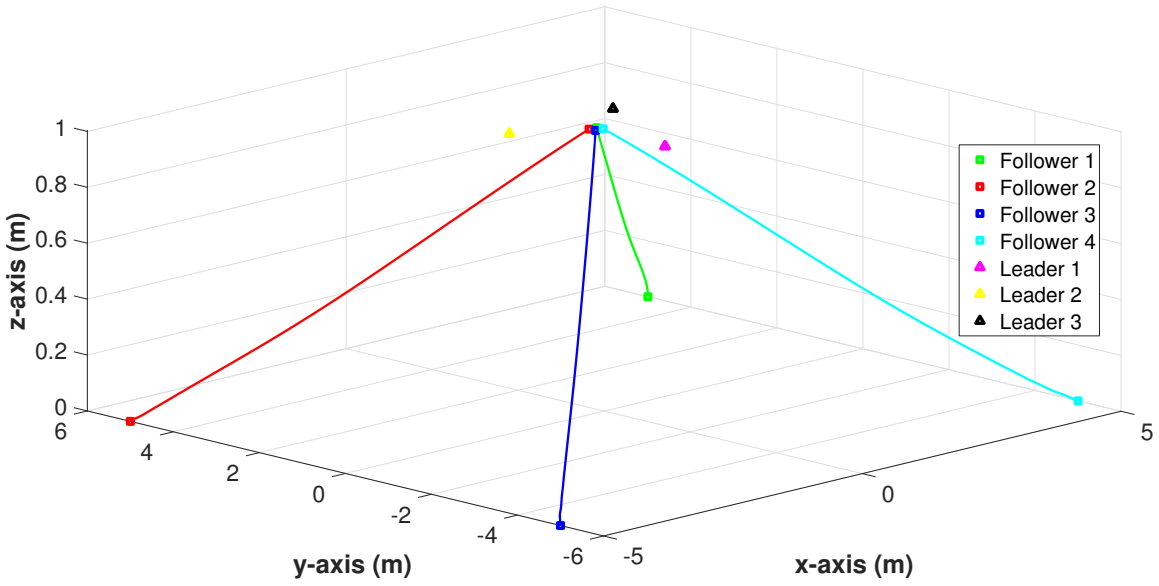


Figure 28: Path tracked by the UAVs in the MLF game.

parameters of (5.55) should be summed with the nominal flight condition parameters, i.e, $\boldsymbol{u}_{hover} = [\theta = 0 \ \phi = 0 \ \psi = c_\psi \ \mu = mg]^T$ before using them as set points for the attitude control loop. The attitude controller is taken from our work [27], and the proof of stability for the backstepping method is omitted.

The adjacency matrix $\boldsymbol{\mathcal{A}^f}$ that connects $N = 4$ followers or nodes of the graph $\mathrm{G^f}$ and the bipartite communication graph parameters $\boldsymbol{G_j^b}$ $j \in \{1, ..., M = 3\}$ that do not contradict with Assumption 9 are selected as

$$\boldsymbol{\mathcal{A}^f} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \ g_{11}^b = g_{12}^b = g_{13}^b = 1. \tag{5.56}$$

The MLF game communication graph is illustrated in Fig. 27. Note that the root node is selected as the first follower $\boldsymbol{y_1^f}$ according to the Assumption 9. There

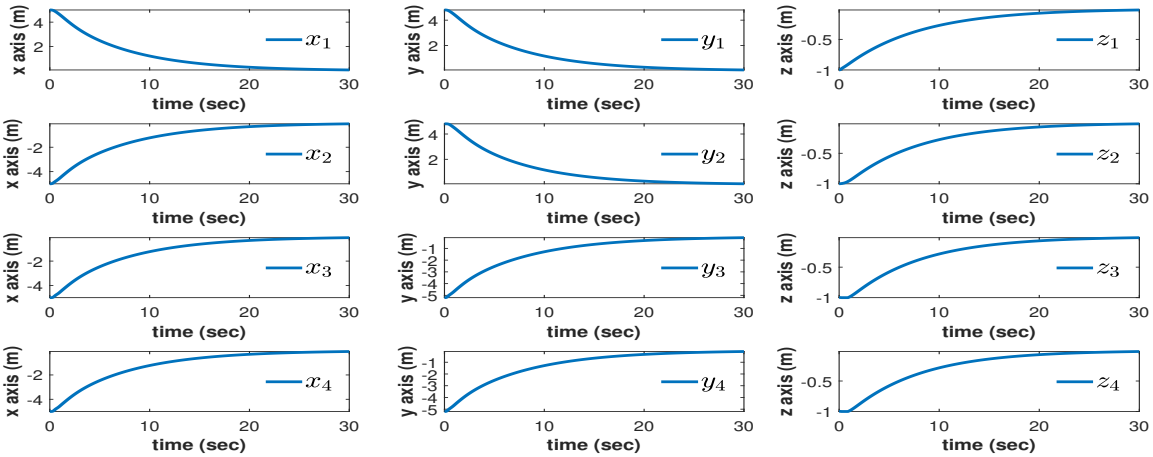


Figure 29: Output containment position error for each follower in three dimensional space.

is also no connection from uninformed followers $(\boldsymbol{y_2^f}, \boldsymbol{y_3^f}, \boldsymbol{y_4^f})$ to the well informed follower $\boldsymbol{y_1^f}$ and the follower graph is an acyclic digraph. The eigenvalues of the pinned Laplacian matrix $(\sum_{j=1}^{M} \boldsymbol{\psi_j})$ are $3.000, 0.667, 0.333, 0.333$ that are all positive real numbers as required.

Now, to verify the correct performance of the proposed methods, the control for each follower is selected as (5.13) and the worst-case disturbance is set to $\boldsymbol{w_1} = -\boldsymbol{N_1}\boldsymbol{\xi_1}$ for the well-informed follower UAV. Then, all seven quadrotor models implemented in the MATLAB Simulink. To derive the Nash strategy gain expressions $\boldsymbol{K_i}$ and $\boldsymbol{N_i}$, MATLAB's *icare* command is used. The objective functional parameters for (5.17) are selected as

$$\boldsymbol{\mathcal{Q}_i} = \begin{bmatrix} 0.1\boldsymbol{I_3} & \boldsymbol{0_3} \\ \boldsymbol{0_3} & 5\boldsymbol{I_3} \end{bmatrix}, \ \boldsymbol{R_i} = 10\boldsymbol{I_4}, \ \gamma = 3,$$

$$c_1 = 0.333, \ c_2 = 1.5, \ c_3 = c_4 = 3. \tag{5.57}$$

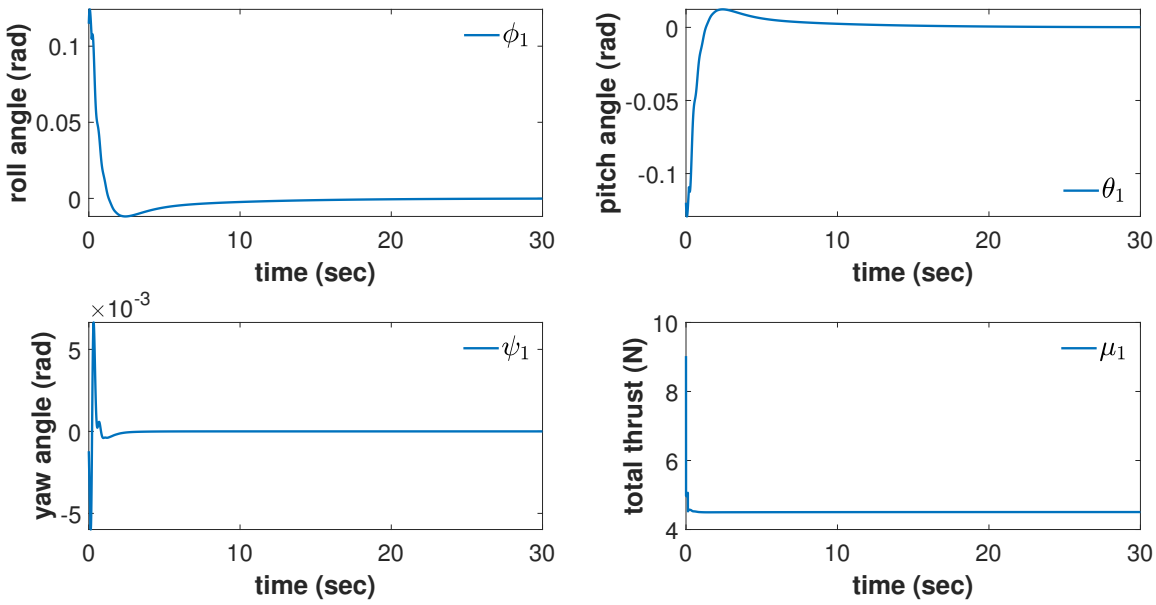


Figure 30: The Euler angles and total thrust of first follower UAV.

The input weighting matrix $\boldsymbol{R_i}$ parameters are selected big with respect to the state weighting matrix $\boldsymbol{\mathcal{Q}_i}$ to make sure that the Euler angles $\theta$ pitch angle and $\phi$ roll angle remains in the interval $[-\pi/6, \pi/6]$. Note that the condition (5.50) is satisfied for all followers with these parameters.

Since the quadrotor's linear model is obtained around the hover flight condition, the leader's positions are stationary in three-dimensional space. This is shown in the Fig. 28. Additionally, we illustrate attitudes of the quadrotor UAVs in Figures 30-33. The leaders' attitudes at the condition that we linearize the quadrotor's dynamics is hover, and hence their attitude data simply are $[\theta = 0 \ \phi = 0 \ \psi = c_\psi \ \mu = mg]^T$.

Realize that the disturbance acting on the well informed first UAV affects not only its stability but also the other uninformed followers. Thence, the uninformed followers' attitudes are nosier with respect to the well-informed UAV and also they become nosier as the nodes of the graph $\mathrm{G^f}$ gets away from the well informed follower node. Nevertheless, the $\mathcal{H}_\infty$ control method is robust enough to overcome the

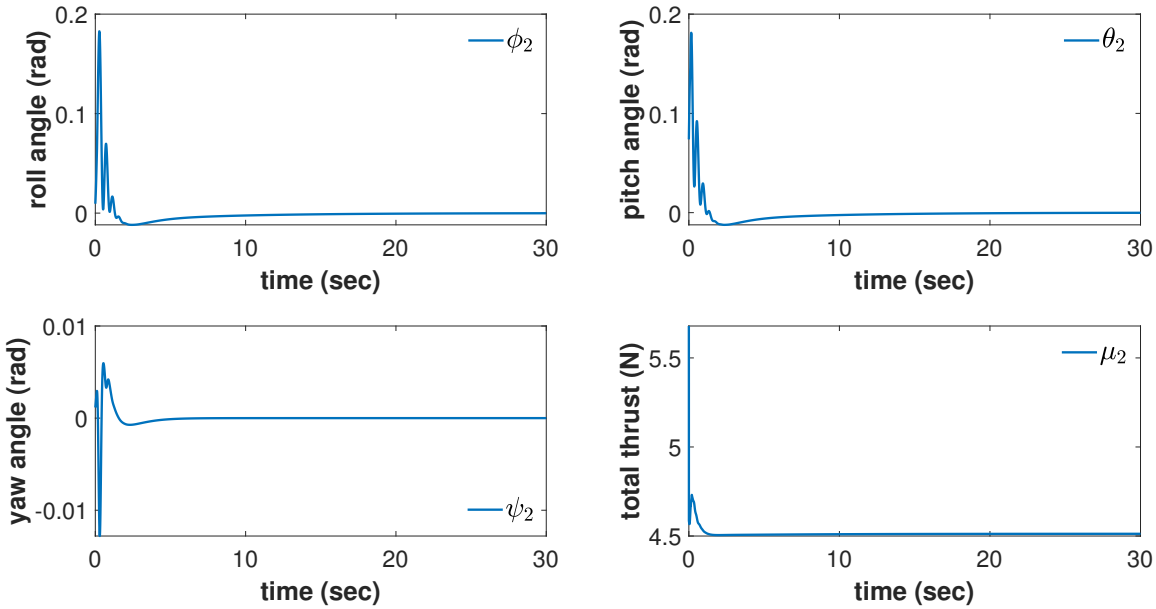


Figure 31: The Euler angles and total thrust of second follower UAV.

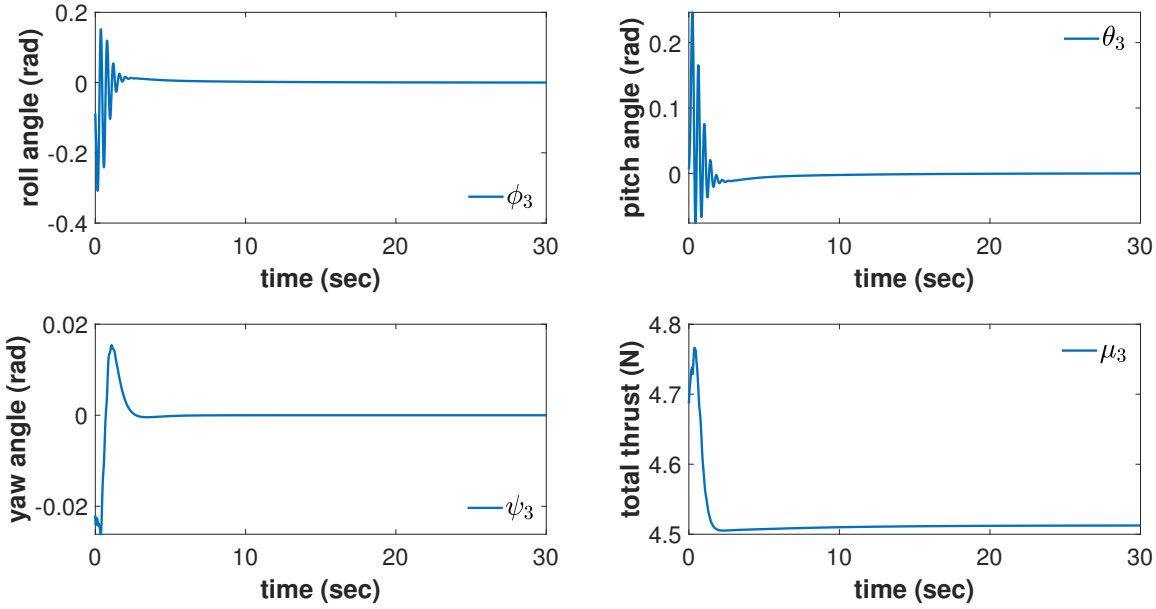


Figure 32: The Euler angles and total thrust of third follower UAV.

disturbance affect as shown in Figures 30-33. Lastly, all of the Euler angle values are converging to zero as expected and the total thrust values are converging to

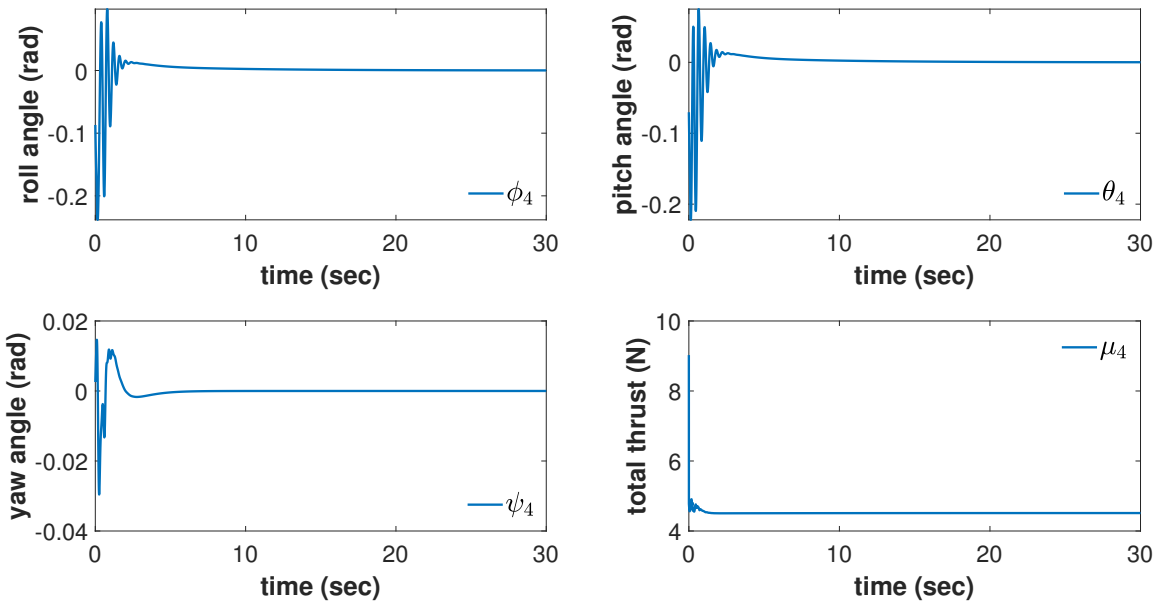


Figure 33: The Euler angles and total thrust of fourth follower UAV.

$mg = 4.5126\ N$ as can be seen Figures 30-33. This is the value of thrust to make the quadrotors motionless in three-dimensional space.

# CHAPTER 6

## Conclusions and Summary

In the first chapter of this dissertation, a method of distributed backstepping method to have formation flight of multiple quadrotors with distributed time delays is discussed. The proposed algorithms are validated by using Vicon Tracker, AR.Drone 2.0 and a master computer. Through rigorous experimentation and stability analysis, we showed that distributed backstepping control method provides a guaranteed performance for follower agents to track the leader agent with a predetermined position offset. We give the trajectories followed by the single quadrotor under the influence of commensurate delay and by the multiple quadrotors under the influence of distributed delays.

Next, we worked on the game theoretic solution of pursuit-evasion intercept problem when the velocity constraints are imposed on both pursuer and evader. By solving the HJI equation corresponds to the novel non-quadratic functional, we showed that game-optimal velocity trajectories are smooth and satisfies the predetermined boundaries. Using the rigorous Lyapunov analysis, we proved that the PE game ends in a finite-time under certain conditions, which indeed implies that intercept or capture occurs in finite-time. To solve the HJI equation, the IRL method with critic NN structure is used. Consequently, we showed the simulation results of the PE game when the evader adopts both game optimal and sub-optimal velocity policy while the pursuer tracks corresponding game optimal velocity trajectory with the nonlinear backstepping tracker. Simulations showed that when the evader adopts

its game optimal velocity policy, it takes more time to be intercepted by the pursuer compared to the scenario, in which the evader employs a sub-optimal velocity policy.

In the third chapter, we proposed a novel augmented Hamiltonian, and develop a new iterative algorithms based on stationarity conditions of the augmented Hamiltonian to obtain optimal gain solutions for $H_\infty$ (OPFB) control problem. These gain solutions guarantees both stability and $L_2$ gain boundedness of an LTI system when the $H_\infty$ static OPFB control method is employed. Convergence properties of two off-line iterative solution algorithms are given. We showed that solving the Riccati equation iteratively obviates the initial stabilizing gain requirement of Algorithm 1. Then based on Lyapunov iterations, an online off-policy IRL algorithm which is a model-free version of the offline Algorithm 1, is developed to solve the optimal $H_\infty$ regulator problem by learning the optimal gain solution without requiring system state, control, and disturbance matrices. Lastly, we applied proposed algorithms to the linearized F-16 lateral dynamics at a particular flight condition to verify the correct performance of proposed algorithms.

In the last chapter, we proposed novel local and global objectives for the MLF output containment game for the LTI multi-agent systems. The local $\mathcal{H}_\infty$ game is solved for the Nash equilibrium strategies where controls are minimizing and disturbances are maximizing players. On the other hand, the graphical game is introduced as a global objective to address mutual interests among the followers. Then, it has been shown that the local and global objectives can be optimized simultaneously under certain conditions. Rigorous $\mathcal{L}_2$ gain bound and asymptotic stability analyses are provided. The proposed methods are tested by means of the MLF quadrotor UAV game simulations. The results indeed verify the efficacy of the proposed methods.

REFERENCES


[1] J. K. Parrish, S. V. Viscido, and D. Grunbaum, "Self-organized fish schools: an examination of emergent properties," *The biological bulletin*, vol. 202(3), pp. 296–305, 2002.

[2] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems , Heidelberg*. Berlin: Springer, 2013, pp. 545–558.

[3] T. Kopfstedt, M. Mukai, M. Fujita, and C. Ament, "Control of formations of uavs for surveillance and reconnaissance missions," *IFAC Proceedings Volumes*, vol. 41(2), pp. 5161–5166, 2008.

[4] S. Waharte, N. Trigoni, and S. Julier, "Coordinated search with a swarm of uavs," in *IEEE annual communications society conference on sensor, mesh and ad hoc communications and networks workshops*. IEEE, June 2009, pp. 1–3.

[5] Y. Kartal, P. Kolaric, V. Lopez, A. Dogan, and F. Lewis, "Backstepping approach for design of pid controller with guaranteed performance for micro-air uav," *Control Theory and Technology*, vol. 18, no. 1, pp. 19–33, 2020.

[6] J. M. Daly, Y. Ma, and S. L. Waslander, "Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays," *Autonomous Robots*, vol. 38(2), pp. 179–191, 2015.

[7] R. Isaacs, *Games of pursuit*. Rand Corporation, 1951.

[8] H. Modares, M.-B. N. Sistani, and F. L. Lewis, "A policy iteration approach to online optimal control of continuous-time constrained-input systems," *ISA transactions*, vol. 52, no. 5, pp. 611–621, 2013.

[9] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online adaptive algorithm for optimal control with integral reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2686–2710, 2014.

[10] J. Gadewadikar, F. L. Lewis, and M. Abu-Khalaf, "Necessary and sufficient conditions for h-infinity static output-feedback control," *Journal of guidance, control, and dynamics*, vol. 29, no. 4, pp. 915–920, 2006.

[11] J. Gadewadikar, A. Bhilegaonkar, and F. L. Lewis, "Bounded l2 gain static output feedback: Controller design and implementation on an electromechanical system," *IEEE Transactions On Industrial Electronics*, vol. 54, no. 5, pp. 2593–2599, 2007.

[12] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free q-learning designs for linear discrete-time zero-sum games with application to h-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.

[13] S. A. A. Rizvi and Z. Lin, "Output feedback q-learning for discrete-time linear zero-sum games with application to the h-infinity control," *Automatica*, vol. 95, pp. 213–221, 2018.

[14] Y. Jiang, K. Zhang, J. Wu, C. Zhang, W. Xue, T. Chai, and F. L. Lewis, "H-infinity based minimal energy adaptive control with preset convergence rate," *IEEE Transactions on Cybernetics*, 2021.

[15] J. Han, H. Zhang, H. Jiang, and X. Sun, "H-infinity consensus for linear heterogeneous multi-agent systems with state and output feedback control," *Neurocomputing*, vol. 275, pp. 2635–2644, 2018.

[16] J. Gadewadikar, F. L. Lewis, K. Subbarao, K. Peng, and B. M. Chen, "H-infinity static output-feedback control for rotorcraft," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 4, pp. 629–646, 2009.

[17] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[18] H. Modares, F. L. Lewis, and Z.-P. Jiang, "H-infinity tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 10, pp. 2550–2562, 2015.

[19] S. Zuo, Y. Song, F. L. Lewis, and A. Davoudi, "Output containment control of linear heterogeneous multi-agent systems using internal model principle," *IEEE transactions on cybernetics*, vol. 47, no. 8, pp. 2099–2109, 2017.

[20] S. Zuo, Y. Song, F. Lewis, and A. Davoudi, "Time-varying output formation containment of general linear homogeneous and heterogeneous multiagent systems," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 537–548, 2018.

[21] H. Liu, G. Xie, and L. Wang, "Necessary and sufficient conditions for containment control of networked multi-agent systems," *Automatica*, vol. 48, no. 7, pp. 1415–1422, 2012.

[22] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, 2012.

[23] M. Liu, Y. Wan, V. G. Lopez, F. L. Lewis, G. Hewer, and K. Estabridis, "Differential graphical game with distributed global nash solution," *IEEE Transactions on Control of Network Systems*, 2021.

[24] M. I. Abouheaf and F. L. Lewis, "Multi-agent differential graphical games: Nash online adaptive learning solutions," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 5803–5809.

[25] Z. Qu and M. A. Simaan, "A design of distributed game strategies for networked agents," *IFAC Proceedings Volumes*, vol. 42, no. 20, pp. 270–275, 2009.

[26] V. G. Lopez, F. L. Lewis, Y. Wan, M. Liu, G. Hewer, and K. Estabridis, "Stability and robustness analysis of minmax solutions for differential graphical games," *Automatica*, vol. 121, p. 109177, 2020.

[27] Y. Kartal, K. Subbarao, N. R. Gans, A. Dogan, and F. Lewis, "Distributed backstepping based control of multiple uav formation flight subject to time delays," *IET Control Theory & Applications*, vol. 14, no. 12, pp. 1628–1638, 2020.

[28] I. Maza, A. Ollero, and E. Casado, "'classification of multi-uav architectures'," *Handbook of unmanned aerial vehicles*, pp. 953–975, 2015.

[29] G. Lee and D. Chwa, "Decentralized behavior-based formation control of multiple robots considering obstacle avoidance," *Intelligent Service Robotics*, vol. 11(1), pp. 127–138, 2018.

[30] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34(4), pp. 916–923, 2018.

[31] H. Duan and H. Qiu, "Unmanned aerial vehicle distributed formation rotation control inspired by leader-follower reciprocation of migrant birds," *IEEE Access*, vol. 6, pp. 23 431–23 443, 2018.

[32] J. Xi, N. Cai, and Y. Zhong, "Consensus problems for high-order linear time-invariant swarm systems," *Physica A: Statistical Mechanics and its Applications*, vol. 389(24), pp. 5619–5627, 2010.

[33] X. Dong, Y. Zhou, and Z. Ren, "et al.: 'time-varying formation control for unmanned aerial vehicles with switching interaction topologies'," *Control Engineering Practice*, vol. 46, pp. 26–36, 2016.

[34] J. F. Carvalho, S. Pequito, and A. P. Aguiar, "et al.: 'composability and controllability of structural linear time-invariant systems: Distributed verification'," *Automatica*, vol. 78, pp. 123–134, 2017.

[35] C. Q. Ma and J. F. Zhang, "Necessary and sufficient conditions for consensusability of linear multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 55(5), pp. 1263–1268, 2010.

[36] W. Rui, D. Xiwang, and L. Qingdong, "et al.: 'adaptive time-varying formation control for high-order lti multi-agent systems'," *2015 34th Chinese Control Conference (CCC)*, pp. 6998–7003, July 2015.

[37] F. Baghbani, M. R. Akbarzadeh-T, and M. B. N. Sistani, "Cooperative adaptive fuzzy tracking control for a class of nonlinear multi-agent systems," in *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, June 2017, pp. 1–6.

[38] G. X. Wen, C. P. Chen, and Y. J. Liu, "et al.: 'neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems'," *IET Control Theory & Applications*, vol. 9(13), pp. 1927–1934, 2015.

[39] W. He, G. Chen, and Q. L. Han, "et al.: 'network-based leader-following consensus of nonlinear multi-agent systems via distributed impulsive control'," *Information Sciences*, vol. 380, pp. 145–158, 2017.

[40] J. Sun and Z. Geng, "Adaptive consensus tracking for linear multi-agent systems with heterogeneous unknown nonlinear dynamics," *International Journal of Robust and Nonlinear Control*, vol. 26(1), pp. 154–173, 2016.

[41] M. Defoort, A. Polyakov, and G. Demesure, "et al.: 'leader-follower fixed-time consensus for multi-agent systems with unknown non-linear inherent dynamics'," *IET Control Theory & Applications*, vol. 9(14), pp. 2165–2170, 2015.

[42] H. Du, Y. Cheng, and Y. He, "et al.: 'second-order consensus for nonlinear leader-following multi-agent systems via dynamic output feedback control'," *International Journal of Robust and Nonlinear Control*, vol. 26(2), pp. 329–344, 2016.

[43] X. Dong, B. Yu, and Z. Shi, "et al.: 'time-varying formation control for unmanned aerial vehicles: Theories and applications'," *IEEE Transactions on Control Systems Technology*, vol. 23(1), pp. 340–348, 2014.

[44] A. Abdessameud and A. Tayebi, "Formation control of vtol unmanned aerial vehicles with communication delays," *Automatica*, vol. 47(11), pp. 2383–2394, 2011.

[45] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, "Adaptive nonlinear control without overparametrization," *Systems and Control Letters*, vol. 19(3), pp. 177–185, 1992.

[46] R. Skjetne, T. I. Fossen, and P. V. Kokotović, "Adaptive maneuvering, with experiments, for a model ship in a marine control laboratory," *Automatica*, vol. 41(2), pp. 289–298, 2005.

[47] R. Kristiansen and P. J. Nicklasson, "Satellite attitude control by quaternion-based backstepping," *Proceedings of the 2005, American Control Conference*, pp. 907–912, June 2005.

[48] S. S. Pavlichkov, S. N. Dashkovskiy, and C. K. Pang, "Uniform stabilization of nonlinear systems with arbitrary switchings and dynamic uncertainties," *IEEE Transactions on Automatic Control*, vol. 62(5), pp. 2207–2222, 2016.

[49] Z. Li and J. Zhao, "Co-design of controllers and a switching policy for nonstrict feedback switched nonlinear systems including first-order feedforward paths," *IEEE Transactions on Automatic Control*, vol. 64(4), pp. 1753–1760, 2018.

[50] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.

[51] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on circuits and systems*, vol. 25(9), pp. 772–781, 1978.

[52] R. J. Plemmons, "M-matrix characterizations. i—nonsingular m-matrices," *Linear Algebra and its Applications*, vol. 18(2), pp. 175–188, 1977.

[53] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of time-delay systems*. Science & Business Media: Springer, 2003.

[54] Y. Kartal, K. Subbarao, A. Dogan, and F. Lewis, "Optimal game theoretic solution of the pursuit-evasion intercept problem using on-policy reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 16, pp. 7886–7903, 2021.

[55] V. Turetsky and J. Shinar, "Missile guidance laws based on pursuit–evasion game formulations," *Automatica*, vol. 39, no. 4, pp. 607–618, 2003.

[56] T. Mylvaganam, M. Sassano, and A. Astolfi, "A differential game approach to multi-agent collision avoidance," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4229–4235, 2017.

[57] J. R. Marden and J. S. Shamma, "Game theory and control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 105–134, 2018.

[58] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[59] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[60] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[61] T. Basar and P. Bernhard, *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.

[62] S. Y. Hayoun, M. Weiss, and T. Shima, "A mixed l 2/l$\alpha$ differential game approach to pursuit-evasion guidance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 2775–2788, 2016.

[63] S. Bhattacharya, T. Başar, and N. Hovakimyan, "A visibility-based pursuit-evasion game with a circular obstacle," *Journal of Optimization Theory and Applications*, vol. 171, no. 3, pp. 1071–1082, 2016.

[64] H. Li, D. Liu, and D. Wang, "Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics," *IEEE Transactions on Automation science and engineering*, vol. 11, no. 3, pp. 706–714, 2014.

[65] Y.-Y. Liu, Z.-S. Wang, and Z. Shi, "Hinf tracking control for linear discrete-time systems via reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 1, pp. 282–301, 2020.

[66] Y. Dong, S. Mingming, and S. Zhaowei, "Satellite proximate interception vector guidance based on differential games," *Chinese Journal of Aeronautics*, vol. 31, no. 6, pp. 1352–1361, 2018.

[67] ——, "Satellite proximate pursuit-evasion game with different thrust configurations," *Aerospace Science and Technology*, vol. 99, p. 105715, 2020.

[68] H. Gong, S. Gong, and J. Li, "Pursuit-evasion game for satellites based on continuous thrust reachable domain," *IEEE Transactions on Aerospace and Electronic Systems*, 2020.

[69] A. Jagat and A. J. Sinclair, "Nonlinear control for spacecraft pursuit-evasion game using the state-dependent riccati equation method," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 3032–3042, 2017.

[70] R. W. Carr, R. G. Cobb, M. Pachter, and S. Pierce, "Solution of a pursuit–evasion game using a near-optimal strategy," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 4, pp. 841–850, 2018.

[71] V. Shaferman and T. Shima, "Cooperative differential games guidance laws for imposing a relative intercept angle," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2465–2480, 2017.

[72] I. Weintraub, E. Garcia, and M. Pachter, "Optimal guidance strategy for the defense of a non-manoeuvrable target in 3-dimensions," *IET Control Theory & Applications*, vol. 14, no. 11, pp. 1531–1538, 2020.

[73] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[74] W. M. Haddad and V. Chellaboina, *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton university press, 2011.

[75] P. Cannarsa and C. Sinestrari, *Semiconcave functions, Hamilton-Jacobi equations, and optimal control*. Springer Science & Business Media, 2004, vol. 58.

[76] V. G. Lopez, F. L. Lewis, Y. Wan, E. N. Sanchez, and L. Fan, "Solutions for multiagent pursuit-evasion games on communication graphs: Finite-time capture and asymptotic behaviors," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 1911–1923, 2019.

[77] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[78] Y. Yang, K. G. Vamvoudakis, and H. Modares, "Safe reinforcement learning for dynamical games," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 9, pp. 3706–3726, 2020.

[79] H. Jiang, H. Zhang, and X. Xie, "Critic-only adaptive dynamic programming algorithms' applications to the secure control of cyber–physical systems," *ISA transactions*, vol. 104, pp. 138–144, 2020.

[80] A. P. Valadbeigi, A. K. Sedigh, and F. L. Lewis, "H infinity static output-feedback control design for discrete-time systems using reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 2, pp. 396–406, 2019.

[81] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[82] D. P. Bertsekas, *Dynamic programming and optimal control: Vol. 1*. Athena scientific Belmont, 2000.

[83] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, 2014.

[84] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[85] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2017.

[86] C. Chen, H. Modares, K. Xie, F. L. Lewis, Y. Wan, and S. Xie, "Reinforcement learning-based adaptive optimal exponential tracking control of linear systems with unknown dynamics," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4423–4438, 2019.

[87] B. Luo, H.-N. Wu, and T. Huang, "Off-policy reinforcement learning for h-infinity control design," *IEEE transactions on cybernetics*, vol. 45, no. 1, pp. 65–76, 2014.

[88] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Neurodynamic programming and zero-sum games for constrained control systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1243–1252, 2008.

[89] H. Zhang, Q. Wei, and D. Liu, "An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games," *Automatica*, vol. 47, no. 1, pp. 207–214, 2011.

[90] Q. Wei, D. Liu, Q. Lin, and R. Song, "Adaptive dynamic programming for discrete-time zero-sum games," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 957–969, 2017.

[91] S. Mehraeen, T. Dierks, S. Jagannathan, and M. L. Crow, "Zero-sum two-player game theoretic formulation of affine nonlinear discrete-time systems using neural networks," *IEEE transactions on cybernetics*, vol. 43, no. 6, pp. 1641–1655, 2012.

[92] L. M. Zhu, H. Modares, G. O. Peen, F. L. Lewis, and B. Yue, "Adaptive suboptimal output-feedback control for linear systems using integral reinforcement learning," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 264–273, 2014.

[93] F. A. Yaghmaie, S. Gunnarsson, and F. L. Lewis, "Output regulation of unknown linear systems using average cost reinforcement learning," *Automatica*, vol. 110, p. 108549, 2019.

[94] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[95] R. Moghadam and F. L. Lewis, "Output-feedback h-infinity quadratic tracking control of linear systems using reinforcement learning," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 2, pp. 300–314, 2019.

[96] Q. Jiao, H. Modares, F. L. Lewis, S. Xu, and L. Xie, "Distributed l2-gain output-feedback control of homogeneous and heterogeneous systems," *Automatica*, vol. 71, pp. 361–368, 2016.

[97] S. A. Arogeti and F. L. Lewis, "Static output-feedback $h_\infty$ control design procedures for continuous-time systems with different levels of model knowledge," *IEEE Transactions on Cybernetics*, 2021.

[98] J. Gadewadikar, F. L. Lewis, L. Xie, V. Kucera, and M. Abu-Khalaf, "Parameterization of all stabilizing h-infinity static state-feedback gains: application to output-feedback design," *Automatica*, vol. 43, no. 9, pp. 1597–1604, 2007.

[99] B. M. Chen, *Robust and H-infinity Control*. Springer Science & Business Media, 2013.

[100] P. Apkarian, D. Noll, and A. Rondepierre, "Mixed h2 h-infinity control via nonsmooth optimization," *SIAM Journal on Control and Optimization*, vol. 47, no. 3, pp. 1516–1546, 2008.

[101] D. Kleinman, "On an iterative technique for riccati equation computations," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 114–115, 1968.

[102] H. W. Knobloch, A. Isidori, and D. Flockerzi, *Topics in control theory*. Birkhäuser, 2012, vol. 22.

[103] D. Moerder and A. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains," *IEEE Transactions on Automatic Control*, vol. 30, no. 9, pp. 900–903, 1985.

[104] B. Datta, *Numerical methods for linear control systems*. Academic Press, 2004, vol. 1.

[105] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[106] H. Modares, F. L. Lewis, and Z.-P. Jiang, "Optimal output-feedback control of unknown continuous-time linear systems using off-policy reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2401–2410, 2016.

[107] C. Li, G. G. Yin, L. Guo, C.-Z. Xu *et al.*, "State observability and observers of linear-time-invariant systems under irregular sampling and sensor limitations," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2639–2654, 2011.

[108] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[109] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, *Cooperative grasping and transport using multiple quadrotors*. Springer, 2013.

[110] T. Kopfstedt, M. Mukai, M. Fujita, and C. Ament, "Control of formations of uavs for surveillance and reconnaissance missions," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 5161–5166, 2008.

[111] S. Waharte, N. Trigoni, and S. Julier, "Coordinated search with a swarm of uavs," in *2009 6th ieee annual communications society conference on sensor, mesh and ad hoc communications and networks workshops*. IEEE, 2009, pp. 1–3.

[112] Y. Kartal, K. Subbarao, A. Dogan, and F. Lewis, "Optimal game theoretic solution of the pursuit-evasion intercept problem using on-policy reinforcement learning," *International Journal of Robust and Nonlinear Control*, 2021.

[113] S. B. Sarsılmaz and T. Yucelen, "Distributed control of linear multiagent systems with global and local objectives," *Systems & Control Letters*, vol. 152, p. 104928, 2021.

[114] C. Chen, F. L. Lewis, S. Xie, H. Modares, Z. Liu, S. Zuo, and A. Davoudi, "Resilient adaptive and $h_\infty$ controls of multi-agent systems under sensor and actuator faults," *Automatica*, vol. 102, pp. 19–26, 2019.

[115] S. Zuo, F. L. Lewis, and A. Davoudi, "Resilient output containment of heterogeneous cooperative and adversarial multigroup systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 7, pp. 3104–3111, 2019.

[116] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[117] D. S. Bernstein, *Matrix mathematics*. Princeton university press, 2009.

[118] X. Dong and G. Hu, "Time-varying formation tracking for linear multiagent systems with multiple leaders," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3658–3664, 2017.

[119] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.

[120] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.

[121] B. D. McKay, F. E. Oggier, G. F. Royle, N. Sloane, I. M. Wanless, and H. S. Wilf, "Acyclic digraphs and eigenvalues of (0, 1)-matrices," *arXiv preprint math/0310423*, 2003.

## BIOGRAPHICAL STATEMENT

Yusuf KARTAL studied Electrical and Electronics Engineering at Bilkent University (Turkey) obtaining the degree of Bachelor's in Science in 2015. Then, he obtained Master in Science degree at Aerospace Engineering Department in University of Texas at Arlington in 2019. He has recently obtained Ph.D. at the same department in University of Texas at Arlington in March 2022. His research interests include multi-robot systems, nonlinear control, game theory, unmanned aerial vehicles, machine learning and distributed control.