INTERNATIONAL JOURNAL FOR INNOVATION EDUCATION AND RESEARCH
ISSN: 2411-2933

Journal URL: https://scholarsjournal.net/index.php/ijier

DOI: https://doi.org/10.31686/ijier.vol11.iss1.4066

Multiagent lqr-based control design and gain tuning for quadcopters fleet

Evandro Martins Araujo Filho, João Viana da Fonseca Neto

Abstract

An LQR-based Control design and gain tuning strategies proposals for a multi-agent system are presented in this article, the agents are connected in an undirected graph. Controller gains tuning are adjusted by selecting the Q and R weighting matrices of the Linear Quadratic Regulator. Agreement (consensus) is one of the fundamental problems in multi-agent control, where a set of agents must agree on a joint state value. In the proposed design, first considering that the behavior of the agreement protocol is undirected and static, the main objective is to highlight the complexity of the relationship between the convergence properties of this protocol and the structure of adjacent interconnections. The effects on the formation due to static geometry are analyzed from the resulting data according to the proximity between the agents, where behavior and stability are analyzed based on the desired formation geometry through the construction of the Laplacian matrix.

Keywords: Formation Control, Agreement Protocols, Multi-agent Control, Consensus, Linear Quadratic Regulator (LQR), quadrotor, Optimality, Quadcopters fleet;

1. Introduction

The aim of cooperative control of multi-agent systems (MAS) is to control multiple dynamic units that share information to achieve a given mutual goal via collective movements and local communications (Li & Dua, 2017). This control has been widely used in recent years due to its various applications in unmanned aerial vehicles (UAV), robotics in manufacturing and military, and space missions using satellites. Currently, many military and non-combat applications involve groups of agents that must cooperate to complete a specific mission without human involvement.

Cooperative control is inspired by natural behaviors; in this case, the focus is on the collective movement of animals in groups. Every animal has individual movement characteristics; however, their collective movement and the communication between individuals result in movements that make multiple animals appear as a single entity, such as flocks of birds, herds of animals, and schools of fish. This synchronized and collective movement helps animals to stay together when searching for food, defend against predators, and stay together during migration (Lewis, Zhang, & Hengster-Morvic, 2014). In the context of cooperative control for formation and from the point of view of control in multi-agent systems, the authors (Tanner, Pappas, & Kumar, 2004) emphasize optimal and adaptive control approaches.

A methodology oriented to the design of Optimal LQR-based multiagent control system design and its gain tuning is presented in this article. In the sense of performing evaluations of model-based, formation protocols for a fleet of quadcopters are associated to verify proposed design methodology.

In PhD thesis (Arokiasami, 2016), the author presents a multi-agent framework for unmanned aerial/ground vehicles design. This thesis is focused on obstacle detection and avoidance via computer vision-based algorithms, as these algorithms are generally platform independent. Obstacle detection is achieved utilizing Hough transform, Canny contour and Lucas Kaneda sparse optical flow algorithm. Collision avoidance is performed utilizing time-to-contact estimation techniques.

A unified framework is presented to capture the maximal disc-guaranteed gain margin (GGM) of the discrete-time system via linear quadratic regulator (LQR) in reference (Fonseca Neto, Abreu, & Silva, 2010). Sufficient and necessary conditions on consensusability are established.

In a multi-agent system that are oriented for formation tasks, each member (agent) may interfere with or operate within the range or radius of another member of the group, resulting in the agents becoming moving obstacles for each other, making formation control more complex. The advantages of interconnected multi-agent systems over conventional systems include reduced cost, greater efficiency, better performance, parameter tuning and robustness, and new features (Tanner, Pappas, & Kumar, 2004).

Graph-based multi-agent systems (MAS) show good results in terms of achieving a unique stable formation (Bamieh, Paganini, & Dahleh, 2002). However, the main drawback of these systems is that they require highly connected communication patterns to guarantee a unique formation that does not depend on the initial conditions (Smith, Egerstedt, & Howard, 2009). In the context of stability, a protocol that promotes a continuous generalization of the behavior when it evolves across switching networks is presented (Guo, Zhou, & Liu, 2018). In addition, these studies usually consider vehicles as a point mass to simplify the dynamic and kinematic models.

Consensus algorithms for autonomous agents is an area of research that is related to the problems and techniques discussed in this paper, where agents $(1, 2, \dots, N)$ with a given formation have reached a consensus if their associated variables $(x_1, x_2, \dots, x_N)$ converge to a common value. A stable formation must be achieved when a consensus problem is considered. Furthermore, vehicles must also reach the same speed in addition to reaching a certain point. In contrast, the vehicle dynamics in the formation problem are governed by differential equations of order n.

The design of control protocols for cooperative control systems is not a trivial activity because the communication topology can restrict the application of distributed control protocols. In these, the control policy of each agent depends only on its own information and the available local information of its neighbors, directly affecting the stability of the multi-agent system in trying to achieve a given desired formation (Bamieh, Paganini, & Dahleh, 2002). The relationships between stability and optimization are usually applied in single agent systems; however, understanding the relationships between stability and optimization in cooperative control systems with multiple agents has gained interest in recent years (Cepeda-Gomez & Perico, 2015) (Fax J. A., 2002) (Dimarogonas & Kyriakopoulos, 2009). These relationships in cooperative control are not the same as those appearing in the single-agent case, and local optimization and global team optimization are not the same (Lewis, Zhang, & Hengster-Morvic, 2014).

Essentially, the problem of formation of multi-agent systems includes two steps. As a first step, it is necessary to determine the desired training. In the second step, the design of the corresponding control algorithm used to achieve this training is performed (Li & Dua, 2017). Motivated by these problems, some basic concepts related to the formation of MAS are presented. Furthermore, a general framework methodology for the optimal formation problem is introduced to establish a solid basis for developing optimal control in the formation of a multi-agent system.

The article is organized as follows. In Section II, the dynamics of a quadcopter are briefly described, followed by the mathematical model identified for its representation. In Section III is presented the state space model for formation control. The model is prepared for LQR-based control strategies that are oriented to multiagent formation. The mathematical formulation of state space modelling and a method for controller tuning through the Linear Quadratic Regulator is presented are described with a focus on state space multiagent optimal control design. In Section V, the numerical simulation results of the proposed method are presented. Finally, in Section VI, the conclusions are drawn.

2. Modelling and LQR Control of Quadcopter Fleet

A quadcopter model, graph theory to model the fleet, Kronecker product and Linear Quadratic Regulator (LQR) to state feedback regulation of quadcopter formation are the fundamental topics for the design methodology proposal.

A quadcopter is one of many types of unmanned aerial vehicles (UAV), and it shows reliability in maneuvering and its control and modeling technical issues accessible for real world needs. Therefore, research on quadcopters is gaining interest in the area of robotic control (Lyu, 2017). Graph theory is one of the most fundamental concepts and approaches for fully understanding the control of multi-agent systems (Mesbahi & Egerstedt, 2010). The main advantage of graph-based control methods is that they are easy to analyze and that it is easy to design associated control methods, which can also incorporate various communication topologies and scalability. Graph-based abstractions of a networked system contain almost no information about what exactly is shared between agents; instead, they provide information about the network connectivity in terms of the topology by representing topology objects in terms of nodes and edges (Guo, Zhou, & Liu, 2018). An extensive research of news methods and algorithms for analysis and implementation of control design in multiagent systems has been developed to several, sectors of our days.

2.1 State Space Variables and Inputs of Quadcopter Agent

The challenge in designing a control system for a quadcopter is that the mathematical model of the system is highly coupled, nonlinear and multivariable (Elkholy, 2014). The dynamics of a quadcopter contain six degrees of freedom and can be described by three degrees of rotation (roll, pitch, and yaw) and three translations along the x, y, and z axes. In most instances, the structure of a quadcopter is symmetrical and consists of four rotors attached at an equal distance from the center of mass of the body, as shown in Figure 1. Each of the rotors is driven by a DC motor. Propellers 1 and 3 rotate in the same direction, while propellers 2 and 4 rotate in the opposite direction to maintain the balance of the system.

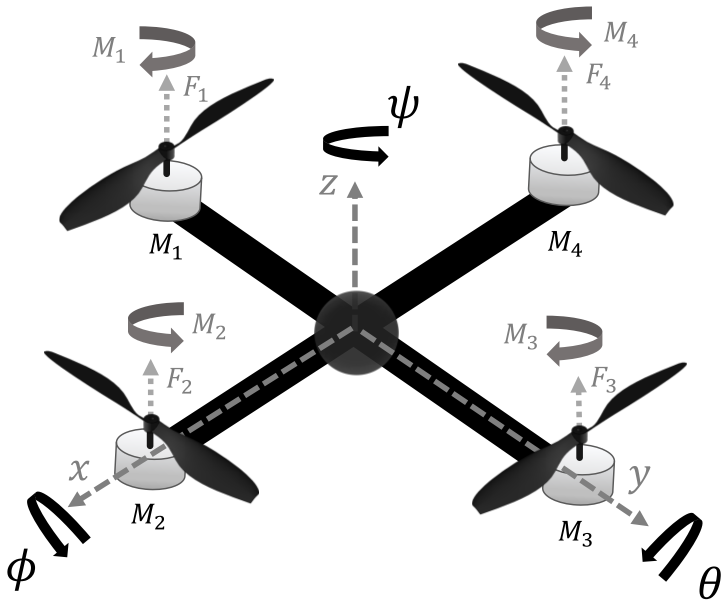

Figure 1. Referential Structure and Euler Angles.

The body structure features the origin at the center of the quadrotor body with the x-axis pointing along propeller 1, the y-axis pointing along propeller 2, and the z-axis pointing to the ground, as illustrated in Figure 1. The motions of the quadrotor can be divided into two subsystems—translational (positions x, y, and altitude) and rotational (*roll*, *pitch*, and *yaw*) (Elkholy, 2014). Based on Newton and Euler formalisms, the forces and moments acting on the quadcopter are investigated.

The challenge in designing a control system for a quadrotor is that the mathematical model of the system is highly coupled and nonlinear (Ataka, Tnunay, & Inovan, 2013). In addition, the system of differential equations has several control and output variables and constants governing the system dynamics, making the control system quite complex (Elkholy, 2014). Therefore, the process of system identification was used to simplify the mathematical model of this system.

The process of building models from experimental data is called system identification, involving building a mathematical model of a dynamic system based on a sample set of measured inputs and outputs, as illustrated in Figure 2. The chosen mathematical model can be characterized in terms of descriptions, such as transfer functions, impulse response, or power series expansions, and then used for controller design (Bhuvaneswari, Praveena, & Divya, 2012).

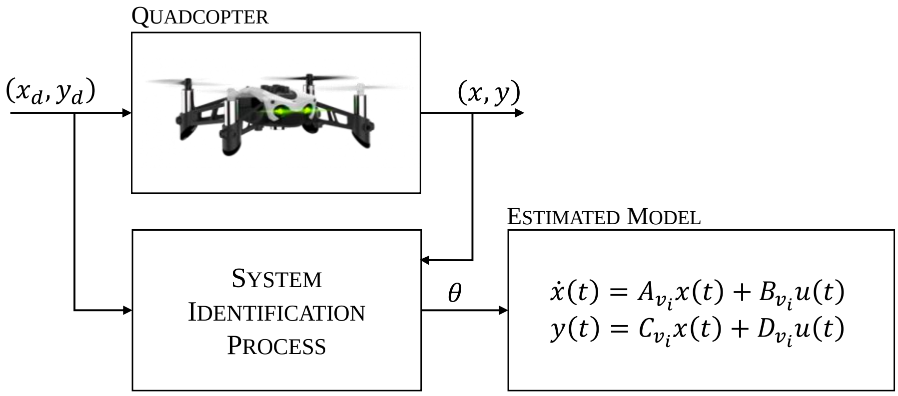

Figure 2. Block diagram of the quadcopter state space identification process.

The block diagram of Figure 2 presents the mathematical model structure used in the identification process of the state space model parameters. This model is obtained in its state space form, with $\theta = \{A_{v_i} \quad B_{v_i} \quad C_{v_i} \quad D_{v_i}\}$. The state, control, and output for the i^{th} agent are represented by $x \in \Re^n$, $u \in \Re^m$, and $y \in \Re^p$, respectively.

2.2 Graph Theory and Formation Modelling

Expanding quadrotor state space model of Figure 2, concepts of graph theory and multiagent (Mesbahi & Egerstedt, 2010) state space to establish formation modelling of quadcopters fleet are presented in this subsection.

A graph is defined as a set of vertices and a group of edges, each connecting a pair of vertices. In mathematical representation, a graph is denoted as $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_N\}$ are a set of nodes (vertices), and $E_{i,j} = \{v_i, v_j\}$ are a set of edges. The degree of a vertex denotes the number of edges that have v_i as its head, and the degree of an edge indicates how many have v_j as its tail. The degree of a vertex in the graph is equal to the number of neighbors of that node; thus, the number of neighbors of node i, i.e., $\mathcal{J}_i = \{v_j : (v_i, v_j) \in E\}$ (Li & Dua, 2017).

A Graph G consists of a finite set V of vertices and a set E of subsets of two elements of V to be referred to as edges. Graph G is considered connected if for any vertex $i, j \in V$, a path of edges exists in G from i to j. Then, for any $i, j \in V$, we define the distance between i and j to be the number of edges along the shortest path joining i and j. The diameter D of a connected graph G is the maximum distance between any two vertices of G.

In an undirected graph G, the degree of a vertex is denoted by $d(v_i)$ and is equivalent to the number of vertices adjacent to the vertex (v_i) in the graph. The degree matrix, denoted by $D(G)$, is a diagonal matrix with diagonal elements equal to the degree of each vertex (Li & Dua, 2017) and is defined as:

$$D(G) = \begin{bmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_n) \end{bmatrix}, \quad (1)$$

where n is the number of vertices.

The adjacency matrix is an optimal attribute for defining the communication degree of a given graph, as it contains information about the agents and their connections (Mesbahi & Egerstedt, 2010). The adjacency matrix of an undirected graph is denoted by $A_d(G)$ and is a symmetric matrix $n \times n$ defined as:

$$A_d(G) = \begin{cases} 1 & v_i v_j \in E, \\ 0 & otherwise. \end{cases} \tag{2}$$

The graph information is essential for computing the dynamics of agreements between agents and is obtained by means of a Laplacian (Mesbahi & Egerstedt, 2010) that is given by

$$L_G = D(G) - A_d(G)\,, \tag{3}$$

where D and A_d are the degree and adjacency matrix of the same graph, respectively. The Laplacian $L_G \in \Re^{n \times n}$ matrix.

For agents to achieve a goal collectively, it is necessary for all of them to have a common variable of interest, called information state, and a set of rules for entering into an agreement, called consensus algorithm (Mesbahi & Egerstedt, 2010). The consensus is defined as an agreement between several agents on a certain shared variable or a common objective by interactions of the group through communication links or sensors; when this condition is satisfied, it is said that the agents have reached a consensus (Dimarogonas & Kyriakopoulos, 2009).

According to the consensus protocols in (Dimarogonas & Kyriakopoulos, 2009), suppose a network consists of N corresponding agents with continuous-time linear dynamics. This model can similarly be considered a linearized version of the nonlinear system. Suppose the system dynamics model in state space description is given by

$$\begin{aligned} \dot{x}_{v_i} &= A x_{v_i} + B u_i\,, \\ y_{v_i} &= C x_{v_i,} \end{aligned} \tag{4}$$

where the state, control, and output for the i^{th} agent are represented by $x_{v_i} \in \Re^n$, $u_i \in \Re^m$, and $y_{v_i} \in \Re^p$, respectively. In addition, A, B, and C are constant matrices and their dimensions change based on the number of agents to satisfy matrix multiplications. The goal of consensus is that each agent communicates only with its neighbors.

Consequently, a group of N agents obtain consensus when $\lim_{t\to\infty} \left\| x_i(t) - x_j(t) \right\| = 0\,,\ \forall i,j = 1,\ldots,N\,.$

Another term for distributed control laws is consensus protocols, which are either dynamic or static based on the choice of problem.

2.3 LQR and Tuning Method for Multiagent Control Design

The Linear Quadratic Regulator (LQR) problem is presented in form of multiagent design, where each agent of the fleet is in state space description that is given by Eq. (4). The LQR design is based on minimization of the performance index (J) that is a time integral of the sum of the states and the control energies. This performance index is given by

$$J(x_{v_i}, Q, u_i, R) = \frac{1}{2}\int_0^{\infty} (x_{v\ i}^T Q x_{v_i} + u_i^T R u_i) dt, \tag{5}$$

where $Q \geq 0$ is a positive semidefinite state weighting matrix and $R > 0$ is a positive input weighting matrix. Due to the fact that is necessary to insert in the controllers, design specifications, and physical constraints. The performance index J of Eq. (5) handles with optimality and systematization of controller actions design. In (Fonseca Neto, Abreu, & Silva, 2010) the optimization function that minimizes the performance index is represented as

$$\min_u J(x_{v_i}, Q, u_i, R), \tag{6}$$

subject to the constraint given by

$$\dot{x}_{v_i} = A x_{v_i} + B u_i . \tag{7}$$

Considering the system given in Eq. (7) with the cost function given by Eq. (5), the optimal control law is given by

$$u^*(t) = -F^* x_{v_i}(t), \tag{8}$$

where F^* is the state feedback gain matrix for the closed loop system and is given by

$$F^* = -R^{-1} B^T P, \tag{9}$$

where P is the solution matrix of the algebraic Riccati differential equation (Lewis, Vrabie, & Syrmos, 2012), which is obtained numerically by

$$-\dot{P} = A^T P + PA - PBR^{-1}B^T P + Q . \tag{10}$$

Using dynamic programming methods, one can solve the Riccati differential equation given in Eq. (10). When the time horizon tends to infinity, the values of P converge (stabilize), i.e., the matrix $\dot{P} = 0$. Consequently, the algebraic Riccati equation is given by

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 . \tag{11}$$

The LQR is optimal for an infinite horizon and the system behavior can be adjusted through the weighting matrices Q and R. Another way to describe the performance function (J) is to penalize the output signal (y_{v_i}) (Fonseca Neto, Abreu, & Silva, 2010), assuming that

$$Q = C^T Q_y C , \tag{12}$$

where $Q_y \geq 0$ is the exit penalty. Substituting this equation into Eq. (5) and considering that the output signal (y_{v_i}) is represented by state variables, as in Eq. (4), the performance index is given by

$$J(x_{v_i}, Q, u_i, R) = \frac{1}{2}\int_0^{\infty} (x_{v_i}^T C^T Q_y C x_{v_i} + u_i^T R u_i) dt = \\ J(y_{v_i}, Q_y, u_i, R) = \frac{1}{2}\int_0^{\infty} (y_{v_i}^T Q_y y_{v_i} + u_i^T R u_i) dt . \tag{13}$$

In this manner, the performance index is described in relation to the output signal. Generally, a relationship between the matrices Q and R is noted, resulting in a factor that is given by

$$\beta = QR^{-1}. \tag{14}$$

The values of β can be selected to provide fast control by selecting only the values of the matrix Q or obtain a low power consumption of the input signal by selecting only the matrix R (Grimstad, 2009).

Schur's triangularization is an alternative for solving the linear systems of the ARE that is given by Eq. (12), although it still has a high computational cost (Horn & Johnson, 2013). The triangularization process converts a square linear system into a triangular system with the same solution. The strategy used by this method for solving a linear system is to apply transformations to this system in order to convert it into another one with a higher triangular form or lower triangular form, but that has the same solution as the original system (Laub, 1979). Defining a matrix U as being a unitary matrix, that is, $U^T = U^{-1}$, superior triangular form is given by

$$\begin{gathered} U^{-1}TU = \tilde{T}\,, \\ \tilde{t}_{ii} = \lambda_i, i = 1,2, \dots, n_i \end{gathered} \tag{15}$$

For $U \in \Re^{n_i}, n_i$ is the order of the matrix and $\tilde{t}_{ii}$ are the elements of the main diagonal of the matrix $\tilde{T}$.

3. State Space Model for Formation Control

The main goal of formation control is to make multiple agents achieve a common goal, such as organizing themselves into a specified geometric shape using their mutual information and updating their positions and velocities relative to each other. As mentioned in the previous section on consensus algorithms, each agent communicates with its neighboring agents (Feng, Zhang, Tong, & Zhang).

3.1 State Space Agent Modelling

Assuming that there are N agents with a dynamic given by

$$\dot{x}_{v_i} = A_{v_i} x_{v_i}(t) + B_{v_i} u_{v_i}(t)\,, \quad i = 1,2,\dots,N, \tag{16}$$

where $u_{v_i} \in \Re^m$ is the input vector for the dynamic system of a given agent and A_{v_i}, x_{v_i}, and B_{v_i} are given by

$$\begin{gathered} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \\ x_{v_i} = [x_{i_1}(t) \quad x_{i_2}(t) \quad \dots \quad x_{i_n}(t)]^T \\ B_{v_i} = \begin{bmatrix} b_{11} & b_{12} & \cdots & a_{1m} \\ b_{21} & b_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix} \\ u_{v_i}(t) = [u_{i_1} \quad u_{i_2} \quad \dots \quad u_{i_m}]^T, \end{gathered} \tag{17}$$

where $A_{v_i} \in \Re^{n \times n}$, $i = 1,2, \dots, N$, $B_{v_i} \in \Re^{n \times m}$, n and m are the system order and the number of inputs for each agent, respectively, A_{v_i} is the state weighting matrix and B_{v_i} is the input weighting matrix.

3.2 State Space Multiagent Modelling

To a better understanding of the proposed design methodology, concepts and associative property of the Kronecker product are is presented in Appendix 1. After algebraic operations with Kronecker algebra, the mathematical model that represents the multi-agent system is given by

$$\dot{x}_F(t) = Ax_F(t) + Bu_F(t)\,, \tag{18}$$

where the variables and parameters of this equation are given by

$$\begin{gathered} A = I \otimes A_{v_i} \\ x_F(t) = [x_{v_1} \quad x_{v_2} \quad \cdots \quad x_{v_N}]^T, \\ B = I \otimes B_{v_i}\,, \quad I \in \Re^{(nN)\times(nN)}, \\ u_F(t) = [u_{v_1} \quad u_{v_2} \quad \cdots \quad u_{v_N}] \end{gathered} \tag{19}$$

3.3 Specified Formation

The desired formation is achieved through Eq. (18) by means of the reference points given by

$$h_p = \left(h \otimes \begin{bmatrix}1\\0\end{bmatrix} + \dot{h} \otimes \begin{bmatrix}0\\1\end{bmatrix}\right) \in \Re^{nN}\;, \tag{20}$$

where the vectors of the reference points are given by

$$h = [h_1 \quad h_2 \quad \dots \quad h_N]\;, \tag{21}$$

where h_i is the desired position vector with coordinates in the Cartesian coordinate system (x, y) for each agent, and is given by

$$h_i = [x_{p_i} \quad y_{p_i}]\;, \tag{22}$$

The vehicles N are in formation h_p if there are vectors $q \in \Re^n$ such that $x_{p_i} - h_i = q, i = 1 \dots N,$ the subscript p refers to the position components of $x_{v_i}(t)$. The desired formation is illustrated in Figure 3.

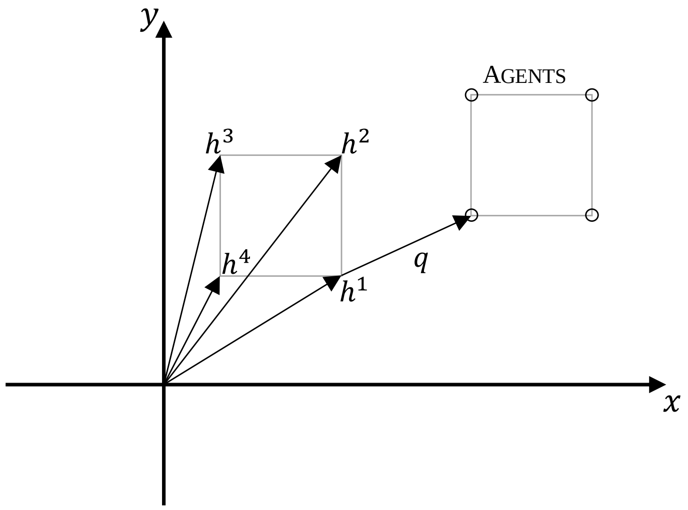

Figure 3. Square formation for system with four agents.

A regular quadrilateral formation is illustrated in Figure 3. In this type of formation control, the error between agents is calculated based on the average relative motion of adjacent agents (Fax & Murray, 2004) and is given by

$$z_i = (x_i - h_i) - \frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i}(x_j - h_j)\,, \qquad i = 1 \dots N\,, \tag{23}$$

where $|\mathcal{J}_i|$ is the elements quantity of the set $\mathcal{J}_i$. This gain does not influence the results considering that the sum of the weights of a given one is equal to one. By substituting the Laplacian matrix into equation Eq. (23), the output vector and the system equation is obtained as

$$\dot{x}_F(t) = Ax_F(t) + Bu_F(t)\,,$$
$$z(t) = L\big(x_F(t) - h_p\big), \tag{24}$$
$$L = L_G \otimes I, \quad I \in \Re^{(nN)\times(nN)}\,.$$

Once the mathematical model of the system has been determined, it is necessary to find a feedback gain matrix F that satisfies Eq. (24) and leads this system to a given formation h_p. Eq. (20) is given by

$$\dot{x}_F(t) = Ax_F(t) + BFL\big(x_F(t) - h_p\big)\,, \tag{25}$$

By substituting the matrices A and B (Eq. (19)) and $F = I_N \otimes F_{v_i}$ into Eq. (25), the following equation is obtained

$$\dot{x}_F(t) = I \otimes A_{v_i}x_F(t) + L_G \otimes B_{v_i}F_{v_i}\big(x_F(t) - h_p\big)\,. \tag{26}$$

The feedback gain matrix F_{v_i} shows some of the communication properties of graph G. Thus, it is necessary to insert the eigenvalues of the directed Laplacian matrix into the given graph. Consequently, matrix U is defined to be a unitary matrix, making $\tilde{L}_G = U^{-1}L_GU$ the upper triangular. Using the direct forms of the matrix, A, B, and F, the fleet model is given by

$$(U^{-1} \otimes I)(A + BFL)(U \otimes I) = I \otimes A_{v_i} + L_G \otimes B_{v_i}F_{v_i}\,. \tag{27}$$

The right-hand side of equation Eq. (27) is an upper triangular block with blocks in the form $A_{v_i} + \lambda B_{v_i}F_{v_i}$ where λ is the eigenvalue of the Laplacian matrix. Consequently, the eigenvalues of $(A + BFL)$ are the same as the eigenvalues of $(A_{v_i} + \lambda B_{v_i}F_{v_i})$, provided that λ is an eigenvalue of L_G.

The desired formation is achieved if Eq. (27) is stable, i.e., each agent must be precisely in formation. A stability analysis of the system is performed using the matrices A_{v_i} and B_{v_i}. As $zero$ is an eigenvalue of L_G of multiplicity 1 (for connected graphs), then each eigenvalue of A_{v_i} will also be an eigenvalue of $A + BFL$ with the same multiplicity. The system will be stable if all other eigenvalues of $(A + BFL)$ have a negative real part.

The formation variable h satisfies $\dot{h} = 0$ if the desired formation is constant; thus, the system given by Eq. (26) can be rewritten as

$$\dot{x}_F(t) = Ax_F(t) + BFLx_F(t) - BFL\left(I\otimes \begin{bmatrix}1\\0\end{bmatrix}\right)h\,,$$
$$\dot{h} = 0\,, \tag{28}$$

this set of equations is written as

$$\dot{y} = My\,, \quad y = \begin{bmatrix}x_F\\h\end{bmatrix}\,, \tag{29}$$

where the matrix M is given by

$$M = \begin{bmatrix}A + BFL & -BFL\left(I\otimes \begin{bmatrix}1\\0\end{bmatrix}\right)h\\ 0 & 0\end{bmatrix}. \tag{30}$$

4. Simulation Results and Discussion

The acquired results are illustrated with several numerical simulations, all of which consider that the graph consists of four nodes and all edges have a weight equal to one.

4.1 Formation control algorithm and Consensus Criteria

The formation control algorithm simulates the results using the consensus criteria, where the adjacency and degree matrices, respectively, are given by

$$A_d = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{31}$$

According to Eq. (31), it is noted that the topology of the multi-agent system used is of the undirected graph type, in which each agent communicates with its neighbor, as illustrated in Figure 4.

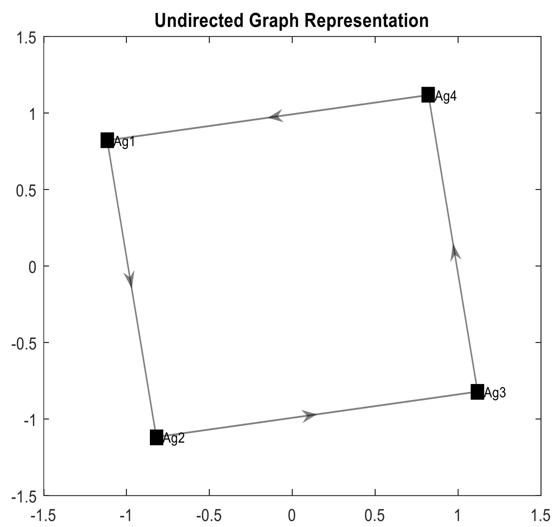

Figure 4. Representation of the undirected cyclic graph.

From the graph illustrated in Figure 4, it is possible to verify the information flow between agents. From Eq. (31), the Laplacian matrix described in Eq. (24) is given by

$$L_g = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}. \tag{32}$$

4.2 Multiagent Dynamics

In computational experiments, it is assumed that the dynamics of all agents are equal, and therefore, the mathematical model of the system obtained through the identification process is used. Thus, the model used for the simulation of the dynamics of each of the agents presented in using the state variables in Eq. (16) is given by

$$A_{v_i} = \begin{bmatrix} 0 & 2.54 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.54 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B_{v_i} = \begin{bmatrix} 1.73 & 0 \\ 0 & 0 \\ 0 & 1.73 \\ 0 & 0 \end{bmatrix}, \quad (33)$$

and the state vector is described by

$$x_{v_i} = \begin{bmatrix} x_{m_i} \\ \dot{x}_{m_i} \\ y_{m_i} \\ \dot{y}_{m_i} \end{bmatrix}, u_{v_i} = \begin{bmatrix} u_{x_i} \\ u_{y_i} \end{bmatrix}, \quad (34)$$

where (x_m, y_m) are the linear positions, $(\dot{x}_m, \dot{y}_m)$ are the linear velocities, and (u_x, u_y) are the inputs of the x and y systems, respectively.

4.3 Formation Specification

The desired formation is specified as being the vertices of a regular quadrilateral with four agents, and the desired formation vector is given by

$$h = [1 \quad 1 \quad -1 \quad 1 \quad -1 \quad -1 \quad 1 \quad -1]^T$$
$$\dot{h} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T. \quad (35)$$

4.4 Controller Gains

The control system for training, described by Eq. (26), used the feedback gain matrix obtained by means of the Linear Quadratic Regulated theory, which is obtained through Eq. (9) and (11) that is given by

$$F_{v_i} \cong \begin{bmatrix} -100 & -101.45 & 0 & 0 \\ 0 & 0 & -100 & -101.45 \end{bmatrix}. \quad (36)$$

4.5 Numerical Results

In the simulation, the initial states of the multi-agent system were considered as uniformly distributed random variables. Thus, the agents must leave their position and maintain a given desired formation and the response of the system is illustrated in Figure 5.

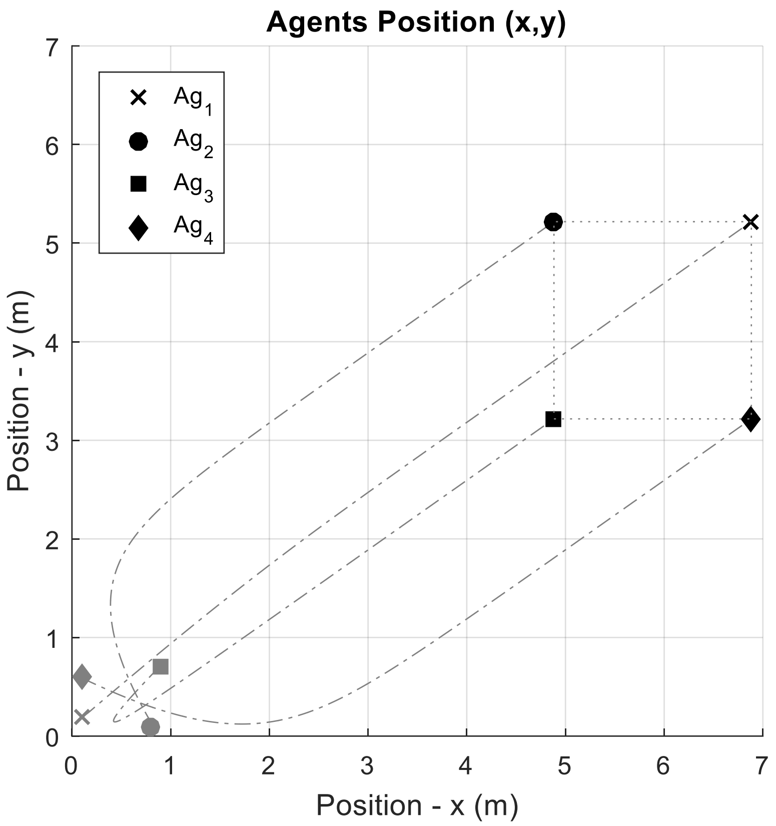

Figure 5. Position of agents in the Cartesian coordinate system for a square formation. Gray symbols indicate the initial positions and the black symbols represent the final positions of the agents.

The Figure 5 illustrates the formation of a multi-agent system, where Ag_1, Ag_2, Ag_3, and Ag_4 represent the dynamics of agents 1, 2, 3, and 4, respectively. Also, in this figure, it is noted that the agents move in space at a constant speed because the vehicles have an initial velocity different from zero; thus, during the training process the velocity is kept constant. In addition, although the agents reach the desired formation, they continue moving in space, i.e., their speed is different from zero until the final instant. This behavior is illustrated by the graph presented in Figure 6.

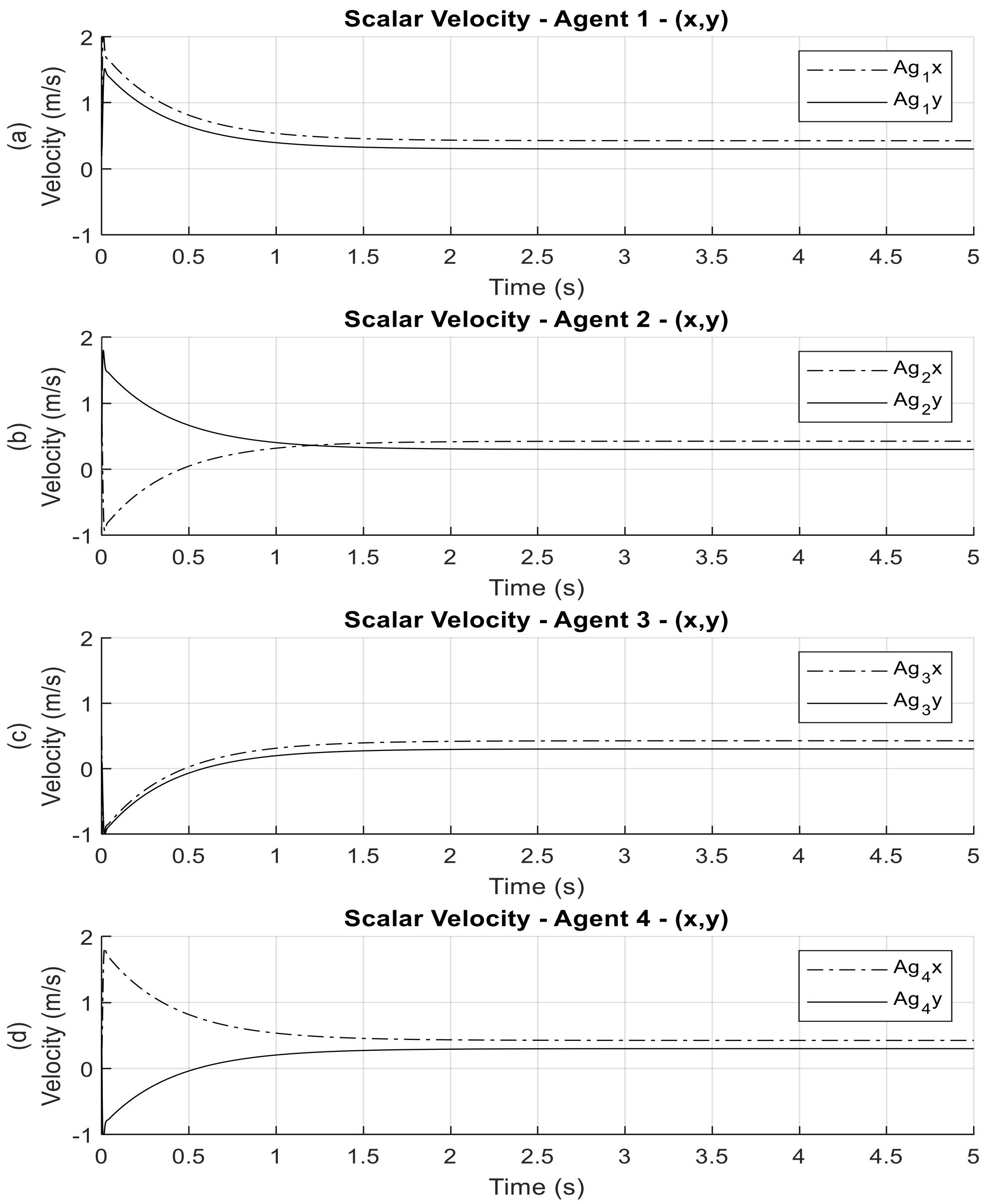

Figure 6. Agent velocity versus time for square formation.

In Figure 6 is shown the relationship between speeds along the (x, y) axes. Note that the desired formation is maintained because the velocities at the final instants are approximately equal and different from zero; thus, the formation continues moving along the axes.

Based on an elaboration of the communication graph of the agents, illustrated in Figure 4, it is possible to verify the way each agent communicates with its neighbors. To maintain the desired formation, it is necessary that each agent maintains a certain distance from its respective adjacent agents. The behavior of the agents for analysis purposes, considering their form of communication and their respective distances, is illustrated in Figure 7. As well as this figure, changes in the Euclidean distance between agents over multiple iterations is shown.

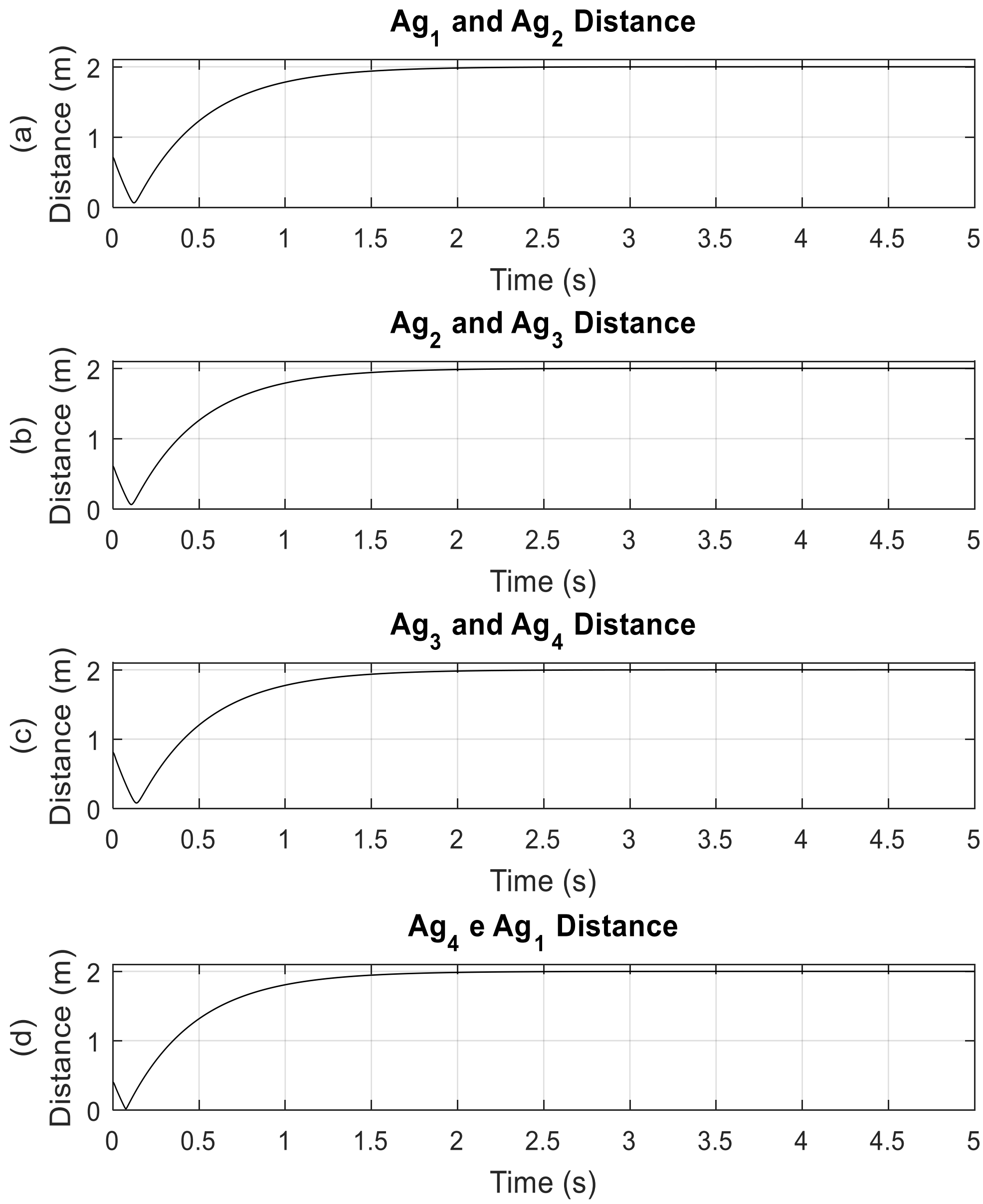

Figure 7. Distance between agents with respect to time for square formation.

6. Conclusion

Strategies and design proposals for a multi-agent control system based on LQR was presented. A general procedure for multiagent formation via optimal control LQR-based approach were presented and evaluated in model mathematical of four quadcopter fleet.

A connection between graph theory and a formation control method was presented. The multi-agent system of model of four quadcopters were used to computationally evaluate the proposal. The LQR control design approach had shown to be a good alternative to real world implementations for control formation of quadcopters. The gains adjustments to the feedback gain matrix were performed using the Linear Quadratic Regulator weighting matrices, which ensures optimality of the closed loop system, i.e., with minimum control effort and stability robustness (phase and gain margins). For simplicity, the analysis was performed in a restricted manner, using only undirected graphs; however, analogous results are valid for directed graphs.

About the graph theory, the following comment are relevant: some cooperative control algorithms for multi-agent systems, such as consensus and formation, have been presented. Graph theory is the main communication stream between agents to facilitate the use of directed, undirected graphs and Laplacian graphs in consensus and formation algorithms. In addition to the results obtained by simulation, an interesting observation is the importance of the Laplacian matrix influencing most of the results because it is a summarized representation of the system.

This design makes possible to explore the more complex dynamics of an agent, i.e., to use a mathematical model of higher order and compare the results for directed and undirected graphs in future work. Also, any graph topology with multiple communication connections can be used to study modifications to the consensus algorithm and formation convergence. Motivation for future work includes implementing control algorithms in hardware for synchronized robots equipped with location-sensing abilities to perform aerial formation or do tasks beyond human capabilities.

7. Acknowledgement

We thank UFMA-PPGEE, the Laboratory of Embedded Systems and Intelligent Controls (LABSECI) for making this research possible and for the technical/scientific contributions given to the development of this work. We acknowledge CAPES and CNPq for promoting and supporting the advanced studies that contributed to the realization of this work. Also, we are especially grateful to FAPEMA for encouraging high-level research in the State of Maranhão.

8. References

1. Arokiasami, W. A. (2016). *Designing a multi-agent framework for unmanned aerial/ground vehicles.* Singapore: PhD. Thesis National University of Singapore.

2. Ataka, A., Tnunay, H., & Inovan, R. (2013). Controllability and observability analysis of the gain scheduling based linearization for uav quadrotor. *International Conference on Robotics Biomimetics, Intelligent Computational Systems (ROBIONETICS)*.

3. Bamieh, B., Paganini, F., & Dahleh, M. (2002). Distributed control of spatially-invariant systems. *IEEE Trans. Automat. Control, 47*, pp. 1091–1107.

4. Bhuvaneswari, N. S., Praveena, R., & Divya, R. (2012). System Identification and Modeling for Interacting and Non-Interacting Tank Systems Using Intelligent Techniques. *International Journal of Information Sciences and Techniques (IJIST), 2*(5).

5. Cepeda-Gomez, R., & Perico, L. F. (2015). Formation Control of Non-holonomic Vehicles under Time Delayed Communications. *IEEE Transactions on Automation Science and Engineering, 12*(3), pp. 819-826.

6. Dimarogonas, D. V., & Kyriakopoulos, K. J. (2009, Março 27). Inverse Agreement Protocols With Application to Distributed Multi-Agent Dispersion. *IEEE Transactions on Automatic Control, 54*(3).

7. Elkholy, H. (2014). *Dynamic modeling and control of a quadrotor using linear and nonlinear approaches.* Master's Thesis, the American University in Cairo: AUC Knowledge Fountain.

8. Fax, J. A. (2002). Optimal and cooperative control of vehicle formations. *Ph.D. dissertation, California Inst. Technol., Pasadena, CA*.

9. Fax, J. A., & Murray, R. M. (2004). Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control, 49*(9), 1465-1476.

10. Feng, T., Zhang, J., Tong, Y., & Zhang, H. (n.d.). Consensusability and Global Optimality of Discrete-Time Linear Multiagent Systems. *IEEE Transactions on Cybernetics, 52*(8), pp. 8227-8238. doi:10.1109/TCYB.2021.3049910

11. Fonseca Neto, J. V., Abreu, I. S., & Silva, F. N. (2010). Neural–Genetic Synthesis for State-Space Controllers Based on Linear Quadratic Regulator Design for Eigenstructure Assignment. *IEEE Transactions on Systems, Man, and Cybernetics, 40*(2), 266-285.

12. Grimstad, B. (2009). Studies In Static Output Feedback Control. *Tese de Mestrado em Norwegian University of Science and Technology*.

13. Guo, Y., Zhou, J., & Liu, Y. (2018). Distributed Lyapunov-based model predictive control for collision avoidance of multi-agent formation. *IET Control Theory & Applications, 12*(18), 2569 - 2577.

14. Horn, R. A., & Johnson, C. R. (2013). *Matrix Analysis* . Cambridge University Press.

15. Laub, A. J. (1979). A Schur Method for Solving Algebraic Riccati Equations. *IEEE Transactions on Automatic Control, Ac-24*(6), 913-921.

16. Lewis, F. L., Vrabie, D. L., & Syrmos, V. L. (2012). *Optimal Control* (Vol. Third Edition). John Wiley & Sons, Inc.

17. Lewis, F., Zhang, H., & Hengster-Morvic, K. (2014). *Cooperative Control of Multi-Agent Systems Optimal and Adaptive Design Approaches.* New York: Springer.

18. Li, Z., & Dua, Z. (2017). *Cooperative Control of Multi-Agent Systems.* London: CRC Press.

19. Lyu, H. (2017). Multivariable Control of a Rolling Spider Drone. *Master's Theses. University of Rhode Island,*. Retrieved from http://digitalcommons.uri.edu/theses/1064/

20. Mesbahi, M., & Egerstedt, M. (2010). *Graph Theoretic Methods in Multiagent Networks.* New Jersey: Princeton University Press.

21. Smith, B. S., Egerstedt, M., & Howard, A. (2009, Junho). Automatic Generation of Persistent Formations for Multi-Agent Networks Under Range Constraints. *CM/Springer Mobile Networks and Applications Journal (MONET), 14*(3), pp. 322-335.

22. Tanner, H. G., Pappas, G. J., & Kumar, V. (2004). Leader-to-Formation Stability. *IEEE Transactions on Robotics And Automation, 20*(3), 443.

Appendix

Appendix 1. Product of Kronecker in Formation

An operator used in graph theory is the Kronecker product, comprising the product between two matrices or vectors, this operation results in a block matrix, and is represented by the symbol $\otimes$. For example, assuming that the matrix $E \in \Re^{n\times m}$ and the matrix $F \in \Re^{p\times q}$ [7], the Kronecker product of $E \otimes F$ is defined as

$$E \otimes F = \begin{bmatrix} e_{11}F & \cdots & e_{1m}F \\ \vdots & \ddots & \vdots \\ e_{n1}F & \cdots & e_{nm}F \end{bmatrix}. \tag{A1}$$

The Kronecker product is a useful tool in modeling and manipulating the equations that rules the dynamics of a given formation. For instance, if $\dot{x}_{v_i} = Ax_{v_i}$ represents the dynamics of a single agent, the dynamics of N agents are represented by $\dot{x} = (I_N \otimes A)x$. Another example, if A is a matrix, $N \times N$ represents a set of data of N agents, and this operation needs to be applied to each value of the position vector n. The manipulation is represented by concatenating N dimension vectors n into a single vector of dimension Nn and multiplying it by $A \otimes I_n$.

The associative property of the Kronecker product facilitates the manipulation of matrices. is represented by

$$(I_N \otimes X)(Y \otimes I_s) = (Y \otimes I_r)(I_N \otimes X) = Y \otimes X\,, \tag{A2}$$

where $X \in \Re^{r\times s}$ and $Y \in \Re^{N\times N}$.