Distributed Formation Control of Double-Integrator Vehicles in the Presence of Unknown Constant Disturbances

Pedro Miguel Dias Trindade
pedro.m.dias.trindade@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2019

Abstract

This dissertation describes the development of robust distributed and cooperative control algorithms for formation tracking by teams of vehicles modeled as double integrators, with application to multirotor vehicles. For this effect, consensus-based protocols are considered, to achieve coordination between the vehicles without explicit communication. A consensus protocol for agents modeled as triple integrators is proposed and analyzed, providing necessary and sufficient conditions for convergence. Also, a new criteria for convergence of the existing second-order consensus protocol is provided. The effect of constant disturbances on the capability to achieve consensus is then analyzed and the third-order consensus protocol is then used to propose integral action to the formation tracking controller used for double integrator agents. The development and implementation of an experimental setup installed in a controlled indoor environment, which allowed for testing, acquisition of experimental data, and validation of the proposed algorithms using physical multirotor vehicles is also described.
Keywords: Consensus-Based Protocols, Formation Control, Integral Action, Multirotor Vehicles.

1. Introduction

Formation control is an important topic of research in the coordinated motion of multiple unmanned autonomous vehicles. Moving in formation can have several advantages on the overall system, such as increased redundancy and robustness, and reduced cost [1]. This problem presents several challenges, mainly related to the lack of total information by each agent, but also to the desire to use a distributed, or a decentralized approach. In decentralized approaches, each agent makes its own decisions independently from the others, therefore a central controller, coordinator or supervisor does not exist, making the problem more challenging. Despite the challenges, a decentralized approach is still the one that presents more potential applications, as it provides scalability and robustness to the system.

A survey on the topic of multi-agent formation control can be found in [2]. There, the authors distinguish the several proposed approaches based on the required sensing capabilities of the agents and on the amount of interaction necessary between them, categorizing the formation control methods into position-, displacement- and distance-based. The position-based approach considers that each agent has access to measurements in the inertial frame (e.g. absolute position measurements). In this case, each agent can be equipped with a control law to drive its position to a desired position, thus achieving the prescribed formation without the need to interact with others. This is however, the most demanding approach in terms of the sensing capability of each agent. The displacement-based approach, considers that the agents can only measure relative quantities (e.g., measurement of the relative position, or displacement, to another agent), and that they have the same sense of orientation. However, it is necessary to have more interactions between agents, in order to overcome the reduced sensing capability. For agents modeled as single integrators, this approach is studied under directed interaction topologies in [3], considering consensus-based protocols. For the case of agents modeled as double integrators, it was studied in [4], under directed and undirected interaction

Figure 1: Two multirotors flying in formation.

topologies. As for the general case of agents with linear dynamics, it has been studied in [5]. Finally, in the distance-based approach, it is assumed that agents only have access to relative measurements, but do not share a sense of orientation, and therefore, formations are stabilized based only on the distance between the agents, not accounting for the orientation of the formation. This approach is the less demanding in terms of sensing capability of the agents, however, it requires the use of more elaborate control laws, and more interactions between the agents. It is commonly studied under the use of gradient control laws, which are defined using artificial potential fields. For single integrator modeled agents, it has been studied in [6] and [7], and for double integrator modeled agents in [8] and [9].

Considering the importance of developing robust control algorithms for the coordinated motion of multiple unmanned autonomous vehicles, this work aims to develop distributed control algorithms for formation tracking, and apply the devised solutions to multirotor vehicles. Noting that it is usual for multirotors to have access to measurements of their orientation, a displacement-based approach is considered. This approach is studied using consensus based protocols, for the case of triple integrator modeled agents, and the results are used to introduce integral action to the formation tracking controller for vehicles modeled as double integrators, to enable disturbance rejection.

2. Notation and Preliminaries

The set of real numbers is denoted by $\mathbb{R}$, the subset of positive real numbers by $\mathbb{R}^+$, real numbers except zero by $\mathbb{R}_{\neq 0}$, and the set of complex numbers by $\mathbb{C}$. For a complex number $z \in \mathbb{C}$, $\mathrm{Re}\,(z)$ denotes its real part, and $\mathrm{Im}(z)$ its imaginary part. $\mathbb{R}^m$ denotes the m-dimensional euclidean space, with norm $\|\mathbf{x}\| = \sqrt{\mathbf{x}\cdot\mathbf{x}}$ for all $\mathbf{x} \in \mathbb{R}^m$, where $\mathbf{a}\cdot\mathbf{b} \coloneqq \mathbf{a}^\mathsf{T}\mathbf{b}$ denotes the inner product between two vectors. The dot notation is used to define the time derivative (as in $\dot{\mathbf{x}}$), and the number of dots its order. $\mathbf{I}_n$ is used to denote the $n\times n$ identity matrix, and $\mathbf{0}_{n\times m}$ an $n\times m$ matrix of zeros. $\mathbf{0}$ is used when the size can be determined.

When working with networks of agents, the network information flow is typically described using graph theory. A directed graph (or digraph) D, consists of a pair of sets $(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ is a nonempty finite set of nodes, and $\mathcal{E} \in \mathcal{V}^2$ is a finite set of ordered pairs of nodes, called edges. Undirected graphs are considered a particular case of digraphs in which the edges are unordered pairs of nodes, hence definitions are provided for digraphs only. If a digraph is weighted, the weight associated to an edge connecting node i to j, is denoted by $k_{ij} \in \mathbb{R}^+$. If the graph is not weighted, then all weights are considered to be one. If there is no edge connecting i to j, $k_{ij} = 0$. When a directed edge connects i to j, i is said to be the parent node, and j the child node. The set of parent nodes of i is called the neighborhood of i, denoted $\mathcal{N}_i \subset \mathcal{V}$. A directed path is defined as an ordered sequence of edges, connecting two nodes. When there is at least one node that has a directed path to all others, the digraph is said to have a directed spanning tree. The adjacency matrix associated to the weighted digraph D, is denoted by $\mathbf{A}_D = [a_{ij}]$, and defined by $a_{ij} = k_{ji}$ for $i \neq j$ and $a_{ij} = 0$ for $i = j$. Then, $\mathbf{L} = [l_{ij}]$ is defined as $l_{ij} = -a_{ij}$, for $i \neq j$, and $l_{ij} = \sum_{j\neq i} a_{ij}$ otherwise. A relevant property of this matrix is the null row sum, meaning that it has at least one null eigenvalue, for which the associated eigenvector is a vector of ones, $\mathbf{1}_n$, where n is the number of nodes.

3. Consensus Protocols

Here, consensus protocols used for distributed coordination of the multiple agents are presented. Firstly, protocols for agents with single and double integrator dynamics are presented, and a protocol for agents with triple integrator dynamics is proposed. Then, a convergence analysis to the proposed protocol is performed, providing necessary and sufficient conditions for convergence.

3.1. Single and Double Integrator Dynamics

Consider a group of n agents with single integrator dynamics, each described by $\dot{\alpha}_i = \mu_i$, with $\alpha_i, \mu_i \in \mathbb{R}$, where μ_i is the control input of the agent. In [3], the following consensus protocol has been proposed for this system

$$\mu_i = -\sum_{j\in\mathcal{N}_i} k_{ji}\left(\alpha_i - \alpha_j\right). \tag{1}$$

The goal of protocol (1), is to drive the agents to a consensus, i.e., $|\alpha_i - \alpha_j| \to 0$ as $t \to \infty$. Note that each agent i only needs to know the difference from its state to the state of its neighbors given by $(\alpha_i - \alpha_j)$, $j \in \mathcal{N}_i$. From the definition of $\mathbf{L}$, it is possible to write (1) in vector form as $\boldsymbol{\mu} = -\mathbf{L}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_n]^\mathsf{T} \in \mathbb{R}^n$ and $\boldsymbol{\mu} = [\mu_1 \cdots \mu_n]^\mathsf{T} \in \mathbb{R}^n$. As shown, for example, in [3], the existence of a directed spanning tree on the digraph, which describes the interaction topology for this system, is a necessary and sufficient condition for achieving consensus.

Consider now that the agents are modeled with double integrator dynamics, that is,

$$\begin{cases} \dot{\alpha}_i = \beta_i \\ \dot{\beta}_i = \mu_i \end{cases}, \tag{2}$$

with $\alpha_i, \beta_i, \mu_i \in \mathbb{R}$, where μ_i is the control input of the agent. The following consensus protocol was

proposed in [4],

$$\mu_i = -\sum_{j\in\mathcal{N}_i} k_{ji}\left[(\alpha_i-\alpha_j)+\gamma\left(\beta_i-\beta_j\right)\right], \qquad (3)$$

with $\gamma\in\mathbb{R}^+$. The goal of protocol (3) is to drive $|\alpha_i-\alpha_j|\to 0$, and consequently $|\beta_i-\beta_j|\to 0$, as $t\to\infty$. In vector form, the control law becomes $\boldsymbol{\mu}=-\mathbf{L}\boldsymbol{\alpha}-\gamma\mathbf{L}\boldsymbol{\beta}$, with $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu}\in\mathbb{R}^n$. Unlike the single integrator case, it is shown in [4] that the existence of a spanning tree is not a sufficient condition for reaching consensus. However, a sufficient but not necessary condition is provided.

3.2. Triple Integrator Dynamics

For agents modeled by triple integrator dynamics, i.e.,

$$\begin{cases}\dot{\vartheta}_i=\alpha_i\\ \dot{\alpha}_i=\beta_i\\ \dot{\beta}_i=\mu_i\end{cases}, \qquad (4)$$

with $\vartheta_i,\alpha_i,\beta_i,\mu_i\in\mathbb{R}$, where μ_i is the control input of the agent, the following consensus protocol is proposed

$$\mu_i = -\sum_{j\neq i} k_{ji}\left[\alpha_{ij}+\gamma\beta_{ij}+\zeta\vartheta_{ij}\right], \qquad (5)$$

where $\gamma,\zeta\in\mathbb{R}_{\neq 0}$, and the subscript ij is used to denote the difference between the state of vehicles i and j. To achieve consensus, the goal is to have $|\vartheta_i-\vartheta_j|\to 0$, $|\alpha_i-\alpha_j|\to 0$ and $|\beta_i-\beta_j|\to 0$ as $t\to\infty$. Note that, in matricial form, (5) becomes $\boldsymbol{\mu}=-\mathbf{L}\boldsymbol{\alpha}-\gamma\mathbf{L}\boldsymbol{\beta}-\zeta\mathbf{L}\boldsymbol{\vartheta}$, with $\boldsymbol{\vartheta},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu}\in\mathbb{R}^n$, hence, the feedback actuated system becomes

$$\begin{bmatrix}\dot{\boldsymbol{\vartheta}}\\ \dot{\boldsymbol{\alpha}}\\ \dot{\boldsymbol{\beta}}\end{bmatrix}=\begin{bmatrix}\mathbf{0}_{n\times n} & \mathbf{I}_n & \mathbf{0}_{n\times n}\\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{I}_n\\ -\zeta\mathbf{L} & -\mathbf{L} & -\gamma\mathbf{L}\end{bmatrix}\begin{bmatrix}\boldsymbol{\vartheta}\\ \boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix}=\mathbf{H}\begin{bmatrix}\boldsymbol{\vartheta}\\ \boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix}. \qquad (6)$$

3.3. Convergence Analysis

This section focuses on the analysis of the convergence properties of the proposed third order consensus protocol (5). To this end, the eigenvalues of $\mathbf{H}$ must be studied. These correspond to the solution of $\det(\lambda\mathbf{I}_{3n}-\mathbf{H})=0$ (where $\det(\lambda\mathbf{I}_{3n}-\mathbf{H})$ is the characteristic polynomial of $\mathbf{H}$). Noting that all the blocks of $\lambda\mathbf{I}_{3n}-\mathbf{H}$ commute, the result described in Theorem 1 in [10] can be used to conclude that

$$\det(\lambda\mathbf{I}_{3n}-\mathbf{H})=\det\left(\lambda^3\mathbf{I}_n+(\gamma\lambda^2+\lambda+\zeta)\mathbf{L}\right). \qquad (7)$$

It is known that if a matrix $\mathbf{L}\in\mathbb{R}^{n\times n}$ has eigenvalues η_i, $i=1,\dots,n$, then the matrix $a\mathbf{L}+b\mathbf{I}_n$ has eigenvalues $a\eta_i+b$. Using this property on (7), and recalling that the determinant of a matrix is equal to the product of its eigenvalues, yields

$$\det(\lambda\mathbf{I}_{3n}-\mathbf{H})=\prod_{i=1}^{n} g_i(\lambda), \qquad (8)$$

where $g_i(\lambda):=\lambda^3+(\gamma\lambda^2+\lambda+\zeta)\eta_i$. By definition, $\zeta\neq 0$, hence it is possible to conclude that $\lambda=0$ is a root of $g_i(\lambda)$ if and only if $\eta_i=0$. Therefore, for each $\eta_i=0$, there are exactly three null eigenvalues in $\mathbf{H}$. Conditions on the eigenvalues of $\mathbf{H}$ that allow to reach consensus are now presented. The following result can be considered an extension of Lemma 4.1 in [4].

Lemma 1
The consensus protocol (5) *for agents modeled by triple integrator dynamics reaches consensus if and only if* $\mathbf{H}$ *has exactly three null eigenvalues, and the remaining eigenvalues have negative real part. More concretely, when reaching consensus (for large* t*),* $\boldsymbol{\beta}(t)\to\mathbf{1}_n\mathbf{r}^\mathsf{T}\boldsymbol{\beta}(0)$, $\boldsymbol{\alpha}(t)\to\mathbf{1}_n\mathbf{r}^\mathsf{T}\boldsymbol{\alpha}(0)+\mathbf{1}_n\mathbf{r}^\mathsf{T}\boldsymbol{\beta}(0)t$ *and* $\boldsymbol{\vartheta}(t)\to\mathbf{1}_n\mathbf{r}^\mathsf{T}\boldsymbol{\vartheta}(0)+\mathbf{1}_n\mathbf{r}^\mathsf{T}\boldsymbol{\alpha}(0)t+\mathbf{1}_n\mathbf{r}^\mathsf{T}\boldsymbol{\beta}(0)\frac{t^2}{2}$*, where* $\mathbf{r}$ *is a non-negative left eigenvector of* $\mathbf{L}$ *associated to the null eigenvalue, and is such that* $\mathbf{1}_n^\mathsf{T}\mathbf{r}=1$.

Considering Lemma 1, the eigenvalues of $\mathbf{H}$ are studied, providing necessary and sufficient conditions for reaching consensus.

Theorem 1
The third order consensus protocol (5) *achieves consensus asymptotically, if and only if the associated digraph has a directed spanning tree,* $\zeta>0$*, and*

$$\begin{cases}\gamma>\max\limits_{\eta_i\neq 0}\sqrt{\dfrac{1-\xi_i^2}{\xi_i\omega_{n_i}}}\\ \zeta<\min\limits_{\eta_i\neq 0}\left[\dfrac{\omega_{n_i}}{\xi_i}\left(\gamma-\sqrt{\dfrac{1-\xi_i^2}{\xi_i\omega_{n_i}}}\right)\right]\end{cases},$$

where $\omega_{n_i}=|\eta_i|$ *and* $\xi_i=\frac{\mathrm{Re}(\eta_i)}{\omega_{n_i}}$ *represent the natural frequency and damping coefficient, respectively, associated with the* i*-th eigenvalue of* $\mathbf{L}$.

From Theorem 1, the following result is obtained.

Corollary 1.1
The consensus protocol (3) *applied to the double integrator multi-agent system reaches consensus asymptotically if and only if the digraph describing the interaction topology of the agents has a directed spanning tree and*

$$\gamma>\max_{\eta_i\neq 0}\sqrt{\frac{1-\xi_i^2}{\xi_i\omega_{n_i}}},$$

where $\omega_{n_i}=|\eta_i|$ *and* $\xi_i=\frac{\mathrm{Re}(\eta_i)}{\omega_{n_i}}$ *represent the natural frequency and damping coefficient, respectively, associated with the* i*-th eigenvalue of* $\mathbf{L}$.

Note that Corollary 1.1 can now replace the result previously presented in [4], since it provides not only sufficient but also necessary conditions for reaching consensus. The following result is now presented, describing the effect of a constant disturbance on the overall convergence of the system.

Proposition 1
In the conditions of Theorem 1, the consensus protocol (5) *achieves consensus for the state variables* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\beta}$ *in the presence of a constant disturbance,* $\mathbf{d} \in \mathbb{R}^n$. *Moreover, the consensus protocol* (3) *is able to achieve consensus for the state variables* $\boldsymbol{\beta}$ *in the conditions of Corollary 1.1. Furthermore, if all entries of* $\mathbf{d}$ *are equal, consensus is achieved for all variables.*

4. Formation Tracking Controller

Consider a group of n vehicles modeled as double integrators, to which the desired formation can be given as a desired position in time for each vehicle i, i.e., a trajectory $\mathbf{p}_i^{\mathbf{d}}(t) \in \mathbb{R}^3$, defined for all $t > 0$, with known and continuous first and second time derivatives. Their dynamics are then described as

$$\begin{cases} \dot{\mathbf{p}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = \mathbf{u}_i \end{cases} \implies \begin{cases} \dot{\tilde{\mathbf{p}}}_i = \dot{\mathbf{p}}_i - \dot{\mathbf{p}}_i^{\mathbf{d}} = \tilde{\mathbf{v}}_i \\ \dot{\tilde{\mathbf{v}}}_i = \dot{\mathbf{v}}_i - \ddot{\mathbf{p}}_i^{\mathbf{d}} = \tilde{\mathbf{u}}_i \end{cases}, \quad (9)$$

where $\mathbf{p}_i, \mathbf{v}_i \in \mathbb{R}^3$ denote the position and velocity of the i-th vehicle, respectively, $\mathbf{u}_i \in \mathbb{R}^3$ is the control input of the vehicle, in this case, its acceleration, and $\tilde{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{p}_i^{\mathbf{d}}$ is the trajectory tracking error. Let $\mathbf{p}_{ij} = (\mathbf{p}_i - \mathbf{p}_j)$ and $\mathbf{p}_{ij}^{\mathbf{d}} = (\mathbf{p}_i^{\mathbf{d}} - \mathbf{p}_j^{\mathbf{d}})$ denote the relative position, and the desired relative position of vehicle i with respect to vehicle j, respectively. In order to track a prescribed formation, the goal is to have $\mathbf{p}_{ij} - \mathbf{p}_{ij}^{\mathbf{d}} \to \mathbf{0}$. However, note that $\mathbf{p}_{ij} - \mathbf{p}_{ij}^{\mathbf{d}} = \tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_j$, meaning that, the goal is equivalent to $\tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_j \to \mathbf{0}$. The dynamics, described in (9), are decoupled, and so, the controllers can be designed independently for each axis. Comparing the dynamics over each axis with the ones described in (2), it is possible to conclude they are the same. Also, note that the control objective, is the same as the one described for protocol (3). Therefore, protocol (3) can be used to achieve formation tracking, and the control input $\tilde{\mathbf{u}}_i$ for the error dynamics becomes

$$\tilde{\mathbf{u}}_i = -\sum_{j \neq i} k_{ji} \left[(\tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_j) + \gamma \left(\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j \right) \right], \quad (10)$$

which is guaranteed to drive the vehicles into formation, under the conditions of Corollary 1.1. The control input for the i-th vehicle can then be recovered, yielding

$$\mathbf{u}_i = \ddot{\mathbf{p}}_i^{\mathbf{d}} - \sum_{j \neq i} k_{ji} \left[\left(\mathbf{p}_{ij} - \mathbf{p}_{ij}^{\mathbf{d}} \right) + \gamma \left(\mathbf{v}_{ij} - \dot{\mathbf{p}}_{ij}^{\mathbf{d}} \right) \right]. \quad (11)$$

The same rationale could be used to define a controller for vehicles modeled as triple integrators using the consensus protocol (5).

4.1. Inclusion of Integral Action

When performing formation tracking, using the control law described in (11), the distributed multi-vehicle system is able to track the prescribed formation, under the conditions of Corollary 1.1. However, real systems are susceptible to a number of non-idealities, such as disturbances, modeling errors and actuator dead-zones, which can deteriorate the ability to achieve their goal. It follows from Proposition (1) that (11) is not able to reject a constant disturbance acting on the system. To cope with these effects, integral action is proposed. This can be achieved considering the integral of the position tracking error, modeled by an extra state on (9), described as $\dot{\tilde{\mathbf{g}}}_i = \tilde{\mathbf{p}}_i$. Note that the system including this new state, becomes a triple integrator system. It is then straightforward to conclude that consensus protocol (5) can be used, and

$$\tilde{\mathbf{u}}_i = -\sum_{j \neq i} k_{ji} \left[\tilde{\mathbf{p}}_{ij} + \gamma \tilde{\mathbf{v}}_{ij} + \zeta \tilde{\mathbf{g}}_{ij} \right]. \quad (12)$$

Noting that

$$\tilde{\mathbf{g}}_{ij} = \int_{t_0}^{t} \tilde{\mathbf{p}}_{ij} \, dt = \int_{t_0}^{t} \left(\mathbf{p}_{ij} - \mathbf{p}_{ij}^{\mathbf{d}} \right) dt,$$

the control input for the vehicle is recovered. It follows from Proposition 1, that this controller is able to track the formation in the presence of a constant disturbance, in the conditions of Theorem 1.

4.2. Inclusion of a Goal Seeking Term

Now that a controller that achieves formation tracking has been designed, the positions of all vehicles must be driven to their desired positions. This is accomplished using goal seeking terms, which take the form of a trajectory tracking controller, to drive the position of a vehicle to its desired position. It could be added to all the vehicles, as in [4], but all vehicles would need access to their own state. Note however that it only needs to be added to one vehicle. Consider adding goal seeking terms to the consensus protocol (5), to drive the errors of all the vehicles to zero. To do so, $\boldsymbol{\mu}$ must be designed for system (4), such that the feedback actuated system becomes $\dot{\mathbf{x}} = \mathbf{H_G x}$, where $\mathbf{x} = \left[\boldsymbol{\vartheta}^{\mathsf{T}} \; \boldsymbol{\alpha}^{\mathsf{T}} \; \boldsymbol{\beta}^{\mathsf{T}} \right]^{\mathsf{T}}$, and $\mathbf{H_G}$ is a stable matrix, i.e., all its eigenvalues have negative real part. To add a goal seeking term, the following is added to μ_i

$$\mu_i^G = -k_i^G \left[K_\alpha \alpha_i + \zeta K_\vartheta \vartheta_i + \gamma K_\beta \beta_i \right], \quad (13)$$

where $K_\alpha, K_\beta, K_\vartheta \in \mathbb{R}^+$ are the gains of the goal seeking term, and $k_i^G \in \{0, 1\}$ is a variable controlling whether the i-th agent has a goal seeking term, or not. It is considered that only one vehicle uses this term. For this reason, the following result is introduced.

Proposition 2
The vehicles reach their desired positions, only if the vehicle with a goal seeking term has a directed path to all others.

In the scope of this work, it is assumed that there is a leader vehicle. This is a vehicle that only has outgoing links in the digraph that describes the interaction topology, and will correspond to a line of zeros in $\mathbf{L}$. Then, if there is a directed spanning tree in the digraph, this is also the only vehicle that has a directed path to all others. The controller for this leader vehicle can be designed independently of the formation tracking controller. If both the trajectory tracking controller and the formation tracking controller are stable, the overall system is stable, and all vehicles are able to achieve their desired position. Without loss of generality, let the leader vehicle be denoted by vehicle 1. Its consensus seeking term is null, and this is the only vehicle which considers a goal seeking term. Hence, the control input for the error dynamics associated to this vehicle becomes $\tilde{\mathbf{u}}_1 = -K_p\tilde{\mathbf{p}}_1 - K_v\tilde{\mathbf{v}}_1 - K_i\tilde{\mathbf{g}}_1$, where $K_p = K_\alpha$, $K_i = \zeta K_\vartheta$ and $K_v = \gamma K_\beta$ are the proportional, integral and derivative gains, respectively. Then, its control input considering the transformation from the error dynamics to the vehicle dynamics, yields

$$\mathbf{u}_1 = \ddot{\mathbf{p}}_1^{\mathbf{d}} - K_p\tilde{\mathbf{p}}_1 - K_v\tilde{\mathbf{v}}_1 - K_i\int_{t_0}^{t}\tilde{\mathbf{p}}_1\,dt. \tag{14}$$

Note that the control law for the leader vehicle, which can be independently designed taking only into account its own state vector, consists on the feedback of all the error states associated to this vehicle. This can then be described by an LQR feedback control law, to tune the gains for this PID trajectory tracking controller.

5. Multirotor Tracking Control

The presented algorithms can be applied to any vehicles modeled as double integrators, however, the goal is to implement them on multirotors. This type of vehicle is characterized by having multiple rotors, all generating thrust aligned with the vertical direction of the vehicle. Their actuation is usually the angular speed of each rotor. Then, an inner-outer loop control scheme is typically used to control the attitude (inner-loop) with actuation on the body torques, and the position of the multirotor (outer-loop) with actuation on the total thrust and virtual control inputs defined by the attitude. To define the outer-loop position controller and ultimately apply the controller defined in Section 4, consider Newton's second law of motion, which states that $\sum_i \mathbf{f}_i = \dot{\mathbf{q}}$, where $\mathbf{f}_i \in \mathbb{R}^3$ represents the i-th force applied to the multirotor, $m \in \mathbb{R}^+$ its mass, $\mathbf{v} \in \mathbb{R}^3$ its velocity, and $\mathbf{q} = m\mathbf{v}$ the linear momentum of the vehicle. Since the mass is constant, and considering the forces presented in Fig. 2,

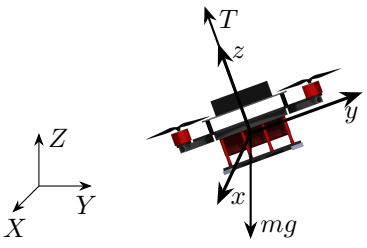


Figure 2: Forces applied to the multirotor.

$$\dot{\mathbf{v}} = \frac{1}{m}\sum_i \mathbf{f}_i = \frac{T}{m}\mathbf{R}\mathbf{e}_3 - g\mathbf{e}_3, \tag{15}$$

where $\mathbf{e}_3 = [0 \quad 0 \quad 1]^\mathsf{T}$, g is the acceleration of gravity, $\mathbf{R}$ represents the rotation matrix from the body to the inertial frame and T represents the norm of the total thrust applied by the rotors. Note that, $\dot{\mathbf{v}} = \mathbf{u}$ is the control input of the double integrator system previously discussed. The goal is then to determine T, and the attitude associated with $\mathbf{R}$, that correspond to the control input $\mathbf{u}$. Note that the matrix $\mathbf{R}$ can be decomposed into three rotation matrices, associated to a sequence of elementary rotations around the principal rotation axis. These rotations refer to the Euler angle representation of the orientation. Different rotation sequences can be used to represent an arbitrary rotation matrix $\mathbf{R}$. In this work, the ZYX Euler angle representation is adopted, for which $\mathbf{R}$ takes the form $\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$. For a rotation matrix, it holds that $\mathbf{R}^{-1} = \mathbf{R}^\mathsf{T}$. It follows that if ($T$,$\phi$,$\theta$) are chosen such that $\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)T\mathbf{e}_3 = \mathbf{u}^*$, where

$$\mathbf{u}^* \coloneqq m\mathbf{R}_z(\psi)^\mathsf{T}\left(\mathbf{u} + g\mathbf{e}_3\right), \tag{16}$$

the double integrator model can be recovered. The angles ϕ and θ, as well as T, associated to $\mathbf{u}$, must then be determined. To do so, it is necessary to know ψ, commonly denoted by yaw angle. Note that the yaw angle varies with time, and can be controlled by providing the inner-loop attitude controller with a desired yaw angle, which was chosen to be constant. To determine ϕ, θ and T, given $\mathbf{u}$, ψ and the transformed control input $\mathbf{u}^* = [u_1^* \; u_2^* \; u_3^*]^\mathsf{T}$, it can be written that

$$\frac{u_1^*}{u_3^*} = \tan(\theta), \qquad \frac{u_2^*}{\sqrt{{u_1^*}^2 + {u_3^*}^2}} = -\tan(\phi).$$

It is then straightforward to determine ϕ and θ, provided that either $u_1^* \neq 0$ or $u_3^* \neq 0$. To determine the total thrust, T, note that $T = \|\mathbf{u}^*\|$.

5.1. Input Limitations

As any other vehicle, a multirotor is subject to limitations imposed by its actuators. In this case, the maximum thrust that the multirotor is capable of is limited. At any time, the multirotor should have a vertical force to balance its weight, i.e., a force of

magnitude mg. Then, it must be able to maneuver around this equilibrium. A descent can be achieve by a decrease in the vertical force that balances the weight. However, to climb, it must be able to produce an higher thrust on the vertical direction. It is defined that it is always necessary to have available, in the vertical direction, a thrust magnitude of $m(g + a_z^{max})$, where $a_z^{max} \in \mathbb{R}^+$ represents the maximum acceleration capability along the vertical axis. Let $T_{max} \in \mathbb{R}^+$ be the maximum thrust that the multirotor can achieve, and $a_{xy}^{max} \in \mathbb{R}^+$ be the maximum acceleration on the horizontal plane. Hence,

$$a_{xy}^{max} = \sqrt{(T_{max}/m)^2 - (g + a_z^{max})^2}. \qquad (17)$$

The saturation along the vertical direction must first be chosen, to ensure that the multirotor is capable of properly maintaining its altitude. Then, the horizontal saturation is a result of the choice made about a_z^{max}. It is concluded that $|u_z| < a_z^{max}$ and $\|\mathbf{u}_{xy}\| < a_{xy}^{max}$, where $u_z \in \mathbb{R}$ and $\mathbf{u}_{xy} \in \mathbb{R}^2$ are the components of the control input $\mathbf{u}$ along the vertical axis, and its projection onto the horizontal plane, respectively.

6. Experimental Setup

6.1. Test Space

The experiments with physical vehicles were performed in a controlled indoor environment, which consists of a flying arena. This is equipped with an OptiTrack Motion Capture system, which provides precise and high-frequency indoor position measurements and also attitude data for several rigid bodies, such as the multirotors. The considered inertial frame is centered in the arena. Although the multirotors are usually equipped with magnetometers, used for attitude estimation, in the considered indoor environment, magnetic readings are too unreliable to be used. For this reason, from here on, an IMU (Inertial Measurement Unit) is considered to be composed only of accelerometers and gyroscopes. However, using only these sensors, it is not possible to obtain a good estimate of the yaw angle. The solution is provided next.

6.2. Multirotors Setup

The physical multirotors used to perform the experiments were the Intel Aero Ready to Fly quadrotors, equipped with the PX4 autopilot. These quadrotors feature an onboard computer with Wi-Fi capabilities, communicating with the autopilot, which was used mainly as a bridge between the autopilot and the network. To be able to use the position control features of the PX4 autopilot to perform takeoff, landing or position hold, the autopilot requires valid position, velocity and yaw estimates. For this reason, the autopilot is provided with the position and attitude measurements from the Motion Capture system. Note that, from the provided attitude measurements, only the yaw measurements are used for estimation. The ROS (Robot Operating System) middleware was used on the onboard computer to inject the Motion Capture data into the autopilot, using a ROS program to decode the received data and another ROS program, MavROS (which provides an interface to communicate with the autopilot), is used to send the decoded data to the autopilot, after translating it into the MavLink protocol. Upon receiving this data, the autopilot fuses it with the IMU measurements using an EKF (Extended Kalman Filter), to acquire valid estimates of the yaw angle, position and velocity. It was necessary to tune this EKF to receive these measurements. Due to the communication overhead, the Motion Capture data reaches the autopilot with some delay. For this reason, the EKF fuses the data in a delayed time horizon. Then the estimate is propagated in time by a Complementary Filter, using only the IMU data. This delay was determined by performing several experiments with different delay values. Then, the one which produces the smallest EKF innovations is used. This delay is expected to have some variation with time (due to dropped messages, network traffic and others), therefore, the confidence of the EKF on these measurements cannot be set too high.

6.3. Simulating Multirotor Vehicles

Due to space constraints, the number of quadrotors that is possible to fly simultaneously is quite limited. For this reason, some quadrotors were simulated in the loop, using the Gazebo simulator as a physics engine, to simulate the physical aspects of the system. The PX4 autopilot is executed for each quadrotor in the simulation. Since the Gazebo simulator is computationally heavy, it was decided to have a computer dedicated to running the simulation, ensuring that this would run near real time.

6.4. Controlling the Vehicles

The control algorithms to test were executed on an external computer, to simplify the implementation, avoiding the setup of a communication network to exchange data among the several vehicles, and simplifying the interface with the simulated quadrotors which would interact with the physical ones. The algorithms were implemented using the Python programming language. To interact with the autopilot, both for the simulated vehicles and for the real ones, the ROS middleware was again used, running the MavROS software.

6.5. Physical Flights

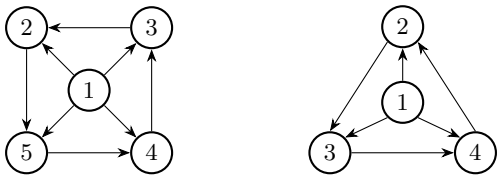
The flights always consider the same phases. First, the takeoff is performed using the capabilities of

the PX4 autopilot, in all vehicles simultaneously. Then, a transition is made into the controller to test. When the experiment ends, all vehicles are given a command to land. The commands to switch between flight phases are manually inserted by the operator during the flight, and sent to all vehicles simultaneously.

Some flights were performed using the trajectory tracking controller described in (14), with $K_i = 0$, to get familiarized with the capabilities of the multirotor used, and to test the capability of the devised experimental setup for testing of multirotor control solutions. To gain more insight into how fast the dynamics of the controllers to test could be, the trajectory tracking controller was tuned. Since the inner-loop dynamics are not taken into account in the design of this controller, its gains cannot be set too high, or the effects of these inner-loop dynamics will become visible, and eventually, destabilize the vehicle. After tuning this controller, the gains were set to $K_v = 4\,\mathrm{s}^{-1}$ and $K_p = 4\,\mathrm{s}^{-2}$. These gains are always used for this controller, which is used by the leader vehicle.

7. Experimental Results

In this section, experimental results obtained using the previously described setup are presented. Simulations proved to yield very similar results to the ones obtained using physical vehicles, hence, only the latter are presented. Two experiments are performed to test the proposed control algorithms, and a video of these experiments is presented in [11]. When presenting results, circles and squares are used to represent the current and initial positions of the vehicles, respectively. The shaded areas in the figures used to present the altitude evolution represent the takeoff and landing parts of the flight, and the remaining area is when the controller to test was acting.



(a) Experiment 1. (b) Experiment 2.

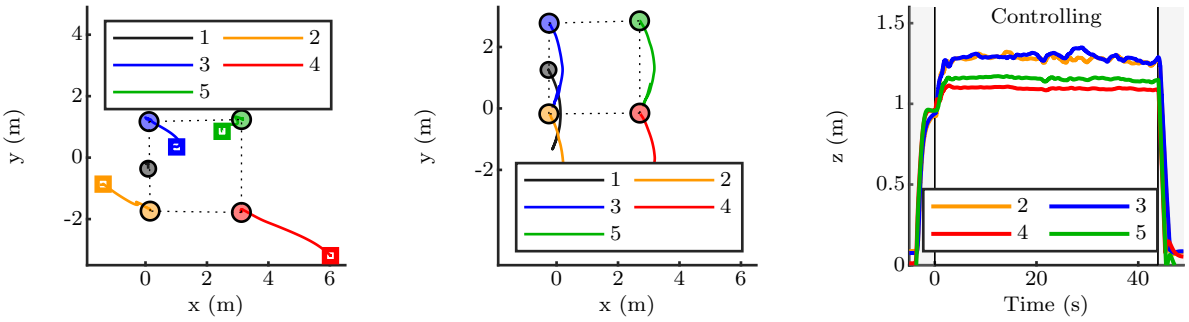
Figure 3: Interaction topologies considered.

7.1. Experiment 1 - Static Formation

In this experiment, the goal is to show the convergence of the vehicles from their initial positions to the desired formation. For this, a virtual leader was considered. The virtual leader is not an actual vehicle, but some other entity of interest, such as a target to follow. The goal of the vehicles is then to control their position with respect to each other and to the virtual leader. An actual object, with known position and velocity, was used as the virtual leader. The position and velocity of this object along the Z-axis were set to zero, ensuring that it only influences the movement of the vehicles in the horizontal plane. Initially, this object is placed at the origin of the inertial frame, and is moved after the vehicles reach the prescribed formation. The goal formation prescribed to the vehicles is a square formation, at an altitude of one meter relative to the virtual leader (which as null altitude). The interaction topology considered is described by the digraph of Fig. 3a. For this graph $\mathbf{L}$ has eigenvalues with imaginary parts, meaning that some oscillations are expected in the movement of the vehicles. The connection weights were set to be $k_{ij} = 1.4$, the derivative gain is $\gamma = 1.2\,\mathrm{s}^{-1}$, and when considering integral action, $\zeta = 0.15\,\mathrm{s}^{-3}$, which can be verified to follow the conditions presented in Theorem 1. Vehicle 1 is the virtual leader previously described. When the virtual leader moves to a position other than the origin, all vehicles are expected to move in the same way. No disturbance was introduced, since this is expected to exist. The physical multirotors used were the vehicles 2 and 3.

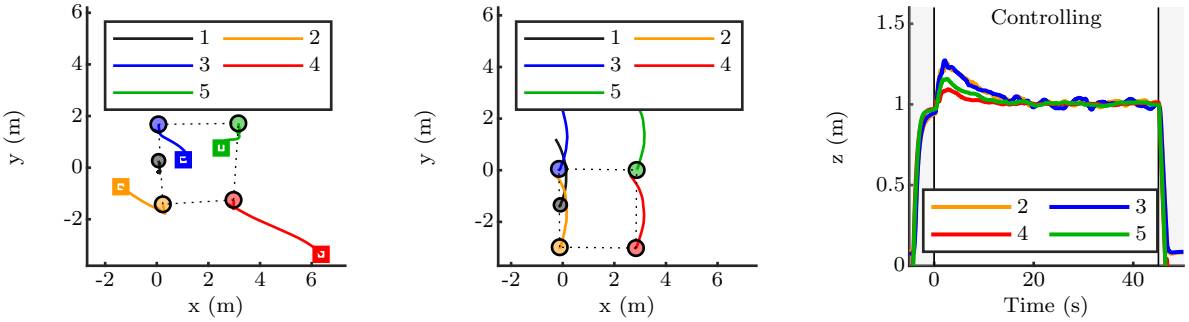
The results obtained without considering integral action are presented in Fig. 4. In Fig. 4a, it is possible to observe the convergence of the vehicles to the desired formation, while the virtual leader remains static. After the initial convergence to the desired formation, the virtual leader is moved. This is depicted Fig. 4b. In Fig. 4c, the time evolution of the altitude of the vehicles is presented. As can be seen, the physical vehicles considered, 2 and 3, both climb to an higher altitude, consequently dragging the simulated vehicles with them. This is evidence of unknown constant disturbances. There are numerous possible causes for these disturbances. Likely causes are: a poor identification of the thrust curve; a mass different than the one considered; the ground effect of the multirotors, due to flying at low altitude, leading to an increase in thrust, not considered in the model.

Regarding the results obtained when using integral action, these are presented in Fig. 5. The initial convergence to the desired formation is presented in Fig. 5a. Some distinctions in relation to Fig. 4a are to be noted. First, note that the initial positions of the vehicles are not exactly the same, hence some differences in the movement of the vehicles were expected. Also, note that there is still an observable error regarding the desired square formation after the initial 10 seconds, most apparent in vehicle 4. This can be an effect of the increased oscillations caused by adding integral action to the controller. It may also be some overshoot on the response. However, this seems more likely to have



(a) Movement at $t = 10\,\mathrm{s}$. (b) Movement at $t = 30\,\mathrm{s}$. (c) Evolution of altitude with time.

Figure 4: Experiment 1 - Results without integral action.



(a) Movement at $t = 10\,\mathrm{s}$. (b) Movement at $t = 30\,\mathrm{s}$. (c) Evolution of altitude with time.

Figure 5: Experiment 1 - Results with integral action.

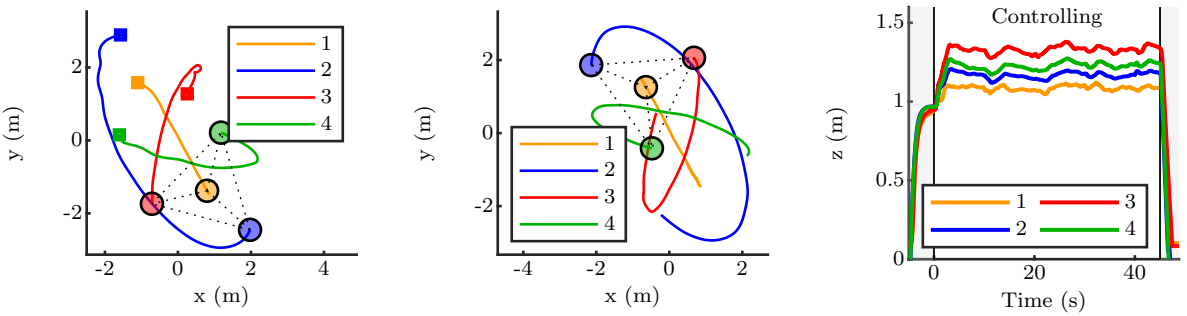
been caused by actuator saturation. Note that this is the vehicle that starts farther away from the desired formation. For this reason, it is expected to have an high control input during the initial instants, which saturates the actuation. Because of this saturation, the controller was not able to decrease the error at the desired rate, causing the integrator to fill up. For this reason, it took some time for the integrator to converge. A possible solution consists in adding anti-windup action to the integrator, to deal with actuator saturation. Figure 5b shows the movement of the vehicles, tracking the object while maintaining the formation. It is also possible to see that the integrator accumulated error above described has disappeared, as the vehicles achieved the prescribed square formation. The time evolution of the altitude is presented in Fig. 5c, where it is now possible to observe that the vehicles were able to achieve consensus on the altitude of the formation and reach the prescribed altitude of one meter, thus rejecting the disturbances.

7.2. Experiment 2 - Time Varying Formation

Now that convergence to a static formation, and the capability to follow a moving target (the virtual leader) have been presented, a more demanding example consisting in a time varying formation is presented. This consists of a leader vehicle, following a trajectory that is also time-varying (going back and forth along a straight line with a sinusoidal velocity profile), and three other vehicles orbiting it.

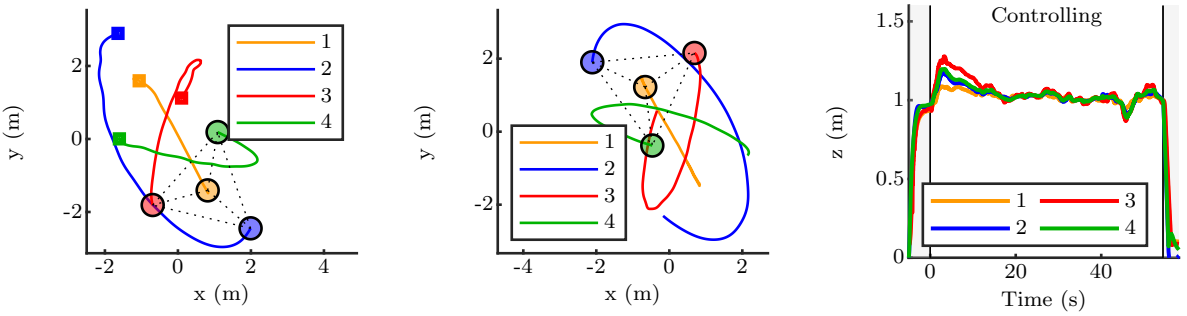
The prescribed motion is slow and was carefully designed to fit the working space, with reasonable margins between the vehicles and the arena limits. The periodic motion considered has a period of 20 seconds, and was defined during two periods, i.e., 40 seconds. After those 40 seconds, the vehicles stop at the last desired position of this movement (which is also the initial position). This consists in a discontinuity on the desired motion, meaning that a new convergence is initiated at that instant. The digraph associated to the interaction topology considered is presented in Fig. 3b. The connection weights were set to be $k_{ij} = 1.7$, the derivative gain to $\gamma = 1.1\,\mathrm{s}^{-1}$, and the integral gain, when used, to $\zeta = 0.15\,\mathrm{s}^{-3}$ as in the previous experiment, which can be verified to follow the conditions presented in Theorem 1. Since this is a more complex experiment, and due to the space constraints of the testing environment, the initial positions of the vehicles were set to be close to their initial desired positions. However, they start with null velocity, so there is still a convergence to the desired time varying formation.

The physical vehicles considered are vehicles 1



(a) Movement at $t = 11\,\mathrm{s}$. (b) Movement at $t = 22\,\mathrm{s}$. (c) Evolution of altitude with time.

Figure 6: Experiment 2 - Results without integral action.



(a) Movement at $t = 11\,\mathrm{s}$. (b) Movement at $t = 22\,\mathrm{s}$. (c) Evolution of altitude with time.

Figure 7: Experiment 2 - Results with integral action.

and 3. The movement on the horizontal plane, when integral action is not considered, is presented in Fig. 6a and in Fig. 6b. As expected, oscillations are present during the initial convergence. After this initial convergence, the vehicles achieve the time varying formation in the horizontal plane, orbiting around the leader vehicle. The movement of the vehicles on the horizontal plane when considering integral action, is presented in Fig. 7a and Fig. 7b. As can be observed, there are no noticeable differences between the movement on the horizontal plane with and without integral action, apart from the initial instants, which is a consequence of slightly different initial positions.

Regarding the evolution of altitude with time, this is presented in Fig. 6c when no integral action is considered, and in Fig. 7c when considering integral action. This is where most distinction can be noted. Note that the presence of disturbances is confirmed when no integral action is present, as the vehicles climb to an altitude different than the desired altitude of one meter, and do not maintain the same altitude. This effect is apparent, mainly in vehicle 3, which is one of the physical vehicles considered, and has influence on the other vehicles. The other physical vehicle considered, vehicle 1, now plays the role of leader, using the trajectory tracking controller. For that reason, it is not influenced by the disturbance of vehicle 3. However, it can be noted that the leader vehicle is also subject to a disturbance, although its effect is smaller. This behavior is expected, as this vehicle used a trajectory tracking controller which was tuned, meaning it has higher gains, and thus it is less influenced by the disturbance. Nonetheless, some influence is still noticeable, since when the experiment was performed without integral action on the formation tracking controller, the integral action of the trajectory tracking controller was also not present, as the gain K_i introduced in (14), was set to zero. However, when performing the experiment with integral action on the formation tracking controller, K_i was set to $0.3\,\mathrm{s}^{-3}$, to reject the disturbance acting on the leader vehicle as well, enabling all vehicles to reach the desired altitude of one meter. It can be noted that, even when tracking a time varying trajectory, when integral action is added, the vehicles are capable of rejecting the disturbance and achieve the same altitude.

7.3. Discussion

The goal of these experiments consisted in assessing the ability of the integral action which was added to the formation tracking controller to reject disturbances, and verify that it provides increased capabilities when working with multirotor

vehicles. These vehicles are susceptible to modeling errors, which can be interpreted as disturbances to the nominal system that is being considered. For this reason, when applying controllers to this type of vehicles, it is important to keep in mind that these must be designed with some robustness to these errors. It was shown, in the presented experiments, that the ability of this vehicles to reach the prescribed formation is considerably influenced by these disturbances. These effects would be even more evident when considering an increased number of vehicles. In the described experiments, only two physical multirotors were used, and still, considerable improvements were observed when adding integral action. It is clear that the proposed integral action has a positive effect when performing formation control with multirotor vehicles.

The effect of disturbances in multirotor vehicles is clearly more apparent on the vertical axis, as evidenced by the presented experiments, due to the effect of the weight, and also because the prescribed movement was sufficiently slow for the drag forces to be negligible. Since the system of double integrator agents previously described is decoupled, a choice could be made to add integral action to the vertical axis alone, as this is where most of the effect is visible. However, if flights were to be made in an open space, an interesting movement for the vehicles would be to move in formation with a constant velocity. In this case, if the velocity was high enough for the drag to have a considerable effect, it would probably prove to be useful to add integral action to the controller on the other axis as well. It was also seen in the first experiment that, when considering integral action, actuator saturations can have an adverse effect on the system. Anti-windup action was proposed to deal with this effect, however, other solutions might be considered. Nevertheless, it was shown that the proposed algorithm is able to reject disturbances in the vehicles, while following a decentralized approach, and considering a limited amount of information.

8. Conclusions

The goal of this dissertation was to study the formation control problem, for application to multirotor vehicles and devise an experimental setup, that would allow to test the proposed algorithms in physical vehicles. The distributed formation control problem was successfully studied, providing a review on some of the already proposed algorithms, and enhancing the existing algorithms for vehicles modeled as double integrators with the feature of integral action, which enabled disturbance rejection and proved to be useful when working with multirotor vehicles. New theoretical results were achieved regarding the consensus-based protocols used, obtaining criteria for the convergence of the proposed third-order consensus protocol, and enhancing the existing criteria provided in the literature for the second-order consensus protocol. An experimental setup was successfully implemented, which enabled to test the proposed algorithms in physical multirotor vehicles. Although the devised setup considered controlling the vehicles from an external computer, modifications to this setup can now be devised to allow executing the controllers onboard. The algorithms were then tested and experimental results were obtained, allowing to successfully validate the proposed algorithms using several vehicles.

References

[1] Yang Quan Chen and Zhongmin Wang, "Formation control: a review and a new consideration," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 2005, pp. 3181–3186.

[2] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424 – 440, 2015.

[3] W. Ren, R. Beard, and T. McLain, "Coordination Variables and Consensus Building in Multiple Vehicle Systems," *Proc. Block Island Workshop Coop. Control*, vol. 309, pp. 439–442, December 2004.

[4] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, vol. 17, July 2007.

[5] G. Wen, Z. Duan, and G. Chen, "Distributed consensus of multi-agent systems with general linear node dynamics through intermittent communications," in *24th Chinese Control and Decision Conference (CCDC)*, May 2012, pp. 1–5.

[6] L. Krick, M. Broucke, and B. Francis, "Stabilization of infinitesimally rigid formations of multi-robot networks," *International Journal of Control*, vol. 82, pp. 477 – 482, January 2009.

[7] D. V. Dimarogonas and K. Johansson, "On the Stability of Distance-based Formation Control," *Proceedings of the IEEE Conference on Decision and Control*, pp. 1200–1205, January 2008.

[8] K. Oh and H. Ahn, "Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space," *International Journal of Robust and Nonlinear Control*, vol. 24, August 2014.

[9] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," *IFAC Proceedings Volumes*, vol. 35, July 2002.

[10] I. Kovacs, D. S. Silver, and S. G. Williams, "Determinants of commuting-block matrices," *The American Mathematical Monthly*, vol. 106, no. 10, pp. 950–952, 1999.

[11] Video presenting the performed experiments. [Online]. Available: https://youtu.be/AqIRXtMwwaQ [Accessed: 2019-11-03]