University of Groningen

Distributed Formation Control in a Multi-agent Network

Bachelor Integration Project

Industrial Engineering and Management

By

Jan Lyckle Bijlsma

s3467635

First Supervisor:	J. M. A. Scherpen
Daily Supervisor:	N. Li
Second Supervisor:	M. Munoz Arias

June 12, 2020

Disclaimer
This report has been produced in the framework of an educational program at the University of Groningen, Netherlands, Faculty of Science and Engineering, Industrial Engineering and Management (IEM) Curriculum. No rights may be claimed based on this report. Citations are only allowed with explicit reference to the status of the report as a product of a student project.

Abstract

This research elaborates on the concepts of formation control, velocity tracking and leader-follower strategy in a port-Hamiltonian framework. First, in chapter 1, background information is provided about graph theory and the port-Hamiltonian framework, followed by the outline of the research. Then, the design and results of the formation controller, velocity tracking controller and leader-follower controller are presented in chapter 2 where the agents are modelled as fully actuated, single integrator agents and the variables of interest are velocity, position and displacement. The leader-follower controller uses the concepts of both formation control and velocity tracking control to create a stable system. The results are obtained via simulations in Matlab and Simulink models. The Simulink models are presented in appendix A and the Matlab scripts can be found in appendix B.

Contents

Introduction

Looking at nature, different types of formations can be distinguished, such as flocks of birds or a school of fish. These formations are a product of nature and provide benefits to the animals. A group of fish can spot a predator easier than one single fish. A flock of birds can spot possible food easier than one single bird. From these examples alone, it can be seen that when individuals form a formation, performance can be enhanced. However, this is not limited to the animal realm, because the same concepts can be applied to artificial individuals. An increasing number of governments and companies are exploring the possibility of deploying unmanned vehicles for replacing humans in labour intensive work (Quaranta, 2016) (National Research Council, Division on Engineering & Physical Sciences, Board on Army Science & Technology and Committee on Army Unmanned Ground Vehicle Technology, 2003) (Shiue & Chang, 2010). These robots, more specifically called Unmanned Grounded Vehicles (UGV) or Unmanned Aerial Vehicles (UAV), move autonomously and can form any formation that is desired. The UGVs in combination with the UAVs show promising possibilities for aid in marine corps combat, construction, agriculture and other logistical challenges (Naval Studies Board, Division on Engineering & Physical Sciences, Committee on Autonomous Vehicles in Support of Naval Operations and National Research Council, 2005)(Krizmancic et al., 2020) (Dreano, 2018)(Bonadies et al., 2016) (Kuru et al., 2019). Furthermore, a multi-agent network with formation problems can be compared to a network of multiple satellites orbiting around the earth keeping a constant relative distance enabling the technology of Global Positioning System (GPS) (Xu et al., 2007) (Parkinson et al., 1996) or robots that can help disable mines from war zones by staying in a formation and keep a constant velocity to minimise disturbances (Healey, 2001). The mentioned examples all form a formation which results in enhanced performances of the network as a whole.

Furthermore, chapter 1 will discuss the research design plan, and is divided into four sections. Section 1.1 elaborates on preliminary background information about graph theory and the framework used for modelling the mechanical system. Section 1.2 discusses the formulation of the problem, including the problem context, problem statement and the system description. The last section of Chapter 1 is the research proposal in section 1.3 and will discuss how the research will look like and which concepts are utilised. Chapter 2 discusses the design of controllers that should solve the problem discussed in section 1.2. The chapter is divided into three sections; section 2.1 is focused on the formation control, section 2.2 describes the velocity tracking problem and the solution and section 2.3 elaborates on how to combine both the formation control and velocity tracking in a leader-follower strategy to solve the presented problems.

Chapter 1

Research Design

1.1 Background Information

Before the problem context or system description can be defined, preliminary background information is necessary in order to fully understand the concepts.

The port-Hamiltonian concept is a method of representing a multi-domain system, i.e. a mechatronics system, which arose due to *"the widespread interest for unifying the modelling frameworks of different physical domains"* (Van Der Schaft, 2019). A port-Hamiltonian system consists of energy storing elements and energy dissipating elements that come together in a central energy routing structure, called a Dirac structure (Van Der Schaft, 2019). The example in figure 1.1 shows the flow (f) and effort (e) variables. Consequently, one can read the flow and effort variables as velocity and force in the mechanical domain and voltages and currents in the electrical domain. The Dirac structure has the property of power conservation, meaning that it links the flow and effort variables in a way that the power input is equal to the power output (Van Der Schaft, 2019). The Dirac structure allows for the introduction of virtual couplings in the port-Hamiltonian system, which are virtual representations of mechanical components, i.e. spring or damper. The dynamics of such a mechanical component are then used as the dynamics of a virtual version between agents. This virtual coupling allows the system to behave as if a spring or damper is present, without the physical presence of it.

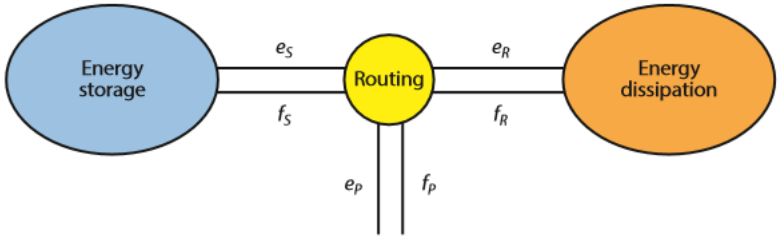

Figure 1.1: Dirac structure that visualises a port-Hamiltonian system. *source*: Van Der Schaft (2019)

Dynamical relations in discrete systems, i.e. mass-spring-damper mechanical systems or multi-body systems, lead to various sorts of network dynamics (Van Der Schaft & Maschke, 2013). The analogy with graph theory allows the coupling of a port-Hamiltonian framework with a multi-agent system. The coupling with graph theory is based on the fact that externally supplied power goes through the vertices of the graph. Due to the conservation law in a port-Hamiltonian framework, the increase of energy is the external power minus the power lost in the dissipating elements which are associated to some of the edges or vertices of the graph (Van Der Schaft & Maschke, 2013). Moreover, the virtual coupling is directly linked to the edges of a graph, as each edge represents a virtual coupling between the agents. The number of edges in a graph determines the number of virtual couplings and between which agents. Furthermore, the Dirac structure of the whole system is directly defined by the *incidence* matrix (B) of the multi-agent network (Van Der Schaft & Maschke, 2013).

The incidence matrix is a subject of graph theory, however, before defining the incidence matrix, basic graph theory should be understood. Let $G = (\nu, e)$ be a graph with nodes $\nu = (v_1, v_2, ..., v_n)$ and edges $e = (e_1, e_2, ..., e_j)$. Vertices i and k, where $i, k \in \nu$, are adjacent or neighbours if they are the ends of a common edge (Ahn, 2020). Using this knowledge, the incidence matrix can be defined. If the edge $e_{ik} = (v_i, v_k)$ is directed from k to i, then q_{ik} will have a $+1$ value and q_{ki} will have a -1 value in the incidence matrix $B = B(G) = (q_{ik})$. If $e_{ik} = (v_i, v_k) = 0$ meaning that there is no edge connecting the two vertices, the entries q_{ik} and q_{ki} are both zero (Merris, 1994). Consequently, the incidence matrix is representation of the vertices and the edges, meaning that the rows represent the vertices and the columns represent the edges. Furthermore, incidence matrix is not the only matrix for graph theory, as the adjacency matrix $\mathbb{A}$ or the degree matrix $\mathbb{D}$ are also matrices based on a graph. The adjacency matrix is a matrix where a_{ik} has the value of 1 if there exists an edge between v_i and v_k; otherwise, a value of zero is appointed. The degree matrix is a diagonal matrix where the value for $a_i i$ is the number of neighbours of node v_i. Finally, using $\mathbb{A}$ and $\mathbb{D}$, a Laplacian matrix can be computed by $\mathbb{L} = \mathbb{D} - \mathbb{A}$.

The Laplacian matrix is a representation solely of the vertices, whereas the incidence matrix takes the edges also into account. Moreover, the directed edge introduces the concepts of a directed and undirected graph. An directed graph has only directed edges and is denoted by $\vec{G} = (\nu, \vec{\varepsilon})$. In such a directed graph, the possibility of a cycle presents itself if a path from edge to edge can be taken where the first and last node is the same, but otherwise all nodes are distinct (Easley & Kleinberg, 2010). For the research, only acyclic graphs are considered. Moreover, the undirected graph, as the name suggest, does not contain directed edges. Finally, a graph can be considered a tree graph, if there exists a node v_i which has a directed path to all other nodes and no cycles are present in the graph (Ahn, 2020).

1.2 Problem Formulation

The problem formulation section will elaborate on the problem that is presented to the researcher. Before the problem statement can be made, the problem context needs to be assessed and the current state of the art is determined, in order to formulate a correct system description.

1.2.1 Problem Context

This section discusses the problem context and what the state of the art is for current literature. Based on the body of knowledge presented in section 1.1, the problem context can be defined. The problem that is presented concerns the lacking of a controller for a single integrator network in a port-Hamiltonian framework that can ensure formation keeping and tracking of a constant velocity via a leader-follower strategy.

The formation control problem is not a new phenomena, as described in the introduction. Moreover, prior research in the formation control and velocity tracking fields are present in current literature. (Peng et al., 2020) (K. Cao et al., 2019) (Tran et al., 2020) (Meng & Jia, 2014). However, none of these studies suggest the use of a single integrator in the port-Hamiltonian framework for modelling the agents. Moreover, none of these studies use the concept of the leader-follower strategy. The amount of new leader-follower studies are on a rise in recent years (Oh et al., 2015) (Scharf et al., 2003) (Scharf et al., 2004). The leader-follower strategy is a control approach where the network is not considered as a whole, but is divided into leaders and followers (Kang et al., 2014) (Zhao, 2018). The concept is based on the fact that the leader move with a desired velocity, either constant or time-varying, and the followers control the distance, displacement or relative position with respect to the leader agent. When the leader agent moves with a constant velocity, it is not concerned with the desired formation, meaning that the velocity of the leader is not altered by the formation controller. The advantage of the leader-follower strategy, compared to just combining the formation and velocity tracking controllers, is that per agent, less information is needed to obtain the correct results. The leader is only required to know the desired velocity and the follower agents are only required to know the desired displacement to form the desired formation. By limiting the required information, the calculating capacity is reduced (Kang et al., 2014). Existing literature on leader-follower strategies use relative positions as control variables in a mechanical framework and do not consider modelling the agents in a port-Hamiltonian framework (Kang et al., 2014) (Zhao, 2018) (Scharf et al., 2003) (Scharf et al., 2004).

The advantage of the port-Hamiltonian framework is that it can describe a mass-damper model (figure 1.2) as a system of energy which, consequently, can be used to verify (asymptotic) stability using Lyapunov's theorem. Moreover, such a mechanical system is useful as the analogy with graph theory can be made. The graph theory is particularly important due to the nature of the formation problem. *"In order to obtain a formation, agents must position themselves relative to the other agents in the system"* (Olfati-Saber & Murray, 2004). It should be noted that the

positioning process of an agent requires information of other agents, the relative displacement, with respect to the other agents. This means that an information flow (edge) must present between two agents (vertices). As this resembles basic graph theory, other characteristics of graph theory can be applied to the system of interest, i.e. (weighted) adjacency matrix, degree matrix, incidence matrix or Laplacian matrix (Ahn, 2020).

Furthermore, agents can be positioned in a formation for increased performance, for example if *coverage* is requested (Vos, 2015), which is a situation where the robots must position themselves in a predefined formation to cover an area as efficiently as possible. The formation can be considered as a consensus between agents (Saber & Murray, 2003). Consensus problems have a long history in computer sciences, in particular distributed computations (Olfati-Saber & Murray, 2004) (Olfati-Saber et al., 2007). According to Olfati-Saber et al. (2007), a consensus in a network of agents is defined as *"to reach an agreement regarding a certain quantity of interest that depends on the state of all agents"* and consequently a consensus algorithm as *"an interaction rule that specifies the information exchange between an agent and all of its neighbours on the network"*. Therefore, it can be derived that a consensus algorithm takes the information of all the agents into account and actuate the agents to a final formation, a consensus.

However, for the current problem, the focus lies primarily on the formation control and velocity tracking, where certain parameters of the agents converge to predefined desired values. Nonetheless, the consensus methods are considered as a useful alternative and used as a basis for the leader-follower strategy.

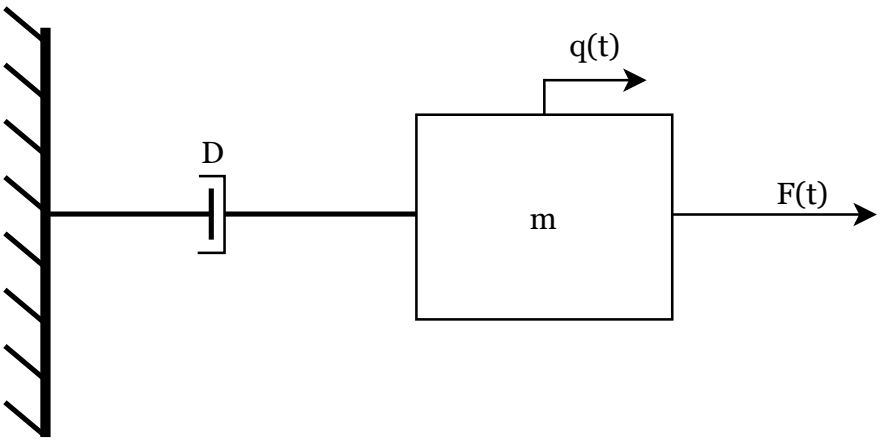

Figure 1.2: A standard mass-damper system

1.2.2 Problem Statement

This section will provide the problem statement based on the problem context. The bigger problem of the research is that agents must be able to form and keep a formation with the help of a controller. The actual problem that is tackled in this research, is the absence of a leader-follower strategy in a port-Hamiltonian framework which can solve the bigger problem of formation keeping. The system that is analysed, is a single integrator system modelled in a port-Hamiltonian framework,

as described by Vos (2015). Due to the chosen variables (position, momenta and relative displacement) that are measured for each agent of the system, the system can be classified as a distributed formation control (Vos, 2015). The problem is defined in the following problem statement.

In order to increase the performance of a multi-agent network, agents must reach a specified formation and hold this formation with a constant velocity. With a leader-follower strategy, this formation and constant velocity can be achieved and reduce the computing capacity needed for every agent (Kang et al., 2014). Current literature on leader-follower strategies does not consider a single integrator system modelled in a port-Hamiltonian framework. Therefore, this research must provide new insights by taking a new approach, the port-Hamiltonian framework, to the leader-follower strategy and, consequently, the formation and velocity tracking problem to create a clearer overview of the energy flows than currently exists.

1.2.3 System Description

The system description section will discuss the system that is under consideration and the corresponding state space variables. The system that is evaluated is a network consisting of n agents which must reach a predefined formation and velocity. The agents in the system can be interpreted as point masses which can move in any direction in order to obtain the correct position for the formation. The agents communicate directly with each other, as in distributed control theory (Vos, 2015), meaning that the system can be considered as a linear position-based algorithm, instead of a non-linear distance-based algorithm (Y. Cao et al., 2012) (Vos, 2015). Using the positions, the relative displacement between the agents is determined, which are the controller variables. Furthermore, agents are modelled as fully actuated agents, meaning that the number of inputs is equal to the degree of freedom of the system, in a tree graph (see section 1.1). The agents can be modelled according to the port-Hamiltonian framework and the mass-damper system, defined by Vos (2015). In this system, with n number of agents, denote $q_i \in \mathbb{R}^n$ as the displacement of the mass m_i and the momentum $p_i \in \mathbb{R}^n$ where $p_i = \dot{q}_i m_i$ for $i = 1, 2, ...n$. Moreover, the mass is assumed to be at rest and subject to an input force denoted as u. With the defined variables, the mechanical system of the agent is defined as

$$\begin{aligned} \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q}(p) \\ \frac{\partial H}{\partial p}(p) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \frac{\partial H}{\partial p}(p) \end{aligned} \tag{1.1}$$

where $p = (p_1, p_2, ...p_n)^T$, $q = (q_1, q_2, ...q_n)^T$ and the Hamiltonian $H(p) = \frac{1}{2} p^T M^{-1} p$, with $M = (m_1, m_2 ... m_n)$, corresponds to the kinetic energy stored in the mass per agent.

1.3 Research Proposal

The research proposal section will elaborate on the research strategy and methods used. The purpose of this section is to determine the outline of the research. First, the research objective will be established, which will be used for the decision on relevant topics for the research questions. Then, the used materials are discussed, followed by the chosen research cycle, validation and planning.

1.3.1 Research Objective

This section will define the research objective and the corresponding criteria. The objective will function as a guide to the research, as the objective determines the research questions. Moreover, when the objective is met, the final designed artefact should pose a solution to the defined problem statement.

The goal of this research is to design a controller with the objectives of formation control and velocity tracking for a group of agents modelled as a single integrator in a port-Hamiltonian framework described in (1.1) and, consequently, implement it in a leader-follower strategy controller that ensures velocity tracking and formation keeping. Designing the formation, velocity tracking and leader-follower controller is done by searching for relevant literature and by experimental testing via mathematical models in order to obtain the correct results and parameters. This goal must be met within three months.

1.3.2 Research Questions

This section defines the research questions that are used in order to guide the researcher in the process of finding relevant literature and answers. Each question is formulated in a way that ensures that the final answers will provide relevant knowledge and help in achieving the research objective.

Central Questions

The central question will be formulated, bearing in mind that the answers must help in achieving the research objective and must provide a steering function to the research. The central questions are therefore defined as:

1. What are the characteristics of a controller that controls the formation of the single integrator system in a port-Hamiltonian framework?

2. What are the characteristics of a controller that controls the velocity tracking problem of the single integrator system in a port-Hamiltonian framework?

3. How are the two controllers combined in order to ensure that all control objectives are met?

4. Which characteristics of the formation and velocity controllers can be used for the leader-follower strategy controller?

Sub Questions

The sub questions provide guidance in answering the central question, as each answer to a sub question must be a part of the final answer to the central question. The sub questions are formulated using the method of unravelling key concepts, where the most important concepts are decomposed in smaller dimensions in order to create a more detailed overview. The concepts of interest for this research are marked with an asterisk (*) in figures 1.3 and 1.4. Using this method, the following sub questions are formulated:

1. What are the control objectives for the formation control?

2. What are the control objectives for the velocity tracking?

3. Which mechanical dynamics are required for the controller, using the concept of virtual coupling?

4. What methods are necessary for proving (asymptotic) stability of the controllers?

5. Which tools and criteria are necessary for the simulation of the closed-loop systems in Matlab?

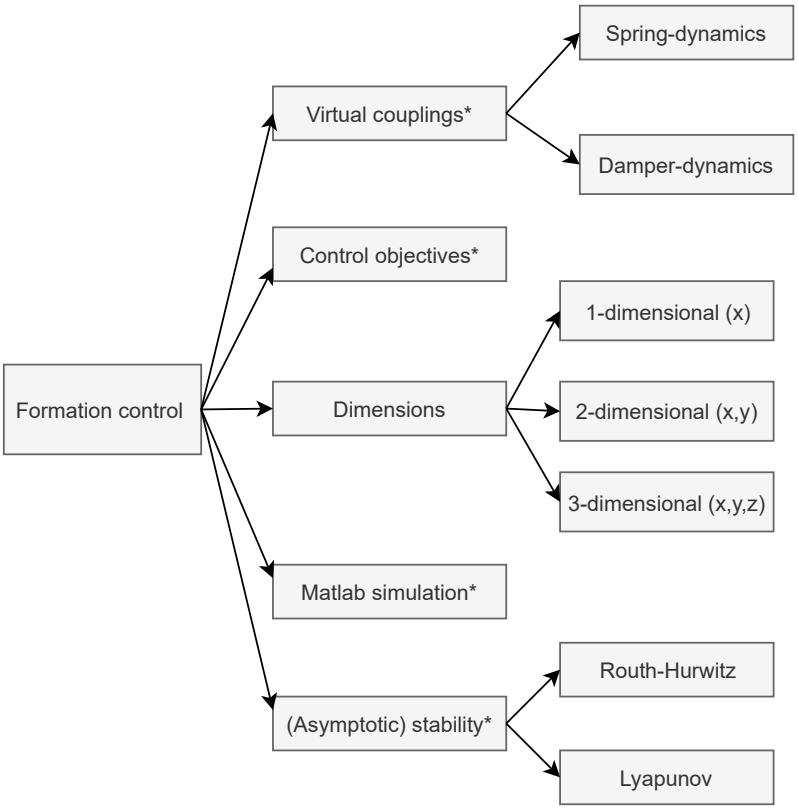

Figure 1.3: Key concepts for the formation control

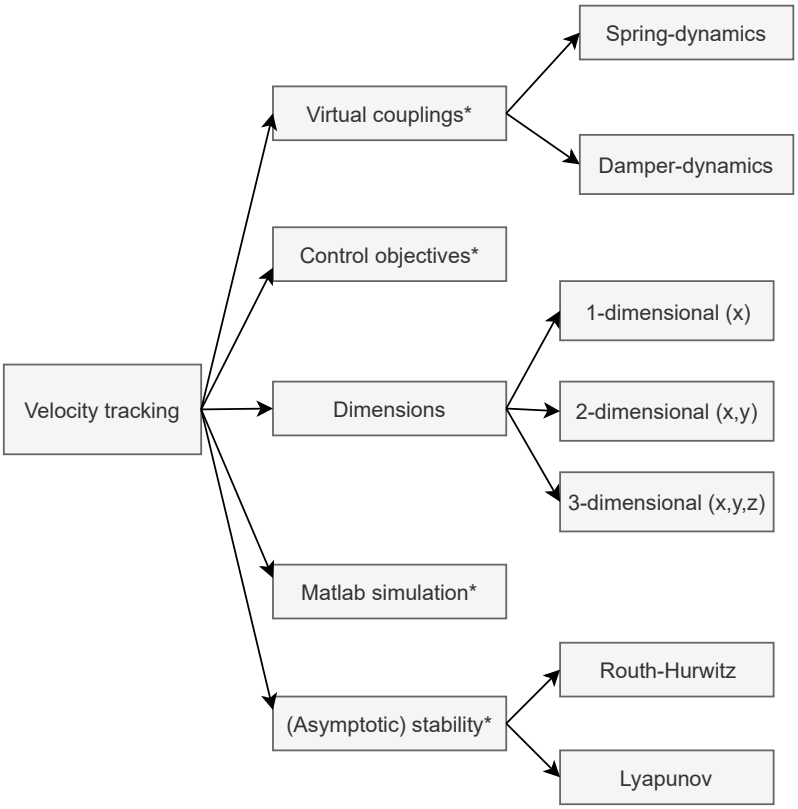

Figure 1.4: Key concepts for the velocity tracking

1.3.3 Research Strategies & Materials

In order to find the most relevant materials for answering the research questions, the research strategy must be determined. The choice of the strategies are based on a set of criteria discussed by Verschuren & Doorewaard (2010): **breadth** or **depth**, **qualitative** or **quantitative** and **empirical** or **desk research**. The formulated problem in section 1.2.2 requires an in-depth research, due to the specific application to a multi-agent network that must reach a formation and track a velocity. However, general knowledge and concepts from the final deliverable can be used in future research on formation control or leader-follower strategies. Furthermore, the research is a quantitative research, as the results consists of calculations, numerical results and tables. Lastly, the research is an empirical research, as the data and results are acquired by the researcher by running computer simulations.

Based on the classification of the research, the strategies that fit the criteria are *experiments* and *desk research*(Verschuren & Doorewaard, 2010). The experiment is a strategy that involves an in-depth, qualitative and empirical research, which fit the criteria exactly. The experiments consists of computer simulations in Matlab, where the controller is tested with different initial conditions. By considering different initial conditions, the controller can be optimised through trial and error. The resulting data from the simulations are used for further improving the controller.

The second strategy, the desk research, is chosen as a useful strategy, because the right theoretical concepts need to be applied to the controller and mechanical system even though the research requires empirical knowledge gathering. The the-

oretical concepts can only be acquired through a literature research, which shows the need for the desk research.

After choosing the strategies, the materials are determined. The research material indicates what type of material and source is used for the research in order to answer the research questions and are determined using the theory of Verschuren & Doorewaard (2010). In order to answer the research questions and consequently achieve the research objective, this research will use *Documents*, *Literature* and *Simulations* as the main sources for acquiring relevant knowledge. The accessing of the sources is done by *content analysis*, *search method* and *computer programs*, respectively (Verschuren & Doorewaard, 2010). Sub questions 1, 2,3 and 4 require documents and literature material, due to theoretical nature of the questions and the already existing literature. Sub question 5 requires simulation material, because it needs actual simulation results to determine the right set-up.

1.3.4 Research Cycle

This section briefly elaborates on the design cycle that is applicable to this research. The chosen cycle is the *empirical cycle* as described by Wieringa (2014) (figure 1.5). The most important aspect of this cycle is the induction and deduction of a hypothesis, which concerns the characteristics of the formation, velocity tracking and leader-follower controller. Based on the hypothesis, the testing and evaluation phase is performed. As mentioned before, the testing is done via computer simulations and based on the evaluation of the results, the controllers are adopted accordingly.

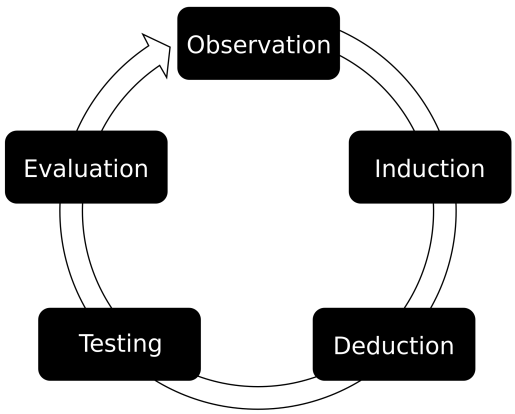

Figure 1.5: Empirical cycle as described by Wieringa (2014)

1.3.5 Validation

This section elaborates on the validation process of the final deliverable of the research. The results of the research will be validated using a mathematical computer model. This model is constructed using the computer software *Matlab2019b*, more specifically, the *Simulink* add-on. The simulations should validate that the proposed controllers are stable and the control objectives are met. If time allows it, more validation using physical agents will be conducted. The controller will be applied as

an input to the physical agents, in this case mobile robots, to see if the formation is indeed stable after time t. This enables the researcher to see if the controller yields the desired results in real-life, when the system is subject to disturbances.

1.3.6 Planning

This section discusses and visualises the planning for the research. First, theory about formation control and velocity tracking is examined before anything can be designed. This process starts in the design plan phase and is extended in the research phase with 1 week. Consequently, a controller for the proposed system in eq. (1.1) must be designed and the proof must be constructed which spans a total period of approximately 3-4 weeks.This includes the controller for the formation control and the velocity tracking. When the controllers are designed, the testing of the controlling algorithm is performed. This should take approximately 1-2 work-weeks. If everything goes according to plan, the last 3-4 weeks will be used for the application of a leader-follower model for the formation control of three agents and testing with physical robots.

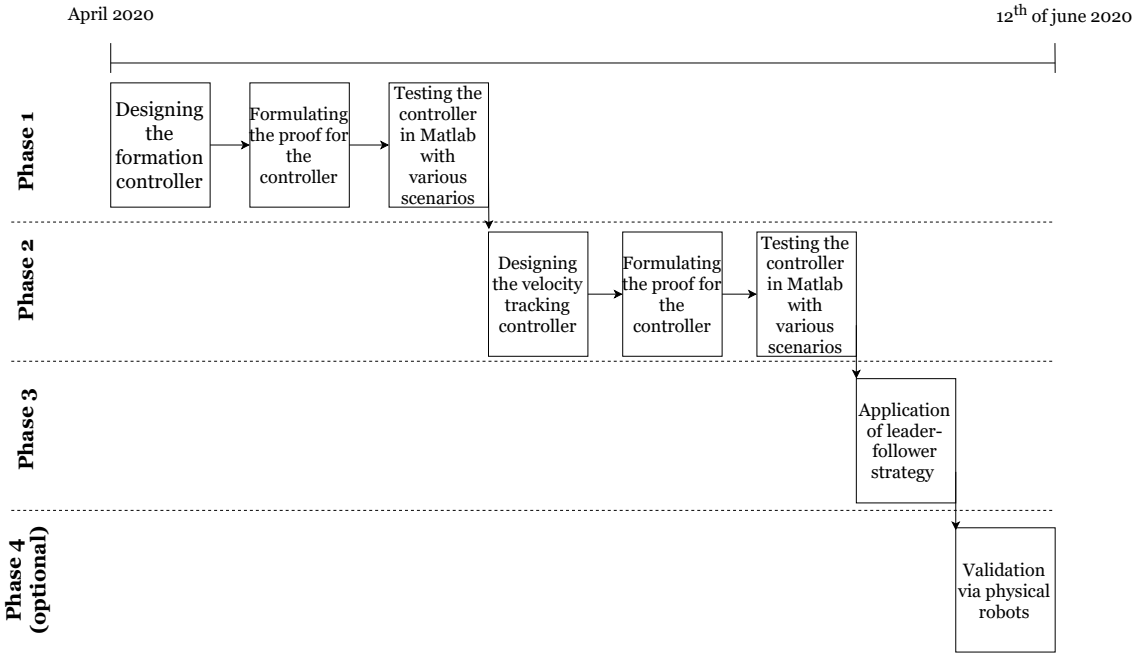

Figure 1.6: Visual representation of the planning

Chapter 2

Controllers design

This chapter elaborates on the controllers characteristics. In order to ensure that the right formation is achieved, a controller must be designed to create a closed-loop system that ensures the desired formation. Moreover, to prevent the formation from drifting, a constant velocity must be tracked by the agents. The constant velocity can be a zero or a nonzero desired velocity. The nonzero desired velocity can be useful if the agents are required to move as a formation. The formation control with constant velocity equal to zero is described in section 2.1. The closed-loop system with a nonzero desired velocity is described in section 2.2. Moreover, section 2.2 also discusses the combination of the two controllers into one closed-loop system to achieve the control objectives of both controllers. In section 2.3, the concepts of formation keeping and velocity tracking are combined into a new leader-follower strategy where the required calculating capacity is reduced per agent.

2.1 Formation control

The thesis written by Vos (2015) elaborates on the design process of a controller for a system similar as described in section 1.2.3. Therefore, the method described in the thesis will function as a basis and guide for the design of the controller for the formation control. Additionally, with the position q_i and momentum p_i defined in section 1.2.3, other variables related to virtual couplings should be defined. With a total number of virtual couplings e, the relative displacement $z_j \in \mathbb{R}^n$ and the desired relative displacement $z_j^* \in \mathbb{R}^n$ for $j = 1, 2..e$, where j is the virtual coupling between the agents. In order to ensure that the desired formation is achieved, two control objectives are formulated:

$$\begin{cases} p \to 0, \\ \\ z \to z^*, \end{cases} \qquad \text{as t} \to \infty \tag{2.1}$$

The formation control objectives are achieved by using virtual couplings which consists of a virtual damper and spring, where the relative displacement z_j is used as the spring elongation, the input velocity is denoted as $w_j \in \mathbb{R}^n$ and the output force as $\tau_j \in \mathbb{R}^n$. The virtual spring is responsible for the convergence to the

desired displacement and the damper is present in the controller to ensure that the agents come to a full stop and, consequently, hold the formation and do not drift. The dynamics of such a spring-damper system in a port-Hamiltonian framework is described by Duindam et al. (2009), van der Schaft & Jeltsema (2014) and Vos (2015) as:

$$\begin{aligned} \dot{z}_j &= w_j \\ \tau_j &= \frac{\partial H_j^c}{\partial z_j} + D_j^c w_j \end{aligned} \tag{2.2}$$

where $D_j \in \mathbb{R}^{nxn}$ is the semi-definite positive dissipation function and $H_j(z_j)$ is the Hamiltonian for the controller which includes a spring as virtual coupling and is denoted as:

$$H_j^c(z_j) = \frac{1}{2}(z_j - z_j^*)^T K_j^c (z_j - z_j^*) \tag{2.3}$$

with positive definite spring constant $K_j^c \in \mathbb{R}^n$. For the ease of notation, $z = (z_1, z_2, ..., z_e)^T$, $z^* = (z_1^*, z_2^*, ...z_e^*)^T$, $w = (w_1, w_2, ..., w_e)^T$, $\tau = (\tau_1, \tau_2, ..., \tau_e)^T$, and for the system matrices $K^c = block.diag(K_1^c, K_2^c, ..., K_e^c)$, $D^c = block.diag(D_1^c, D_2^c, ..., D_e^c)$. The Hamiltonian can be denoted as $H^c = \sum_{j=1}^{e} H_j^c(z_j) = \frac{1}{2}(z - z^*)^T K^c (z - z^*)$.

As mentioned in section 1.2.3, the agents are modelled as a tree graph, with corresponding incidence matrix B. Then, according to Vos (2015), the coupling of the agents and the virtual coupling can be done by:

$$B = \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \tag{2.4}$$

$$\begin{aligned} u &= -(B \otimes I_n)\tau \\ w &= (B^T \otimes I_n)y \end{aligned} \tag{2.5}$$

where u is the input force of mechanical system, y is the output of the mechanical system (1.1) and w is the input velocity of the controller.

In the closed-loop system, since the output of the controller is the input of the mechanical system and the output of the mechanical system is the input of the controller, then, using (1.1), (2.2) and (2.5), the closed-loop system can be described.

Controller state variable:

$$\begin{aligned} \dot{z} &= w \\ &= (B^T \otimes I_n)y \\ &= (B^T \otimes I_n)\frac{\partial H}{\partial p} \end{aligned}$$

Mechanical system state variable:

$$\begin{aligned}\dot{p} &= -I_n \frac{\partial H}{\partial q} + I_n u \\ &= -(B \otimes I_n)\tau \\ &= -(B \otimes I_n)\left(\frac{\partial H}{\partial z} + D^c w\right) \\ &= -(B \otimes I_n)\left(\frac{\partial H}{\partial z} + D^c (B^T \otimes I_n)\frac{\partial H}{\partial p}\right) \\ &= -(B \otimes I_n)\frac{\partial H}{\partial z} - (B \otimes I_n) D^c (B^T \otimes I_n)\frac{\partial H}{\partial p}\end{aligned}$$

Resulting in the closed-loop system

$$\begin{bmatrix}\dot{p} \\ \dot{z}\end{bmatrix} = \begin{bmatrix}-(B \otimes I_n) D^c (B^T \otimes I_n) & -(B \otimes I_n) \\ (B^T \otimes I_n) & 0\end{bmatrix} \begin{bmatrix}\frac{\partial H^r}{\partial p} \\ \frac{\partial H^r}{\partial z}\end{bmatrix} \tag{2.6}$$

where the Hamiltonian $H^r(z,p)$ is equal to the sum of the Hamiltonian in (1.1) and the Hamiltonian in (2.3) (van der Schaft & Jeltsema, 2014), yielding $H^r(z,p) = \sum_{i=1}^{n} H_i(p_i) + \sum_{j=1}^{e} H_j^c(z_j) = \frac{1}{2} p^T M^{-1} p + \frac{1}{2}(z - z^*)^T K^c (z - z^*)$. The closed-loop system (2.6) and Hamiltonian $H^r(z,p)$ meet the control objectives (2.1). Consider the following proposition.

Proposition 2.1. *By coupling the agents in the system with an controller with dynamics described in* (2.2) *via coupling* (2.5)*, the resulting closed-loop system described in* (2.6) *ensures that* $p \to 0$ *and* $z \to z^*$*, when* $t \to \infty$*, thereby achieving the control objectives* (2.1).

Proof. Taking $H^r(z,p)$ as a Lyapunov function candidate, it can be verified that $H^r(z,p) \geq 0$. In the equilibrium point

$$\begin{aligned}H^r(z^*, 0) &= \frac{1}{2} 0^T M^{-1} 0 + \frac{1}{2}(z^* - z^*)^T K^c (z^* - z^*) \\ &= 0\end{aligned}$$

Since K^c is a positive constant and both displacement error and momentum are quadratic, it is verified that $H^r(z,p)$ is greater than zero for all values of $z \neq z^*$ and $p \neq 0$. Furthermore, the time derivative of $H^r(z,p)$, using the chain rule, is defined as

$$\begin{aligned}\dot{H}^r(z,p) &= \frac{\partial H^r}{\partial p}\frac{dp}{dt} + \frac{\partial H^r}{\partial z}\frac{dz}{dt} \\ &= \frac{\partial H^r}{\partial p}\left(-(B \otimes I_n) D^c (B^T \otimes I_n)\frac{\partial H^r}{\partial p} - (B \otimes I_n)\frac{\partial H^r}{\partial z}\right) + \frac{\partial H^r}{\partial z}(B^T \otimes I_n)\frac{\partial H^r}{\partial p} \\ &= -\frac{\partial^T H^r}{\partial p}(B \otimes I_n) D^c (B^T \otimes I_n)\frac{\partial H^r}{\partial p}\end{aligned}$$

It is easily verified that $\dot{H}^r \leq 0$, due to the minus sign and the quadratic form of the partial derivative. Therefore, using LaSalle's invariance principle, the function converges to the largest invariant set where $\dot{H}^r = 0$. Consequently, $\frac{\partial H^r}{\partial p} = 0$, which results in $p = 0$ and $\dot{p} = 0$. Thus, substituting $p = 0$ and $\dot{p} = 0$ in (2.6), yields the equation

$$\begin{aligned} 0 &= -(B \otimes I_n) D^c (B^T \otimes I_n) \cdot 0 - (B \otimes I_n) \frac{\partial H^r}{\partial z} \\ &= -(B \otimes I_n) K^c (z - z^*) \end{aligned} \tag{2.7}$$

In order for this equation to hold, either B or $(z - z^*)$ must be zero, due to the fact that K^c is a positive non-zero design constant. However, since B is the incidence matrix of an acyclic graph, the columns of matrix B are linearly independent, meaning that Rank(B)=2. Consequently, if there exists a equation as

$$(B \otimes I_n)(z - z^*) = 0$$

where $Rank(B) = n =$ number of columns and $(z - z^*)$ is a scalar, only the trivial solution ($(z - z^*) = 0$) exists. Therefore, it means that, in order for (2.7) to hold, $(z - z^*) = 0 \;\; \Rightarrow \;\; z = z^*$. Therefore, it is proven that the controller yields a closed-loop system that meets all the controller objectives. □

2.1.1 Simulations

Consider a network with number of agents $n = 3$, which are moving in the x- and y-direction, due to the triangular shape as desired formation. Every agents has a mass $m_i = 0.167kg$ for $i = 1, 2, 3$. The network consists of $e = 2$ edges, which are represented as homogeneous virtual couplings, which have a spring constant $K_j^c = 0.7kg/s^2$, a damping coefficient $D_j^c = 2kg/s$ and desired displacement $z_{j,x}^* = 0.3m$, $z_{1,y} = \sqrt{0.27} \approx 0.520m$, $z_{2,y} = -\sqrt{0.27} \approx -0.520m$ for $j = 1, 2$. The desired relative displacement is equal to the nominal spring length. The incidence matrix depends on the number of agents and the number of states space variables. Due to the x- and y-component of each agents, the incidence matrix is equal to

$$(B \otimes I_2) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

The simulations are run using Simulink and Matlab 2019b. The Simulink model used for the simulation can be found in appendix A.1. The agents are considered as point masses, meaning that they can move freely in the framework. The simulation is run for $t = 30s$ and the initial conditions are set at $q_x(0) = (1.03, 1.20, 1.32)m$, $q_y(0) = (0, 0, 0)m$, $p_x(0) = (0, 0, 0)kg\ m/s$, $p_y(0) = (0, 0, 0)kg\ m/s$. The results are shown in figures 2.1, 2.2 and 2.3.

Figure 2.1 shows the displacement of the two virtual couplings after $t = 30s$. The graph shows two lines, one going to a final value $0.3m$ and one going to a final value $\approx 0.52m$, corresponding to the desired relative displacement z_x and z_y, respectively. Due to the symmetric form of a triangle, $z_{1,x}$ and $z_{2,x}$ overlap. This can also be seen in figure 2.2, as the distance between agent 2 and agent 1, and the distance between agent 2 and agent 3, is both $+0.3m$.

Figure 2.2 shows the x- and y- position of the three agents after $t = 30s$. Due to the settling time of the displacement, the formation is achieved after $t_s = 11.0917s$. The dashed lines indicate that the agents reach a triangular formation. Due to the fact that the agents come to a full stop, the travelled distance is limited as the agents no longer move when the formation is achieved.

Figure 2.3 shows that the velocity of the three agents converge to a final value of 0. This implies that $p = 0$ as $p = \text{velocity} \cdot \text{mass} = 0 \cdot 0.167 = 0$. The sign of the velocity indicate the direction, as can be seen when the x-directional velocity of the agent 1 is compared to the x-directional velocity of agent 3, which move in complete opposite directions. Velocity $v_{1,y}$ is not distinguishable in figure 2.3, as it follows the same trajectory as velocity $v_{3,y}$.

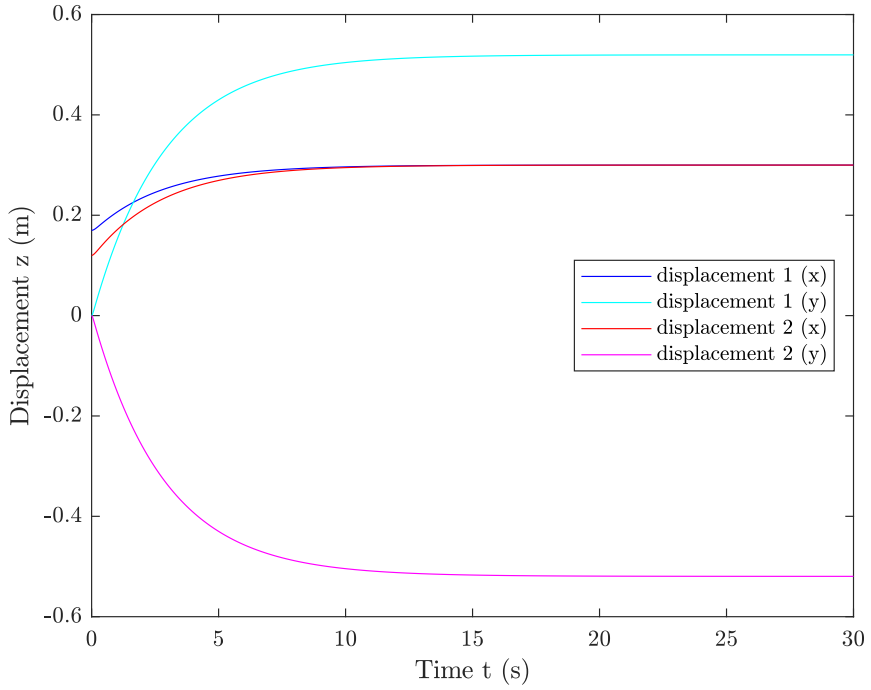

Figure 2.1: The relative displacement of the agents after implementation of the formation controller.

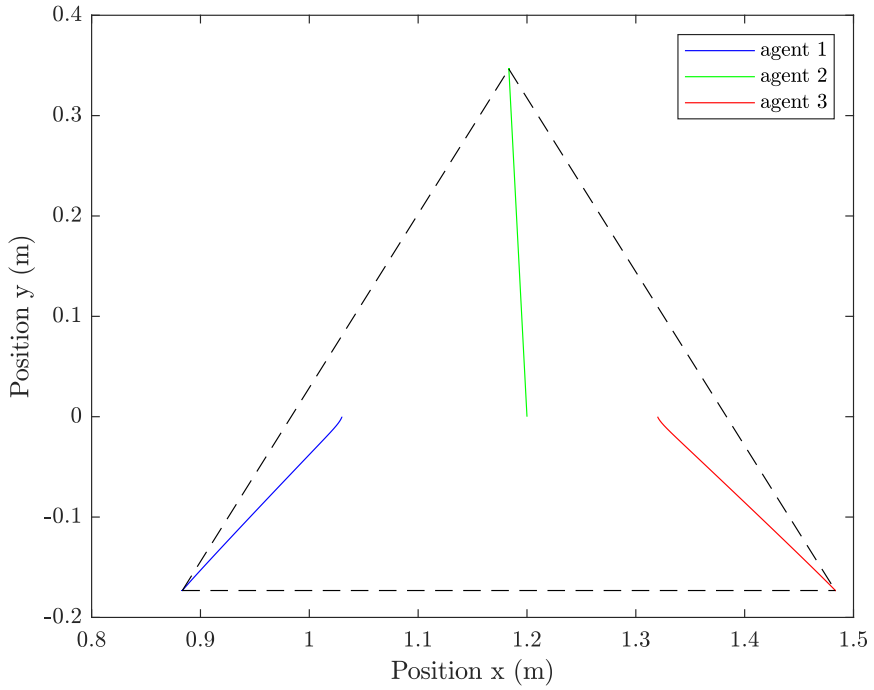

Figure 2.2: The position of the agents after implementation of the formation controller.

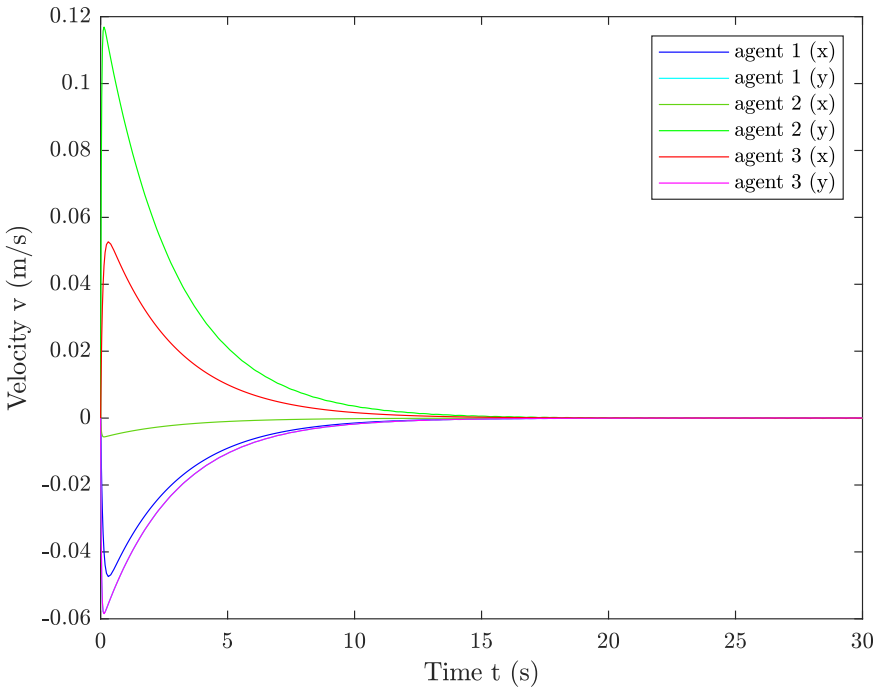

Figure 2.3: The velocity of the agents after implementation of the formation controller.

2.2 Velocity Tracking

For the velocity tracking, again, the thesis of Vos (2015) is taken as a basis and a guide for obtaining the correct error variables and state space representations of the total system, including the mechanical system and the controller. The velocity tracking controller is designed independently from the formation controller, meaning that the formation controller is not considered for the closed-loop system. However, the two controllers are combined, where the formation controller is considered in the closed-loop system.
The concept of velocity tracking is the convergence of the velocity to the desired velocity. Consequently, this yields the following control objective:

$$\begin{cases} p \to m \cdot v^*, & \text{as t} \to \infty \end{cases} \tag{2.9}$$

In order to track the desired velocity, the error between the actual velocity and the desired velocity must converge to zero. Since the agents are required to form any 2-dimensional shape, both the velocity in the x- and y-direction are considered, yielding the desired velocity $v_i^* = (v_{i,x}^*, v_{i,y}^*)^T$. For simplicity of notation, $p_i = (p_{i,x}, p_{i,y})^T$, $q_i = (q_{i,x}, q_{i,y})^T$. In order to ensure the tracking of the desired velocity, the error state variables are determined

$$\begin{bmatrix} \bar{q}_i \\ \bar{p}_i \end{bmatrix} = \begin{bmatrix} q_i - v_i^* t \\ p_i - m_i v_i^* \end{bmatrix} \tag{2.10}$$

Using the error variables, the following Hamiltonian can be defined

$$\begin{aligned} H_i^t &= \frac{1}{2} \bar{p}_i^T m_i^{-1} \bar{p}_i \\ &= \frac{1}{2} p_i^T m_i^{-1} p_i - p_i^T v_i^* + \frac{1}{2} v_i^{*T} m_i v_i^* \end{aligned}$$

To simplify notation, $H^t = \sum_{i=1}^n H_i^t(p_i) = \frac{1}{2} \bar{p}^T M^{-1} \bar{p}$. Using the notations of (1.1) to define the input and output of the mechanical system, in order to ensure that the error system converge to zero (Vos, 2015), the output of the controller must be equal to $u_{i,x}^c = -D_i^t(y_{i,x} - v_{i,x}^*)$ and $u_{i,y}^c = -D_i^t(y_{i,y} - v_{i,y}^*)$, where $u_i^c = (u_{i,x}^c, u_{i,y}^c)^T$, $y_i = (y_{i,x}, y_{i,y})^T$ and D^t is the damping coefficient for the system with $D^t = block.diag(D_1^t, D_2^t, ...D_n^t)$. Therefore, the controller dynamics can be described as

$$\begin{aligned} \dot{q}_i^c &= y_i \\ u_i^c &= -D_i^t(y_i - v_i^*) \end{aligned} \tag{2.11}$$

To simplify notation, $u^c = (u_1^c, u_2^c, ...u_n^c)^T$, $q^c = (q_1^c, q_2^c, ...q_n^c)^T$, $y = (y_1, y_2, ...y_n)^T$. Combining (1.1) with the controller dynamics of (2.11), yields the following error closed-loop system

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{\bar{p}} \end{bmatrix} = \begin{bmatrix} 0 & I_2 \\ -I_2 & -D^t I_2 \end{bmatrix} \begin{bmatrix} \frac{\partial H^t}{\partial \bar{q}} \\ \frac{\partial H^t}{\partial \bar{p}} \end{bmatrix} \tag{2.12}$$

The closed-loop system ensures that $v_{i,x} \to v_x^*$ and $v_{i,y} \to v_y^*$ for $i = 1, 2, ...n$. The closed-loop system enables velocity tracking of a non-zero desired velocity. Moreover, combining the velocity tracking controller (2.11) with the controller in (2.2) yields a closed-loop system that meets the control objectives of (2.1) and (2.9). Consider the following theorem.

Theorem 2.2. *Take $u_{i,1}$ as u_i of* (2.5) *and $u_{i,2}$ as u_i of* (2.11)*, then, using $u_i = u_{i,1} + u_{i,2}$, the system in* (1.1) *converges to $z = z^*$, $p = m \cdot v^*$ when $t \to \infty$, thereby achieving all control goals of* (2.1) and (2.9)

Proof. Define $\bar{z} = z - z^*$ to create an error state space system. For simplicity of notation, define $\bar{p} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_n)$ and $\bar{q} = (\bar{q}_1, \bar{q}_2, ..., \bar{q}_n)$. Then, the closed-loop system, obtained from (2.6) and (2.12), is defined as

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{\bar{p}} \\ \dot{\bar{z}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & -\bar{D} & -(B \otimes I_n) \\ 0 & (B^T \otimes I_n) & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \bar{H}}{\partial \bar{q}} \\ \frac{\partial \bar{H}}{\partial \bar{p}} \\ \frac{\partial \bar{H}}{\partial \bar{z}} \end{bmatrix} \tag{2.13}$$

with $\bar{D} = D^t + (B \otimes I_n)D^c(B^T \otimes I_n)$ and error Hamiltonian $\bar{H}(\bar{p}, \bar{z}) = \frac{1}{2}\bar{p}^T M^{-1}\bar{p} + \frac{1}{2}\bar{z}^T K^c \bar{z}$. Take the error Hamiltonian as a candidate Lyapunov function. It is easily verified that $\bar{H} \geq 0$

$$\bar{H}(0,0) = \frac{1}{2}0^T M^{-1}0 + \frac{1}{2}0^T K^c 0 = 0$$

Since K^c and m are both a positive non-zero constant, and both $\bar{p}$ and $\bar{z}$ are quadratic, $\bar{H}$ will be greater than zero for any input $\neq 0$. The time derivative, using the chain rule, is given by

$$\begin{aligned} \dot{\bar{H}} &= \frac{\partial \bar{H}}{\partial \bar{q}}\frac{d\bar{q}}{dt} + \frac{\partial \bar{H}}{\partial \bar{p}}\frac{d\bar{p}}{dt} + \frac{\partial \bar{H}}{\partial \bar{z}}\frac{d\bar{z}}{dt} \\ &= \frac{\partial \bar{H}}{\partial \bar{q}}\frac{\partial \bar{H}}{\partial \bar{p}} + \frac{\partial \bar{H}}{\partial \bar{p}}\left(-\frac{\partial \bar{H}}{\partial \bar{q}} - \bar{D}\frac{\partial \bar{H}}{\partial \bar{p}} - (B \otimes I_n)\frac{\partial \bar{H}}{\partial \bar{z}}\right) + \frac{\partial \bar{H}}{\partial \bar{z}}(B^T \otimes I_n)\frac{\partial \bar{H}}{\partial \bar{p}} \\ &= -\frac{\partial^T \bar{H}}{\partial \bar{p}}\bar{D}\frac{\partial \bar{H}}{\partial \bar{p}} \end{aligned}$$

Due to the minus sign and the quadratic form for the partial derivative, it can be verified that $\dot{\bar{H}} \leq 0$. Invoking LaSalle's invariance principle, yields that (2.13) converges to the largest invariant set where $\dot{\bar{H}} = 0$ and, consequently, $\frac{\partial \bar{H}}{\partial \bar{p}} = 0$. This results in $\bar{p} = 0$ and $\dot{\bar{p}} = 0$, achieving the first control objective of velocity tracking. Filling in $\bar{p} = 0$ and $\dot{\bar{p}} = 0$ in (2.13), gives the equation

$$\begin{aligned} 0 &= \frac{\partial \bar{H}}{\partial \bar{q}} - \bar{D} \cdot 0 - (B \otimes I_n)\frac{\partial \bar{H}}{\partial \bar{z}} \\ &= -(B \otimes I_n)K^c \bar{z} \end{aligned}$$

This equation can only hold if B or $\bar{z}$ is equal to zero, as K^c is a positive non-zero design constant. However, since B is the incidence matrix of a acyclic graph, the

columns are linearly independent and $Rank(B) = 2$. Since the number of columns of B is equal to the rank, for the equation $(B \otimes I_n)\bar{z} = 0$, with matrix $(B \otimes I_n)$ and scalar $\bar{z}$, only the trivial solution exist where $\bar{z} = 0$. Thereby completing the proof. □

2.2.1 Simulations

Consider again a network with number of agents $n = 3$, where the mass of every agent $m_i = 0.167kg$. For the closed-loop system, the damping coefficient of the controller is defined as $D_i^t = 2kg/s$ and with desired velocity $v_{i,x}^* = 0.2m/s$ and $v_{i,y}^* = 0.3m/s$ for $i = 1, 2, 3$.

The simulations are run using Simulink and Matlab 2019b. First, the velocity tracking closed-loop system is simulated individually. The Simulink model used for the simulation can be found in appendix A.2. The agents are considered as point masses, meaning that they can move freely in the framework. The simulation is run for $t = 3s$, to show the rise and settling time. The initial conditions are set at $q_x = (1.03, 1.20, 1.32)m$, $q_y = (0, 0, 0)m$, $p_x = (0, 0, 0)kg\ m/s$, $p_y = (0, 0, 0)kg\ m/s$. The initial conditions are set to be equal to the initial conditions of the formation controller, to yield comparable results. The result is shown in figure 2.4. It can be seen in figure 2.4 that the two lines converge to $v_{i,x} \to v_{i,x}^* = 0.2m/s$ and $v_{i,y} \to v_{i,y}^* = 0.3m/s$ for $i = 1, 2, 3$. Since the desired velocity is equal for all agents, the results of the other agents follow the same line. Dots and dashed lines are used to show the individual results as much as possible.

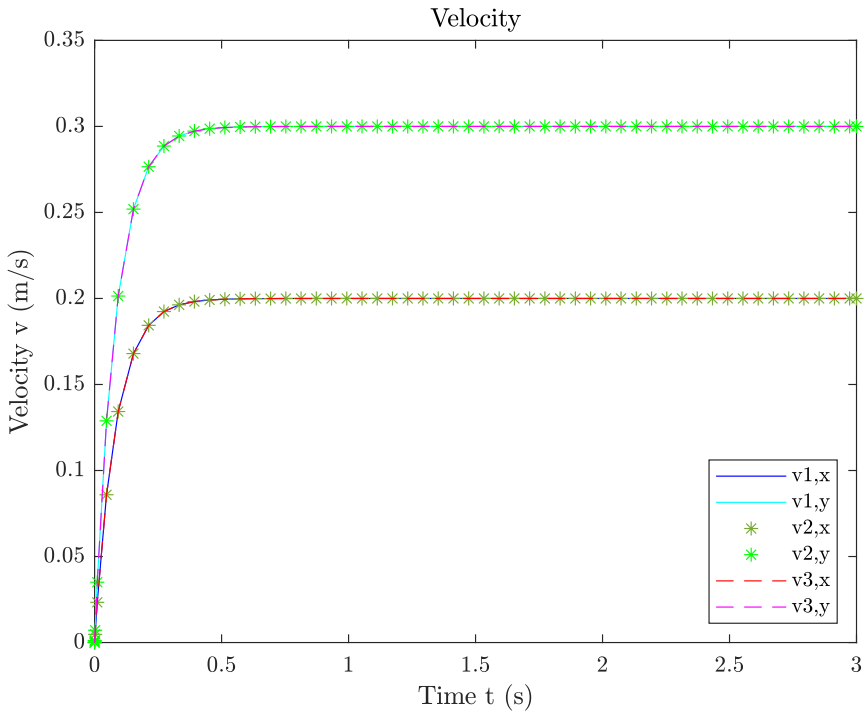

Figure 2.4: The final velocity in x- and y-direction for all the agents in the system, only considering velocity tracking

As per theorem 2.2 and the corresponding proof, if the controller for formation control and velocity tracking are combined, all control objectives (2.1)(2.9) are met. The proof is verified with simulations which are performed using the same initial conditions as for the simulation of the formation control and velocity tracking, with $t = 30s$. The Simulink model can be found in appendix A.3. The results are shown in figures 2.5, 2.6, and 2.7.

Figure 2.5 shows the convergence of the displacement to the desired displacement, indicating $z \to z^*$. In the combined scenario ($t_s = 20.9702s$), the speed of convergence is slower compared to the formation controller ($t_s = 11.0917s$). This difference can be explained by the the movement with a constant velocity of the agents, which hinders the stabilisation of a constant relative displacement.

Figure 2.6 shows a linear graph, which is due to the fact that the agents track a constant velocity; thus constantly move away from their original point. It may seem that agents 1 and 2 cross paths, but this is not true due to the fact that agent 2 moves ahead of agent 1. Furthermore, it can be seen that, at the end-points, the triangular formation is achieved.

Figure 2.7 displays $v_{i,x} \to v^*_{i,x} = 0.2m/s$ and $v_{i,y} \to v^*_{i,y} = 0.3m/s$ for $i = 1, 2, 3$, indicating that the velocities of the agents all converge to the desired value for the final velocity. The peak of the y-velocity agent 2 is explained by the fact that agent 2 is positioned ahead of the other agents.

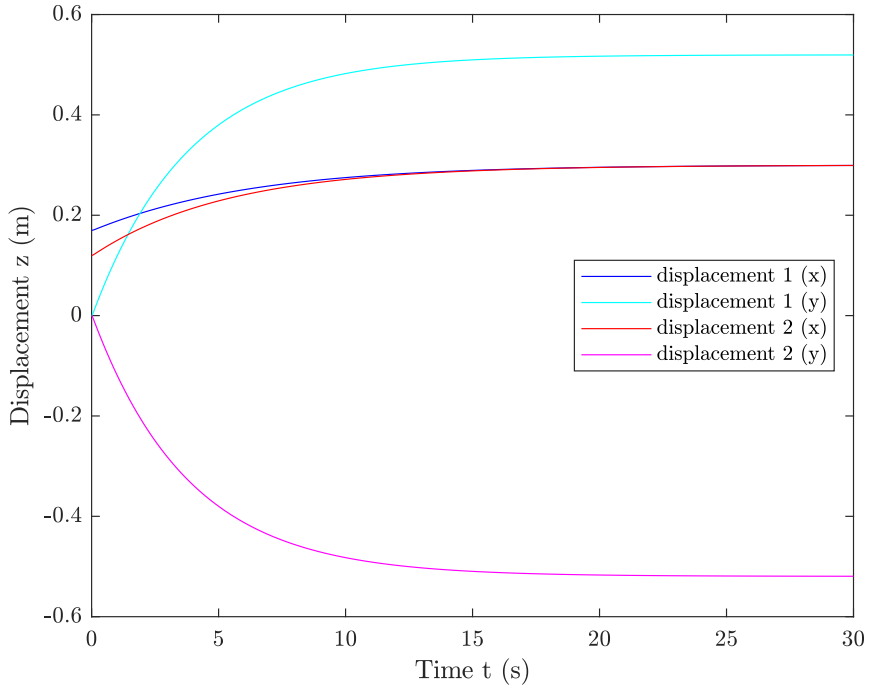

Figure 2.5: The relative displacement of the agents after implementation of the formation controller and the velocity tracking controller

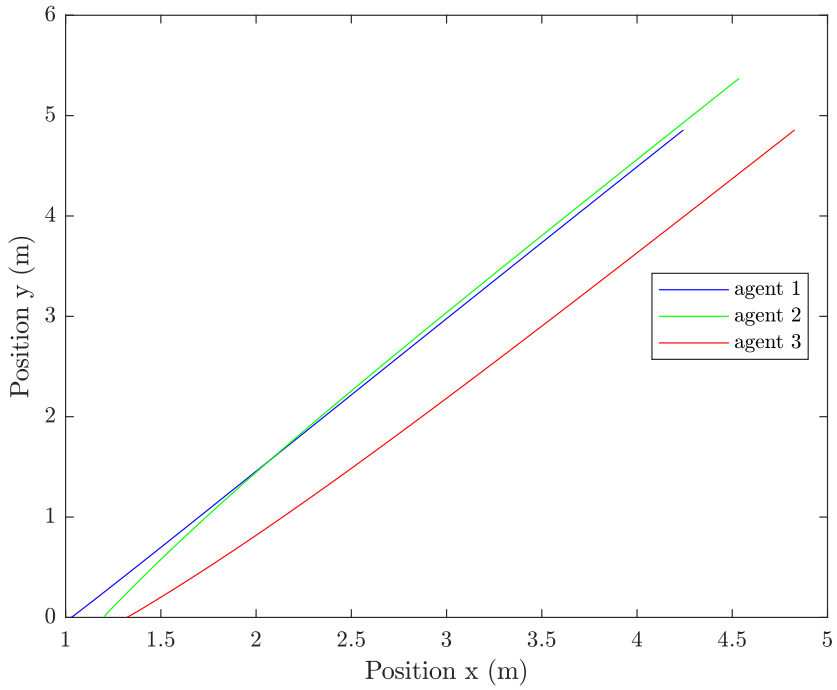

Figure 2.6: The position of the agents after implementation of the formation controller and the velocity tracking controller

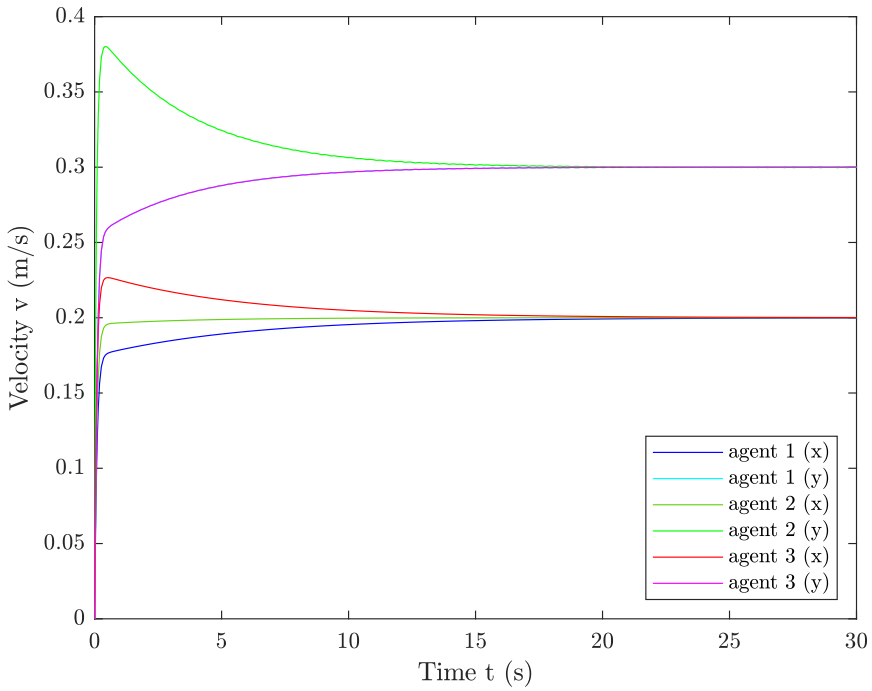

Figure 2.7: The velocity of the agents after implementation of the formation controller and the velocity tracking controller

2.3 Leader-follower

In section 2.1 and section 2.2, all the agents are required to know the desired displacement and the desired velocity. With the leader-follower strategy, it is assumed that the follower agents do not know the desired velocity. The necessity for all this data leads to larger CPUs required for every agent. To overcome this issue, the implementation of a leader-follower strategy can utilised. The leader-follower strategy specifies one or more leaders within the network which knows the desired velocity and will try to converge to that velocity. The other agents present in the network do not know the desired velocity, and only measure the relative displacement with respect to the leader. This leads to follower agents requiring less equipment and the whole network to be more cost-efficient (Kang et al., 2014). Depending on the number of agents in the network (n), the number of leaders (n_ℓ) is determined. Based on the number of leaders, the number of followers $n_f = n - n_\ell$ can be determined. It is assumed that all followers are directly connected to the leader. The objectives for the leader follower strategy, based on (2.1) and (2.9), are defined as

for the leader

$$\begin{cases} p_\ell \to m \cdot v^*, & \text{as } t \to \infty \end{cases} \tag{2.14}$$

for the follower

$$\begin{cases} z \to z^*, \\ \\ p_f \to p_\ell, \end{cases} \quad \text{as } t \to \infty \tag{2.15}$$

The proposed controller uses the same concepts as the formation controller (2.2) and is applied to all the follower agents, whereas the leader agent is controlled by the velocity tracking controller in section 2.2. Therefore, the first control objective, concerning the velocity of the leader, is not considered in the following controller. By assuming that all the followers are directly connected to the leader, the incidence matrix (2.4) is not applicable. Instead, an adjusted matrix is used, denoted as B^*, which is a matrix that consists of only the incidence matrix rows that belong to the respective follower. In order to track the leader's velocity, the follower agents have error variables defined as

$$\begin{bmatrix} \bar{p}_f \\ \bar{z}_f \end{bmatrix} = \begin{bmatrix} p_f - p_\ell \\ z_f - z_f^* \end{bmatrix} \tag{2.16}$$

where z_f is the displacement between the leader and follower, z_f^* is the desired displacement, p_f is the momenta of the follower agent and p_ℓ is the leader agent's momenta and is considered as an constant. For each follower, the formation control dynamics are

$$\begin{aligned} \dot{z}_{f,j} &= w_{f,j} \\ \tau f, j &= \frac{\partial H^{zf}}{\partial z_{f,j}} + D^c w_{f,j} \end{aligned} \tag{2.17}$$

where $H^{zf} = \frac{1}{2}(z_{f,j} - z^*_{f,j})^T K^c (z_{f,j} - z^*_{f,j})$ is the Hamiltonian, $\tau_f = (\tau_{f,1}, \tau_{f,2}, ...\tau_{f,e})^T$is the output force of the leader controller , $w_f = (w_{f,1}, w_{f,2}, ...w_{f,e})^T$ is the input velocity and $D^c = block.diag(D^c_1, D^c_2, ..., D^c_e)$is the damping coefficient to improve transient properties. In order to simplify notation, $z_f = (z_{f,1}, z_{f,2}, ...z_{f,e})^T$, $z^*_f = (z^*_{f,1}, z^*_{f,2}, ...z^*_{f,e})^T$. The velocity of the leader can be considered as a constant, because the agent is not dependent on any other agent and thus will converge to v^* due to the velocity tracking controller (2.11) being applied to the leader agent. The follower agents are only able to measure the displacement with respect to the leader agent and velocity of the leader, which means that the desired velocity is not known for the follower agents. Furthermore, it is assumed that the follower agents are capable of measuring their own velocity. Using this information, the coupling equations, based on (2.5), are

$$\begin{aligned} w_f &= (B^{*T} \otimes I_n)(y_f - \frac{p_\ell}{m}) \\ u_f &= -(B^* \otimes I_n)\tau_f \end{aligned} \tag{2.18}$$

Where $y_f = (y_{1,f}, y_{2,f}, ...y_{n_f,f})^T$ is the output velocity of the mechanical system, m is the mass of the leader, in this case equal to the mass of the followers. Using the controller dynamics (2.17), error variable (2.16) and the coupling equations (2.18), the following closed loop exists

$$\begin{bmatrix} \dot{\bar{p}}_f \\ \\ \dot{\bar{z}} \end{bmatrix} = \begin{bmatrix} -(B^* \otimes I_n)D^c(B^{*T} \otimes I_n) & -(B^* \otimes I_n) \\ \\ (B^{*T} \otimes I_n) & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H^f}{\partial \bar{p}_f} \\ \\ \frac{\partial H^f}{\partial \bar{z}} \end{bmatrix} \tag{2.19}$$

With Hamiltonian $H^f(\bar{p}_f, \bar{z}) = \sum_{i=1}^{n_f 1} H_i(\bar{p}_{i,f}) + \sum_{j=1}^{e_f} H^{zf}_j(z_j) = \frac{1}{2}\bar{p}^T_f M^{-1} \bar{p}_f + \frac{1}{2}\bar{z}^T K^c \bar{z}$. Applying the closed loop system to the first follower agent, will meet the controller objectives. Consider the following theorem.

Theorem 2.3. *Using the velocity tracking controller* (2.11) *for the leader agent and the controller* u_f (2.18) *for the follower agents, the total system* (1.1) *converges to* $p_\ell \to mv^*$, $p_f \to p_\ell$, $z \to z^*$, *when* $t \to \infty$. *Thereby achieving the control objectives* (2.14) (2.15) *for the whole system.*

Proof. Taking $H^f(\bar{p}_f, \bar{z})$ as a Lyapunov function candidate, it can be verified that $H^f(\bar{p}_f, \bar{z}) \geq 0$. In the equilibrium point

$$\begin{aligned} H^f(0,0) &= \frac{1}{2} \cdot 0 \cdot M^{-1} \cdot 0 + \frac{1}{2} \cdot 0 \cdot K^c \cdot 0 \\ &= 0 \end{aligned}$$

The follower momentum and the displacement have a quadratic form which means that those parameters will always be semi-positive. Moreover, the positive design parameter K^c and the absolute value for the mass M, yields an equation $H^f(\bar{p}_f, \bar{z})$ which will have a positive value for every input $\neq 0$. Furthermore, the time derivative of the Hamiltonian is given by

$$\begin{aligned}\dot{H}^f &= \frac{\partial H^f}{\partial \bar{p}_f}\frac{d\bar{p}_f}{dt} + \frac{\partial H^f}{\partial \bar{z}}\frac{d\bar{z}}{dt} \\ &= -\frac{\partial^T H^f}{\partial \bar{p}_f}(B^* \otimes I_n)D^c(B^{*T} \otimes I_n)\frac{\partial H^f}{\partial \bar{p}_f}\end{aligned}$$

It is verified that $\dot{H}^f \leq 0$ due to the quadratic form of the partial derivative and adjusted incidence matrix, the positive design constant D^c and the minus sign. Invoking LaSalle's invariance principle, the system will converge to the state where $\dot{H}^f = 0$, which can only hold if $\frac{\partial H^f}{\partial \bar{p}_f} = 0$ and, consequently, $\bar{p}_f = 0$ and $\dot{\bar{p}}_f = 0$, meaning that $p_f \to p_\ell$, thereby meeting the first control objective. Filling in these values in (2.19), yields

$$\begin{aligned}0 &= -(B^* \otimes I_n)D^c(B^{*T} \otimes I_n) \cdot 0 - (B^* \otimes I_n)\frac{\partial H^f}{\partial z} \\ &= -(B^* \otimes I_n) \cdot \bar{z} \cdot K^c\end{aligned}$$

For this equation to hold, $\bar{z}$ or B^* must be equal to zero, as K^c is a non-zero design constant. Since the column of B are linearly independent, it can be concluded that the columns of B^* are also linearly independent. Using the same concept as the proof 2.1 and section 2.2, then, it can be concluded that if B^* is linearly independent and the above equation is of the form $(B^* \otimes I_n)\bar{z} = 0$, the trivial solution exists. Therefore, it can be concluded that the displacement error $\bar{z} = 0$. Furthermore, it can be said that the leader-follower strategy results in $p_\ell \to mv^*$ via the velocity tracking controller and $p_f \to p_\ell$ and $\bar{z} \to 0$ via the adjusted formation controller, thereby completing the proof. □

2.3.1 Simulations

Consider a network with number of agents $n = 3$ which are moving in x- and y-direction. Every agents has a mass $m_i = 0.167kg$ for $i = 1, 2, 3$. The network consists of $e = 2$ number of edges, which are represented as homogeneous virtual couplings, which have a spring constant $K_j^c = 0.7kg/s^2$, a damping coefficient $D_j^c = 2kg/s$ and desired displacement $z_{j,x}^* = 0.3m$ and $z_{1,y} = \sqrt{0.27} \approx 0.520m$, $z_{2,y} = -\sqrt{0.27} \approx -0.520m$ for $j = 1, 2$. For the velocity tracking of the leader agent, the damping coefficient of the controller is defined as $D_i^t = 2kg/s$ and desired velocities $v_{i,x}^* = 0.2m/s$, $v_{i,y}^* = 0.3m/s$ for $i = 1$

The simulations are run with Matlab2019b and Simulink. The simulink model used for the simulation can be found in appendix A.4. The initial conditions for the leader agent are $q_x = 1.20m$, $q_y = 0m$, $p_x = 0kg\ m/s$, $p_y = 0kg\ m/s$. The initial conditions for the first follower agent is $q_x = 1.03m$, $q_y = 0m$, $p_x = 0kg\ m/s$, $p_y = 0kg\ m/s$. The initial conditions for the remaining follower agent is $q_x = 1.32m$, $q_y = 0m$, $p_x = 0kg\ m/s$, $p_y = 0kg\ m/s$. The results can be found in figures 2.8 2.9 2.10.

Figure 2.8 visualises how the three agents move with respect to a global framework. As the figure shows, when the leader agent is only tracking the desired velocity

and the follower agents are only subject to a formation controller, the triangular formation is achieved. The time displayed is 15 seconds, as the maximum settling time for the displacement is equal to $t_s = 10.6149s$. The figure shows that, when the formation is achieved, it does not change, indicating a steady state. Furthermore, it should be noted that, although it may seem as if the leader agents and the first follower cross paths, this is not true. The graph should be read, bearing in mind that the leader agent moves ahead of the follower agents, meaning that the agents will never cross paths. This is illustrated in the graph as the end-position of the leader is above the follower agents.

Figure 2.9 clearly follows a different trajectory compared to the velocity graphs of the formation controller and the velocity tracking controller (figures 2.3 and 2.4), but do converge to the desired velocities $v^*_{i,y} = 0.3m/s$ and $v^*_{i,x} = 0.2m/s$. The graph shows that the velocity of the leader converges relatively quick to the desired velocity, which is supported by the settling time of $t_s = 2.0282s$. The velocities of the followers take more time to converge to the desired values, ($t_s = 9.7721s$). Both followers have the same y-velocity trajectory, meaning that they have the same maximum settling time.

Figure 2.10 shows the displacement between the leader and the follower agents. Again, the trajectories are different from the formation controller. Nonetheless, the displacement still converge to the desired displacement. *Displacement 1 (y)* is not distinguishable in the graph, because it follows the same trajectory as *displacement 2 (y)*. Both follower agents reach a displacement of $\approx -\sqrt{0.27}$, as can be seen in figure 2.8.

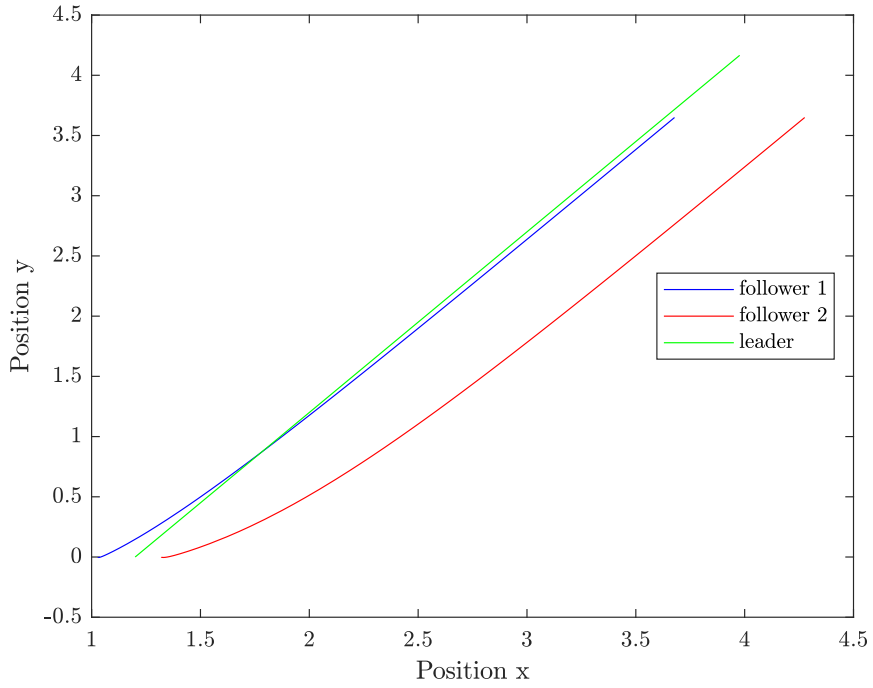

Figure 2.8: The position graph of the three agents using the leader follower strategy

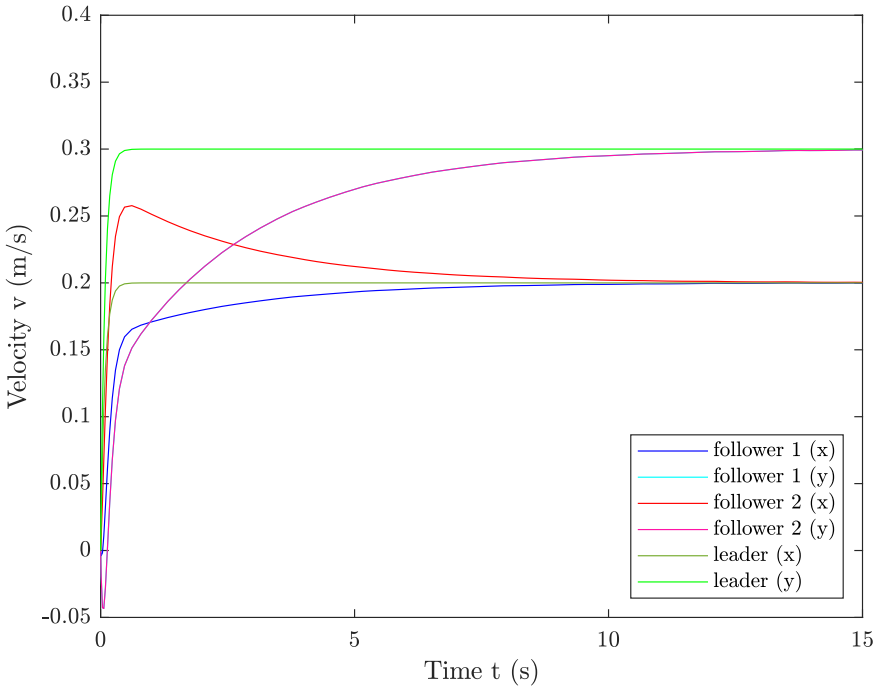

Figure 2.9: The velocity graph of the follower agents

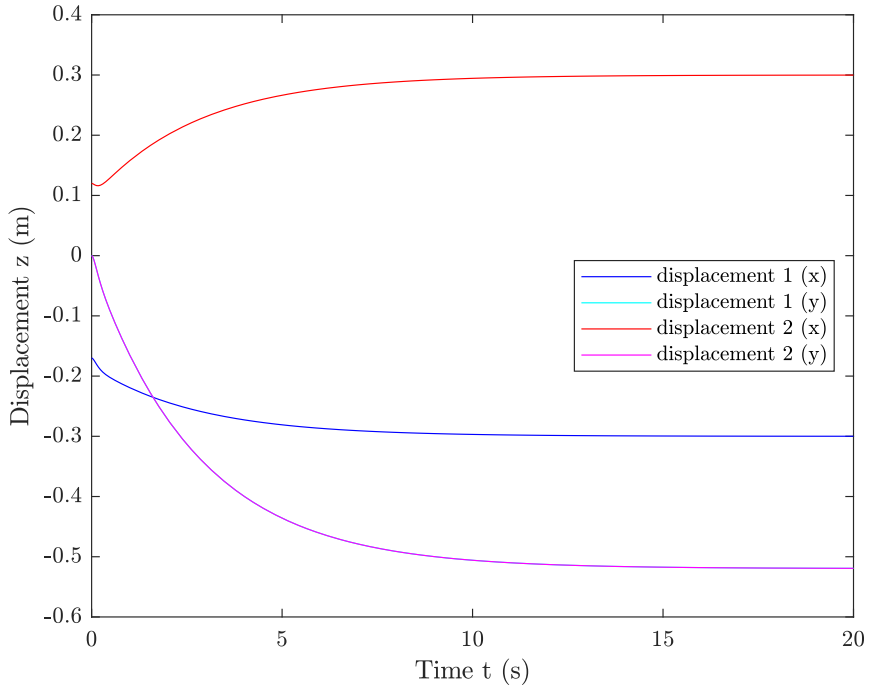

Figure 2.10: The displacement graph of the three agents using the leader follower strategy

Chapter 3

Conclusion & Future Research

3.1 Conclusion

This section will provide the conclusions that can be drawn from the research and the obtained results. The results show that when the correct controller is implemented to the mechanical system, the control objectives are met and the control variables converge to the desired value, which is confirmed by the simulation results. The implementation of the system in a port-Hamiltonian framework provides an overview about the energy flows within a network and allows for checking of the stability via the Lyapunov method. Moreover, it is confirmed that with the proposed formation controller (2.2), any 2-dimensional shape can be achieved, based on the number of agents in the network. Furthermore, it can be concluded that the controller yields a stable closed-loop system which means that the formation is held when $t \to \infty$. The controller for the velocity tracking (2.11) achieves the control objective of tracking a desired velocity and when the formation and velocity tracking controllers are combined and both the controllers are implemented as an input to the mechanical system, it can be concluded that the closed loop system is stable and meets all control objectives of formation control and velocity tracking.

Furthermore, the two controllers are used for the leader-follower strategy in the port-Hamiltonian framework. The leader agent is merely controlled by the velocity tracking controller (2.11) and the followers are controlled by the adjusted formation controller (2.17). The formation controller for the followers only measures the displacement with respect to the leader agent and the velocity of the leader. However, the adjusted formation controller does not use the information of the desired velocity, yet still converge to the desired velocity. Based on the simulation results, it can be concluded that, with the leader-follower strategy, where multiple agents in a network are controlled by different controllers, a stable system is achieved which meets all control objectives. Therefore, by meeting the control objectives, it can be concluded that the final design meets the objective of the research, and, therefore, will be a partial solution to the problem presented in the problem statement.

3.2 Future Research

The Future research section will elaborate on possible additional research related to the topic of formation control in a port-Hamiltonian framework. The first point of interest is the disturbance rejection in the model. Due to time constraints and the complexity of the disturbance characteristics, the model presented in this research does not consider disturbances of any sort, meaning that the robustness of the controllers is not tested. By not testing the robustness, it is not implied that the controllers are not robust at all. However, it is recommended for future research to consider disturbance rejection for the controllers to accommodate to real-life disturbance problems.

The second point of interest is the time-varying desired velocity for the velocity tracking controller. This research only considered the parameter 'desired velocity' to be a constant value over time. If the controllers can be designed to track a time-varying velocity, the controllers become more widely applicable. The advantage of a time-varying velocity is that it can support velocities in different direction if the rigid body is required to follow a certain path that requires direction changes. Moreover, for other practical use, i.e. mine sweeping, a velocity is required that can adopt to the environment and react accordingly. Again, due to time constraints and the complexity of the time-varying desired velocity, it is not included in this research.

The third point of interest is the possibility of heading, which is again related to the movement of the system as a rigid body. In order to implement heading, the agents must be wheeled robots and the front- and rear-end of the wheeled robots should be defined, to establish the reference points for the heading. The heading of an agent determines how the agent is positioned, with respect to a global framework. If a wheeled robot is required to rotate around it's own axis, the heading of the robot determines how much rotation is required. The research did not consider the heading, because of the time constraint and the non-availability for wheeled robots, due to the Covid-19 pandemic.

The fourth and final point of interest, is the use of a double-integrator model for the agents. The research only considers a single-integrator agent, meaning that the agent is only subject to a damping force. The double-integrator agent is modelled as a mass which is subject to a damping force and a spring force. The double-integrator representation is closer to reality but also more complex. Due to time constraints and the absence of the required knowledge, the modelling of the double-integrator is omitted.

All the aforementioned points can help in improving the quality and validation of the controller and final results. Therefore, it is recommended for future research to consider and include the four improvement topics.

Appendix A

Simulink models

A.1 Formation control

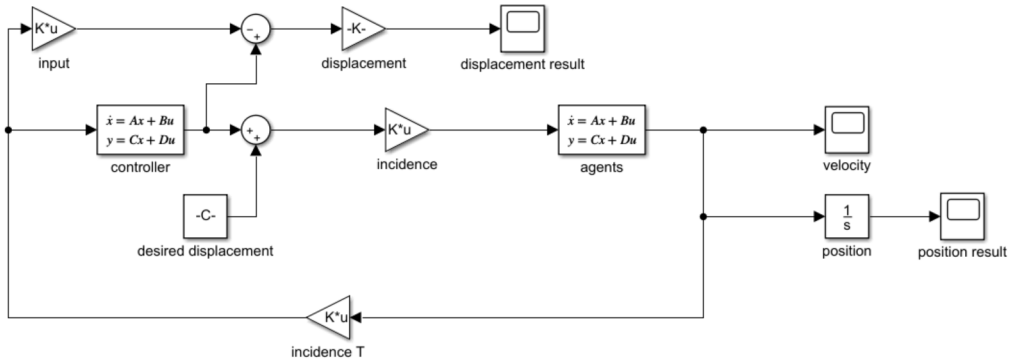

Figure A.1: The Simulink model used for the formation controller

A.2 Velocity tracking

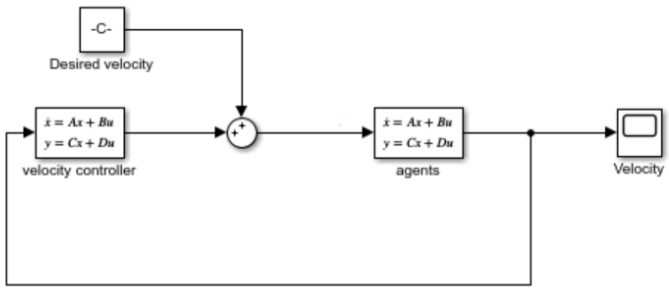

Figure A.2: The Simulink model used for the velocity tracking controller

A.3 Formation control and velocity tracking

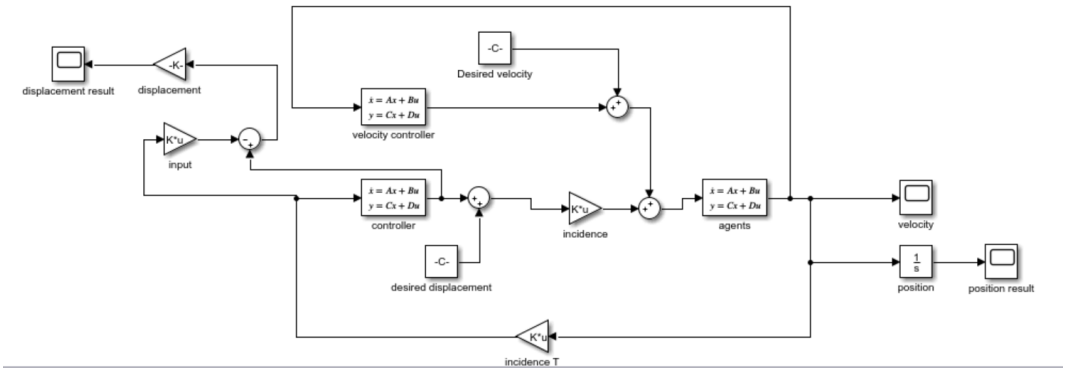

Figure A.3: The Simulink model used for the combination of the formation control and the velocity tracking

A.4 Leader-follower strategy

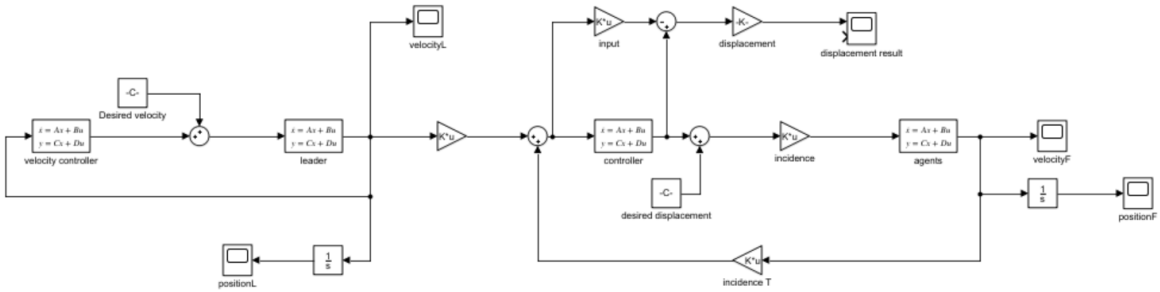

Figure A.4: The simulink model used for the leader-follower strategy, where the formation controller and the velocity tracking controller are used separately

Appendix B

Matlab scripts

B.1 Formation control

```
n=3;
e=2;
Kc=0.7;                     %spring constant for virtual coupling
Dc=2;                       %damping coefficient for virtual coupling
m=0.167;                    %mass of each agent
z1_star=[0.3;sqrt(0.27)];
z2_star=[0.3;-sqrt(0.27)];   %desired relative displacement with
                            %the x-coordinate defined first


A_plant=zeros(4*n);          %creates a matrix of 12x12 with only zeros
for i=1:(2*n)                   %replaces entries of A with the value of 1/m
        j=i+(2*n);
        A_plant(i,j)=1/m;
 end

B_plant=[zeros(2*n);eye(2*n)];
C_plant=[zeros(2*n) 1/m*eye(2*n)];
D_plant=zeros(2*n);
%B_plant yields a 12x6 matrix as the B-matrix for the state space
%representation
%C_plant yields a 6x12 matrix as the C-matrix for the state space
%represenation
%D_plant is a 6x6 matrix with only zeros as entries

 C_controller=Kc*eye(2*e);      %4x4 identity matrix multiplied with the
                                %spring constant

 D_controller=Dc*eye(2*e);      %4x4 identity matrix multiplied with the
                                %damping coefficient
```

Figure B.1: Matlab script used for the formation control with initial conditions and the matrices for the mechanical system and the controller

B.2 Velocity tracking

```
n=3;
e=2;
v_x=0.2;                                %desired velocity in the x-direction
v_y=0.3;                                %desired velocity in the y-direction
v_star=[v_x;v_y;v_x;v_y;v_x;v_y];        %total desired velocity vector

Dt=2;                                   %damping coefficient in the system

A_velocity=zeros(2*n);                    %6x6 matrix with only zeros as entries
B_velocity=eye(2*n);                      %6x6 identity matrix
C_velocity=zeros(2*n);                    %6x6 matrix with only zeros as entries
D_velocity=-Dt*eye(2*n);                  %6x6 identity matrix multiplied with
                                        %minus the damping coefficient
```

Figure B.2: Matlab script used for the velocity tracking describing the initial conditions and matrices of the controller

References

Ahn, H.-S. (2020). *Formation control*. Springer.

Bonadies, S., Lefcourt, A., & Gadsden, S. A. (2016). A survey of unmanned ground vehicles with applications to agricultural and environmental sensing. In *Autonomous air and ground sensing systems for agricultural optimization and phenotyping* (Vol. 9866, p. 98660Q).

Cao, K., Li, X., & Xie, L. (2019). Preview-based discrete-time dynamic formation control over directed networks via matrix-valued laplacian. *IEEE transactions on cybernetics*.

Cao, Y., Yu, W., Ren, W., & Chen, G. (2012). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, *9*(1), 427–438.

Dreano, F. (2018, May 29). *System and method for enhancing distribution logistics and increasing surveillance ranges with unmanned aerial vehicles and a dock network.* Google Patents. (US Patent 9,984,347)

Duindam, V., Macchelli, A., Stramigioli, S., & Bruyninckx, H. (2009). *Modeling and control of complex physical systems: the port-hamiltonian approach*. Springer Science & Business Media.

Easley, D., & Kleinberg, J. (2010). *Networks crowds and markets: Reasoning about a highly connected world*. Cambridge University Press.

Healey, A. J. (2001). Application of formation control for multi-vehicle robotic minesweeping. In *Proceedings of the 40th ieee conference on decision and control (cat. no. 01ch37228)* (Vol. 2, pp. 1497–1502).

Kang, S.-M., Park, M.-C., Lee, B.-H., & Ahn, H.-S. (2014). Distance-based formation control with a single moving leader. In *2014 american control conference* (pp. 305–310).

Krizmancic, M., Arbanas, B., Petrovic, T., Petric, F., & Bogdan, S. (2020). Cooperative aerial-ground multi-robot system for automated construction tasks. *IEEE Robotics and Automation Letters*.

Kuru, K., Ansell, D., Khan, W., & Yetgin, H. (2019). Analysis and optimization of unmanned aerial vehicle swarms in logistics: An intelligent delivery platform. *IEEE Access*, *7*, 15804–15831.

Meng, D., & Jia, Y. (2014). Formation control for multi-agent systems through an iterative learning design approach. *International Journal of Robust and Nonlinear Control*, *24*(2), 340–361.

Merris, R. (1994). Laplacian matrices of graphs: a survey. *Linear algebra and its applications*, *197*, 143–176.

National Research Council, Division on Engineering & Physical Sciences, Board on Army Science & Technology and Committee on Army Unmanned Ground Vehicle Technology. (2003). *Technology development for army unmanned ground vehicles*. National Academies Press. (Chapter 3)

Naval Studies Board, Division on Engineering & Physical Sciences, Committee on Autonomous Vehicles in Support of Naval Operations and National Research Council. (2005). *Autonomous vehicles in support of naval operations*. National Academies Press.

Oh, K.-K., Park, M.-C., & Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, *53*, 424–440.

Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*(1), 215–233.

Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, *49*(9), 1520–1533.

Parkinson, B. W., Enge, P., Axelrad, P., & Spilker Jr, J. J. (1996). *Global positioning system: Theory and applications, volume ii*. American Institute of Aeronautics and Astronautics.

Peng, Z., Hu, J., Shi, K., Luo, R., Huang, R., Ghosh, B. K., & Huang, J. (2020). A novel optimal bipartite consensus control scheme for unknown multi-agent systems via model-free reinforcement learning. *Applied Mathematics and Computation*, *369*, 124821.

Quaranta, P. (2016). *Unmanned ground vehicle developments.* (Vol. 40) (No. 5). Monch Publishing Group.

Saber, R. O., & Murray, R. M. (2003). Consensus protocols for networks of dynamic agents. *Proceedings of the 2003 American Control Conference*, *2*.

Scharf, D. P., Hadaegh, F. Y., & Ploen, S. R. (2003). A survey of spacecraft formation flying guidance and control (part i): Guidance. *Proceedings of the 2003 American control conference*, 1733–1739.

Scharf, D. P., Hadaegh, F. Y., & Ploen, S. R. (2004). A survey of spacecraft formation flying guidance and control. part ii: Control. In *Proceedings of the 2004 american control conference* (Vol. 4, pp. 2976–2985).

Shiue, Y.-C., & Chang, C.-C. (2010). Forecasting unmanned vehicle technologies: Use of patent map. In *2010 second international conference on computer research and development* (pp. 752–755).

Tran, V. P., Garratt, M., & Petersen, I. R. (2020). Switching time-invariant formation control of a collaborative multi-agent system using negative imaginary systems theory. *Control Engineering Practice*, *95*, 104245.

Van Der Schaft, A. (2019). Port-hamiltonian modeling for control. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*.

van der Schaft, A., & Jeltsema, D. (2014). Port-hamiltonian systems theory: An introductory overview. *Foundations and Trends® in Systems and Control*, *1*(2-3), 173–378.

Van Der Schaft, A., & Maschke, B. (2013). Port-hamiltonian systems on graphs. *SIAM Journal on Control and Optimization*, *51*(2), 906–937.

Verschuren, P., & Doorewaard, H. (2010). *Designing a research project* (Vol. 2). Eleven International Publishing The Hague.

Vos, E. (2015). *Formation control in the port-hamiltonian framework* (Unpublished doctoral dissertation). University of Groningen.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Xu, Y., Fitz-Coy, N., Lind, R., & Tatsch, A. (2007). μ control for satellites formation flying. *Journal of Aerospace Engineering*, *20*(1), 10–21.

Zhao, S. (2018). Affine formation maneuver control of multiagent systems. *IEEE Transactions on Automatic Control*, *63*(12), 4140–4155.