
**School of Electronics, Electrical Engineering and
Computer Science**

Queen's University Belfast

ELE8060 MSc Project

**Project Title: Development and Implementation of a
Tensegrity-based Formation Controller**

Student Name: *Juan C. Amortegui Cuevas*

Student Registration Number: *40130659*

Project Supervisor: *Dr W Naeem*

Project Moderator: *Prof S McLoone*

September 4, 2014

Declaration of Originality

I certify that this dissertation is my own original work, except where stated or referenced.

Signed

Date

Acknowledgements

I would like to take this opportunity to express my deepest appreciation to those people who made the realisation of this project possible.

I would like to thank my supervisor Dr Wasif Naeem for his guidance, continuous support and constructive advice throughout this project.

I would also like to express my gratitude to Sook Lau for enlightening me with her knowledge when I needed it the most.

I would like to acknowledge the academic and technical support of the Queen's University Belfast, UK for being excellent hosts during my Erasmus exchange in Belfast as well as the international office of the Escola Tècnica Superior d'Enginieria Industrial de Barcelona (ETSEIB) of the Universitat Politècnica de Catalunya for their help before and during this process. Particularly, an enormous thanks to the staff in the School of Electronics, Electrical Engineering and Computer Science (EEECS) for providing the hardware and facilities needed for my project.

I would also like to thank the Erasmus Programme, and all the people involved, for giving us this life-enriching experience and for the financial support provided.

I am endlessly indebted to my family for supporting emotionally and financially, I would not be here if it had not been for you. I also want to thank my friends Louie, Jagger, Rocco and Donal for cheering me up when I felt sad, bored or angry after having a not very fruitful day.

Thank you all.

Abstract

The formation control or cooperative work of multiple unmanned vehicles has proved to be more effective in terms of time consumption and overall performance in a wide variety of applications. This project aims to review some of the research which has been done up to date and in the subsequent chapters a tensegrity-based formation controller is developed and implemented to assess its global behaviour. This development starts with the design of the heading angle control of a vessel based on the first order Nomoto model which is later transformed into a waypoint problem. Later on, a globally decentralised formation topology and the tensegrity concept is introduced in the context of formation control where a sliding mode controller, a linear-quadratic regulator servo controller and a conventional proportional-integral-derivative controller are designed and simulated to regulate the elongation of a virtual spring-mass-damper system which connects two vehicles. This is done by applying a force to this system which, from the point of view of the reference vehicle, can be regarded as a repelling force if the formation distance is smaller than the reference and attracting otherwise. The results obtained are satisfactory proving stability and feasibility of the proposed strategy and the limitations and drawbacks of each control technique used are described. However, it is observed that the LQR servo controller yields a better balance between performance and robustness. Afterwards, the tensegrity concept is implemented on a leader-light-guided-follower formation topology using the LEGO® Mindstorms NXT platform. The distance keeping evolution between the follower and the leader is assessed positively for both cases, a PID and LQR servo controller, however, this latter with better tracking while the leader is moving.

Table of Contents

1	Introduction and Aims of the Project	1
2	Literature Review	2
3	Control of an Unmanned Vehicle	6
3.1	Heading Angle Tracking Problem	6
3.1.1	Sliding Mode Controller	8
3.1.2	Proportional-Derivative Controller	11
3.1.3	Linear-Quadratic Regulator [22]	13
3.2	Waypoint problem	17
3.3	Chapter Summary	22
4	Formation Control of Multiple Unmanned Vehicles	23
4.1	Definition of the Formation Architecture	23
4.2	The Tensegrity Concept in the Context of Formation Control	24
4.3	Simulation Results	26
4.4	Chapter Summary	34
5	Implementation of a Tensegrity-based Formation Controller	35
5.1	Overview of the Lego Mindstorms NXT Kit and its Development Tools	35
5.2	Definition of the Strategy	37
5.3	Experimental Results	42
5.4	Chapter Summary	47
6	Conclusions and Future Work	48
References		49

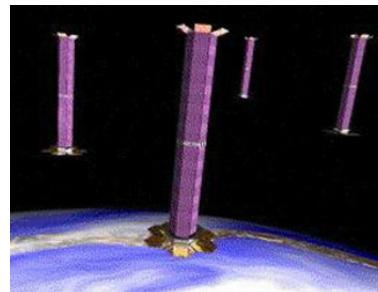
1. Introduction and Aims of the Project

A group of robots cooperatively working following a certain formation to achieve a task can increase the task performance and reduce the execution time in detriment of problem complexity. Current research is being done in order to obtain an efficient controller in which there is a tendency to use a decentralised architecture requiring good error propagation rejection, stable performance regardless of the number of vehicles involved, easy maintenance and controller design, as well as a good obstacle avoidance strategy [1]. The advantages of cooperative work in unmanned vehicles, including ground, aerial, space, marine and underwater robots, are of great interest for nowadays' industry and society. For instance, take the case of a field control and supervision, coastal surveillance and rescue, the exploration of new areas for mapping purposes. In the case of satellite formation flying, multiple satellites can work together to accomplish the same objective of a larger and more expensive one. This brings benefits such as easier design, maintenance, reparation or cheaper replacement, besides, it adds the advantage of multiple viewing angles of the targets at different times, which is a desired feature in the field of satellite communications, meteorology or astronomy, just to name some. Other applications include manipulation cooperation and military missions.

The aim of this project is to design, implement and compare three different formation control techniques based on the tensegrity concept. A conventional PID controller, a sliding mode controller and an LQR servo controller are designed and simulated. The implementation aims to evaluate the overall performance of the system in the case of a PID and LQR servo controller. The LQR and SMC are known for providing the mentioned requirements and most importantly a high disturbance rejection and high insensitivity to parameter uncertainty.



(a) Formation control of military drones [2]



(b) Formation control of satellites [3]



(c) Formation control of a fleet of marine vessels [4]

Figure 1.1: Various applications of formation control

2. Literature Review

It has been shown that by using multiple vehicles to achieve complex tasks the overall performance and time execution have been improved. Furthermore, by using a group of unmanned vehicles it can be assured that the system is more robust against sudden faults from each vehicle involved in a certain task. These faults can be due to sensor malfunctions, communication failures and unexpected abrupt disturbances in the environment. Some of the situations in which cooperative work is convenient are those in which there are force, time, space or dynamics constraints. However, the main motivation to develop intelligent formation control algorithms is to avoid the direct human intervention in certain tasks which may be dangerous, too complicated or tedious, i.e. rescue operations, terrain and maritime exploration and reconnaissance, cooperative work implying huge amounts of force and space missions. Most times cooperative work of multiple unmanned vehicles is more suitable than developing a single robot dedicated to a particular task as it increases expenses. There are several research lines or motivations within the field of cooperative work. For instance, all the different approaches to the formation which aim to achieve a topology with less time consumption and less energy expenditure but not forgetting robustness and complexity. On the other hand, there is the control and models which describe the systems nature and behaviour and the formation control methodologies which is devoted to the way in which the vehicles are supposed to achieve a certain desired shape or task.

In a cooperative mission, there should be a communication flow to and or from each vehicle, informing at least about its relative or absolute position. If a centralised architecture is used then there is a central unit or designated vehicle which guides the rest of the group members. This approach reduces the problem complexity to a simple tracking problem in which the distance error to the leader must be minimised. Conversely, if the architecture is decentralised, all the vehicles can communicate and share information with one another. This approach permits the whole system to continue to work when a single vehicle ceases to function but instead adds more complexity to the control law and the communication flow is more complex and dense. Considering the above advantages and disadvantages of a each architecture one can conclude that a globally decentralised architecture may be a convenient solution. In this approach, each vehicles does not receive or sends information to all the group members but only to its closest neighbours, therefore, the possible constraint arisen from the communication load is relaxed.

Considering the research on formation control done as of yet, the main strategies this

problem has been approached are four:

- 1. Behaviour-based approach: It is a decentralised strategy inspired by the natural animal behaviour where each vehicle behaves depending on the control actions of its neighbours. Clear examples are flocks or herds. It consists in defining motion primitives or simple tasks which a robot must achieve, e.g. avoid obstacles, distance-keeping, simple movements or algorithms to search for a goal. These motion primitives are then used as terms of a weighted law which establishes the final action to solve more complex patterns such as motor control in a group of robots within a hazardous environment. Nonetheless, the main drawback of this approach is its complexity to model and, therefore, ensure stability.
- 2. Leader-following approach: This method requires the existence of a leader robot which is either teleoperated, autonomously moving or carrying out a planned motion. This approach reduces the problem complexity as it transforms a formation control problem into a simpler tracking problem, where the leader's mission is to achieve a certain group task and the followers, provided the leader's position, will simply move to a new position in space which is offset with respect to the leader. This approach makes the use standard control techniques possible because stability is guaranteed. However, because of the existence of a single point of failure the system will fail as soon as the leader falls or is unable to achieve its task. For example, in [7], a vision-based localisation for a leader-follower formation control is designed, simulated and implemented. The work encompasses unicycle Scarab robots which are equipped with a panoramic camera that provides the view angle to the other vehicles. In this case, the state of the leader-follower formation is estimated using a Kalman filter and is then controlled by an input-state feedback controller.
- 3. Virtual structure approach: In this strategy, a group of vehicles is considered as a unique rigid body, therefore, an intrinsically centralised architecture. Each vehicle has its own relative position within the body and tracks the desired trajectory which is translated from the desired motion of the rigid body. Its great advantage is the simplicity it offers when it comes to the formation coordination, however, due to its centralisation the single point of failure is a critical issue for the implementation of such strategies.
- 4. Artificial potential field approach: This method is based on virtual forces which regulate the formation of multiple neighbouring vehicles. For instance in [6] a direct potential field based approach to multi robot formation navigation is presented. This work assumes a context where the formation is not strict and the robots can split in the presence of obstacles. Hence, the virtual forces which obstacles, goal targets

and other vehicles within the formation can exert onto other vehicles can influence each robot's path by changing its steering with repelling or attraction forces.

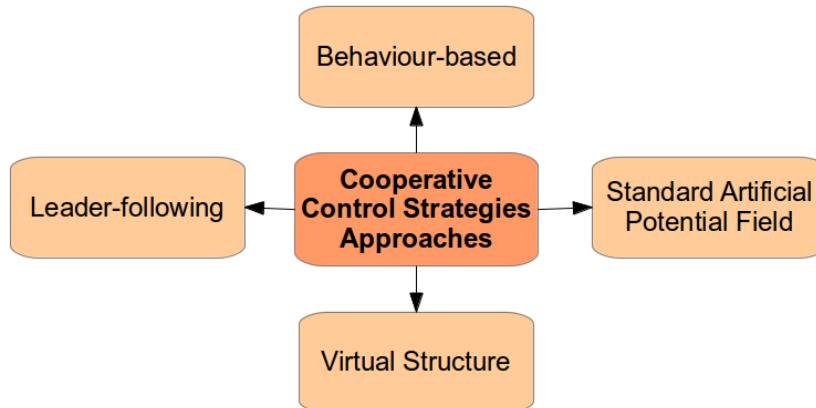


Figure 2.1: Main approaches to cooperative work of multiple robots

It should be noticed that *Cooperative Control* solely implies the work done by two or more vehicles in order to achieve a task whereas *Formation Control* implies a cooperative work in which the vehicles must maintain a constant distance with respect one another, therefore, forming a geometrical shape.

In [5] a decentralised architecture was used to design a behaviour-based formation controller in which three different control strategies were implemented and tested. As the type of vehicle which is desired to be controlled is a differentially driven mobile robot, some problems due to the nonholonomic constraints arise, thus, making it impossible to stabilise the system with continuous static state feedback. These constrains the robot to turn only any angle which is in its possible range. The problem is stated as to drive the total error to zero asymptotically. Such error is described as the error due to the total error between the current position of the robots and the desired formation pattern and the formation error itself. The first approach is the *coupled dynamics formation control* which basically presents a bidirectional topology where the relative position and velocity information of two neighbours as well as the desired positions in the target path are known by each vehicle of the group. The second approach does not need to take the relative velocity into account and is referred to as *coupled dynamics formation control with passivity-based interrobot damping*, where the relative velocities of the neighbours are estimated rather than measured, thus, reducing the communication load.

Different controllers have been implemented to deal with the robot plants, for instance, a LMI-based H_∞ controller was presented in [8], distributed model-predictive control [9], sliding mode control [10][11]. [12] presented a sliding mode control approach incorporating a neural network to modify the parameters of exponent reaching law of sliding mode

control in order to get an optimal balance between convergence speed of the sliding quantity and fuel consumption. This idea opens up many other possibilities in order to conceive auto-tuning controllers, as presented in [14] where a Q-learning fuzzy controller was used achieving a reduction of the steady state error and eliminating the complexity of manual tuning.

The concept of tensegrity has been applied to many application in our recent days, especially in the architecture and civil and mechanical engineering. However, it has caught the attention of formation control researchers. For instance, in [23] the author presents a nonlinear tensegrity-based formation methodology modelled by a virtual tendon-driven system which aims to regulate the shape of formation of multiple vehicles in the presence of disturbances.

[15] presents a bio-inspired formation of Micro Aerial Vehicles (MAVs) based on migrating birds for the application for distributing supplies to a disaster area. As opposed to other topologies, this work does not use shape vectors as the shape is freeform as in nature. It considers three different stages for the flock formation: shape keeping, shape entrance and leader change control. A virtual spring-mass-damper system which connects two MAVs together with a centripetal force consideration. It claims that a V-shape formation, as the ones performed by flocks, structure is able to compensate for wind flow and a less energy expenditure.

In [16] a hybrid topology between a decentralised and centralised strategy is proposed to control the formation of a group of several single aircrafts. The entire formation is considered as a unique virtual structure and at the same time it is subdivided in sub virtual structures which are composed of aircrafts. The architecture used within the virtual structures is a decentralised one while inside each sub virtual structure all the aircrafts are controlled by a centralised architecture. This hybrid topology is called multi-layer virtual structure and an LQR is responsible for the formation of each virtual structure or aircraft.

3. Control of an Unmanned Vehicle

This chapter is devoted to the design of an autopilot which is the first step towards the formation control of multiple vehicles. For simulation purposes a vessel is used as the unmanned vehicle (UV) in question. There are three translation and other three rotation independent movements which a ship can make on the sea, resulting in a total of six degrees of freedom. Namely, the first are known as surge, sway and heave for movements along the x , y and z axes, respectively. The rotation motions are roll, pitch and yaw around the x , y and z axes, respectively. These motions are depicted in Figure 3.1. To represent the complete manoeuvring dynamics of a ship it is required the use of tedious coupled nonlinear equations which may lead to unnecessary complex control solutions. Hence the use of simple models is preferred when it comes to autopilot design, therefore, the first and second order Nomoto models are utilised in this project. These relate the input rudder angle with the output heading angle of an unmanned vehicle. This control problem is approached in three different ways: a conventional PD controller, a sliding mode controller and an LQR-based controller. However, it is more intuitive and common to solve the waypoint problem in which a planar x and y desired coordinates are the objective which can be easily related to the heading angle tracking.

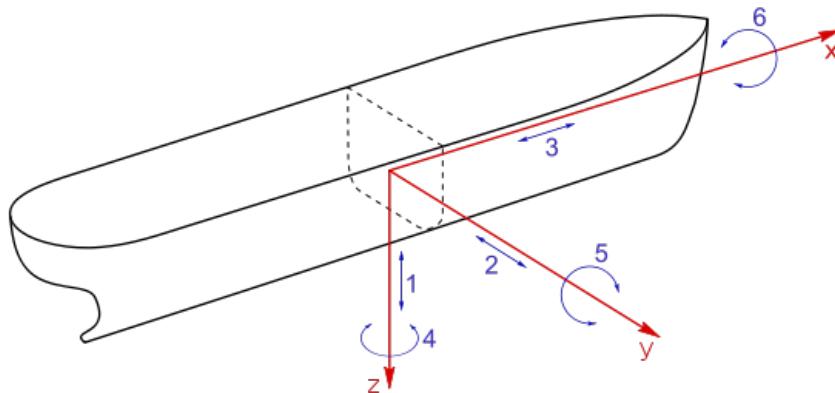


Figure 3.1: Ship response in waves: 1. Heave, 2. Sway, 3. Surge, 4. Yaw, 5. Pitch, 6. Roll. Extracted from [13]

3.1. Heading Angle Tracking Problem

The models of Nomoto [19] are the result of the elimination of the sway velocity from the model of Davidson and Schiff presented in [20]. These models relate the output heading angle to the input rudder angle of an unmanned vehicle. It is generally used for modelling the manoeuvring dynamics.

Equation (3.1) is the Nomoto second order model representation in the Laplace domain:

$$\frac{r(s)}{\delta(s)} = \frac{K(1 + T_3 s)}{(1 + T_1 s)(1 + T_2 s)} \quad (3.1)$$

where δ is the rudder angle, and r the yawing rate.

In time-domain:

$$T_1 T_2 \ddot{\psi} + (T_1 + T_2) \dot{\psi} + \psi = K(\delta + T_3 \dot{\delta}) \quad (3.2)$$

where $\dot{\psi} = r$. ψ is known as the heading angle. Having said this, equation 3.1 can also be written as:

$$\frac{\psi(s)}{\delta(s)} = \frac{K(1 + T_3 s)}{s(1 + T_1 s)(1 + T_2 s)} \quad (3.3)$$

The constant K is the static yaw rate gain and T_1, T_2, T_3 are time constants.

For control purposes, it is desired to express this model in a state-space representation, therefore, for convenience equation 3.2 can be expressed as:

$$\ddot{\psi} + a_1 \dot{\psi} + a_2 \psi = b_0 \dot{\delta} + b_1 \delta \quad (3.4)$$

Where:

$$a_1 = \frac{T_1 + T_2}{T_1 T_2}, \quad a_2 = \frac{1}{T_1 T_2}, \quad b_1 = \frac{K}{T_1 T_2}, \quad b_0 = \frac{T_3 K}{T_1 T_2}$$

Now, let x_1, x_2, x_3 be:

$$\begin{aligned} \dot{x}_1 &= \dot{\psi} \\ \dot{x}_2 &= \ddot{\psi} \\ \dot{x}_3 &= \ddot{\psi} - b_0 \delta \end{aligned} \quad (3.5)$$

Combining equation 3.4 and 3.5:

$$\begin{aligned} \dot{x}_1 &= \dot{\psi} \\ \dot{x}_2 &= \ddot{\psi} \\ \dot{x}_3 &= -(a_1 \ddot{\psi} + a_2 \dot{\psi}) + b_1 \delta \end{aligned} \quad (3.6)$$

Recall that $\ddot{\psi} = x_2$ and $\dot{x}_3 = \ddot{\psi} - b_0 \delta$, then $\ddot{\psi} = \dot{x}_3 + b_0 \delta$, therefore, 3.6 can be rewritten as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 + b_0 \delta \\ \dot{x}_3 &= -a_2 x_2 - a_1 x_3 + (b_1 - a_1 b_0) \delta \end{aligned} \quad (3.7)$$

Finally, the state-space representation in the form:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

of the system 3.7 is presented in 3.8.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ b_0 \\ b_1 - a_1 b_0 \end{bmatrix} \delta \quad (3.8)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \delta$$

The work presented in [17] considers a Mariner class vessel with the following constants $K = 0.185 \text{ 1/s}$ and $T_1=118 \text{ s}$, $T_2=7.8 \text{ s}$, $T_3=18.5 \text{ s}$ which was first presented in [18]. For simulations purposes said parameters are used in this project.

However, a first order approximation of the previous plant is obtained by letting the effective time constant be equal to: $T = T_1 + T_2 - T_3$, therefore, the equation which governs the heading angle dynamics is:

$$T\ddot{\psi} + \dot{\psi} = K\delta \quad (3.9)$$

Its transfer function is:

$$\frac{\psi(s)}{\delta(s)} = \frac{K}{s(1+Ts)} \quad (3.10)$$

The state-space representation of the first order Nomoto model is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-1}{T} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix} \delta \quad (3.11)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \delta$$

where $x_1 = \psi$ and $x_2 = \dot{\psi}$. The first order Nomoto model has become popular due to its simplicity in course-keeping control with relatively low modelling error.

3.1.1. Sliding Mode Controller

In the recent years the research on more robust controllers which compensate for unmodeled dynamics, parameter uncertainty and external disturbances has gained more interest. The sliding mode control (SMC) is a robust technique which aims to solve these problems. A sliding variable, which defines the so called surface, is introduced to the system and the control objective is to drive this variable to zero asymptotically [21].

Let us design a sliding mode controller for the first order Nomoto model. Manipulating

equation 3.9 the system can be rewritten as:

$$\begin{aligned}\frac{dz_1}{dt} &= z_2 \\ \frac{dz_2}{dt} &= \frac{1}{T}(K\delta - z_2)\end{aligned}\quad (3.12)$$

The equations in (3.12) can be represented in the form of (3.13):

$$\dot{z} = f + gu \quad (3.13)$$

where

$$f = \begin{bmatrix} z_2 \\ -\frac{z_2}{T} \end{bmatrix} \quad g = \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix}$$

Consider the sliding surface in (3.14):

$$S(t) = \lambda(z_1^* - z_1) + \beta \frac{d}{dt}(z_1^* - z_1) = \lambda(z_1^* - z_1) + \beta z_2 \quad (3.14)$$

In order to check if a sliding mode controller can be applied to the system using the surface defined in (3.14), the transversality condition must hold: $\Delta S g \neq 0$. As seen in (3.15), such condition is fulfilled for any value of $\beta, K, T \neq 0$.

$$\Delta S g = \begin{bmatrix} -\lambda & -\beta \end{bmatrix} \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix} = -\beta \frac{K}{T} \quad (3.15)$$

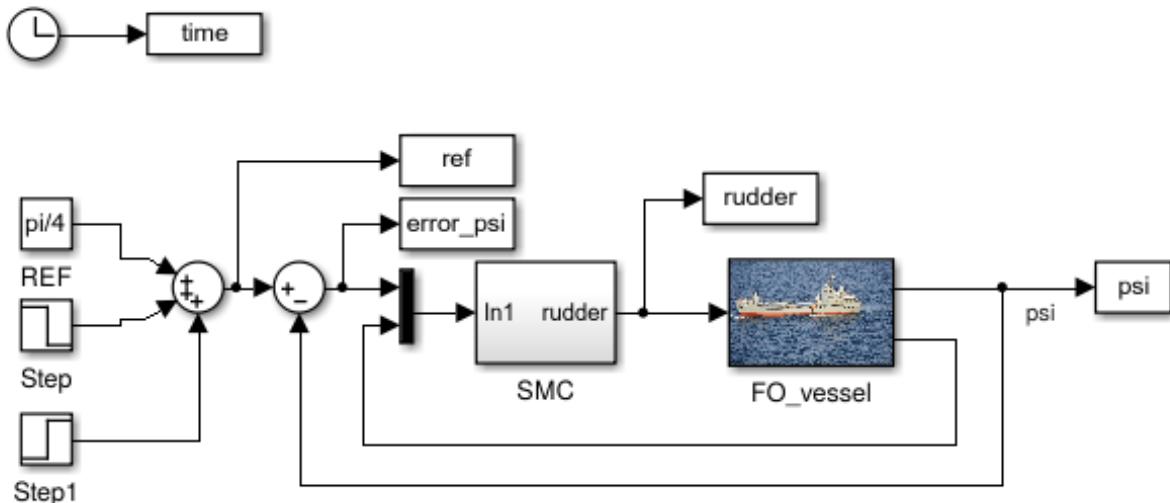


Figure 3.2: *Simulink* model of the ship controlled by a SMC

Figure 3.2 illustrates the plant and the sliding mode controller which has been used for simulation purposes. It was considered that the rudder angle, which is the control signal,

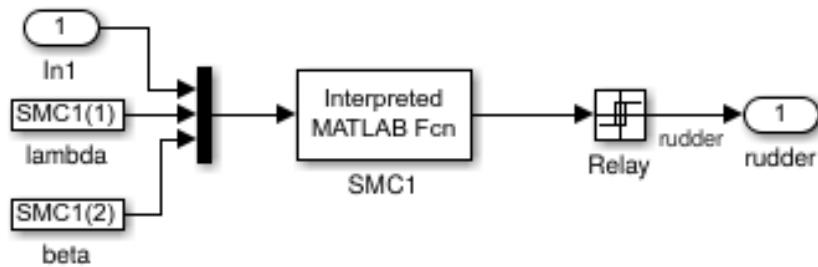


Figure 3.3: *Simulink* model of the SMC

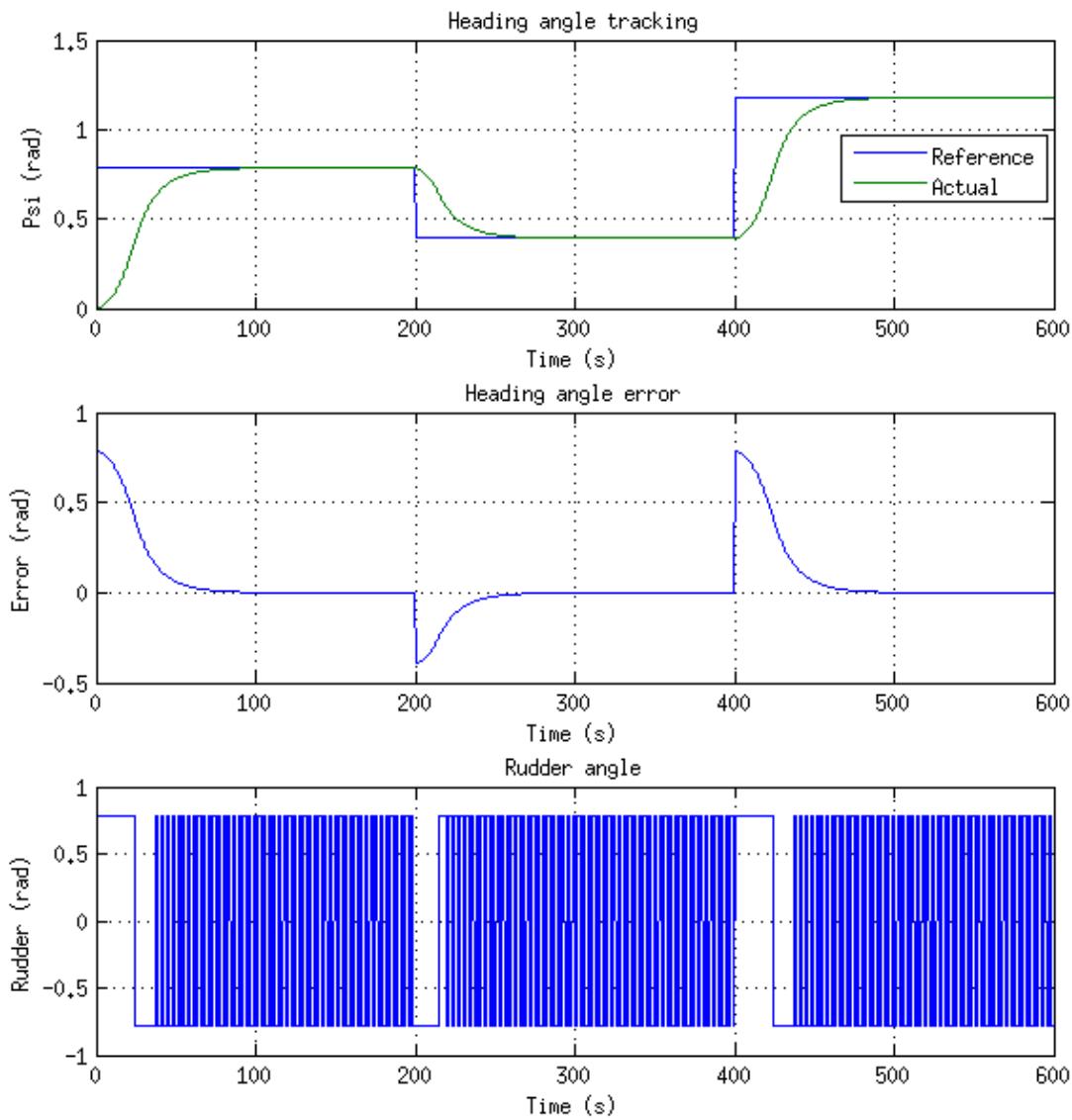


Figure 3.4: Results yielded by a SMC

is constrained between $\pm 45^\circ$ which is more realistic. This constraint can be easily applied in the sliding mode block as these two limits are the relay upper and lower limits. The structure of the SMC is shown in Figure 3.3, where the MATLAB function is the sliding

surface from Equation 3.14.

Basically, the result of the surface operation inputs a relay or hysteresis comparator whose transition limits were empirically chosen to be ± 0.1 . The smaller these transition limits are, the greater the switching frequency is. This may become problematic as the program complexity grows and with it the simulation time. Moreover, if these limits are relaxed, the undesirable *chattering problem* becomes more noticeable. An ideal SMC would perform infinite switching frequency, however, the discrete-nature of the computer simulation makes it impossible. Such limitation of the sliding mode controller has been and continues to be subject of research. Furthermore, in some practical applications, such as the control of aerial and marine vehicles, the chattering problem should be avoided because of the natural dynamics of such systems impede that a high-frequency control signal could be used. An intuitive example can be a ship or an aircraft moving back and forth at a high frequency. For this reason, the scope of this project does not encompass the elimination of the chattering problem as for simulation purposes it is irrelevant.

The results displayed in Figure 3.4 show that the system achieves all the setpoints and remains stable once in steady state. Note that the coefficients λ and β , which are the design parameters of the SMC, are set to $\lambda = 7$ and $\beta = 100$. It should be pointed that these values were obtained empirically in order to find a trade-off between the settling time and the overshoot of the heading angle time response. The settling time is about 70 seconds and no overshoot is present. Notice how the rudder angle, or control signal, switches between $\pm 45^\circ$. In practice this fast moving control signal would be implausible, instead a smooth rudder should be used.

3.1.2. Proportional-Derivative Controller

The plant shown in Equation 3.10 is a type 1 system, therefore, a PD controller is able to compensate for the tracking error. The PID structure utilised by the *Simulink* block is of the form:

$$K(s) = P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \quad (3.16)$$

This software provides automatic tuning which comes integrated within the Control Toolbox. The user is presented with a tuning window, as depicted in Figure 3.5, where a trade-off between performance and robustness is made. Given the plant and that the control signal is constrained, the proportional, integral and derivative gains are set to 1.1, 0 and 35, respectively. The filter coefficient N is set to 6.9 and determines the location of the pole in the derivative filter.

Figure 3.6 and 3.7 show the *Simulink* model and results obtained by the PD controller, respectively. Notice that the settling time in this case is approximately 100 seconds and the overshoot is less than 2%, almost unnoticeable. It is clear that the control signal is smoother than the one obtained by the SMC.

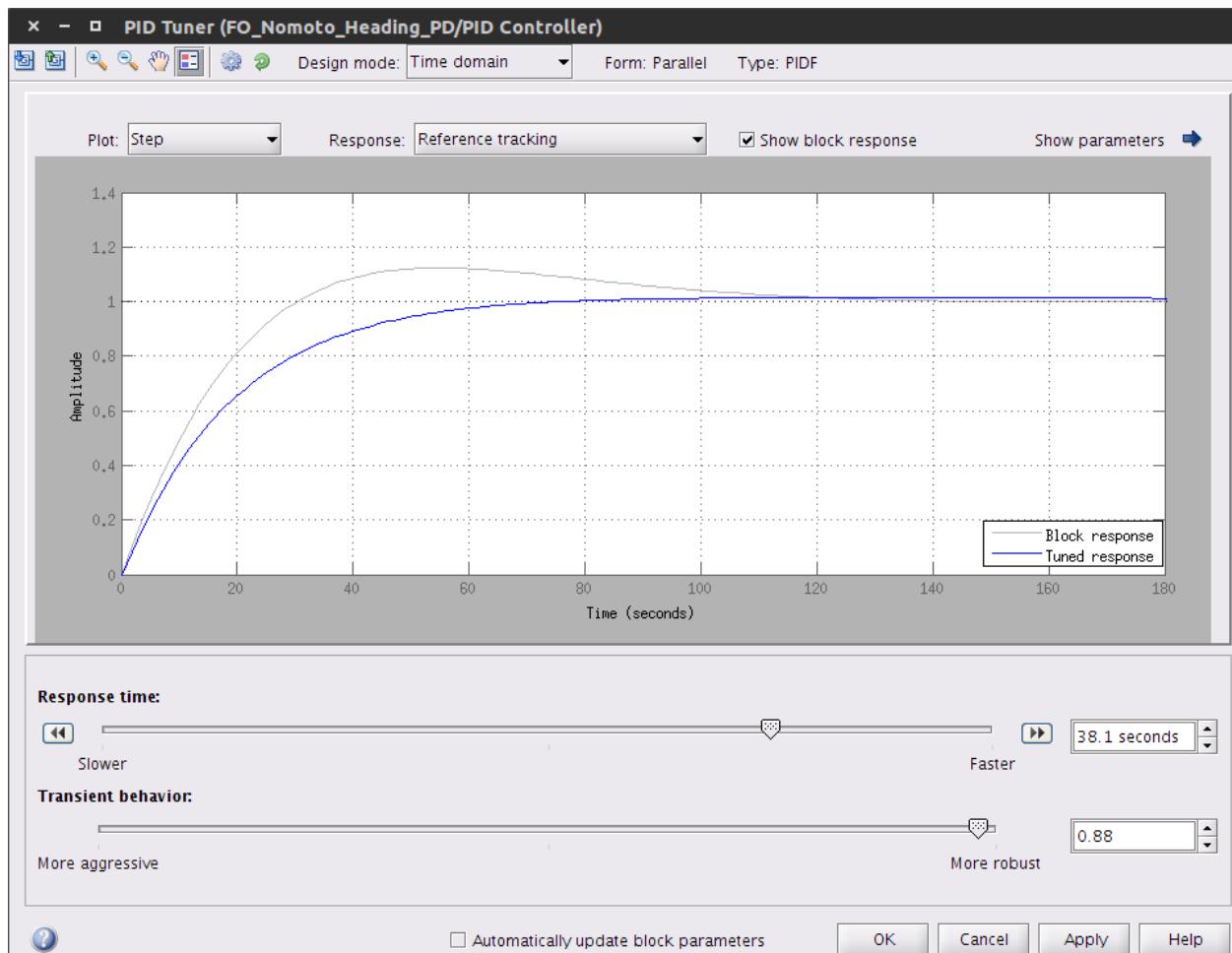


Figure 3.5: PID automatic tuning window

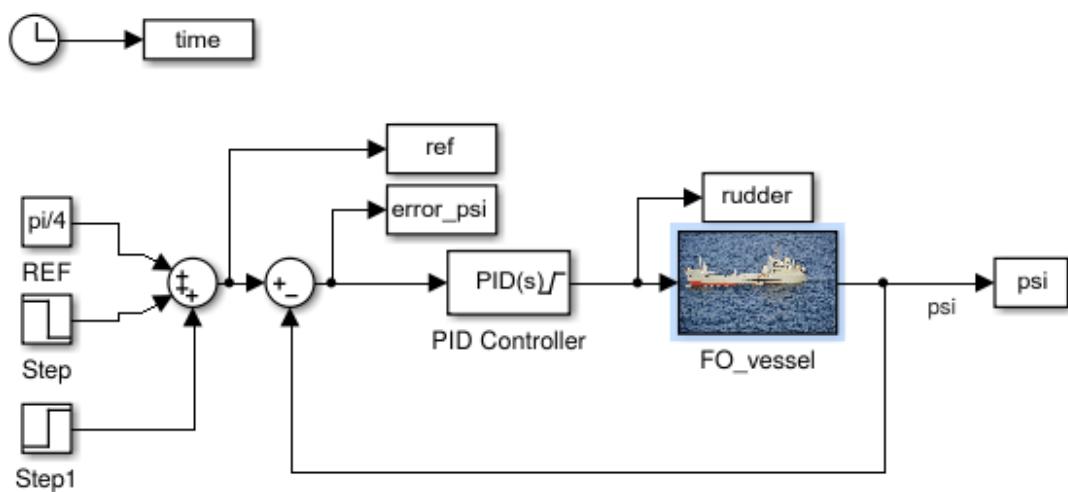


Figure 3.6: Simulink model of the ship controlled by a PID controller

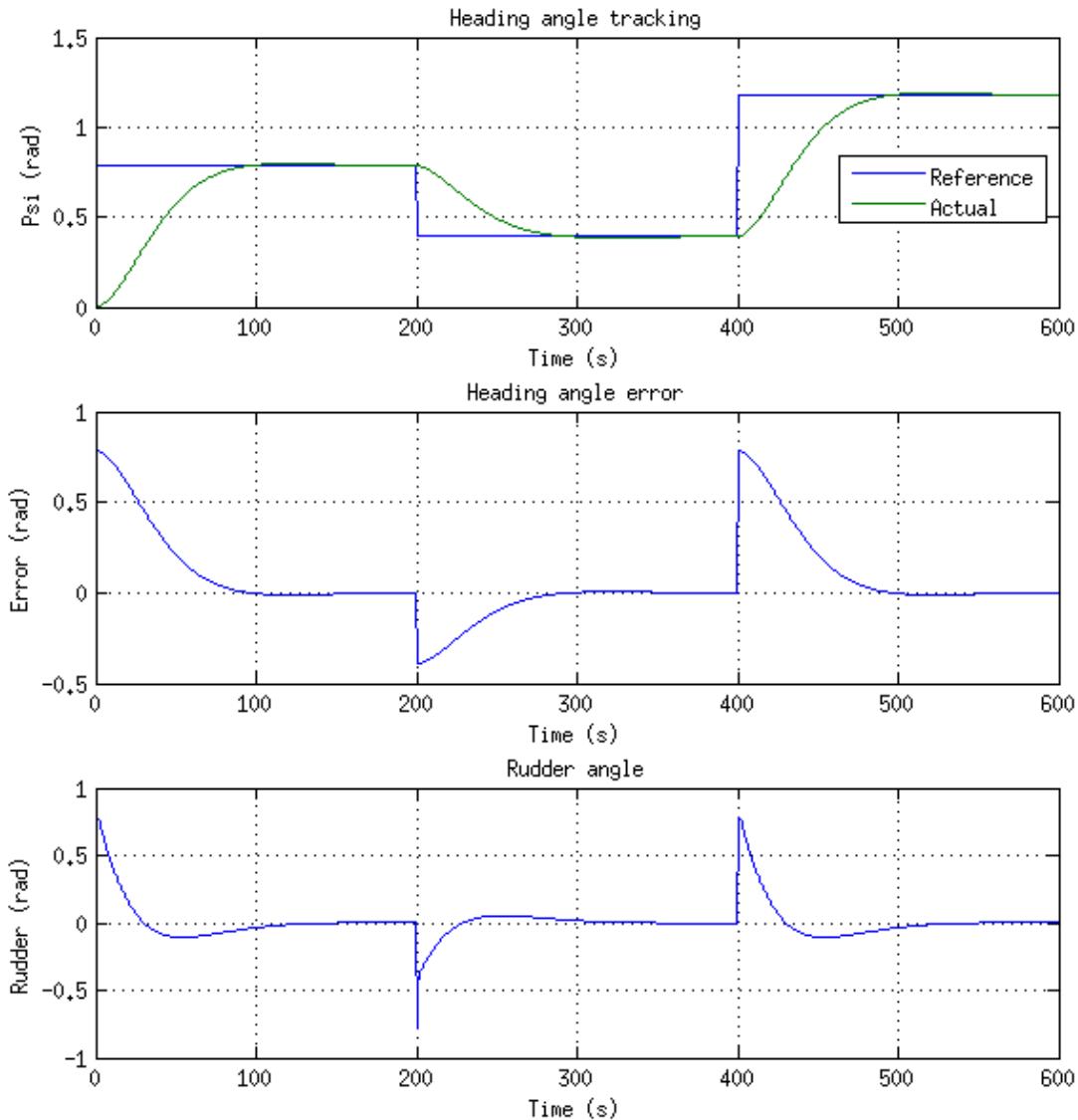


Figure 3.7: Results yielded by a PD controller

3.1.3. Linear-Quadratic Regulator [22]

Given the state-space representation of any system in the form of:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

an optimal regulator is to be designed where a vector K is determined so that the control signal:

$$u = -Kx(t) \quad (3.17)$$

minimises the following cost function:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (3.18)$$

where Q is a positive-definite (or positive-semidefinite) Hermitian or real symmetric matrix and R is a positive-definite Hermitian or real symmetric matrix. Note that the second term on the right-hand side of Equation 3.18 accounts for the expenditure of the energy of the control signals. The matrices Q and R determine the relative importance of the error and the expenditure of this energy. These aspects make the LQR a suitable control technique for applications where the energy of the control signals is critical or restricted, such as the case of an aerial or marine vehicle. K is given by:

$$K = R^{-1} B^T P \quad (3.19)$$

and P is found by solving the continuous time algebraic Riccati equation:

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (3.20)$$

As stated in [22], to ensure the existence of the optimal vector $u(t)$, the system must be completely state controllable. Due to the fact that the cost function, sometimes called performance index, is a linear function of the state variables, it is necessary to feed back all state variables. In reality, this may not be realisable, therefore, a state estimator is implemented. For this project it is assumed that all state variables are observed. Let us check that the system 3.11 is state controllable:

$$\left[\begin{array}{c|cc} B & AB \end{array} \right] = \left[\begin{array}{cc} 0 & \frac{K}{T} \\ \frac{K}{T} & \frac{-K}{T^2} \end{array} \right] \quad (3.21)$$

As the determinant of matrix 3.21 is different from zero, it is said to be nonsingular, thus, the system is completely state controllable.

MATLAB provides the function *lqr* to calculate the optimal gain K given the matrices A , B , Q and R : $K = lqr(A, B, Q, R)$. These weights are design parameters and one of the advantages of using this control technique is that it is easy to tune empirically.

Nonetheless, a regulator aims to drive the system states to zero or a fixed reference which does not change over time. When the desired setpoint is time varying, said system becomes a control system. In such case, the following control action is employed:

$$u = -Kx + k_I \dot{\xi} \quad (3.22)$$

where:

$$\dot{\xi} = r - y$$

It is easy to modify the LQR problem by augmenting the plant to include this error as a new state:

$$\begin{bmatrix} \dot{\xi} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & -C \\ 0 & A \end{bmatrix} \begin{bmatrix} \xi \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} \delta \quad (3.23)$$

The new augmented matrices A and B are then used to obtain the gain K making use of the MATLAB command mentioned before. The first element of this array of gains corresponds to the heading angle error state variable. Figure 3.8 illustrates the control scheme used once the array of gains K is found.

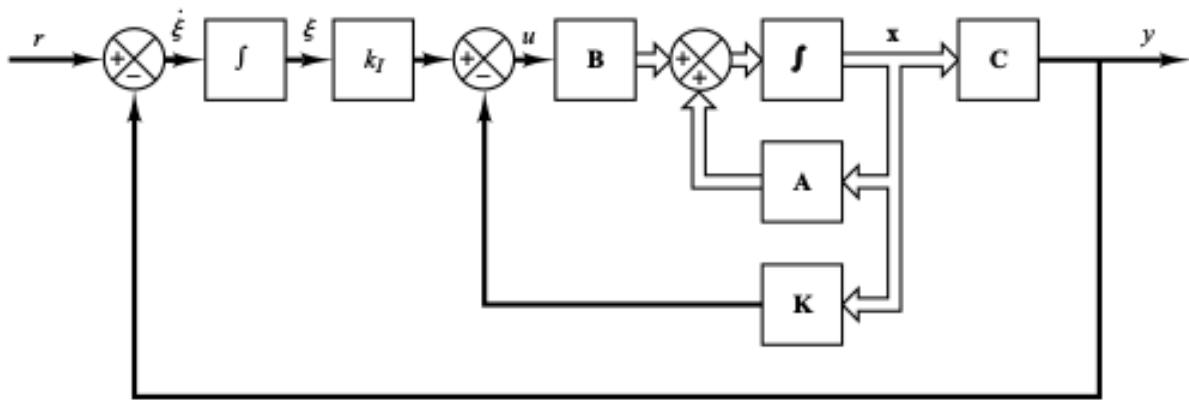


Figure 3.8: Servo control based on an LQR [22]

Figure 3.9 shows the Simulink model of the LQR servo system to control the heading angle of the first order Nomoto model. The results yielded by the LQR are depicted in Figure 3.10. Note that the LQR problem definition assumed that the control signal u was not constrained, unlike this system, therefore, finding suitable weights Q and R is a more laborious task. Q and R are set to:

$$Q = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad R = 100$$

Recall that the first state variable corresponds to the heading angle error, the second to the heading angle, and, finally, the heading rate. The high value of R is due to the control signal constraint as great expenditure of energy is penalised.

The settling time obtained with an LQR servo controller for the first step input is about 90 seconds which is about the same as the SMC and PD controllers, also notice there is no overshoot whatsoever. However, note that in the case of an LQR, the input rudder angle is considerably smoother which is a desirable behaviour. In general terms, the LQR-based controller has a better performance than a PID and compared to the SMC, the chattering problem is not an issue.

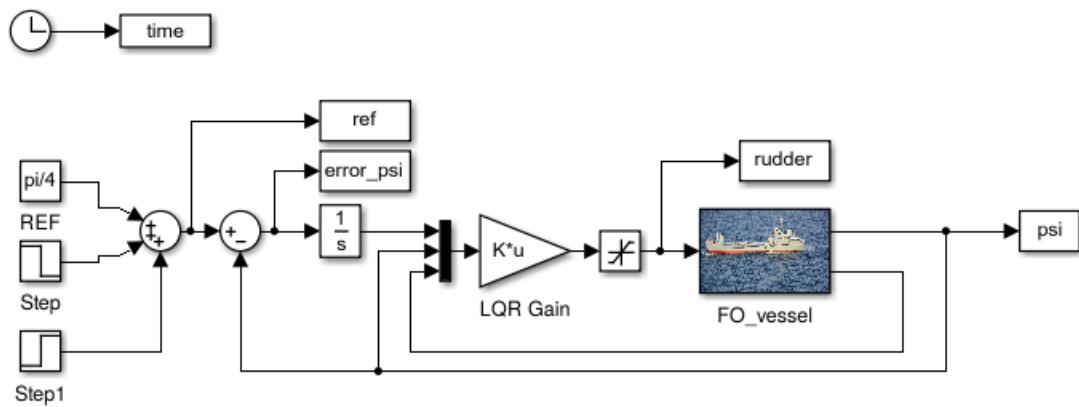


Figure 3.9: Simulink model of the ship controlled by an LQR servo controller

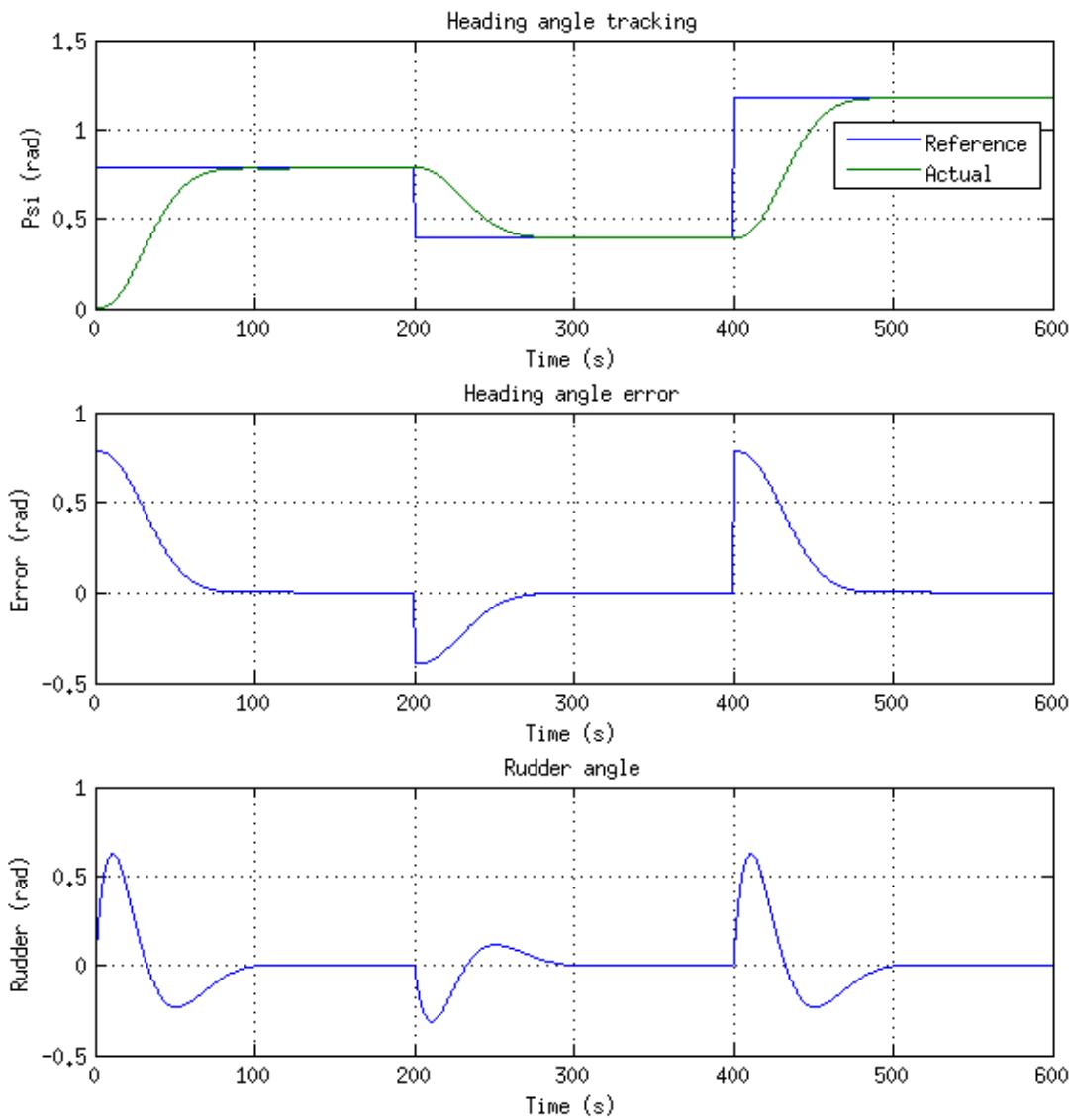


Figure 3.10: Results yielded by an LQR servo controller

3.2. Waypoint problem

In vehicle guidance it is more common to address a waypoint problem. In such problem the desired x and y coordinates are given to the controller and the vehicle must reach these points, thus, the desired heading angle is deduced in function of the desired waypoints. The model of the ship shown in Figure 3.11 is given by (3.24).

$$\dot{x} = v \cos \psi, \quad \dot{y} = v \sin \psi, \quad \dot{\psi} = r \quad (3.24)$$

where v is the forward velocity and r the yawing angle rate, (x, y) is the centre of the vehicle, ψ is the angle between heading direction and the x -axis. In addition to the kinematic model, it is assumed that the rudder angle is bounded between $\frac{\pi}{3}$ and $-\frac{\pi}{3}$. Due to the nature of the ship, the vehicle is considered as a nonholonomic system which introduces the nonholonomic constraint given by (3.25).

$$\dot{x} \sin \psi - \dot{y} \cos \psi = 0 \quad (3.25)$$

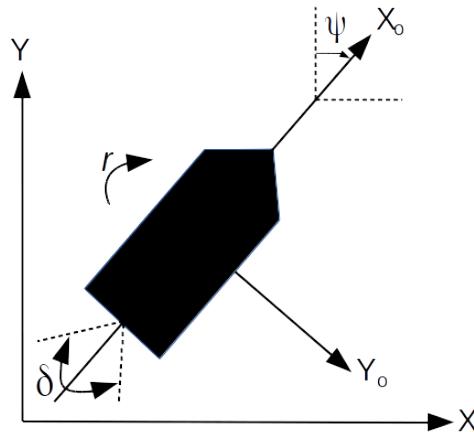


Figure 3.11: Graphical representation of a ship, showing the rudder δ and heading angle ψ

Let (x_0, y_0) and (x_1, y_1) be the current and desired ship coordinates, respectively. From Figure 3.13 it is easy to notice that the heading angle can be calculated as shown by (3.26).

$$\psi = \arctan \left(\frac{y_1 - y_0}{x_1 - x_0} \right) \quad (3.26)$$

The general structure which defines a waypoint tracking is illustrated in Figure 3.12. The actual coordinates are calculated as shown by (3.27).

$$x = \int v \cos \psi dx + x_0, \quad y = \int v \sin \psi dy + y_0 \quad (3.27)$$

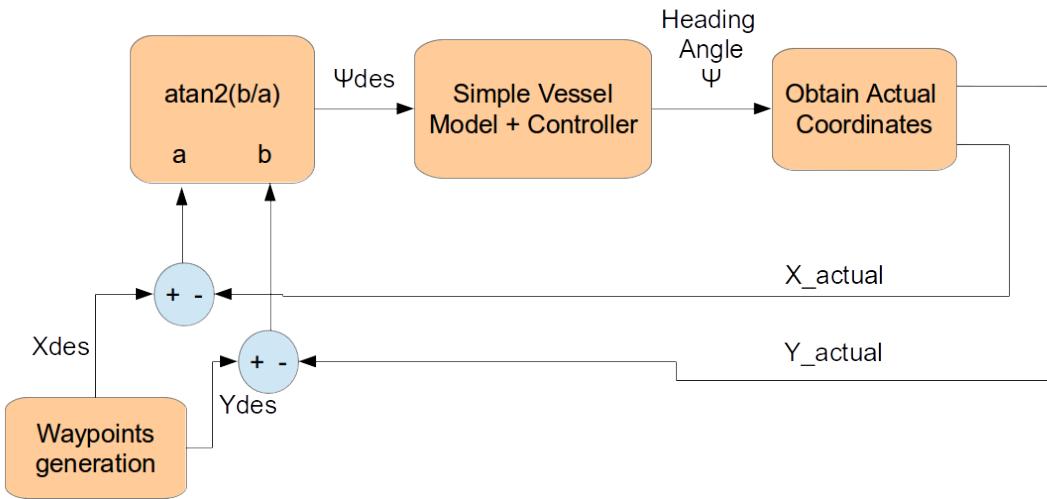
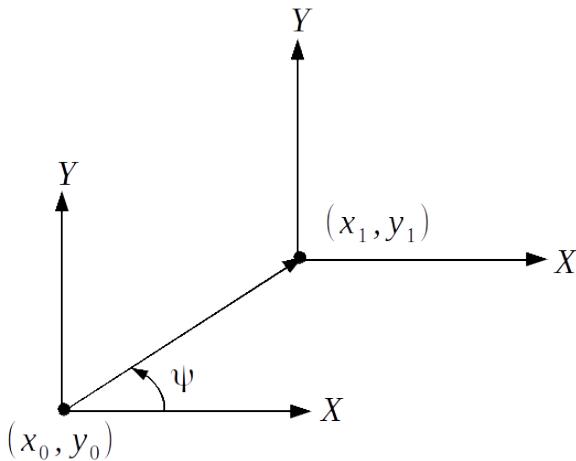


Figure 3.12: Waypoint tracking system block diagram.


 Figure 3.13: Heading angle ψ obtained from current and desired coordinates.

The previous *Simulink* models built for the heading angle control are modified so to transform the heading tracking problem into a waypoint tracking problem. Said modification can be appreciated in Figure 3.14, in the case of a SMC, which is similar for the PD and LQR controllers.

Figures 3.15 to 3.20 show the results obtained. In all cases the waypoints are reached and again the SMC and LQR outperform the PD. Also, notice that in terms of manoeuvring the LQR yields better results. Observe that the waypoints are the same in all cases as well as the initial position, which is (0,0). The speed is set to 1 m/s. A tolerance area of a 5 meter radius is defined around each waypoint, thus, the desired waypoint can be changed automatically whenever the vehicle reaches a point within this area. It is worth mentioning that sharper and more abrupt turns should be avoided as they may lead the system to navigate with oscillatory motions and sometimes instability.

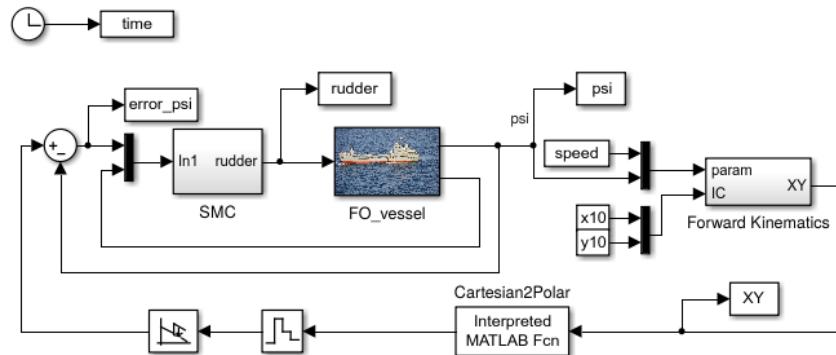


Figure 3.14: *Simulink* model of the sliding mode waypoint controller

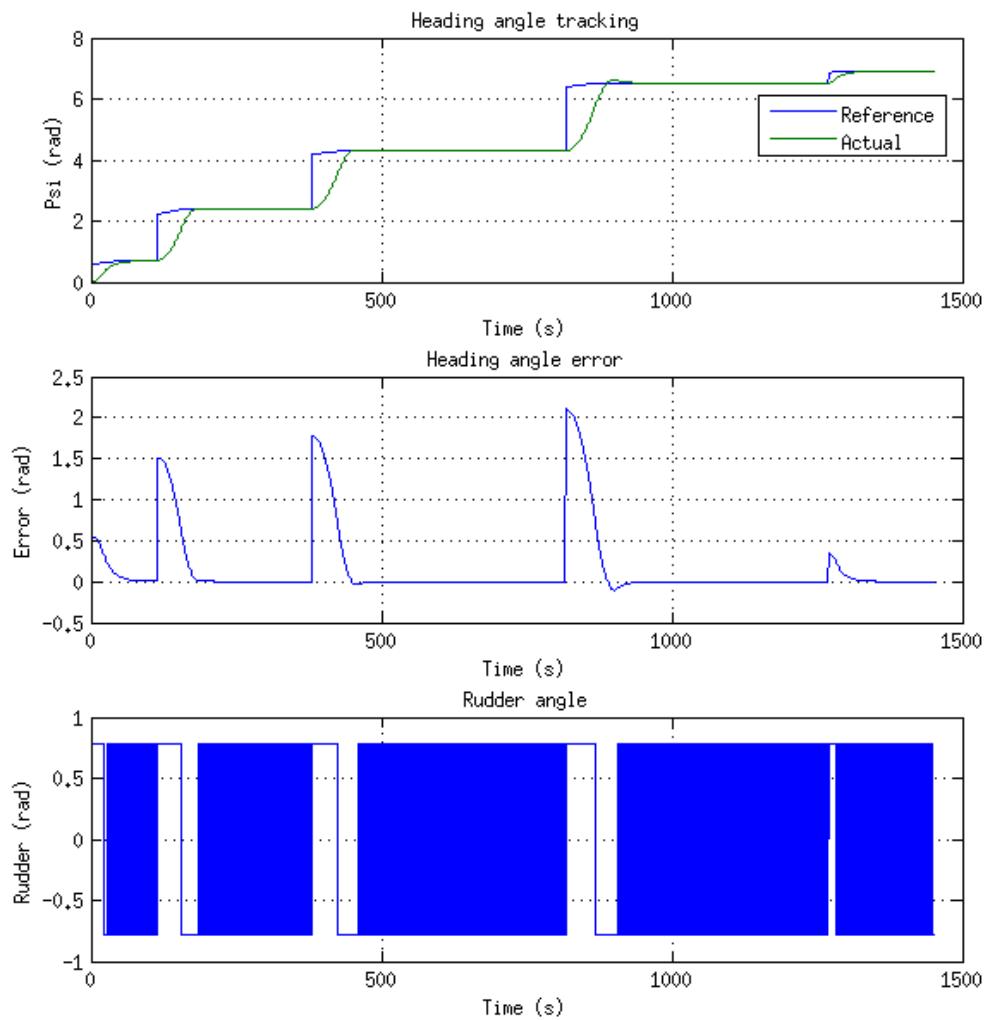


Figure 3.15: Waypoint tracking results yielded by a SMC

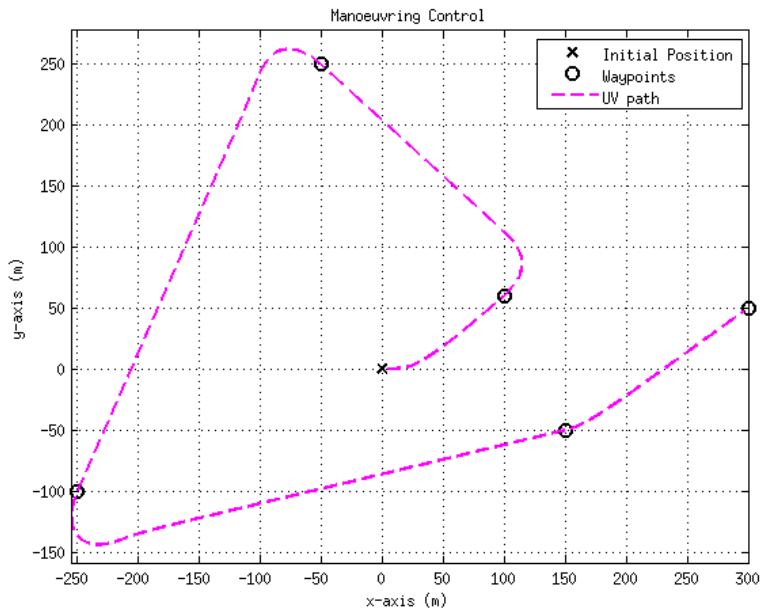


Figure 3.16: Motion trajectory of the SMC-based manoeuvring of a ship

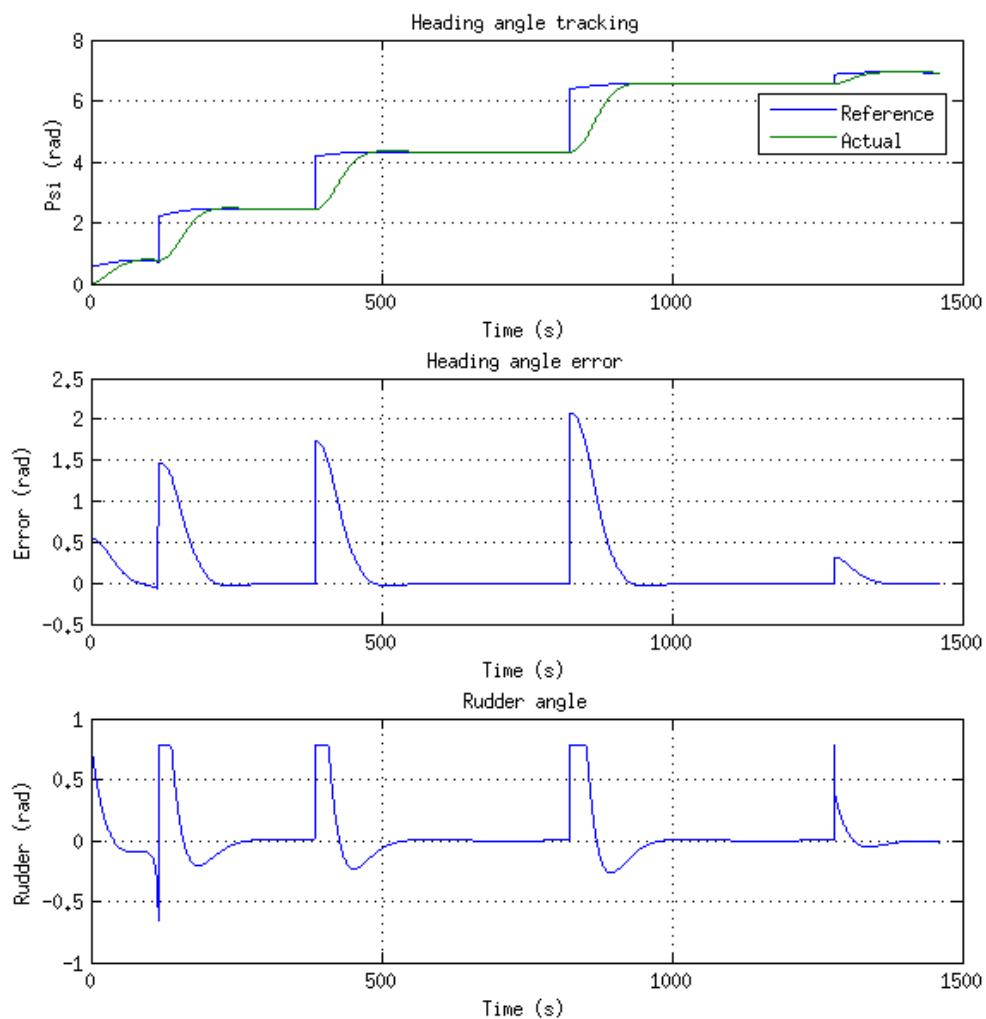


Figure 3.17: Waypoint tracking results yielded by a PD controller

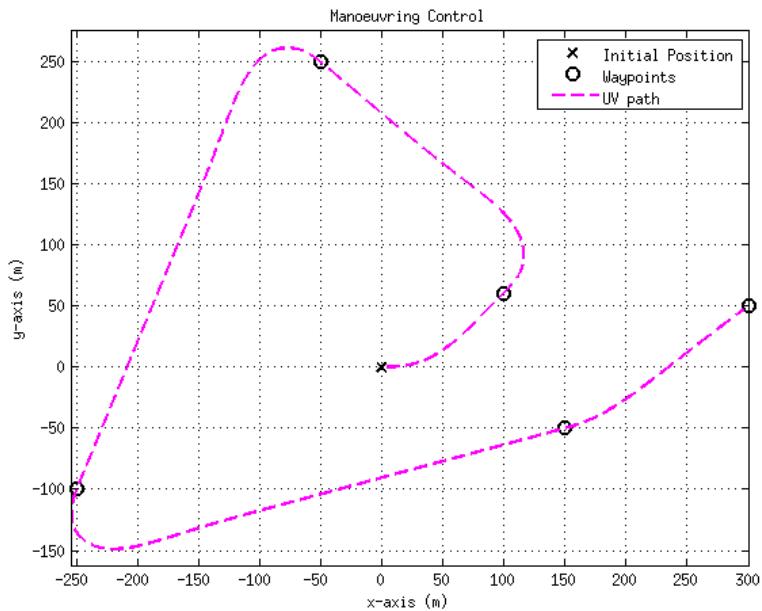


Figure 3.18: Motion trajectory of the PD-based manoeuvring of a ship

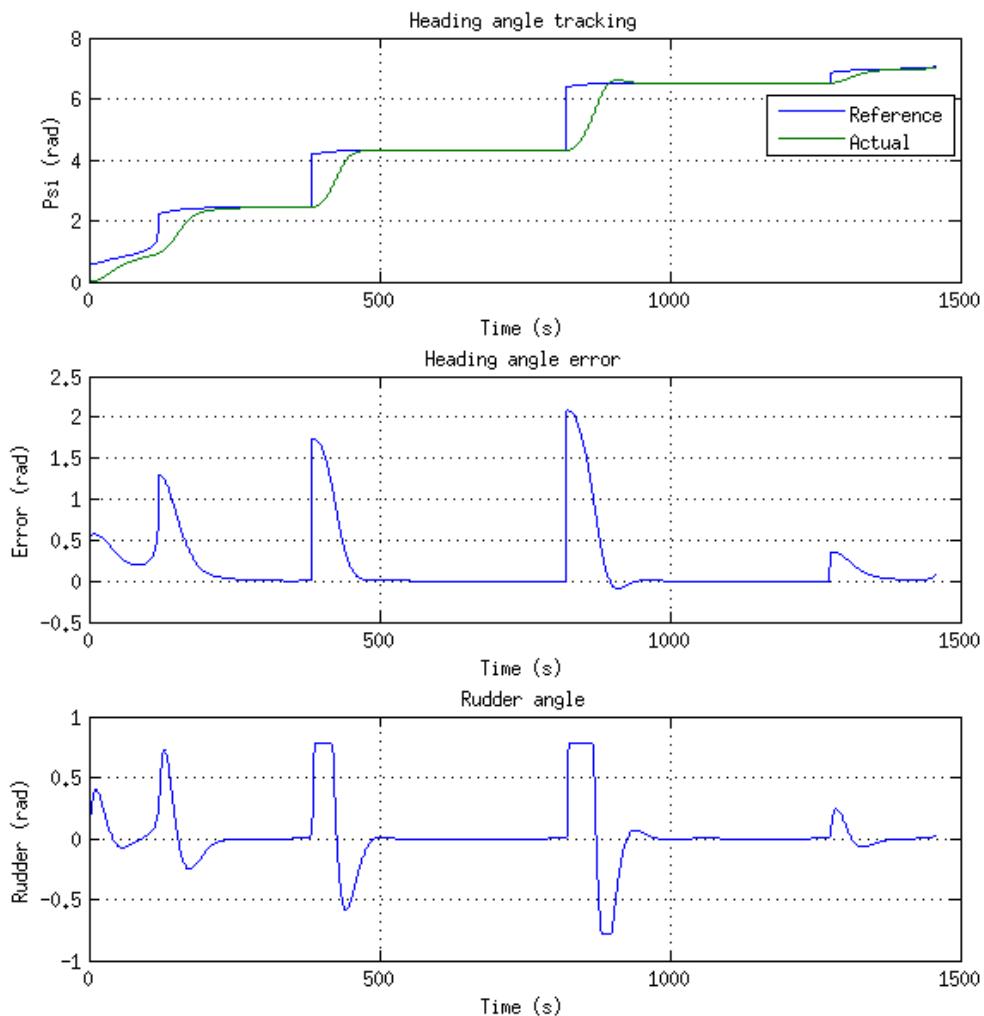


Figure 3.19: Waypoint tracking results yielded by a LQR

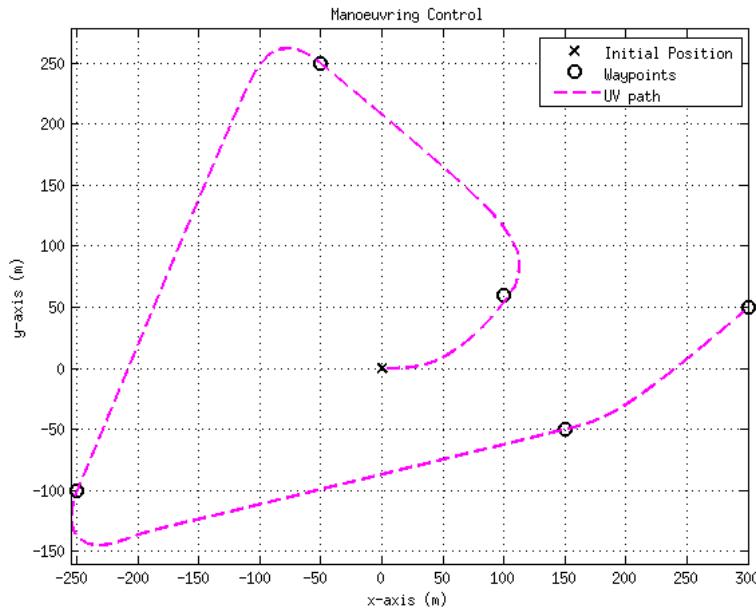


Figure 3.20: Motion trajectory of the LQR-based manoeuvring of a ship

3.3. Chapter Summary

This chapter was devoted to the control of the yaw angle of the first order Nomoto model by means of a sliding mode controller, a proportional-derivative controller and a linear-quadratic regulator. It was observed that despite being more robust the SMC's main drawback is the chattering present in the control signal. Conversely, the LQR yielded a smooth slick rudder angle and yet a fast response, while the PD achieved an acceptable result, driving the system to stability, the settling time and its poor robustness are considered to be its main disadvantages. The second part of the chapter focused on the waypoint problem as it is more common to specify spatial coordinates in vehicle guidance. With a constant speed of 1 m/s the three control approaches achieved all the waypoints as expected.

4. Formation Control of Multiple Unmanned Vehicles

This chapter presents the tensegrity-based formation of multiple unmanned vehicles with a globally decentralised topology. In the previous chapter different techniques to control a single UV were studied and simulated, the next step is to incorporate the other UVs. The kinematic model of a point-mass is considered to validate the concept of tensegrity instead of independent dynamics for each vehicle. In order to generate the heading angle a virtual UV is utilised, considering the second order Nomoto model. The x and y coordinates of this virtual vehicle will be used as the point of reference for the following UV which at the same time will be the reference frame for the next UV and so on and forth. The concept of tensegrity is brought to the formation where through a virtual spring-mass-damper system connected between two consecutive UVs, the speed of the follower vehicle is extracted.

4.1. Definition of the Formation Architecture

In spite of making use of a kinematic model, it is possible to define a globally decentralised formation architecture. One of the advantages of using such strategy discussed in Chapter 2 is that there is no single point of failure because given that one UV fails to function or goes beyond the safe area it can be left behind and the formation structure would be redefined with the remaining robots. The proposed approach, extracted from [23], suggests that an UV will keep a certain distance with a relative angle with respect to one of its neighbours as illustrated in Figure 4.1.

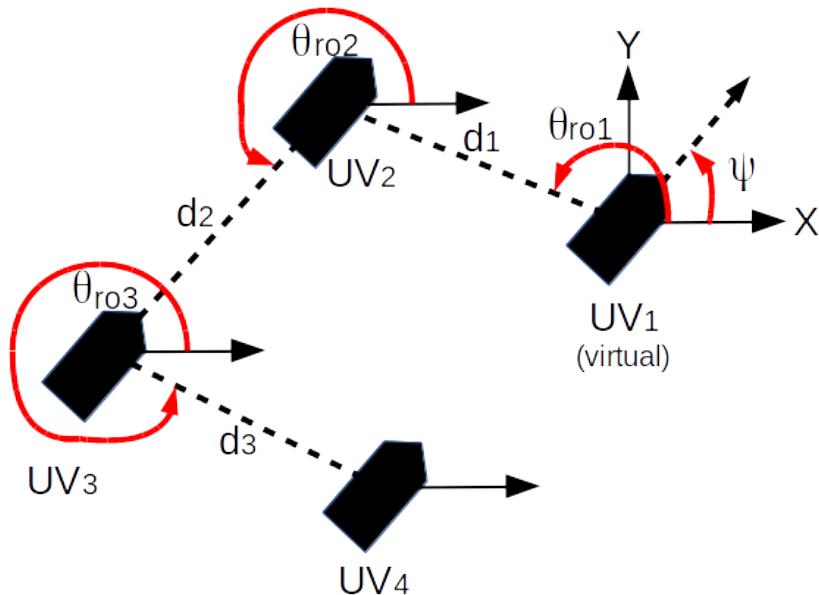


Figure 4.1: Formation topology

The kinematic equations which govern the model in Figure 4.1 are:

$$\begin{aligned}x_{n+1} &= d_n \cos(\theta_{ron} + \psi) + x_n \\y_{n+1} &= d_n \sin(\theta_{ron} + \psi) + y_n\end{aligned}\quad (4.1)$$

For $n = 1, \dots, m$, where m is the number of UVs. The general scheme for the proposed formation topology is illustrated in Figure 4.2, where the speed v , initial position $q_1(0)$ and desired coordinates $q_1(d)$ input the virtual Nomoto model and this one outputs the heading angle ψ and the actual position q_1 of the virtual UV. These are then input to the kinematic model as well as the relative orientation θ_{ro1} of UV_2 and the actual distance d_1 with respect to UV_1 . This distance is the elongation of the SMD system 1 which has as the input the desired formation distance L_{f1} for UV_2 with respect to UV_1 . The output of the kinematic model 1 is the reference position for vehicle 2 and the process continues successively up to UV_m .

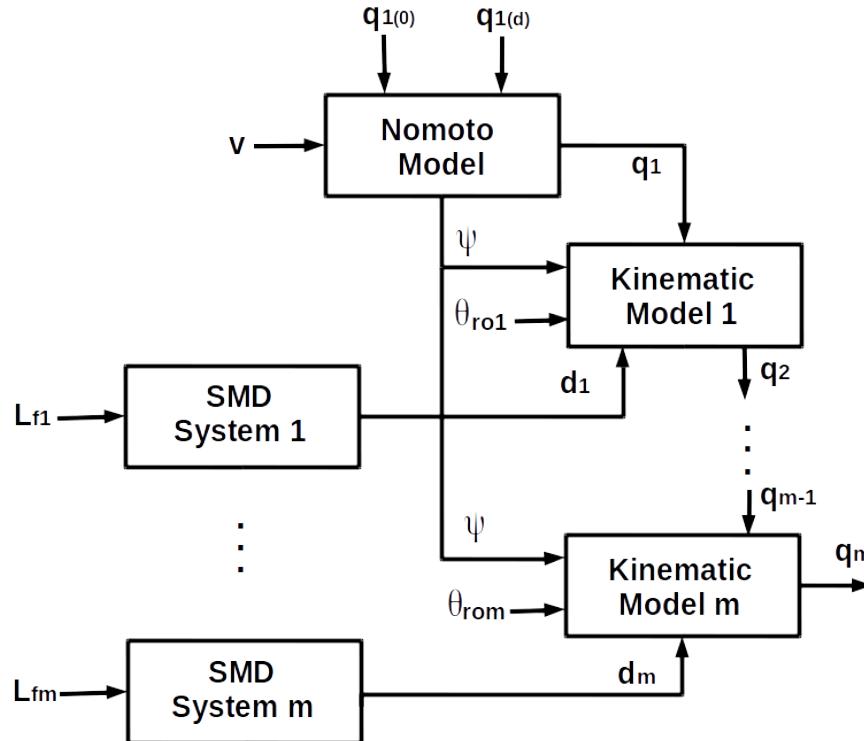


Figure 4.2: Simplified block diagram of the formation strategy

4.2. The Tensegrity Concept in the Context of Formation Control

The spring-mass-damper (SMD) reaction forces makes it a suitable model for a distance keeping architecture in which the equilibrium point of the SMD system is reached when the formation error tends to zero. Due to the SMD system regeneration force the elongation response is convenient for the formation of multiple unmanned vehicles, therefore, the

distance keeping problem is reduced to the stabilisation of the given SMD system around an elongation reference which is virtually equivalent to the formation distance.

The spring-mass-damper configuration proposed is depicted in Figure 4.3, where a spring in parallel with a damper connect a fixed reference frame (the leader) to a mass (the follower) and an external force F is exerted upon this mass.

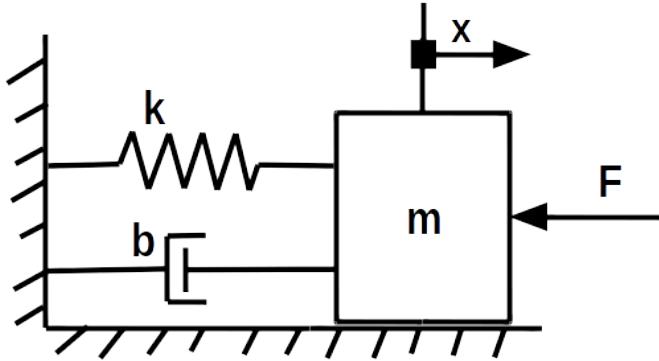


Figure 4.3: Spring-mass-damper system

Where k and b are defined as the elasticity and damping coefficients, m and x represent the mass and its position, respectively.

It is assumed that k , b and m are constant and that the spring force F_s is proportional to the position of the mass as well as the damping force F_b is proportional to the derivative of the position. The system dynamics can be represented by the following differential equation, according to Newton's second law:

$$m\ddot{x} = -kx - b\dot{x} + F \quad (4.2)$$

Let the state variables x_1 and x_2 be the position and the velocity, then Equation 4.2 can be rewritten as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{m}(-kx - b\dot{x} + F) \end{aligned} \quad (4.3)$$

Note that the control action u is the force applied to the mass, thus, the system can be expressed in state-space representation as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F \end{aligned} \quad (4.4)$$

Notice that all state variables are fed back so the implementation of a state estimator for LQR control purposes is not required. Moreover, the system is full state controllable as

matrix 4.5 is nonsingular, therefore, there exists an optimal vector $u(t)$ which stabilises the system.

$$\left[\begin{array}{c|cc} B & AB \end{array} \right] = \left[\begin{array}{cc} 0 & 1 \\ \frac{1}{m} & \frac{-b}{m} \end{array} \right] \quad (4.5)$$

For the design of the sliding mode controller let us define the following sliding surface:

$$S(t) = \lambda(x_1^* - x_1) + \beta \frac{d}{dt}(x_1^* - x_1) = \lambda(x_1^* - x_1) + \beta x_2 \quad (4.6)$$

Following the same procedure as in the previous chapter, the transversality condition holds as $\Delta S \neq 0$ as appreciated in equation 4.7, for any value of β greater than 0.

$$\Delta S g = \left[\begin{array}{cc} -\lambda & -\beta \end{array} \right] \left[\begin{array}{c} 0 \\ \frac{1}{m} \end{array} \right] = \frac{-\beta}{m} \quad (4.7)$$

4.3. Simulation Results

The autopilot used in this section is based on the second order Nomoto model presented in Chapter 3. For simplicity this system is controlled by an LQR servo controller in all cases as the results in this section are focused on the tensegrity concept integration within the formation system. The elasticity [N/m] and damping [N/(m/s)] coefficients are set to 1 as well as the mass [kg] m . The formation distance for each vehicle can be considered as the natural spring elongation of the SMD system, therefore, the control signal or force applied to each SMD system will remain around zero when the formation error tends to zero. If the actual Euclidean distance between any UV is greater than the setpoint, the force applied, which is of negative sign, can be regarded as an attracting force towards the reference vehicle. Conversely, if the distance between two UVs is below the reference, a positive force will be exerted upon the follower which can be regarded as a repelling force.

The *Simulink* model of the complete set up is depicted in Figure 4.4. Each SMD control system shown in this Figure consists of the SMD model itself and the corresponding controller, for instance, in the case of an LQR servo controller this diagram can be seen in Figure 4.5. The proposed formation shape is equivalent to the one shown in Figure 4.1 where all the relative distances are identical. The relative orientation of 135°, 225° and 315° for UV_2 to UV_4 with respect to their reference point. Initially, these are set to 26 m and every 100 s it is changed to 40 m, 20 m, 26 m, 36 m, 26 m, and 16 m. To avoid sudden control actions which may lead, in practice, to inter-vehicle collision, the distance setpoints changes are not in the form of a step, instead, a first order filter makes sure the change is slower. In this kind of applications external disturbances or forces are common, therefore, the performance of the controllers regarding disturbance rejection is vital.

The weighting parameters Q and R for the LQR design are:

$$Q = \begin{bmatrix} 40 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 10000 \end{bmatrix} \quad R = 5$$

The MATLAB command *lqr* returns the following optimal gain:

$$K_{flqr} = \begin{bmatrix} -2.83 & 20.37 & 44.19 \end{bmatrix}$$

The PID structure is the same as in the previous chapter and the tuning of for this design was done empirically. The gains which yield the best results in terms of error minimisation and settling time are:

$$P = 0.5 \quad I = 0 \quad D = 5 \quad N = 10$$

Finally, the SMC parameters are:

$$\lambda = 1 \quad \beta = 1 \quad UpperLimit = 10000 \quad LowerLimit = -10000$$

Figures 4.7 to 4.18 show the results obtained. Observe that for all the three approaches these formation references are achieved, however, one difference is easily spotted, the LQR servo controller rejects the external force disturbances more effectively than the PID and SMC. Also, note that the PID controller struggles with reacting to the formation changes.

Note, for instance, that in Figure 4.10 at time instant 300 s the control signal or force applied is considerably large as the desired formation distance changes from 40 to 20 m, this is due to the regulation made by the controller caused by the change of the equilibrium spring elongation, so that the variation is as soft as possible and with no overshoot. As for the settling time, it is only necessary to look at the first 100 s of the simulations as it is clear that the sliding mode controller achieves the desired formation distance in less than 6 s, as shown in Figure 4.16, at expense of the speed which presents a peak of over 20 m/s in order to establish the initial formation. While the LQR and PID achieve the 26 m in 25 and 50 s, respectively, with a speed peak of less than 3.5 m/s.

Particular attention should be paid to the sign of this force when there is an external disturbance. The disturbance forces are simulated through a square signal of amplitude 2, pulse width of 3% and period 100 s which is then added to a repeating stair sequence with sample time of 50 s and values between -1 and -5, therefore, the external disturbances can vary from -3 to 2 N. Above all, the LQR servo controller yields a less variant speed control which is a desirable feature for vehicle motion. The PID controller performs changes of speed with a more noisy behaviour. On the other hand, due to the relay limits

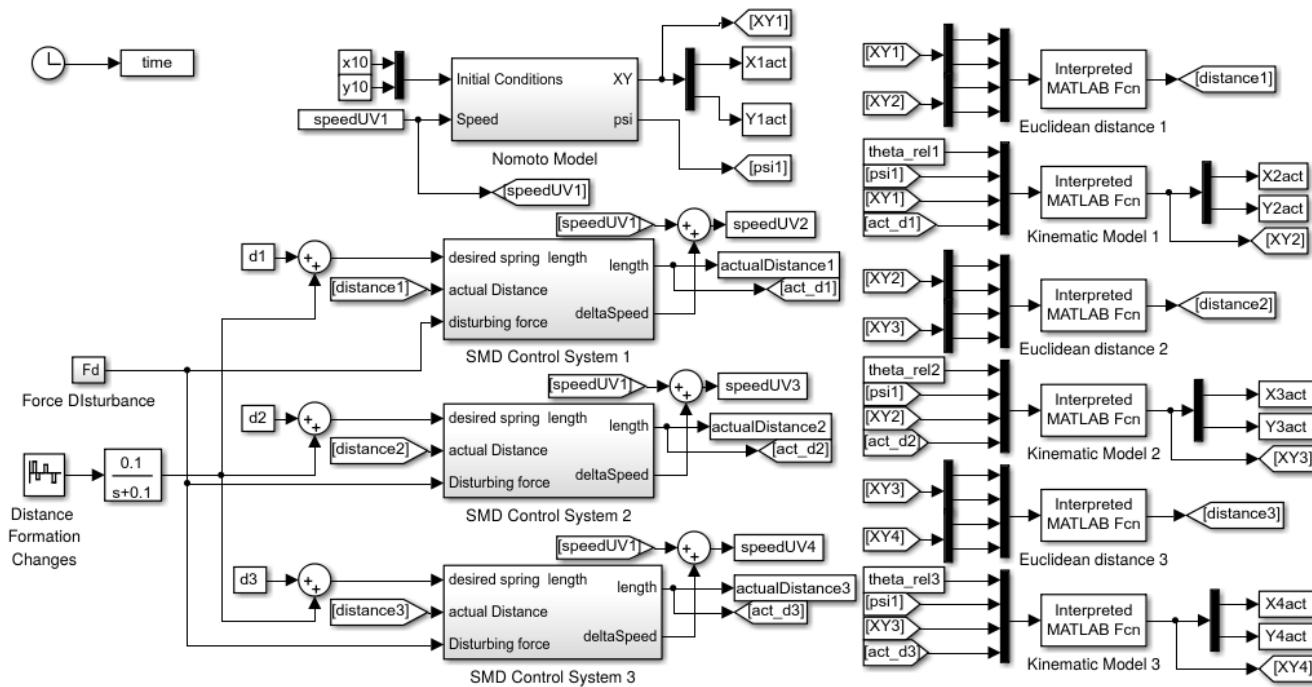


Figure 4.4: Simulink model of the formation of 4 UVs

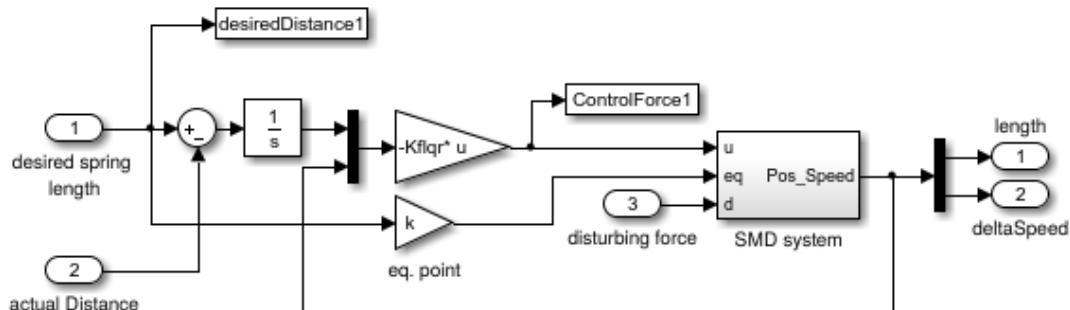


Figure 4.5: Simulink model of a spring-mass-damper controlled by an LQR servo

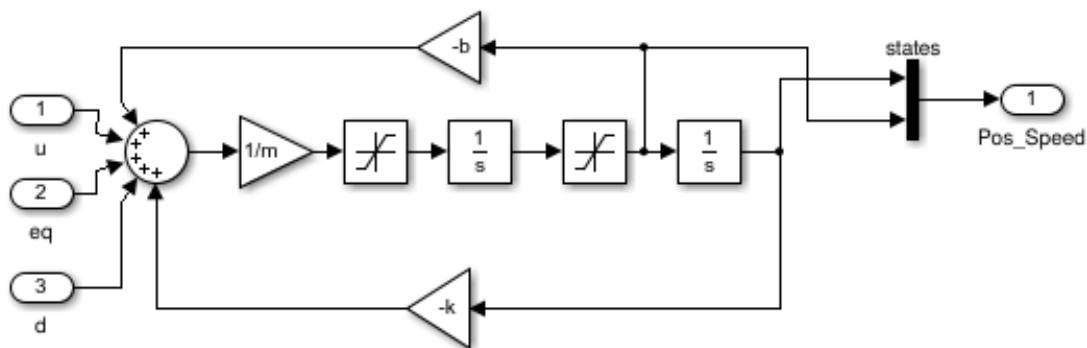


Figure 4.6: Simulink model of a spring-mass-damper

of the SMC it is able to reject the force disturbances with little effort. The chattering problem is attenuated by means of a second order filter placed after the relay which outputs the switching-like control signal. A closer look at this force signal is shown in Figure 4.17.

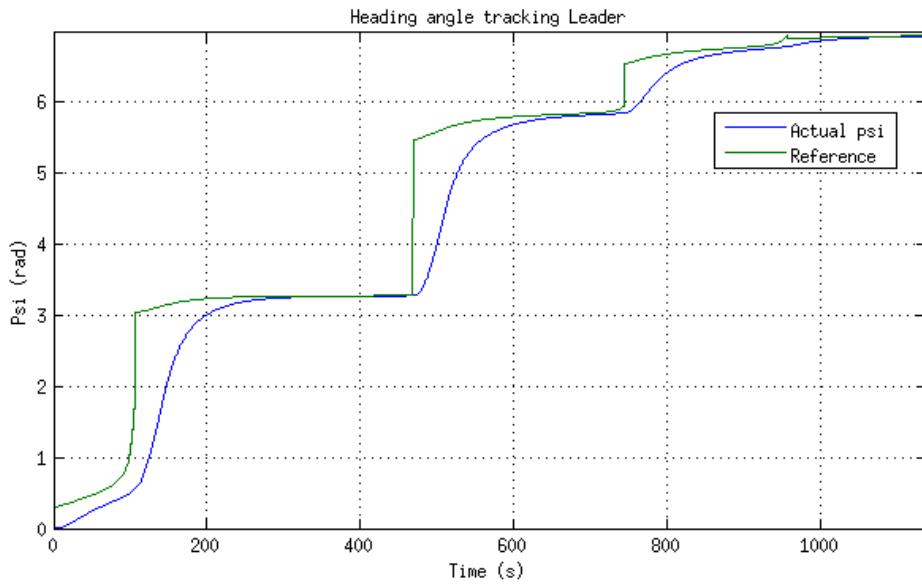


Figure 4.7: Heading angle control of the virtual UV

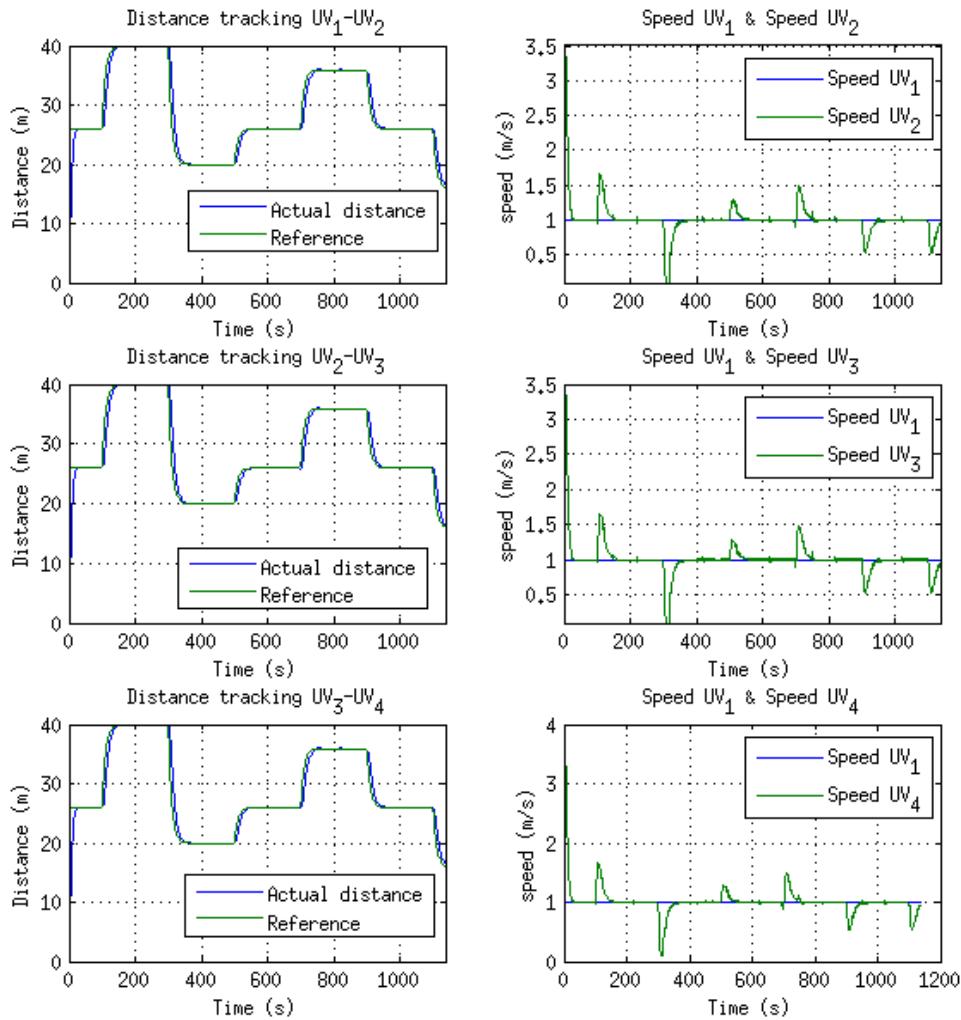


Figure 4.8: Formation distance and speed of each UV controlled by an LQR servo

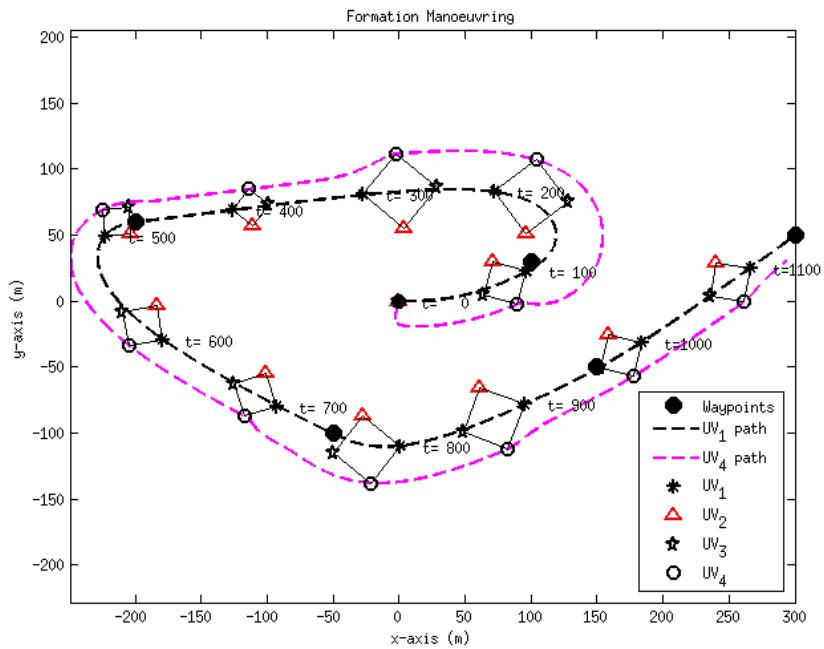


Figure 4.9: Simulation of the formation manoeuvring of the group of 4 UVs obtained by an LQR servo controller

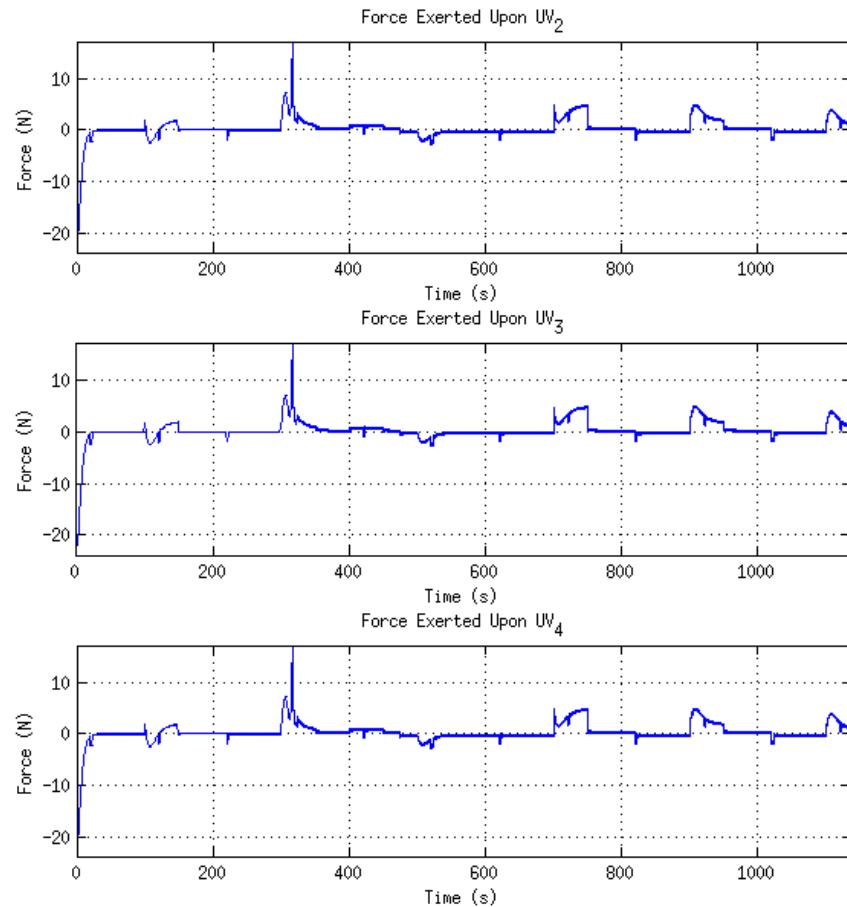


Figure 4.10: Force exerted upon each UV in the case of an LQR servo controller

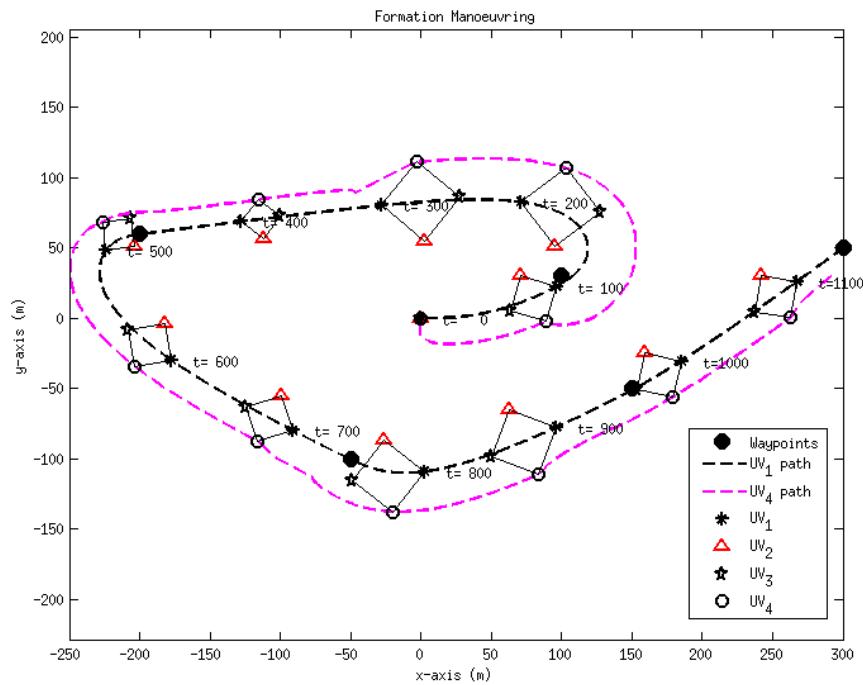


Figure 4.11: Simulation of the formation manoeuvring of the group of 4 UVs obtained by a PID contr

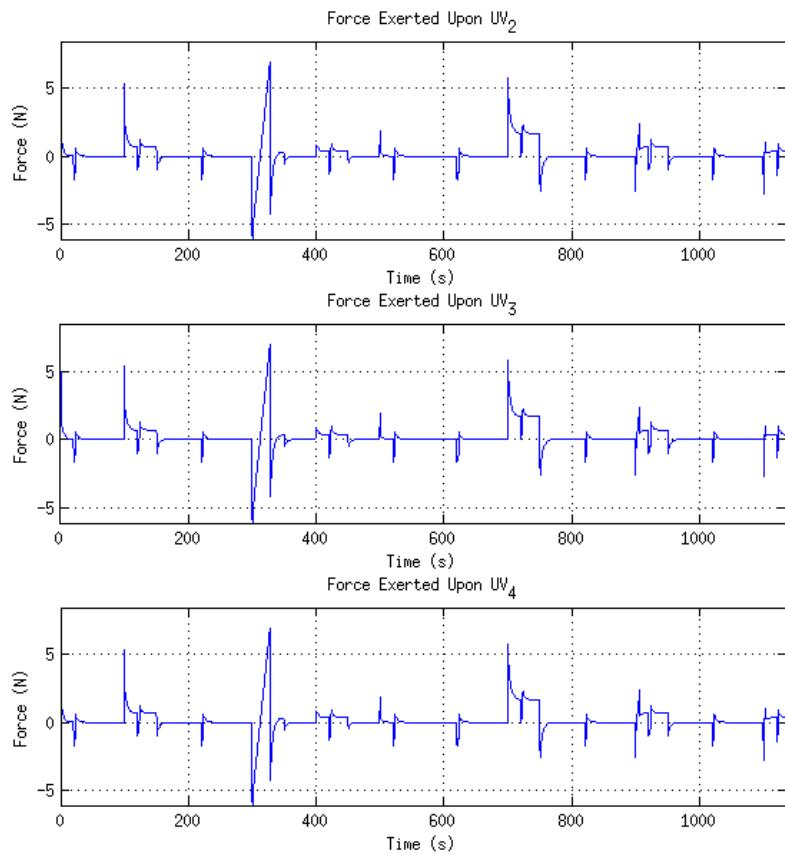


Figure 4.12: Force exerted upon each UV in the case of a PID controller

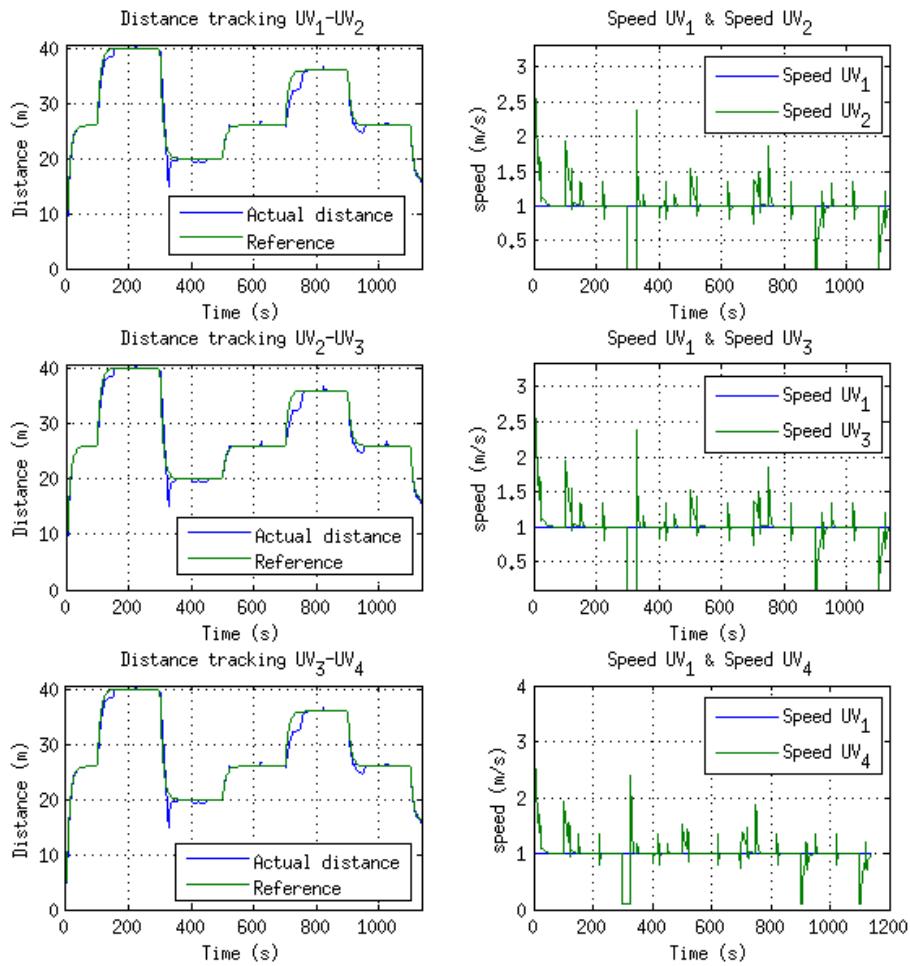


Figure 4.13: Formation distance and speed of each UV controlled by a PID controller

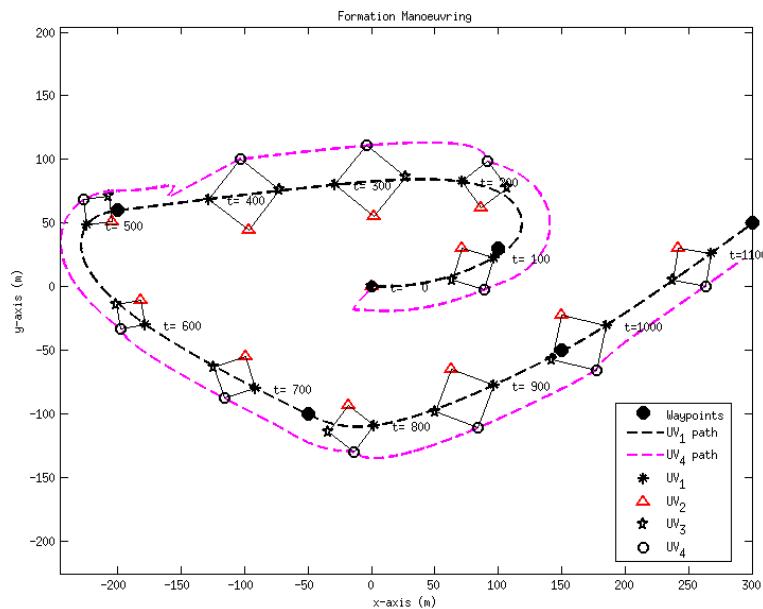


Figure 4.14: Simulation of the formation manoeuvring of the group of 4 UVs obtained by a SMC

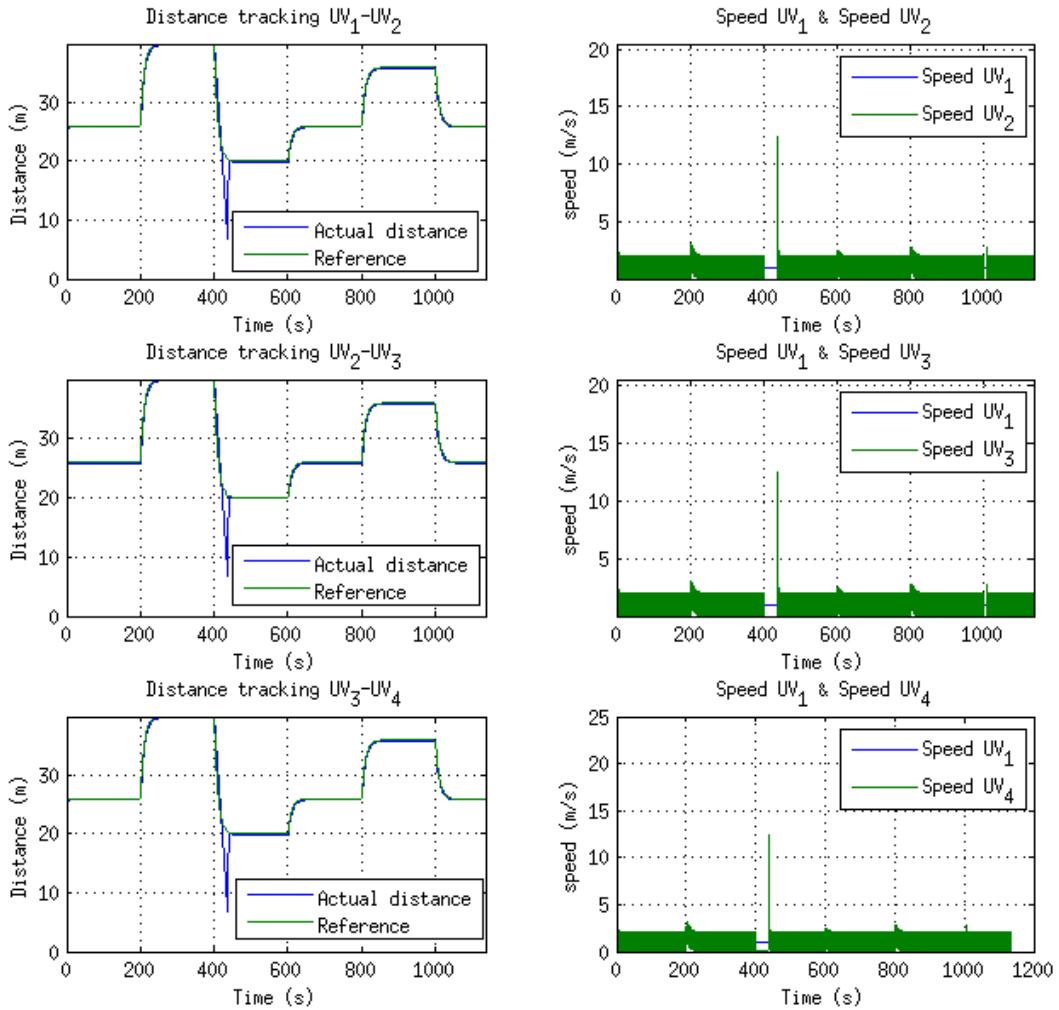


Figure 4.15: Formation distance and speed of each UV controlled by a SMC

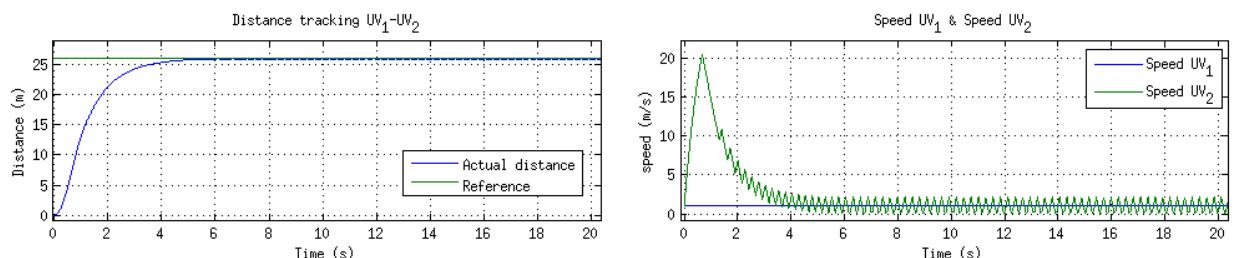


Figure 4.16: Closer look at the formation distance and speed of UV_1 in the case of a SMC

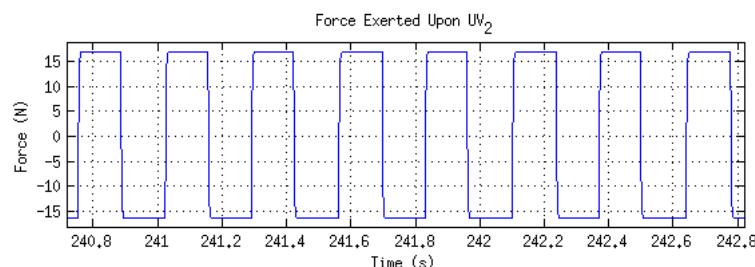


Figure 4.17: Closer look at the force exerted upon UV_1 in the case of a SMC

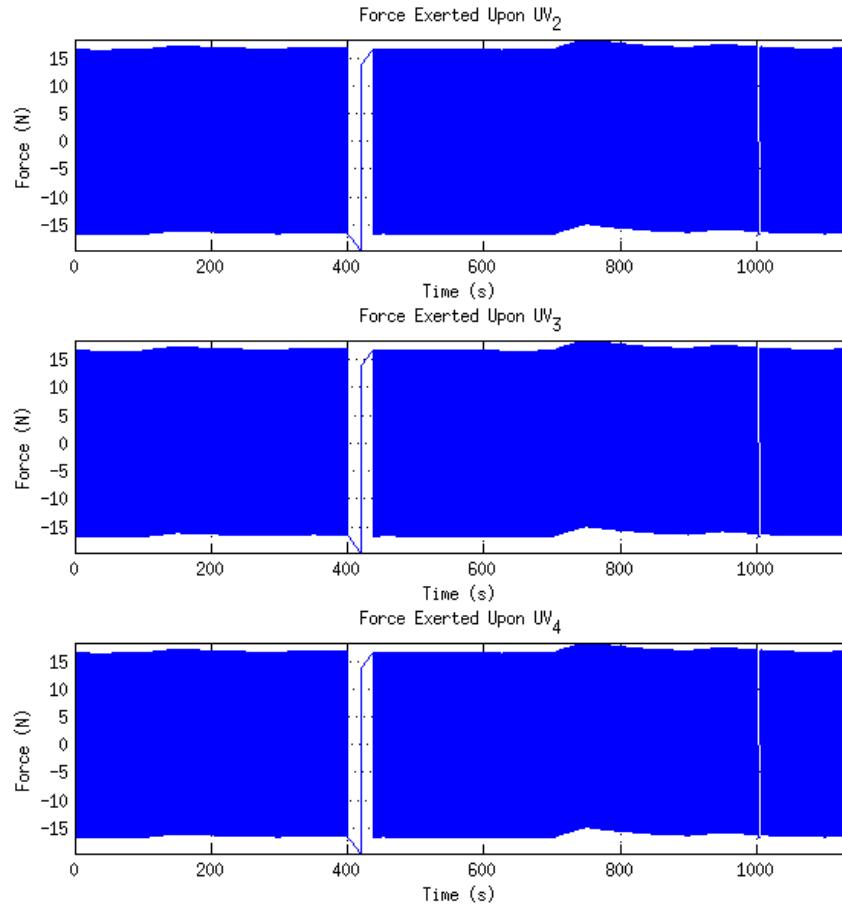


Figure 4.18: Force exerted upon each UV in the case of a SMC

4.4. Chapter Summary

A globally decentralised formation topology was presented in this chapter. A virtual UV based on Nomoto's second order model is controlled by an LQR servo. The relative distance and orientation of one vehicle is calculated with respect to one of its neighbours and it is assumed that one vehicle cannot be used as a reference to more than one vehicle, thus, if one of them ceases to function, the complete formation will not fail as the formation shape can be redefined. A kinematic model is used to validate the manoeuvring formation based on the tensegrity concept introduced in this chapter. Basically, a virtual SMD system is the link between two UVs whose spring elongation is the actual Euclidean distance between the two connected vehicles and its speed is the derivative of this elongation. The results presented prove that even though the SMC the chattering problem was attenuated, the simulation time increased substantially. Nonetheless, the simulations based on the LQR servo and PID controllers required less time. The LQR compared to the PID is more robust against disturbances and achieves a more suitable speed for the UVs. This is thanks to the state and control weights which can be specified in the LQR problem.

5. Implementation of a Tensegrity-based Formation Controller

In this chapter an experimental set up to test the tensegrity-based formation control concept is presented. This experiment encompasses the use of two Lego Mindstorms NXT tricycle robots, available at the Autonomous Systems Laboratory of the Queen's University Belfast, one acts as the leader and the other one as the follower. Both LQR-based and PID controllers, presented in the previous chapter, are implemented to track the formation distance between the follower and the leader. This chapter is structured in three sections, starting with an *overview of the Lego Mindstorms NXT environment*, then the *definition of the strategy* is defined and finally the *experimental results* are presented and discussed.

5.1. Overview of the Lego Mindstorms NXT Kit and its Development Tools

The LEGO® Mindstorms NXT, or simply NXT, has become a useful platform to build, test and verify new ideas in the field of robotics. Originally created as a programmable robotic toy for children, it soon caught the attention of researchers and developers for its versatility who saw great potential in it.

The set includes a NXT programmable LEGO® brick, servo motors, 6-wire cables, a variety of sensors such as ultrasonic, light, colour, touch and sound, and of course the world renowned LEGO® pieces comprising of bricks of different sizes and shapes, wheels, gears, pins, plates, tiles, among others. The input parameter to control the servo motors is the percentage of power applied to it, regarded in this work as the percentage of the maximum speed.

The programmable NXT brick, shown in Figure 5.1, is the central processing unit of the system. It possesses three interactive output ports to control the servo motors, and four input ports to read the data from the sensors. The NXT brick also includes a USB port which is used to program the device. It also incorporates Bluetooth 2.0 to allow wireless communication between two bricks or with a computer for programming purposes.

The default programming environment, depicted in Figure 5.2, is a linear graphical interface powered by LabVIEW™. Notwithstanding its intuitive and simplistic usability, it may become tedious and restrictive when the complexity of the program is relatively high.

MathWorks developed an add-on for *Simulink* to program the NXT brick. A useful feature which should be highlighted is the external mode which allows real-time signal monitoring and parameter tuning. For these reasons and its simplicity and versatility to solve more complex problems, *Simulink* is used. A complete set of blocks to interact with the NXT is available on the library as can be seen in Figure 5.3.

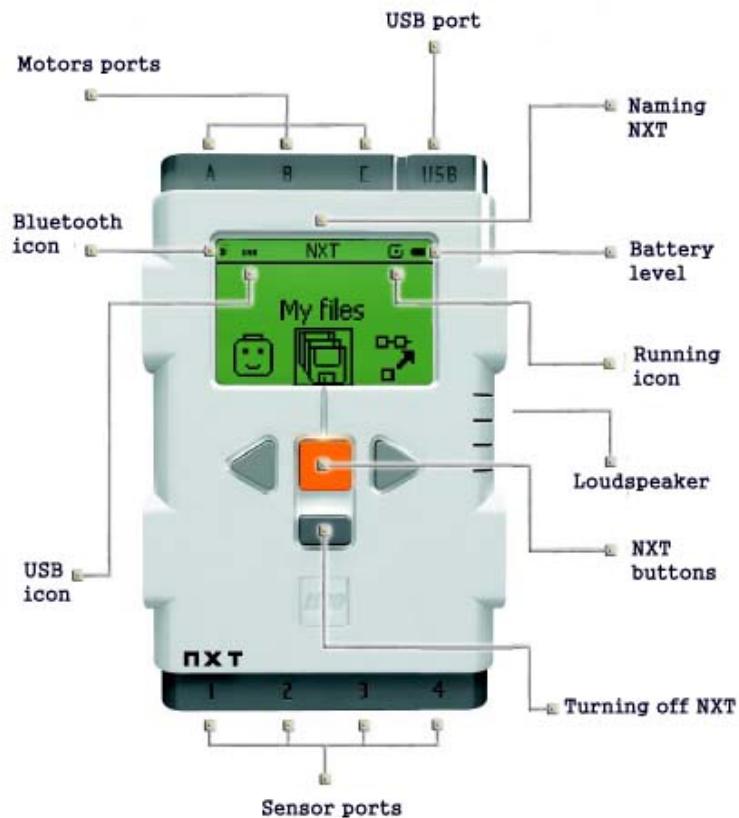


Figure 5.1: LEGO® Mindstorms NXT programmable brick showing the I/O ports and HMI components

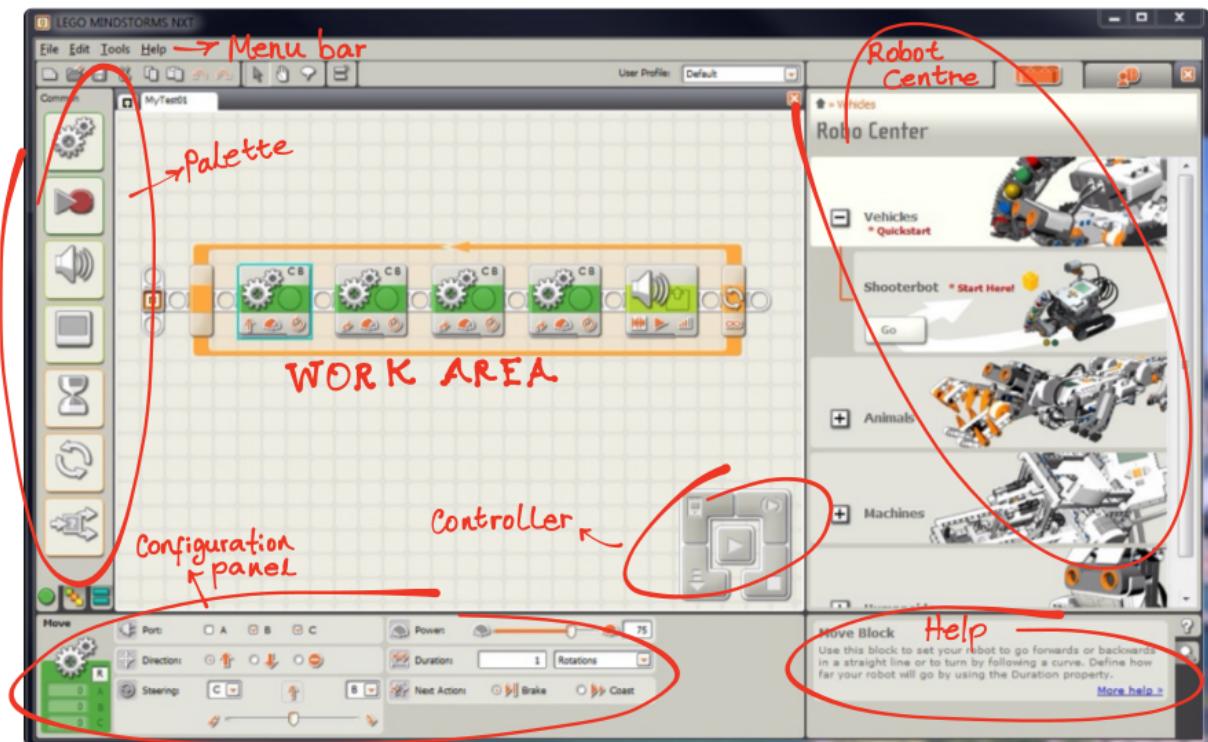


Figure 5.2: General overview of the Lego Mindstorms NXT software

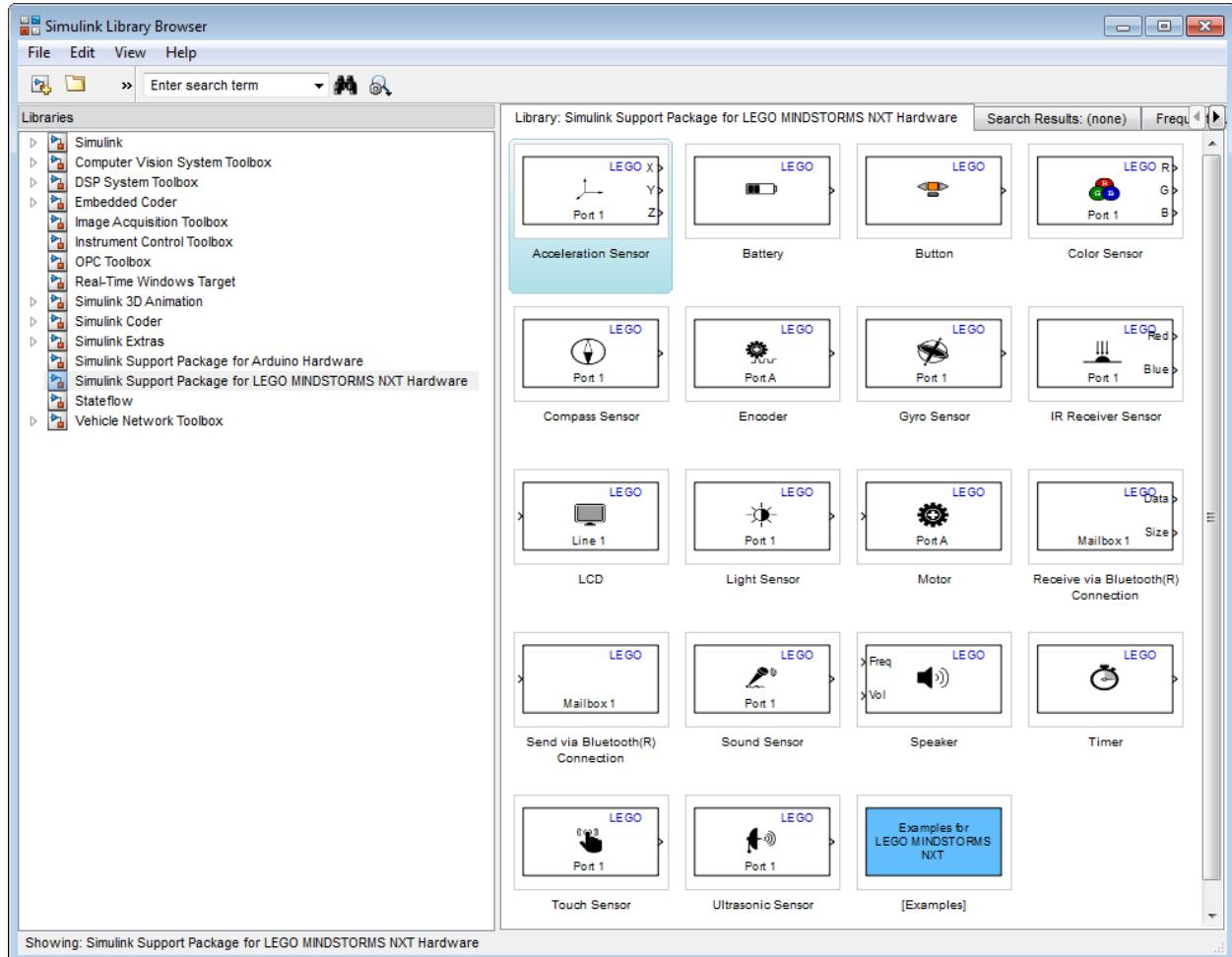


Figure 5.3: Block library available on the *Simulink* add-on

5.2. Definition of the Strategy

It should be recalled that the main purpose of this experiment is to test the general behaviour of the tensegrity-based formation control concept. The LEGO® Mindstorms NXT set can be used to build two differential wheeled robots, namely two tricycle robots. It comprises of two driving wheels at the front and one rear caster wheel. This structure was chosen because of its simplicity and that its motion is easy to program and can be well controlled. One of these tricycles is the leader which moves according to a programmed path and the other acts as the follower.

The only distance sensor available on the LEGO® Mindstorms NXT set, therefore the one selected, is an ultrasonic sensor whose range goes from 0 to 255 cm. In order to track the leader, a light-tracking mechanism was utilised. A horizontal platform at the back of the leader was attached with the purpose of placing a light source on it. For convenience, a smartphone with an LED flash was used as a torch. The follower should be able to track the light and turn accordingly. This was accomplished by building a structure over this vehicle where a servo motor is fixed. Attached to it are two colour sensors, one adjacent to the other, and the ultrasonic sensor which is placed on top of the mechanism.

The colour sensors, which can also be set as light intensity sensors, were used as they proved to be more sensitive to light changes. The first attempt was to just use one sensor to track the light, however, this required the use of a perturb and observe (P&O) algorithm which had a negative impact on the stability of the overall system because the oscillatory movements provoked by this algorithm made the ultrasonic sensor measurements to be also oscillating and, thus, unreliable. Hence the addition of the second sensor adjacent to the one already in place. Does not only this eliminate the need to use a P&O algorithm and its characteristic oscillatory behaviour but also reduces the complexity of the algorithm to a control problem where the difference of both sensor values is the error which inputs a PID controller and the control action is the speed of the servo motor which is holding the light-tracking mechanism. In addition to this, a pile of LEGO® bricks forming a wall is placed between the two sensors so that when the light is close, the system can react to small turns of the leader. Different photographs of the leader and the follower are shown in Figures 5.4 and 5.5, respectively.

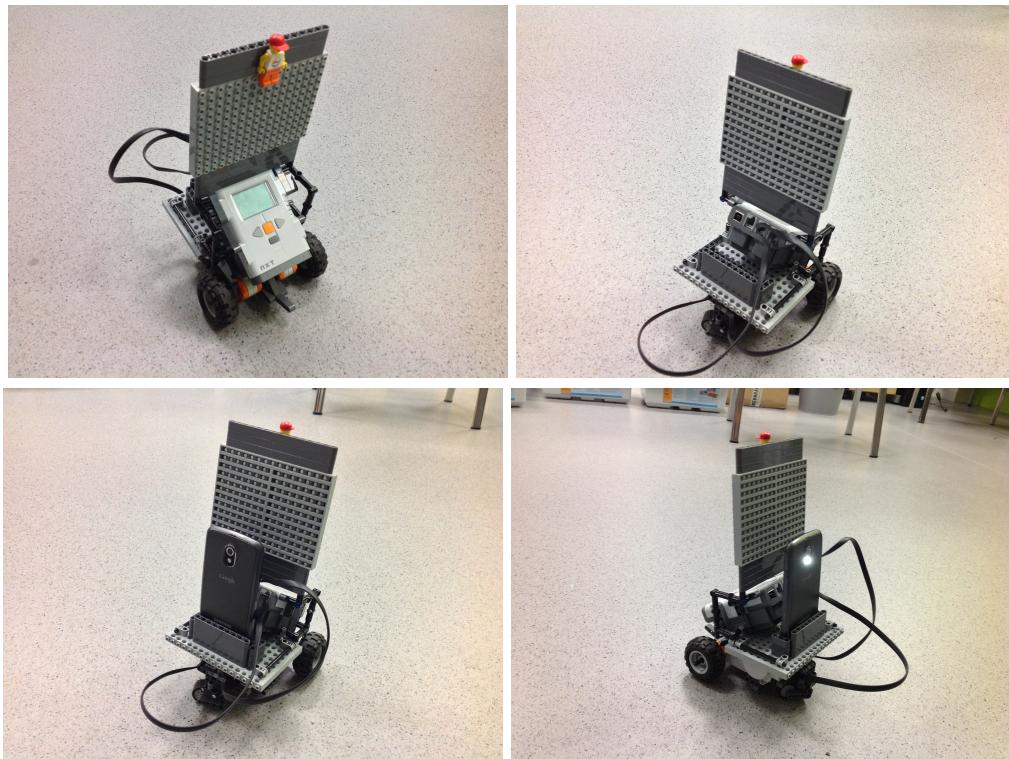


Figure 5.4: Pictures of the leader showing the wall and the light source

The ultrasonic sensor put on top of the light-tracking mechanism is measuring the distance to whatever object is in front of it. Therefore, it was considered that any distance greater than 100 cm will stop the robot from going any further, however, the light-tracking mechanism will still be active, making the robot turn towards the focus of maximum light intensity. This ensures reincorporation of the follower if the leader is spotted within its visual range. Similarly, any distance smaller than 20 cm, signifying a high threat of collision, will make all three motors stop. In order to avoid abrupt changes in the speed of the follower due to

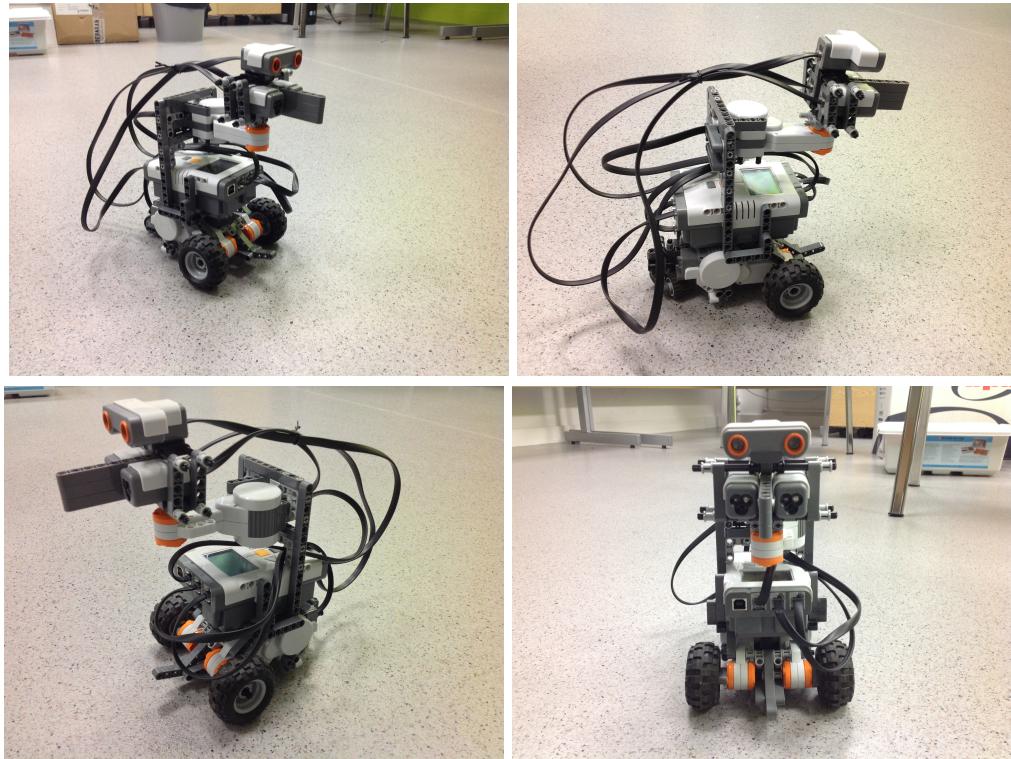


Figure 5.5: Pictures of the follower showing the light-tracking mechanism and the ultra-sonic sensor on top

the steering motions which affect the distance measurements, a rate limiter was added, this helps to prevent abrupt reactions of the follower which may lead to shaky steering or swinging distance tracking. The power at which the left and right motors can be set is limited to $\pm 60\%$, to allow manoeuvrability when a turn is made.

The *Simulink* program downloaded on the follower is shown in Figure 5.6. The program can be divided into three functional blocks as shown in Figure 5.7. First of all, the spring-mass-damper control system from which the speed of the robot is generated and then passed onto the robot manoeuvring section and finally the light-tracking system. The LQR-based controller and plant subsystem can be seen in Figure 5.8 where the LQR gains are represented by the constant array K. Its values are -0.8165, 5.2837, 12.3504, for the formation error, position and speed states, respectively. These gains were obtained with the MATLAB command LQR, explained in the previous chapter. A gain is needed to condition the speed of the spring-mass-damper system so that this signal can be used to power the motors. Notice that the negative sign of this gain is due to the position of the motors which generates a change of direction with respect to what is considered going forwards. Similarly, the PID controller and plant subsystem is depicted in Figure 5.9. Notice that the only input to this block is the formation error. The PID parameters are 2, 2.2 and 0.5, respectively, and were empirically obtained.

The *Simulink* program downloaded on the leader is shown in Figure 5.11. It approaches the problem as a simple steering angle control which in this case is 0 degrees. The right

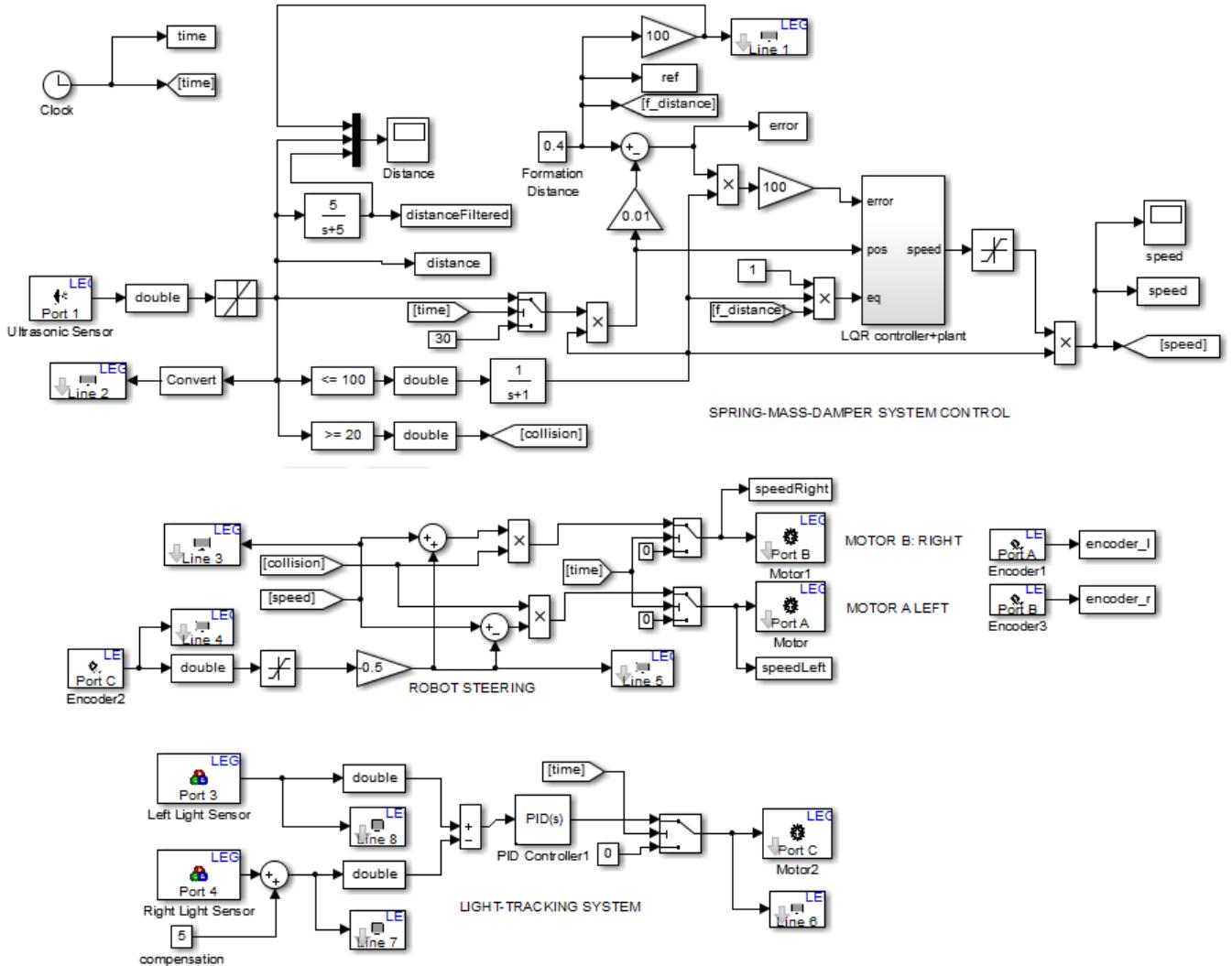


Figure 5.6: *Simulink* program downloaded on the follower using a light-tracking algorithm and LQR-based formation controller

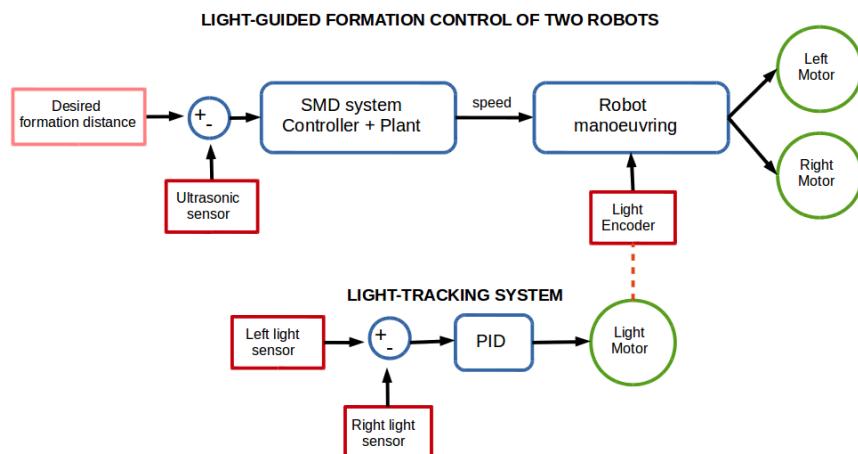


Figure 5.7: Simplified block diagram of the program downloaded on the follower

wheel is powered to 30% while the left motor utilises the steering control signal. The Bluetooth capability of the NXT bricks brings the possibility of communication be-

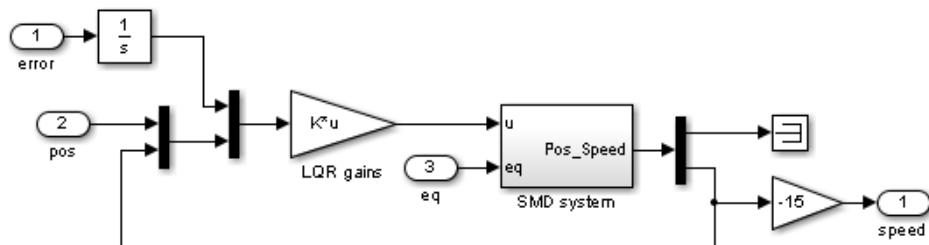


Figure 5.8: LQR-based controller and plant subsystem

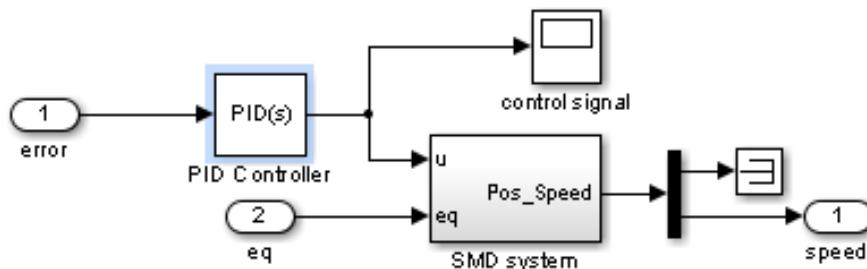


Figure 5.9: PID controller and plant subsystem

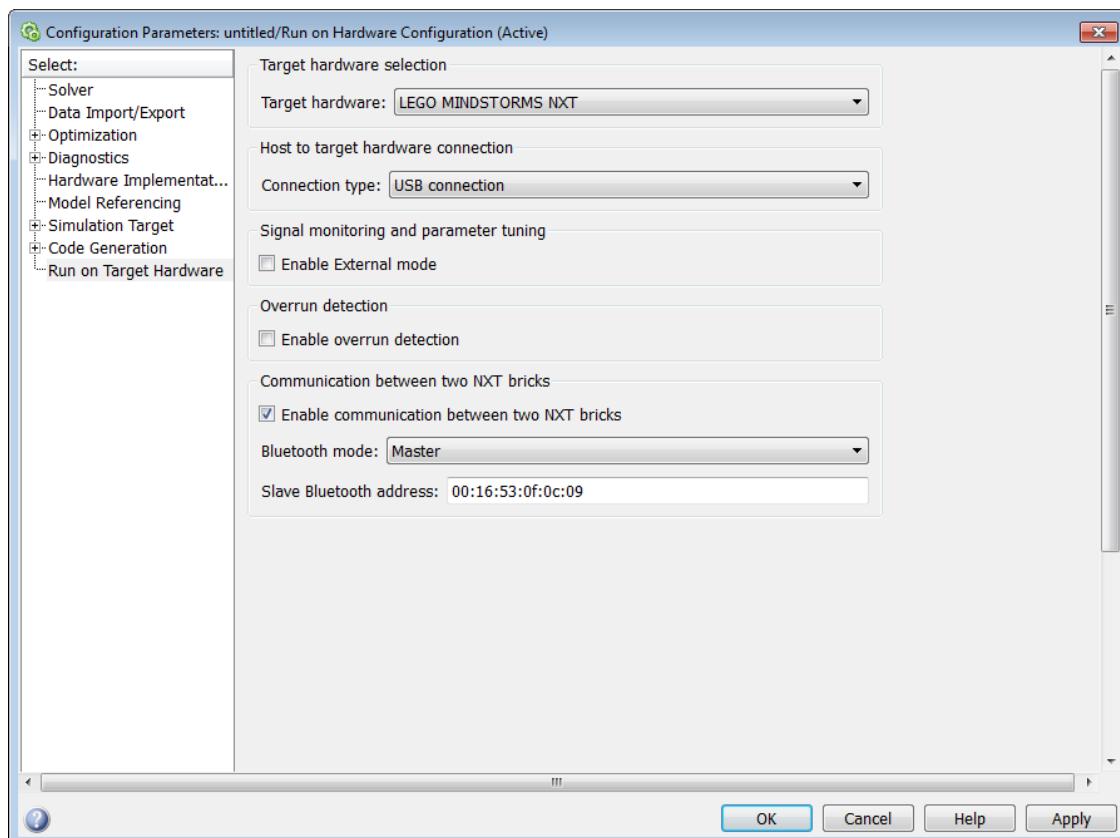


Figure 5.10: Configuration parameters of master NXT brick

tween them. It is common to find that a leader provides the follower with the formation distance and in this case it is possible to establish a Bluetooth connection for such pur-

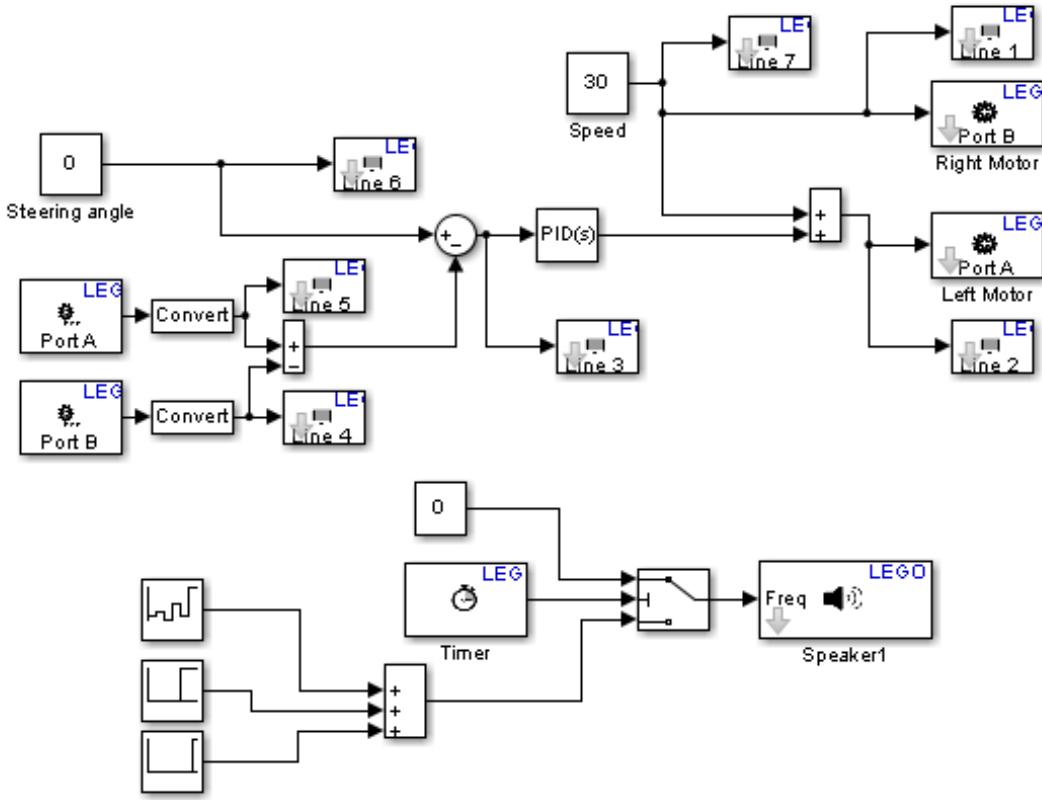


Figure 5.11: *Simulink* program downloaded on the leader: Straight-line trajectory

pose. Nevertheless, the NXT brick is not able to connect to another brick and the master computer at the same time, therefore, an alternative program is done, which presents a slight modification of the program presented in Figure 5.6 where the formation distance block is replaced with a Bluetooth reception block. It should be mentioned that when both bricks are programmed, the "*communication between two NXT bricks*" option should be enabled and whether the device being programmed is the slave or the master. In the latter case, the slave Bluetooth address must be specified as shown in Figure 5.10. The results were satisfactory and because it is not possible to retrieve data from the follower the overall behaviour of this experiment can be seen in this video.

5.3. Experimental Results

The experiment was carried out at the Autonomous Systems Laboratory of Queen's Belfast University under a controlled light environment to ensure proper results regarding the light-tracking algorithm.

It should be pointed out that the leader moves according to a programmed trajectory and does not possess any capability to navigate autonomously, which is why the assumption of the absence of obstacles is made. This is due to the scope of the project which does not include this issue but only the behaviour of the tensegrity-based formation.

The follower is connected via Bluetooth to the master computer from which different sig-

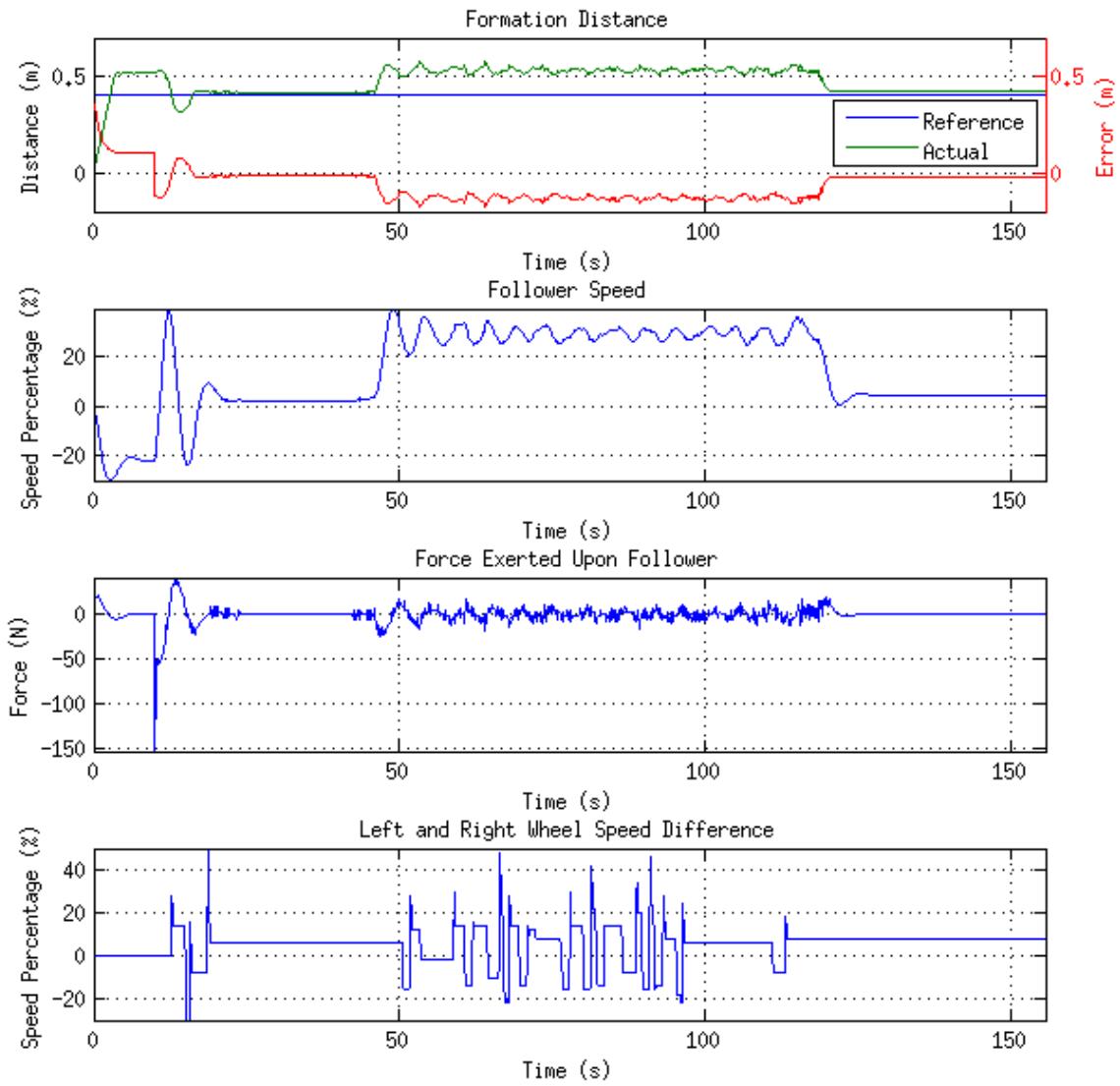


Figure 5.12: Results obtained with the PID controller, formation distance set to 0.4 m

nals can be monitored and any parameter of the program can be tuned. Once the LQR and PID gains are set up according to overshoot, settling time and tracking requirements, the experiment takes place as follows:

- Follower is turned on and placed behind the leader, which is on but immobile.
- Follower connects to master computer and before it starts moving a 10-second time period is allowed for the sensors values to settle.
- The follower program starts and corrects the formation distance, set in this case to 0.4 m. The leader is still at rest.
- Once both robots are in formation, the program on the leader is started and begins to move in straight line with a speed percentage of 30%.

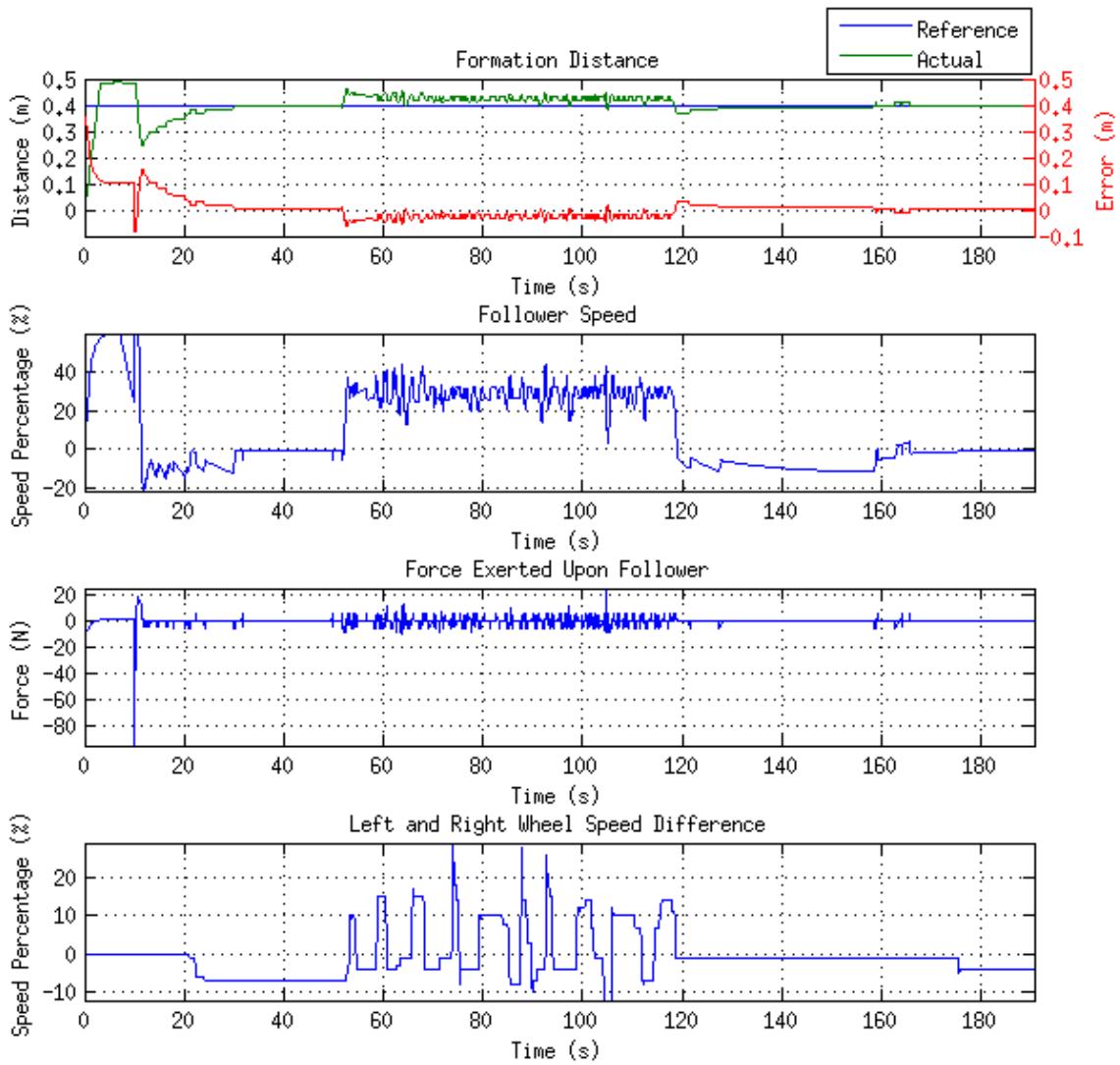


Figure 5.13: Results obtained with the LQR-based controller, formation distance set to 0.4 m

- After some time, the leader is stopped and some time is allowed for the follower to minimise the formation error.
- End of the experiment.

The previous process is done using the LQR-based controller, whose results can be seen in Figure 5.13, and the PID controller, whose results are depicted in Figure 5.12. In order to assess the transient behaviour the formation distance is changed from 0.4 m to 0.3 m in another experiment similar to the one previously described and the results obtained from this experiment are displayed in Figures 5.14 and 5.15 for the LQR and PID controller, respectively.

Results include, from top to bottom, the reference of the formation distance and its actual value as well as the formation error. Following is the speed percentage of the follower, the

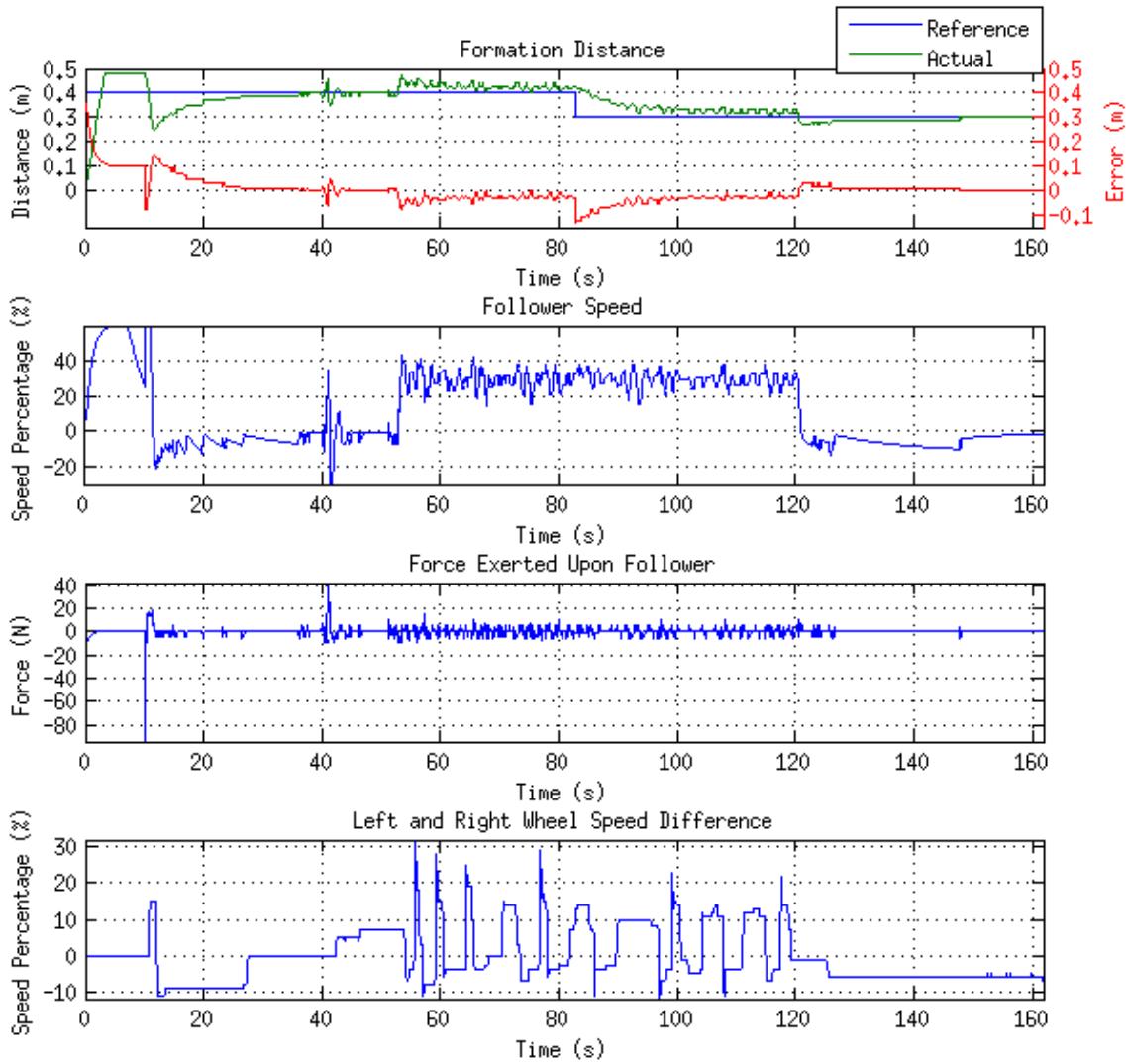


Figure 5.14: Results obtained with the LQR-based controller, formation distance changed from 0.4 m to 0.3 m

force exerted upon it and the difference of the power or speed percentage of the left and right wheel. The latter is an indication of the steering of the follower, if the same power is applied to both motors, and assuming they are identical, it can be expected that the robot moves in a straight line and this differential should remain around zero, which is also expected if the robot is at rest. On the contrary, if this differential is positive or negative it means that the robot is turning right or left, respectively. In reality, applying the same power to both motors might not necessarily mean they will turn at the same speed or with the same torque due to friction, deterioration and other environmental factors which were not accounted for.

Figures 5.12 and 5.15 reveal that a PID controller performance is acceptable as no overshoot is present but failed to maintain the formation error within an admissible range while the robots were active. Said error remained within 0.1 m for the two setpoints 0.4 and 0.3

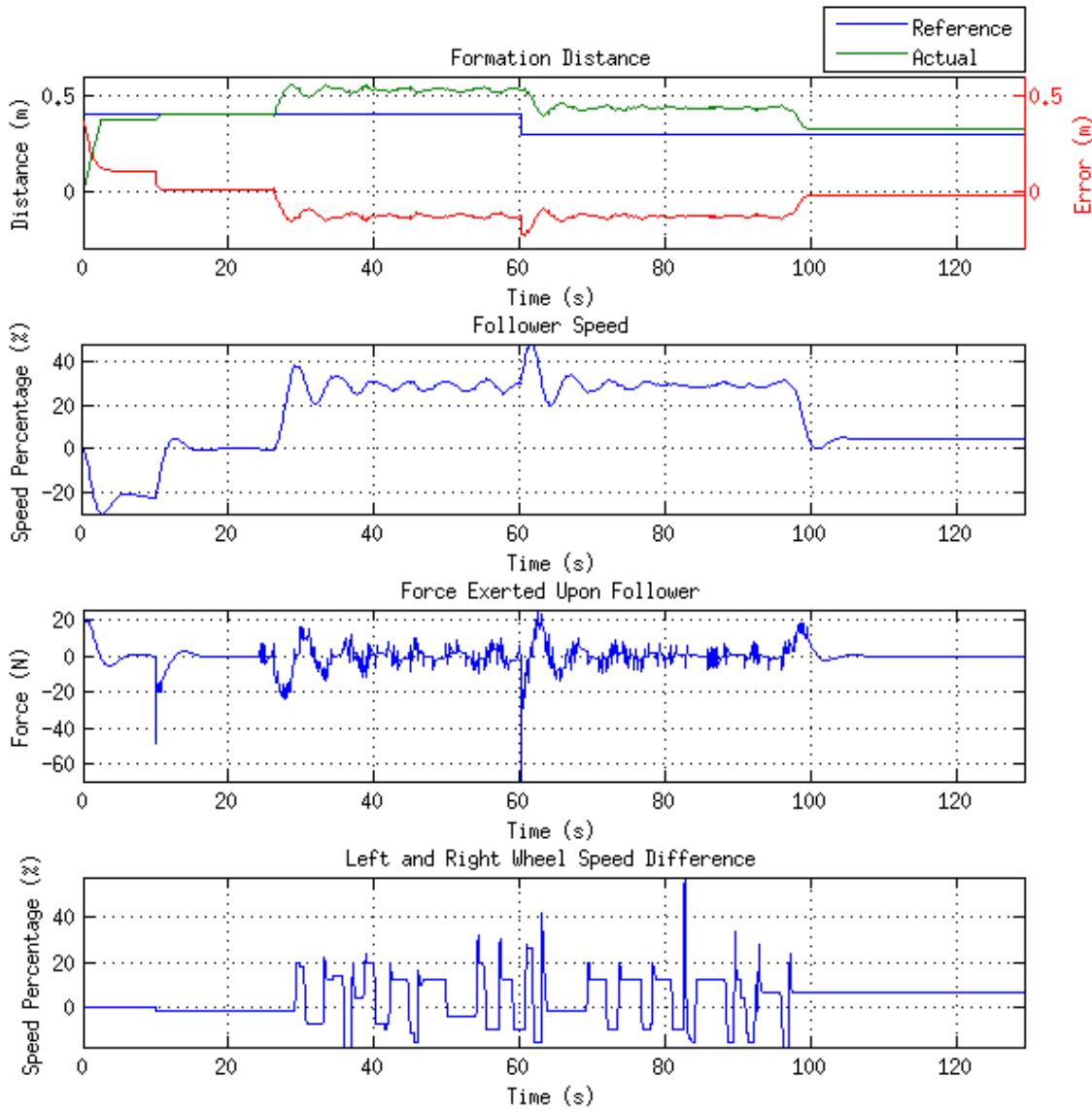


Figure 5.15: Results obtained with the PID controller, formation distance changed from 0.4 m to 0.3 m

m. Conversely, the LQR-based controller proved to be able to maintain the error below 0.05 m for any reference which is half of what the PID achieved. Recall that the speed applied to the motors is extracted from the spring-mass-damper system whose elongation is being controlled by the LQR-based and PID controllers. The results presented here show that the speed, in the case of a PID, is smoother as opposed to the LQR case. It can also be observed that the average speed is 30% which is the speed at which the leader is moving. Furthermore, notice the force exerted upon the follower, it performs pronounced negative peaks as to attract it towards the leader when the distance between them is greater than the reference and positive peaks otherwise.

5.4. Chapter Summary

Throughout this chapter the physical implementation of the tensegrity concept was introduced in a formation control of two NXT-based mobile robots. One acting as the leader and the other as the follower. The proposed strategy was to build a light-guided follower that was able to keep a certain distance from the leader. Both NXT robots were programmed using the add-on available on *Simulink*. The LQR and PID approaches used to control the formation distance proved to be effective, however, the LQR yielded better results in achieving a smaller formation error while in motion. The overall behaviour of the system validated the use of the tensegrity concept as a means to obtain a smoother and more natural formation tracking.

6. Conclusions and Future Work

The work presented in this report mainly develops, compares and implements different control techniques to validate the concept of tensegrity within the formation control context. Namely, a PID, SMC and an LQR servo controller were designed and simulated to regulate the formation distance of the UVs involved. The first problem addressed in Chapter 3 was the control of one single unmanned vehicle based on the first order Nomoto model and afterwards the waypoint problem. A globally decentralised formation architecture for m vehicles is introduced in Chapter 4 where one virtual vehicle generates the heading angle which is then used by the kinematic models to calculate the paths of the other UVs. The mechanism established to determine the speed at which the vehicles move is a SMD system whose distance is the Euclidean distance between a pair of UVs. Generally speaking, the desired formation configuration, an acceptable disturbance rejection and settling time were achieved in all cases. However, the balance found between performance and robustness of the LQR servo controller and its simplicity to implement makes it the best option, as it was concluded in Chapters 4 and 5. The former In this latter, the tensegrity concept is tested and validated with 2 LEGO® Mindstorms NXT tricycle robots. For the experiment a light-guided robot tracks the leader and keeps a certain distance from it thanks to the ultrasonic sensors which is attached on top of the light-tracking system. The leader has a light-emitting source attached at the back and has no motion restrictions. The results validated the tensegrity concept and it could be observed that an LQR servo controller is able to track the distance in motion with a smaller error to that obtained with a PID.

Further work may involve the use of a more complex dynamic model of a vehicle where not only the heading angle and its rate can be regulated but also its acceleration. This may imply the use of nonlinear models for which further research should be done in order to determine whether a nonlinear control technique, such as the sliding mode one, or a linearisation should be applied. If the sliding mode controller line was to be continued, a strategy should be suggested in order to eliminate the chattering problem and with it the large amount of time needed to run the simulations. It should also be considered a formation topology where all the vehicles involved have independent dynamics rather than using a kinematic model. Furthermore, the study of the tensegrity should be extended to a 3-dimensional formation of multiple unmanned vehicles, i.e. aircraft, and additional variables should be taken into account, such as communication architectures, data flow constraints, the reincorporation or rejection of UVs which enter or get away from a safe area, as well as an integrated obstacle avoidance policy.

References

- [1] Youmin Zhang and Hasan Mehrjerdi, *A Survey on Multiple Unmanned Vehicles Formation Control and Coordination: Normal and Fault Situations*, 2013 International Conference on Unmanned Aircraft Systems (ICUAS) May 28-31, 2013, Grand Hyatt Atlanta, Atlanta, GA.
- [2] The Times of Israel, *Illustrative photo of IAF F-15s flying in formation. (photo credit: Moshe Shai/Flash90)*, Retrieved September 1 2014, <http://www.timesofisrael.com/un-reports-increased-iaf-activity-over-lebanon/>, March 2013.
- [3] Wikipedia, The Free Encyclopedia, *Satellite formation flying*, Retrieved September 1 2014, http://en.wikipedia.org/wiki/Satellite_formation_flying, August 2007.
- [4] Wikipedia, The Free Encyclopedia, *The Royal Navy fleet auxiliary ship Lyme Bay (L 3007) leads a formation of ships including the mine countermeasure ships USS Gladiator (MCM 11)*, Retrieved September 1 2014, [http://commons.wikimedia.org/wiki/File:US_Navy_090905-N-3165S-626_The_Royal_Navy_fleet_auxiliary_ship_Lyme_Bay_\(L_3007\)_leads_aFormation_of_ships_including_the_mine_countermeasure_ships_USS_Gladiator_\(MCM_11\).jpg](http://commons.wikimedia.org/wiki/File:US_Navy_090905-N-3165S-626_The_Royal_Navy_fleet_auxiliary_ship_Lyme_Bay_(L_3007)_leads_aFormation_of_ships_including_the_mine_countermeasure_ships_USS_Gladiator_(MCM_11).jpg), November 2009.
- [5] Jonathan R. T. Lawton, Randal W. Beard, Senior Member, IEEE, and Brett J. Young, *A Decentralized Approach to Formation Maneuvers*, IEEE Transactions on Robotics and Automation, Vol. 19, No. 6, December 2003.
- [6] F.E. Schneider and D. Wildermuth, *A Potential Field Based Approach to Multi Robot Formation Navigation*, International Conference on Robotics, Intelligent Systems and Signal Processing. Changsha, China. October 2003.
- [7] G.L. Mariottini et al., *Vision-Based Localisation for Leader-Follower Formation Control*, IEEE Transactions on Robotics, vol. 25, No 6. December 2009.
- [8] A. Yang, W. Naeem, G. W. Irwin, and K. Li, *Stability Analysis and Implementation of a Decentralized Formation Control Strategy for Unmanned Vehicles*, IEEE Trans. On Control Systems Tech.
- [9] Adeleh Mohammadi, Mohammad Bagher Menhaj, and Ali Doustmohammadi , *Distributed Model Predictive Control and Virtual Force Obstacle Avoidance for Formation of Nonholonomic Agents*, 2nd International Conference on Control, Instrumentation and Automation (ICCIA) , pp. 440-445. Dec. 2011.

- [10] M. Defoort, T. Floquet, A. Kökösy, and Wi. Perruquetti , *Sliding-Mode Formation Control for Cooperative Autonomous Mobile Robots*, IEEE Trans. On Industrial Electronics, vol. 55, pp. 3944-3953, 2008.
- [11] F. Fahimi, *Sliding Mode Formation Control for Under-Actuated Autonomous Surface Vehicles*, IEEE Trans. On Robotics, vol. 23, no. 3, pp. 617-622, 2007.
- [12] Xu Bo , Youtao Gao, "Sliding Mode Control of Space Robot Formation Flying, *Proceedings of the 4th International Conference on Autonomous Robots and Agents*, pp. 561-565, 2009.
- [13] Wikipedia, The Free Encyclopedia, *Movimiento y Oscilación del Buque*, Retrieved 27 August 2014, http://es.wikipedia.org/w/index.php?title=Movimiento_y_oscilaci%C3%B3n_del_buque&oldid=66729659, May 2013.
- [14] Pang Rui, *Multi-UAV Formation Manoeuvring Control Based on Q-Learning Fuzzy Controller*, 2nd International Conference on Advanced Computer Control (ICACC), vol. 4, pp. 252-25., 2010.
- [15] Mohammad Fadhil Bin Abas et al., *MAV Circular Leader-Follower Formation Control Utilising Mass-Spring-Damper with Centripetal Force Consideration*, Journal of Robotics and Mechatronics, vol. 25, No.1, 2013.
- [16] Feng Wang, Aouf Nabil, Antonios Tsourdos, *Centralized/Decentralized Control for Spacecraft Formation Flying Near Sun-Earth L2 Point*, 4th IEEE Conference on Industrial Electronics and Applications, 2009.
- [17] Ching-Yaw Tzeng and Ju-Fen Chen, *Fundamental Properties of Linear Ship Steering Dynamic Models*, Journal of Marine Science and Technology, Vol. 7, No. 2, pp. 79-88 (1999).
- [18] Fossen, T.I., *Guidance and Control of Ocean Vehicles*, John Wiley and Sons, NY (1994).
- [19] Nomoto, K., Taguchi. T., Honda, K. and Hirano, S., *On the Steering Qualities of Ships*, Technical Report, International Shipbuilding Progress, vol. 4, (1957).
- [20] Davidson, K.S.M. and Schiff, L.I., *Turning and Course Keeping Qualities*, Transactions of SNAME, vol. 54, (1946).
- [21] Shtessel, Y., Edwards, C., Fridman, L., Levant, A., *Sliding Mode Control and Observation*, Springer, 2014.
- [22] Ogata, Katsuhiko, *Modern Control Engineering*, Prentince Hall, Fifth Edition, 2010.

[23] Sook Yen Lau and Wasif Naeem, *Tensegrity-Based Formation Control of Unmanned Vehicles*, UKACC International Conference on Control, 2012.