A Comparison Study of MPC and Control Barrier Functions Algorithms for Multi-Agent Systems in the presence of Obstacles

João Miguel Carvalho Pina Neves

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Daniel de Matos Silvestre
Prof. Carlos Jorge Ferreira Silvestre

Examination Committee

Chairperson: Prof. Maria Luísa Torres Ribeiro Marques da Silva Coheur
Supervisor: Prof. Daniel de Matos Silvestre
Member of the Committee: Prof. Bruno João Nogueira Guerreiro

May 2022

Acknowledgments

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through Institute for Systems and Robotics (ISR), under Laboratory for Robotics and Engineering Systems (LARSyS) project UIDB/50009/2020, through project PCIF/MPG/0156/2019 FirePuma and through COPELABS, University Lusófona project UIDB/04111/2020.

Abstract

Formation Control problems have been widely studied in the industry. Model Predictive Control, which is an optimization looking forward at horizon h, and other methods have seen use in industrial and other settings for decades. More recently, Control Lyapunov Functions have seen an increase in interest. These functions follow a greedy approach to optimization of a system. This thesis will serve as a comparative work between the different methods used for formation control, namely on their measured output, computation time and simulation accuracy. It will be focused on Lyapunov Functions, MPC with horizon = 1 and MPC with a large horizon. MPC with a horizon = 1 has been chosen to properly compare with the Lyapunov method, Lyapunov Functions do not optimise over a horizon, they look at a single step.

Keywords

Control Lyapunov Functions; Control Barrier Functions; Model Predictive Control; Linear Constraints; Formation Control.

Resumo

Problemas de Controlo de Formações (*Formation Control*) têm vindo a ser estudados recentemente na indústria. Model Predictive Control que é uma optimização a olhar para um horizonte h, e outros métodos têm uso comum em situações industriais e outras situações durante algumas décadas. Mais recentemente, Funções de Controlo de Lyapunov (*Control Lyapunov Functions*) tem vindo a gerar interesse como método. Estas funções seguem uma aproximação oportunística para a optimização do sistema. Esta dissertação tem como objectivo comparar os diferentes métodos de Controlo de Formações, nomeadamente, o output gerado, tempo de computação e erro da simulação. Iremos focar-nos nas Funções de Lyapunov, MPC com horizonte = 1 e MPC com horizonte grande. MPC com horizonte = 1 foi escolhido para ter uma comparação apta com as Funções de Lyapunov, visto que estas não optimizam o sistema num horizonte, mas apenas num passo da iteração.

Palavras Chave

Funções de Controlo de Lyapunov; Funções de Controlo de Barreira; Model Predictive Control; Restrições Lineares; Controlo de Formações.

Contents

Acronyms

CBF	Control Barrier Function
CLF	Control Lyapunov Function
LQR	Linear-Quadratic Regulator
MPC	Model Predictive Control

1

Introduction

Contents

1.1 Motivation of the Dissertation

A formation control problem is defined by a multi-agent system which must fulfill some objective, while acting cooperatively and avoiding undesired effects such as collisions between agents. In the literature, a variety of approaches are taken depending on the individual agents specific sensing capability and level of communication between agents. In this document, multiple types of formation control will be studied, as well as the solutions to formation control, namely, the use of Control Lyapunov Functions and Control Barriers Functions, and MPC. Notably, the use of MPC is well studied, due to its wide adoption in industrial settings, where it provides stability through the optimisation. While MPC has many attractive qualities in many different areas of study, it also has its drawbacks. MPC optimisations are computationally-heavy for certain nonlinear systems. MPC can, however, also be naturally applied in formation control systems, where it has had a recent surge in interest for flight formations [1–3], where the optimisation can be used to minimise the error in formations and thus achieve a desired objective. However, these systems require strict, time-sensitive computation of trajectory in order to maintain system safety. This rise in usage can be explained due to the surge in high performance computation CPU boards, allowing the computationally-heavy MPC optimisation to achieve enough time efficiency to outweigh this disadvantage.

Recently, there has been a growing interest [4–6] in the use of Control Barrier Functions in combination with MPC in order to achieve a system that both guarantees the safety by using the Lyapunov Function's stability and benefits from MPC's favourable optimization qualities.

To this end, in this dissertation we aim to do a comparative work between the different methods in order to ascertain their unique advantages and disadvantages. Notably, we wish to focus on the Control Lyapunov Function method as compared to a MPC with a horizon = 1, in order to draw parallels between these options, and come to conclusions about their functionality, as well as whether or not the results achieved with both methods are meaningfully different. The different methods will be simulated and then their outputs will be measured, namely, the simulation time, error of the formation, distance travelled, and actuation of the system.

We will then draw conclusions based on the output we receive from our measurements.

An implementation of the formation control, as well as the videos of the simulations, are also provided for future study in our GitHub Repository.

1.2 Organization of the Document

This thesis is is organized as follows: In chapter 2, we will go over the methodology used for formation control. We will discuss the MPC and Lyapunov Functions method of formation control, which leads into chapter 3, where we will discuss what is going to be measured in the comparisons, and what the formations are. Chapter 4 will show the results of the different formations. It will also note findings that will be relevant to the comparisons, Chapter 5 will draw conclusions on the different methods, their differences and similarities.

2

Formation Control

Contents

2.1 Formation Control Background

We begin by categorising formation control into different types based on the type of sensed and controlled variables, in a similar fashion to what is presented in [7]. Consider the following N-agents as the formation in a continuous time system:

$$\begin{cases} \dot{x}_i = f_i(x_i, u_i), \\ y_i = g_i(x_1, ..., x_N), \quad i = 1, ..., N \\ z_i = h_i(x_i), \end{cases} \tag{2.1}$$

where the position of agent i is denoted as $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{p_i}$ denotes the actuation, $y_i \in \mathbb{R}^{q_i}$ denotes the measurement, and output is denoted by $z_i \in \mathbb{R}^r$. Further, $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{p_i} \to \mathbb{R}^{n_i}$, $g_i : \mathbb{R}^{n_1} \times ... \times \mathbb{R}^{n_N} \to \mathbb{R}^{q_i}$, and $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^r$. Let $z^* \in \mathbb{R}^{rN} \to \mathbb{R}^M$ be given, which can be a function of time, for dynamic formations. The desired formation for the agents is specified by the following constraint:

$$F(z) = F(z^*). \tag{2.2}$$

Equation (2.1) describes a continuous time system. The equation can be translated to a discrete time system:

$$\begin{cases} x_i^{(k+1)} = f_i(x_i^{(k)}, u_i^{(k)}), \\ y_i^{(k)} = g_i(x_1^{(k)}, ..., x_N^{(k)}), \quad i = 1, ..., N \\ z_i^{(k)} = h_i(x_i^{(k)}), \end{cases} \tag{2.3}$$

where $x_i^{(k)}$ specifies the position of agent i at the control instant k. It is of note that the functions f_i, g_i, h_i are not the same as the ones in the continuous formulation (2.1).

However, this document will use a slight abuse of notation to describe them in order to simplify the understanding between both systems. A physical formation control system that works in discrete time will have an underlying system in continuous time in order to manage the physical elements of the system. The discrete time system will provide an objective to the continuous time system which will then control the actions of the agent. With this formulation, we define the 3 main types of formation control:

2.1.1 Position-based control

Agent i senses their position x_i with respect to a global coordinate system. Each agent i actively controls x_i in order to achieve a global formation. To achieve a desired formation, each agent i requires only their individual desired position x_i^*, and each agent can then act independently to arrive at their desired position. Measurements y_i contain some absolute variables that are sensed with respect to a global coordinate system. The constraint (2.2) is given as:

$$F(z) := z = F(z^*). \tag{2.4}$$

where $z = [z_0 ... z_N]$. Agents actively control z_i. Communication is necessary in the event that collisions between agents are possible. In conclusion, this type of formation control requires complex sensors to actively sense the agent's position in regards to a global coordinate system, but also requires less communication between agents.

A solution for position-based formation control extended for double-integrator agents which requires inter-agent interaction is studied in [8]. The agents are modeled by $\ddot{x}_i = u_i$, with respect to a global coordinate system. The objective is for the agents to move to a desired position while keeping their formation's shape during the translation. The following second-order consensus protocol is proposed:

$$u_i = -\sum_{j=1}^{n} g_{ij} k_{ij} \left[(\xi_i - \xi_j) + \gamma \left(\zeta_i - \zeta_j \right) \right], \quad i \in N \tag{2.5}$$

.

where $g_{ij} = 1$ if i and j communicate and 0 otherwise, and k_{ij} is a scalar. The goal of the consensus protocol is to ensure $||\xi_i - \xi_j|| \to 0$ and $||\zeta_i - \zeta_j|| \to 0$ when $t \to \infty$. Specifically, if ξ_i is the position of agent i, and ζ_i is the velocity of agent i, then (2.5) represents the acceleration of the ith agent. The consensus protocol is then analysed for time-invariant information exchange topologies. A decentralised position-based approach for multiple non-holonomic systems is studied in [9], which is then expanded to include uncertainty in [10]. [11] studies the trajectory tracking problem for robots in unicycle-type kinematic model with the assumption that each agent can sense its own position in regards to a global coordinate system. This model is then expanded in [12], which additionally proposes mutual coupling between robots. Additionally, the concept of virtual structure was introduced in [13], which leads to the concept of feedback control [14], which is a solution to coordinated motion control (formation control), based on a virtual structure, i.e. position based.

This will be the formation control type used for this dissertation's comparisons in the Experimental Results section.

2.1.2 Displacement-based control

Agents actively control displacements of their neighbouring agents $z_j - z_i$ to achieve the desired formation. Measurements y_i contain relative variables that are sensed with respect to a global coordinate system. As opposed to position-based control, they do not contain any absolute variables that need to be sensed with respect to the global coordinate system. The formation is specified by the desired displacements with respect to a global coordinate system $z_j - z_i$. The constraint (2.2) is given as:

$$F(z) := [...(z_j - z_i)^T...]^T = F(z^*) \tag{2.6}$$

for $i, j = 1, ..., N$. As a result, this type of formation control requires moderately complex sensors to measure the displacement of the agent and its neighbouring agents, as well as a moderate amount of communication between agents to communicate the displacement.

Decentralized displacement-based formation control has been proposed in [15]. The article proposes that each agent be assigned a control law that is the sum of two parts, a repulsive potential field, responsible for the agent's collision avoidance, and a attractive potential field, responsible for the agent's convergence on the objective. The law proposed is as follows

$$u_i = -\sum_{j \in N_i} \frac{\partial W_{ij}}{\partial x_i} - \sum_{j \in M_i} \frac{\partial V_{ij}}{\partial x_i} \tag{2.7}$$

where W_{ij} is a continuously differentiable function of the Euclidian distance between agents i and j, defined on $\beta_{ij} \in [0, d^2]$, where d is the desired displacement between agents and $W_{ij} \rightarrow \infty$ when $\beta_{ij} \rightarrow d^2$. This function is the attractive potential field. V_{ij} is also a continuously differentiable of the Euclidian distance between agents i and j. V_{ij} attains the maximum value when $\beta_{ij} \rightarrow 0$. The potential force can either be bounded, in which case the value of V_{ij} is finite, or unbounded, in which case $V_{ij} \rightarrow \infty$ when $\beta_{ij} \rightarrow 0$. This function serves as the repulsive potential field. It is of note that x_i is not a measure of the agent's actual position, given that agents sense the displacement between their neighbours and not their positions, however the agent's local coordinate system is aligned with the global coordinate system.

The control law's stability is then studied with the use of a Lyapunov function and proven as stable, and then the system is applied to nonholonomic kinematic unicycle-type agents as well as in cases of dynamic graphs, i.e., edges being formed and broken in time.

A displacement-based approach to formation control of robots that is robust to position measurement errors is described in [16]. The displacement based formation of "cyclic pursuit", that is, to have an agent i follow agent $i + 1$ is studied with wheeled vehicles in [17]. Different methods of displacement-based control with simple agents are studied in [18], culminating in the article on displacement-based formation control detailing the necessary conditions for formation control on graphs with wheeled vehicles found in [19].

2.1.3 Distance-based control

Inter-agent distances $||z_j - z_i||$ are actively controlled to achieve the desired formation, which is given by the desired inter-agent distances. Individual agents are assumed to be able to sense relative positions of their neighbouring agents with respect to their own local coordinate systems. The orientation of their local coordinate systems are not necessarily aligned with each other. The constraint (2.2) is given as:

$$F(z) := [...||z_j - z_i||...]^T = F(z^*) \tag{2.8}$$

for $i, j = 1, ..., N$. This type of formation control requires relatively simple sensors, given that inter-agent distances can be measured and controlled without considering a global coordinate system's origin or orientation. However, inter-agent communication between neighbours is relatively high in order to achieve the desired formation.

The stability of distance-based control in a 2-dimensional system is studied in [20]. It is shown that if the formation control law is a negative gradient, then it is provably correct when the formation graph is a tree. The proposed control law is of the form

$$u_i = -\sum_{j \in N_i} \frac{\partial \gamma\left(\beta_{ij}(x)\right)}{\partial x_i} = -\sum_{j \in N_i} 2\rho_{ij}\left(x_i - x_j\right), i \in \mathcal{N} \tag{2.9}$$

where N_i is the set of the neighbours for agent i, $\beta_{ij}(x)$ is the Euclidean distance between agent i and agent j. γ is the formation potential, a function of the distance between i and j, $\gamma = \gamma(\beta_{ij})$, that is continuously differentiable, $\gamma(d_{ij}^2)$ and $\gamma(\beta_{ij}) > 0$ for all $\beta_{ij} \neq d_{ij}^2$. d_{ij} is the desired inter-agent distance. The system's stability is then examined by using a Lyapunov Function, and it is then proven that the system reaches a static configuration, i.e., it is stable.

It is of note that certain types of formation control do not necessarily fit into these 3 categories, for example, a flocking formation based on simple interaction among individual agents [21–23], estimation-based control formations [24, 25], angle-based control [26] and behaviour-based control [27, 28]. The results reached in [20] were also applied to flocking formations based on double-integrator agents.

2.2 Linear Quadratic Regulator

The Linear-Quadratic Regulator (LQR) solution was achieved with Matlab's $dlqr$ function. The error is calculated and applied to the system in order to optimise it. This is a simple form of formation control, applying the error directly to the displacement of the agent's position to their optimal position.

2.3 Control Lyapunov Functions

2.3.1 Lyapunov Functions

Lyapunov stability is defined using the following formulation in continuous time:

$$\dot{x} = f(x) \tag{2.10}$$

The equilibrium point $x = 0$ is

- stable if, for each $\epsilon > 0$, there is $\delta = \delta(e) > 0$ such that

$$||x(0)|| < \delta \Rightarrow ||x(t)|| < \epsilon, \forall t \geq 0 \tag{2.11}$$

- unstable, otherwise
- asymptotically stable if it is stable and a δ can be chosen such that

$$||x(0)|| < \delta \Rightarrow \lim_{x \to \infty} x(t) = 0 \tag{2.12}$$

Applying this to Lyapunov's Stability Theorem:
Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that

- $V(0) = 0, V(x) > 0, \quad \forall x \in D \backslash \{0\}$
- $\dot{V}(x) \leq 0, \quad \forall x \in D$

Then, $x = 0$ is stable. Moreover, if

$$\dot{V}(x) < 0, \forall x \in D \backslash \{0\} \tag{2.13}$$

then $x = 0$ is also asymptotically stable.

2.3.2 Control Lyapunov Functions

The application of Lyapunov Functions to formation control problems stems from the use of Control Lyapunov Functions (CLF) and Control Barrier Functions (CBF). These use the attractive qualities of Lyapunov Functions, namely, Lyapunov Stability, to formation control problems. In the following section, we define these functions.

A Control Lyapunov Function (CLF) is a positive definite function $V(x)$ for a system $\dot{x} = p(x) + j(x)u$ that satisfies, for every $x \neq 0$:

$$\inf_{u \in U} [L_p V(x) + L_j V(x) u] < 0 \tag{2.14}$$

where U is the set of all possible actuations. The sum $p(x) + j(x)u$ can be translated into the function $f_i(x_i, u_i)$ from (2.1), with agent number i being omitted for reading clarity. It is of note that in this formulation, the expression for the agent's state $p(x)$ is separated from the expression for the agent's actuation $j(x)u$. This is useful due to the simplified computation applied to the optimisation problem that will be presented in this section.

The CLF can then be used in an optimisation-based approach for a formation control system by optimising the set of stabilising controllers $K_V(x)$:

$$K_V(x) = \{u \in U : L_pV(x) + L_jV(x)u \leq -\sigma(x)\} \tag{2.15}$$

where $\sigma(x) > 0$ is the desired rate at which the CLF should decay. which leads to the optimisation:

$$\begin{array}{cl} \underset{u \in U}{\min} & \frac{1}{2}||u||^2 \\ \text{s.t.} & L_pV(x) + L_jV(x)u \leq -\sigma(x) \end{array}$$

at every point the smallest input u is selected, which ensures the CLF decays at the specified rate $\sigma(x)$. As mentioned before, $L_pV(x)$ and $L_jV(x)u$ are separable, therefore the computation required to complete this optimisation problem is lessened due to the problem being simplified into 2 parts. The computation can then be done as such:

$$\begin{array}{cl} \min & \tilde{u} \\ \text{s.t.} & L_p(x) + \tilde{u} \end{array}$$

where $u = g(x)^{-1}\tilde{u}$. This allows the computation to remain convex due to $p(x)$ and $g(x)$ being constant values when x is fixed. This property is especially useful on systems where $p(x)$ is non-convex. Following this approach, and generalising to a formation control problem with agent $i \in N$, i can approach the desired position x_i by optimising u_i in accordance to the equation:

$$\begin{array}{cl} \underset{u_i \in U_i}{\min} & \frac{1}{2}||u_i||^2 \\ \text{s.t.} & L_pV(x_i) + L_jV(x_i)u_i \leq -\sigma(x_i) \end{array}$$

An example of this application is documented in [29], where the Control Lyapunov Functions are studied in regards to their use in formation control of a class of robots.

Control Lyapunov Functions serve as a method for agents in a formation to reach their desired position through solving an optimisation problem. However, collision avoidance is ignored if they are used exclusively as a system of formation control. Therefore, there are applications of Lyapunov Functions that serve an opposite role to Control Lyapunov Functions, that is, to avoid a position or set of positions, which are the Control Barrier Functions.

2.3.3 Control Barrier Functions

Control Barrier Functions (CLB) serve as systems applied to formation control for avoidance of undesirable outcomes, such as collisions between agents or with the environment. Specifically, we consider the following sets:

$$C = x \in \mathbb{R}^n : c(x) \geq 0$$
$$\partial C = x \in \mathbb{R}^n : c(x) = 0$$
$$O = x \in \mathbb{R}^n : c(x) < 0$$

Set C denotes safe positions for agents. Set ∂C denotes the boundary between safe and unsafe positions. Set O denotes unsafe positions, such as those that may cause collisions between agents. The use of CBFs serves to ensure that $c(x_i) \notin O, \forall i \in N$. Therefore, the function $c(x)$ is a CBF if there exists a locally Lipschitz extended class K_∞ function α such that:

$$\sup_{u \in \mathbb{R}^m} [L_p c(x) + L_j c(x) u] \geq -\alpha(h(x)) \tag{2.19}$$

It is of note that $h(x)$ is only allowed to decrease in the interior of the safe set $int(C)$, but not on the boundary ∂C, that is, C is forward invariant. With this, a set of safe controllers can be defined:

$$K_{CBF}(x) = \{u \in \mathbb{R}^m : L_p h(x) + L_j h(x) u + \alpha(h(x)) \geq 0\} \tag{2.20}$$

which ensures $c(x) \in C$. Combining the Control Lyapunov Functions and Control Lyapunov Barriers, we obtain a control system in the following Quadratic Programming (QP) formulation:

$$\begin{array}{cl} \displaystyle\min_{(u,\delta) \in \mathbb{R}^{m+1}} & \dfrac{1}{2}\|u\|^2 + \dfrac{1}{2}\kappa\delta^2 \\ \text{s.t.} & L_p V(x) + L_j V(x) u + \gamma(V(x)) \leq \delta \quad \textbf{(CLF)}, \\ & L_p h(x) + L_j h(x) u + \alpha(h(x)) \geq 0 \quad \textbf{(CBF)} \end{array}$$

the CBF constraint guarantees that $u^* \in K_{CBF}(x)$ keeps the system trajectories invariant with respect to the safe set $int(C)$. The relaxation variable δ in the CLF constraints softens the stabilisation objective, maintaining the necessary feasibility of the QP.

Control Lyapunov Functions are studied as a form of formation control in [29], which is then exemplified by two robots carrying a beam controlled by CLF. In [30], Control Lyapunov Functions are used along with a leader-follower dynamic of formation control with car-like robots. The use of Control Lyapunov Function in QP is well studied: [31], as well as the use of Control Barrier Functions: [32, 33]. Additionally, a robust solution using Control Lyapunov Function and Control Lyapunov Barrier based Quadratic Programs (CLF - CLB - QP) with the addition of measurement noise have also been studied [34].

2.4 Model Predictive Control

Model Predictive Control (MPC) can easily be applied to formation control problems, as each instance can be considered as an optimisation problem which is then applied to an optimiser. Centralised solutions to the optimisation problem are well studied, due to the control objectives and operating constraints being represented explicitly in the optimisation problem solved at each control instant. The main challenges stem from their application in decentralised systems, with distributed computation, noisy data and occasionally sensing data only in specified intervals.

Some examples of previous work on distributed MPC are reported in [35–37]. Notably, [36], [37] both report examples in physical systems, namely decentralised water management systems, which receive noisy data. Considering the linear discrete time multi-agent dynamic system

$$x_0^{(t+1)} = Ax_0^{(t)} \quad x_i^{(t+1)} = Ax_i^{(t)} + Bu_i^{(t)} \quad \forall i \in N, t \geq 0, \tag{2.22}$$

The application of MPC to the formation control problem is straightforward. According to [38], the MPC control problem can be characterised as follows. At decision instant k, the controller samples the state of the system $x(k)$. The following optimisation problem is then solved to find the control action.

$$\begin{aligned} \min_{X(k),U(k)} \quad & J(X(k),U(k)) \\ \text{s.t.} \quad & x^{(k+i+1|k)} = f(x^{(k+i|k)}, u^{(k+i|k)}) \quad (i = 0, ..., K-1), \\ & G(X(k),U(k)) \leq 0, \\ & x^{(k|k)} = x^{(k)} \end{aligned} \tag{2.23}$$

where

$$X(k) = \{x^{(k+1|k)}, ..., x^{(k+K|k)}\}$$
$$U(k) = \{u^{(k|k)}, ..., u^{(k+K-1|k)}\}$$

for control horizon K. The same notation for x and u is used for equation (2.3), with the addition of the prediction notation. That is, $x^{(k+1|k)}$ is the predicted state (or position) of an agent at control instant $k+1$, based on the information at instant k.

The standard MPC formulation can then be summarised as a series of static optimisation problems in order to obtain a distributed formulation $\{SP_k|k = 0, 1, 2, ...\}$ of the form:

$$\begin{aligned} SP_k : \min_{S} \quad & J(S) \\ \text{s.t.} \quad & G(S) \leq 0, \\ & H(S) = 0 \end{aligned} \tag{2.24}$$

where S is the vector of decision variables, including state and input variables over the prediction horizon. Distributed MPC consists of the decomposition of SP_k into a set of P subproblems, $\{SP_{ki}|i = 1, 2, ..., P\}$. Subproblems are then assigned to different agents in order to ensure that:

- Each problem is much smaller than the whole, that is, each SP_{ki} has far fewer decision variables than SP_k.
- Each problem is coupled to only a few other subproblems, meaning SP_{ki} shares variables with only a few other subproblems.

It is proven in [38] that the distributed MPC problems compose a solution to the overall MPC problem if they follow the proposed theorem, which is simplified here:

If, for all i:

- $\sum J_i = J$ when J is a scalar, or $\sum J_i = W^T J$ when J is a multiobjective vector, where W is a vector of nonnegative weights;
- J_i and G_i are convex;
- H_i is linear;
- The agents within each neighbourhood work sequentially;
- The equality constaints can be relaxed without emptying the feasible region of SP_k.
- J_i is bounded from below in the feasible region;
- The starting point is in the interior of the feasible region;
- Each agent cooperates with its neighbours in that it broadcasts its latest iteration to these neighbours;
- Each agent uses the same interior-point method (barrier method) with the same Lagrange multipliers to generate its iterations.

Then, in the case where J is a single objective, SP_k has a unique solution and $\{\cup s^l_{ki} | l = 0, 1, 2, ...\}$ will converge to this solution. If J is a multiobjective vector, $\{\cup s^l_{ki} | l = 0, 1, 2, ...\}$ will converge to a point on the Pareto surface of SP_k.

With this result, we can conclude that there are computational advantages to solving a set of subproblems by separating them through a set of agents working within the same structure. Notably, we can also conclude that distributed MPC is feasible and viable as a method for formation control if the formation follows the theorem's requirements. Similar results have also been achieved in [39], namely the advantage of cooperative control strategies over non-cooperative strategies in system performance as well as in [40], which proposes a system that incorporates centralised control performance while operating in a decentralised manner. Other examples of formation control using MPC are [41, 42], applied to vehicles and robotics, respectively. An example of a problem involving spacecraft formations with limited communication is studied at [43]. Surveys on MPC are common [38], [44] due to their wide use in optimisation problems in industrial settings where high performance is required.

2.4.1 Collision Avoidance

Collision Avoidance in this system is achieved by the method detailed in [45]. In section 4.5.3 of the book, a method to achieve linear constraints from non-linear obstacles is presented. The method relies on creating constraints that define the planes tangent to the obstacle.

A simplified version of the proposed algorithm is used in this thesis:

Algorithm 2.1: Obstacle Avoidance Strategy

```
begin
    Solve optimisation problem without obstacle constraints
    Determine planes tangent to obstacle facing each point in the trajectory
    Solve optimisation problem with tangent planes as linear constraints
```

- **Solve optimisation problem without obstacle constraints**, saving the agent's positions in each iteration.
- **Determine planes tangent to obstacle facing each point in the trajectory**, using the saved positions. This requires a calculation to discover the tangent line which passes through the unsafe set's limits.
- **Solve optimisation problem with tangent planes as linear constraints**, which leads to a system where safety is assured.

This strategy incurs a time penalty, since it requires the problem to be solved without obstacle constraints, and then with obstacle constraints. These constraints slow down computation considerably, as is shown in the later results section.

3

Experimental Measurements

Contents

In the following chapter, we will discuss how the simulations were executed, as well as what measurements were taken to measure the performance of the various methods.

3.1 Obtained Measurements

The following measures were taken in runtime to assess the viability of each method of formation control

Time: The time of execution will be measured.

Formation Error: The current difference between the agent's positions and the desired formation will be measured.

Distance Travelled: The agent's total travel distance will be measured.

Energy Spent: The actuation will be measured at each iteration. This is meant as an approximation to battery power spent.

3.2 Formations

The regularly used formations place the agents in a virtual circle around a point. The point is then moved, causing the formation to follow it.

Notably, each agent can act independently. The formation's definition supplies each agent with their optimal position, meaning no communication between agents is required. The optimisation is collective and computed once per timestep. Naturally, it would be possible for each agent to individually compute their optimal actuation instead.

Functions to create these formations and then supply them to each solver are available in the repository for future work.

The formations tested either move in a straight line or move in a "sine wave" pattern. Certain formations will also shift the agent's positions at chosen intervals. The velocity of the formation will also be variable.

The trajectory that the formation took will be illustrated along with the final results. The color of each agent's trajectory will be different to better distinguish them.

3.3 Obstacles

For the Model Predictive Control (MPC) and Control Lyapunov Function (CLF) methods, the simulations will have a varying number of obstacles to increase complexity. The number of obstacles chosen was 0, 1 and 5 to differentiate complexity. Each will then be evaluated according to the measurements outlined in Section 3.1. The obstacles are circles of varying radii, which are placed in order to partially or completely block the formation's path.

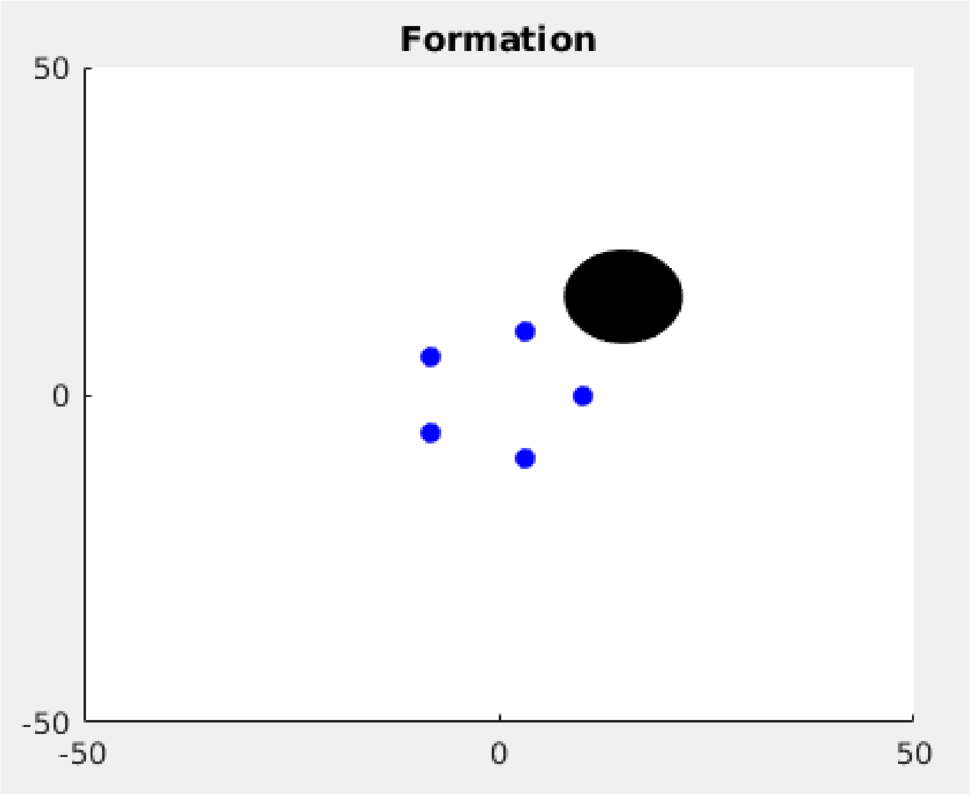


Figure 3.1: A formation with 5 agents and 1 obstacle.

The method to avoid obstacles is dependent on the method used for the simulation. Both Lyapunov Functions and MPC have a specific collision avoidance practice. Notably, LQR will not be simulated with any obstacles.

4

Simulation Results

Contents

The following sections will describe the simulation results. The sections are divided into different simulations, where the formation has no obstacles in its past, one obstacle, and many obstacles.

4.1 No Obstacles

In the following sections, the simulation results for the formations will be shown.

4.1.1 Linear-Quadratic Regulator

4.1.1.A Straight Line Movement

We begin with the most basic simulation, a formation that moves in a straight line, with no obstacles.

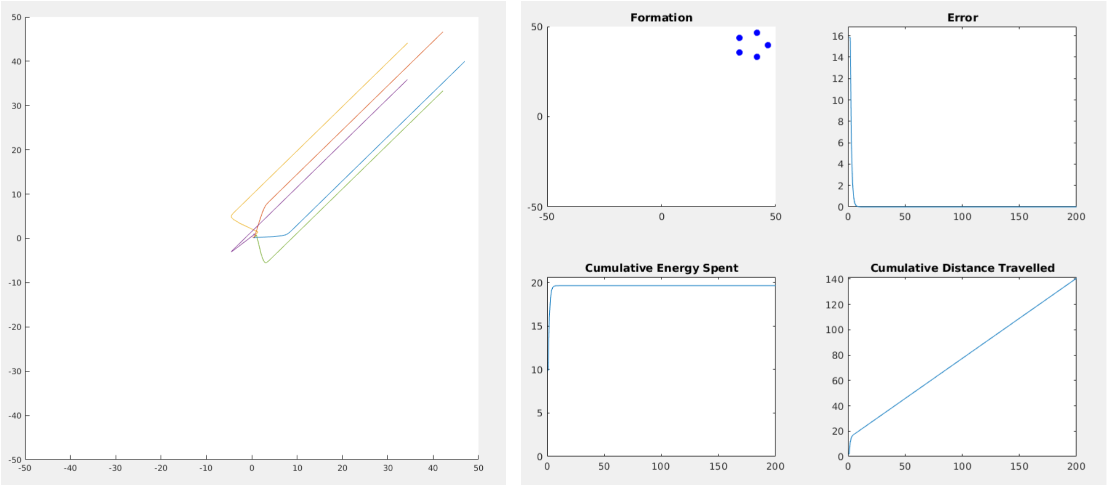


Figure 4.1: The final results of the LQR simulation moving in a straight line.

Used Formation: `straight_5_200_8.mat`

Elapsed Time: $0.7994s$

Units Travelled: 140.41

Energy Spent: 20.20

The behaviour worth noting is the initial actuation, where the agents move towards their optimal positions in the formation. Afterwards, the system becomes stable, and no actuation is necessary, since the agent's velocity remains in accordance with the formation's movement.

4.1.1.B Wave Movement

In this simulation, the formation follows a sine wave movement. This adds more complexity, since it requires actuation to maintain the formation, rather than it being stable after a certain point.

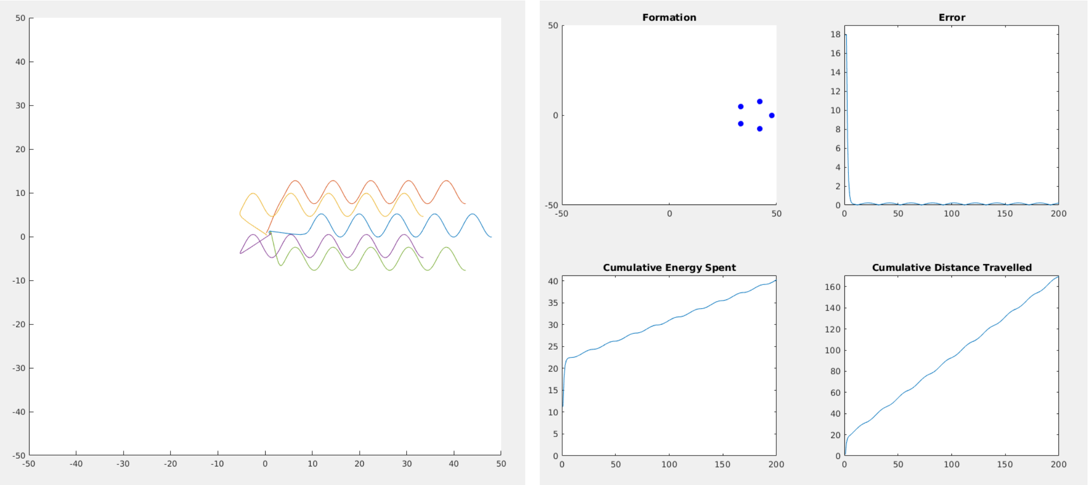


Figure 4.2: The final results of the LQR simulation moving in a wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $0.7604s$

Units Travelled: 169.70

Energy Spent: 40.30

It is worth noting that the elapsed time is smaller than in 4.1.1.A. This difference can be explained by factors external to the computation itself, as it is dependent on the system it is run on.

4.1.1.C Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

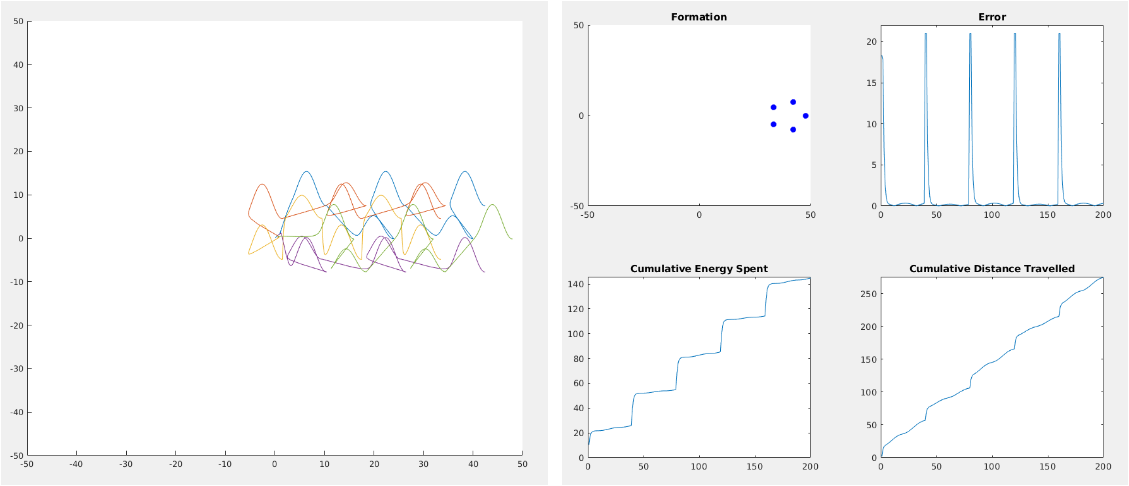


Figure 4.3: The final results of the LQR simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $0.7576s$

Units Travelled: 274.64

Energy Spent: 144.82

The spikes of energy spent every 40 iterations are easy to notice. The formation errors are quickly resolved with the actuation.

4.1.2 Control Lyapunov Functions

The next section will have the same experiments using the CLF method.

4.1.2.A Straight Line Movement

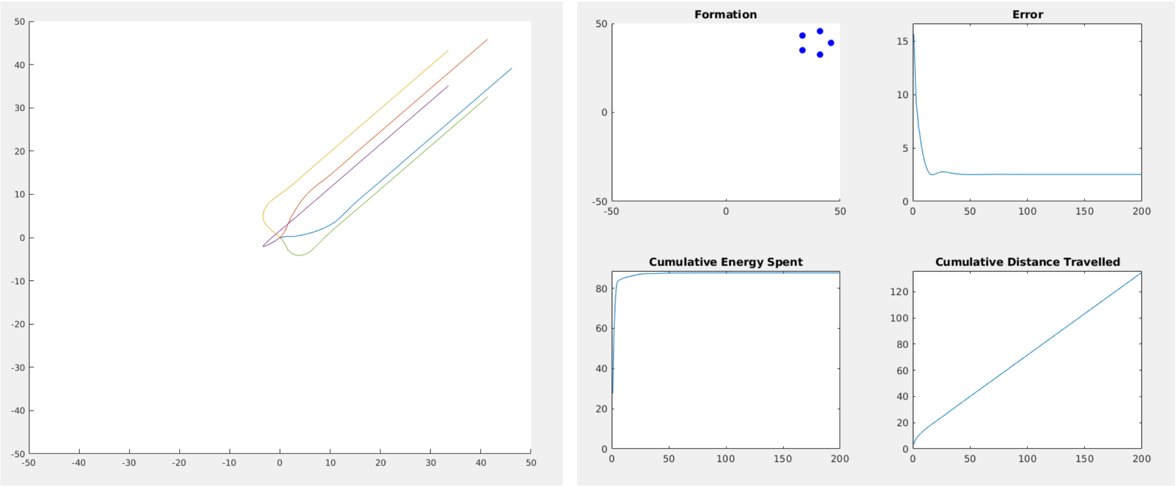


Figure 4.4: The final results of the CLF simulation moving in a simulation moving in a straight line.

Used Formation: `straight_5_200_8.mat`

Elapsed Time: $5.5138s$

Units Travelled: 134.91

Energy Spent: 87.62

The minimum error of the formation is noticeable in 4.4. This minimum error is caused by the CLF having a different system dynamic to the LQR that was previously discussed, as well as the CLF having found a local minimum and not being able to find a more stable actuation. This minimum error will remain throughout the next simulations as well. The much higher energy total spent should be noted in this case, compared to 4.1.1.A, being 4.33 times higher.

4.1.2.B Wave Movement

In this simulation, the formation follows a sine wave movement.

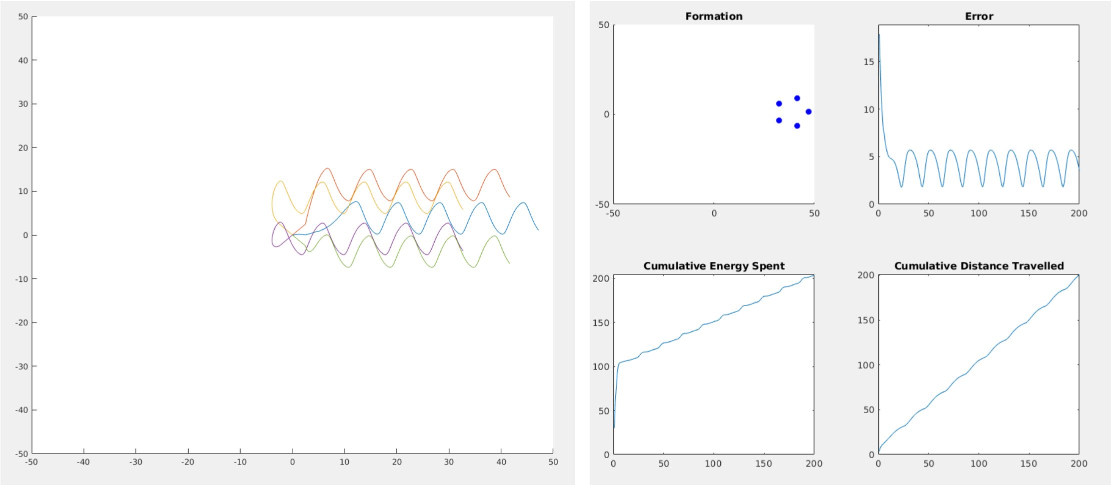


Figure 4.5: The final results of the CLF simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: 6.0689

Units Travelled: 200.02

Energy Spent: 203.60

The trajectory not being a perfect wave should be noted. The peaks and valleys of the trajectory are noticeably not uniform. The energy spent remains much higher than the total in the LQR system (5.05 times higher).

4.1.2.C Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

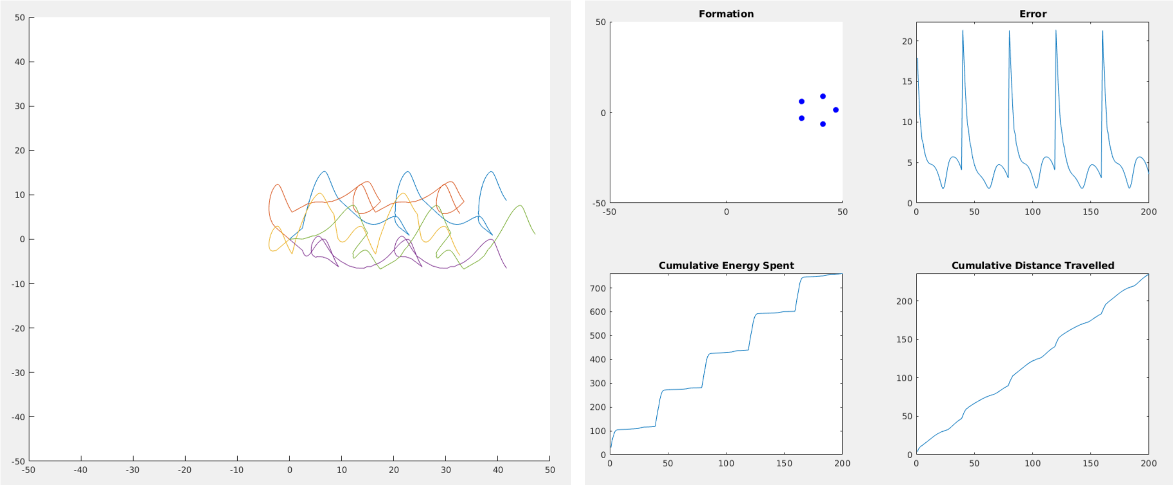


Figure 4.6: The final results of the CLF simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $6.6582s$

Units Travelled: 234.68

Energy Spent: 759.97

The spikes of energy spent every 40 iterations are easy to notice. It is worth noting that the energy spent on this version is 5.25 times higher than the LQR version of the same formation. The somewhat erratic movement at the shift is also to be noted.

4.1.3 Model Predictive Control with horizon = 1

The following sections will display the results of the simulation for the MPC method. This

4.1.3.A Straight Line Movement

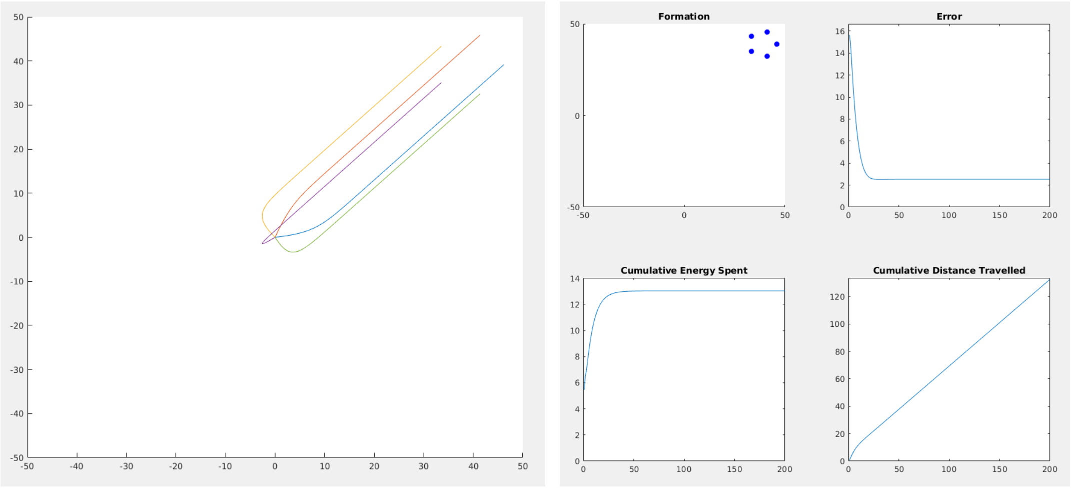


Figure 4.7: The final results of the MPC simulation moving in a simulation moving in a straight line.

Used Formation: `straight_5_200_8.mat`

Elapsed Time: $9.9809s$

Units Travelled: 132.60

Energy Spent: 13.03

The same minimum error can be seen as in 4.1.2.A. This is due to both methods being vulnerable to finding local minimums and not acting towards a global minimum. However, it should be noted that the energy spent by this system is lower than the LQR version, being 0.64 times the value of energy spent.

4.1.3.B Wave Movement

In this simulation, the formation follows a sine wave movement.

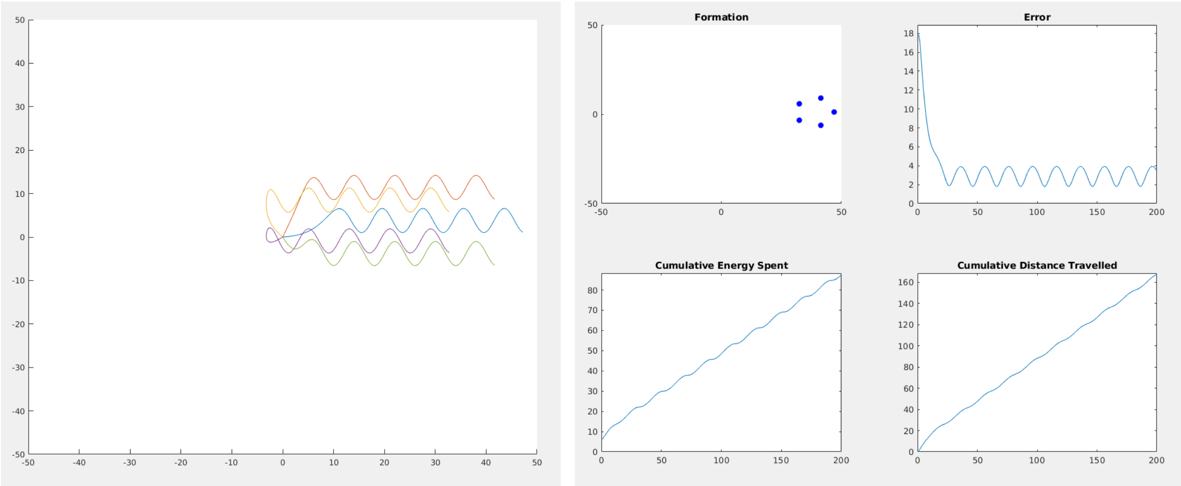


Figure 4.8: The final results of the MPC simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $11.9757s$

Units Travelled: 167.58

Energy Spent: 87.32

4.1.3.C Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

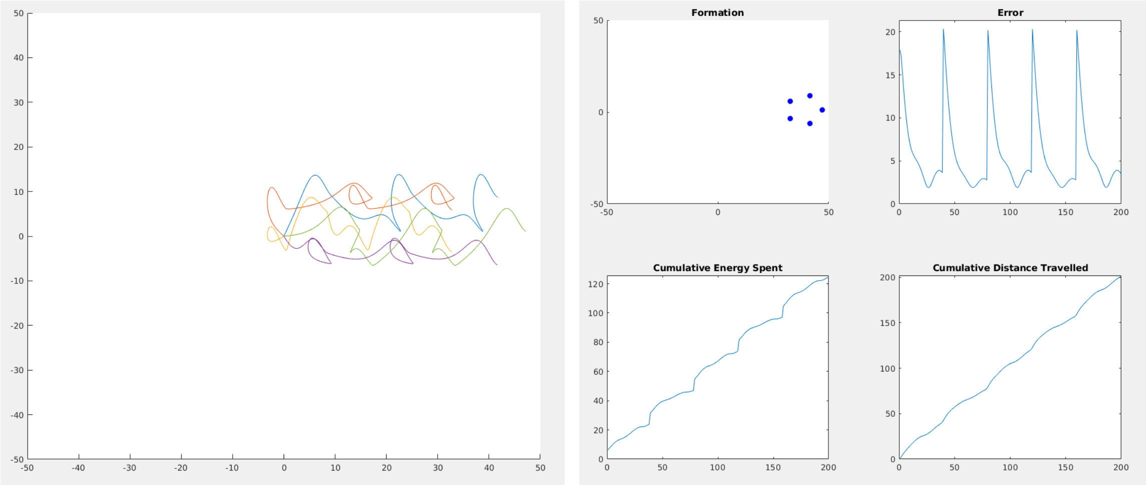


Figure 4.9: The final results of the MPC simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $9.5033s$

Units Travelled: 201.24

Energy Spent: 124.74

The movement as compared to 4.1.2.B is noticeably smoother. However, the trajectory maintains a minimum error.

4.1.4 MPC with large horizon

The following section will display the results when using the MPC method of formation control with a high horizon ($h = 10$). This is the usual way to use MPC, as it allows for the optimization to "look ahead" and avoid local minimums, as well as minimize more accurately.

4.1.4.A Straight Line Movement

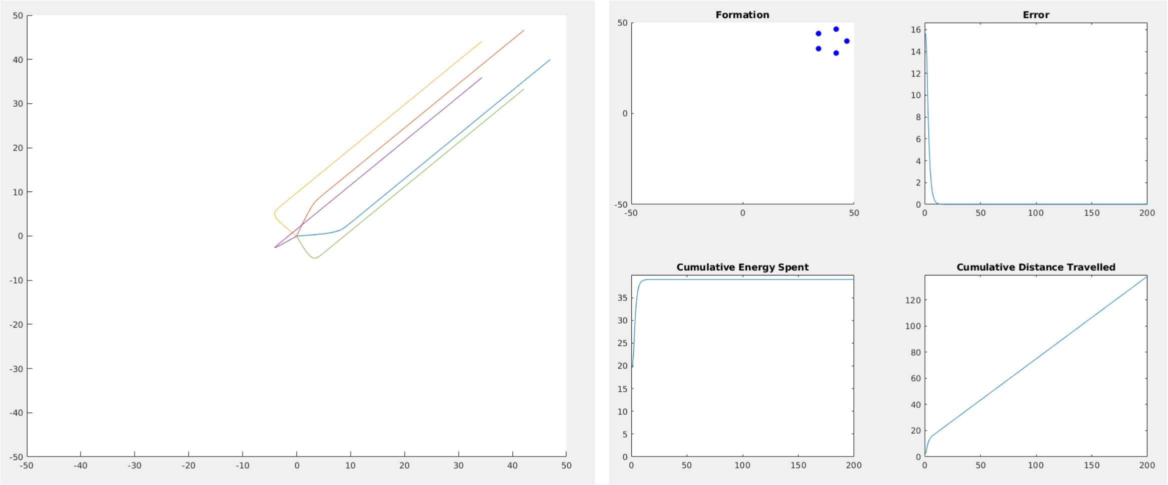


Figure 4.10: The final results of the MPC simulation moving in a simulation moving in a straight line.

Used Formation: `straight_5_200_8.mat`

Elapsed Time: $12.0441s$

Units Travelled: 138.19

Energy Spent: 38.98

This simulation differs from the previous ones, we can notice that instead of a non-zero error being found as a stable local minimum, the "look ahead" allows the simulation to virtually remove this error and achieve a very stable formation. This shows the obvious benefits of using a high horizon in these simulations.

4.1.4.B Wave Movement

In this simulation, the formation follows a sine wave movement.

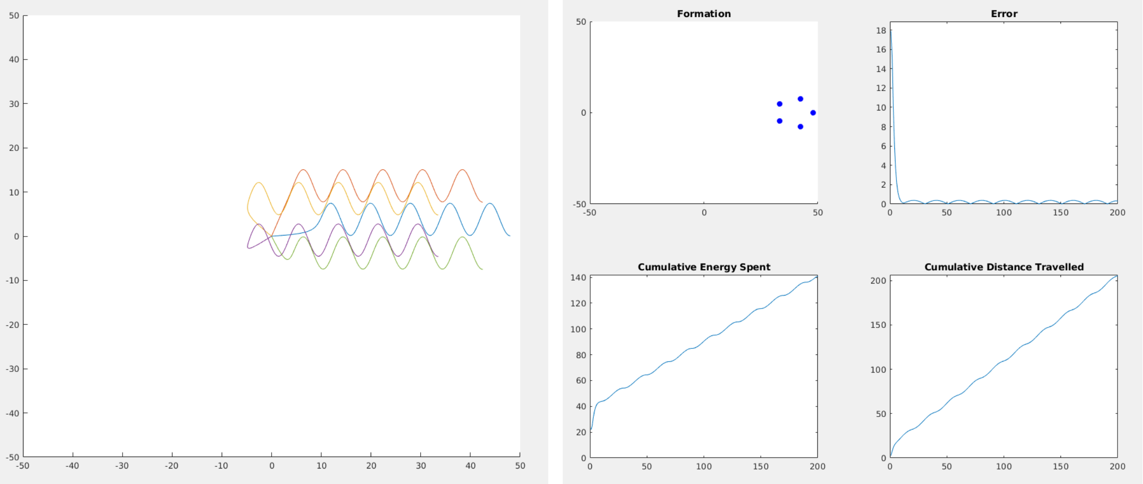


Figure 4.11: The final results of the MPC simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $12.1193s$

Units Travelled: 205.49

Energy Spent: 140.99

The benefits of using a high horizon are also noticeable here. The error remains minimal, with a small increase at each "peak" of the wave trajectory. The trajectory also maintains a smooth curve following the sine wave.

4.1.4.C Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

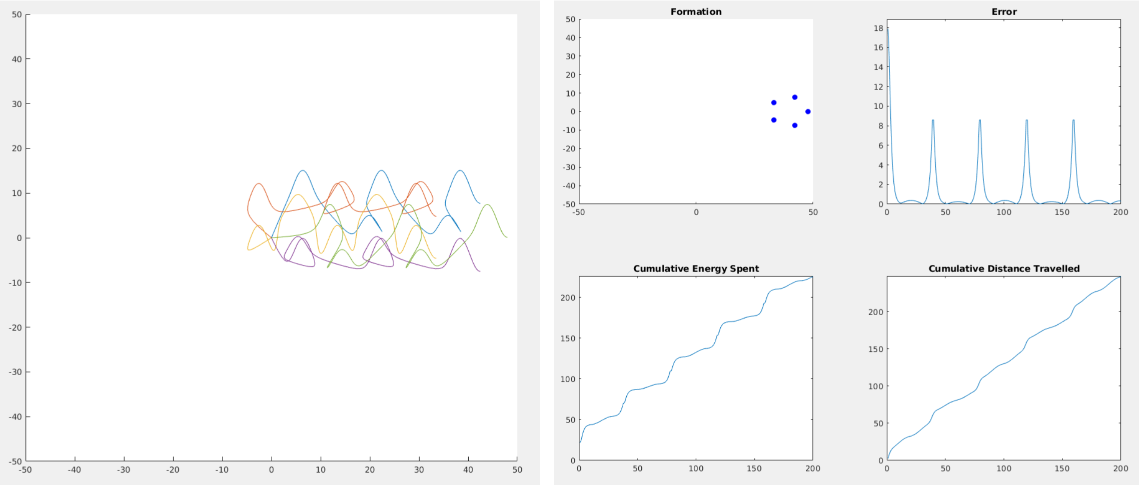


Figure 4.12: The final results of the MPC simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $12.3726s$

Units Travelled: 246.90

Energy Spent: 225.19

This simulation illustrates another benefit of the high horizon being used. Even though the formation requires a sudden movement to shift the agent's position, the simulation accounts for this and moves the agents before the actual formation changes, allowing for a much softer curve in the error graph. All other simulations had a high peak when this shift happened.

4.1.5 Results

The following are charts detailing the measurements for the computations in the previous sections.

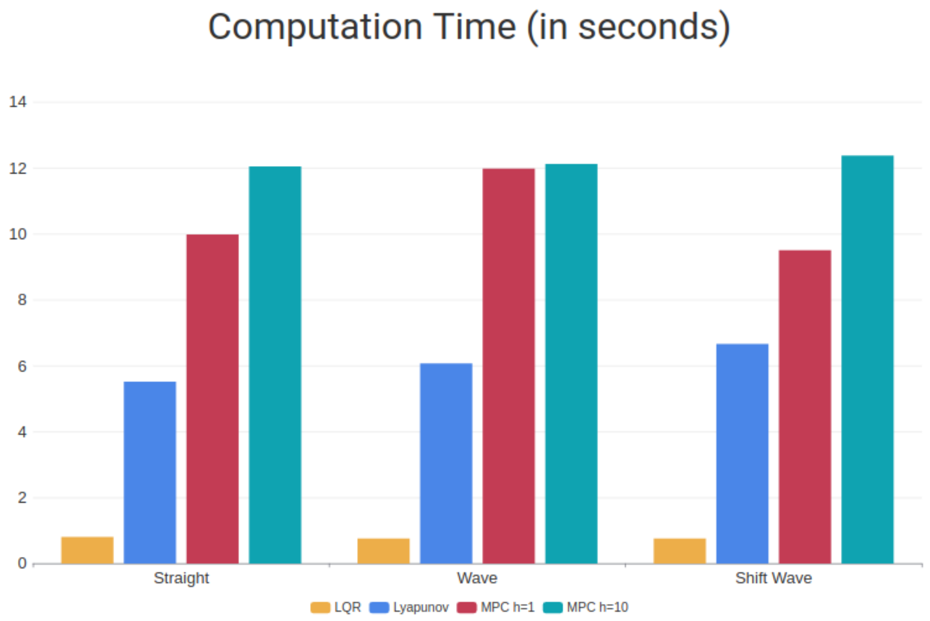


Figure 4.13: Chart of time for each computation

The LQR method, used as a baseline, clearly shows a much faster computation time. It should be noted that despite having a much larger horizon, the MPC with $h = 10$ does not run much slower than the $h = 1$ version. The greedy approach from the Lyapunov Function method also shows a clear advantage for execution time compared to the others.

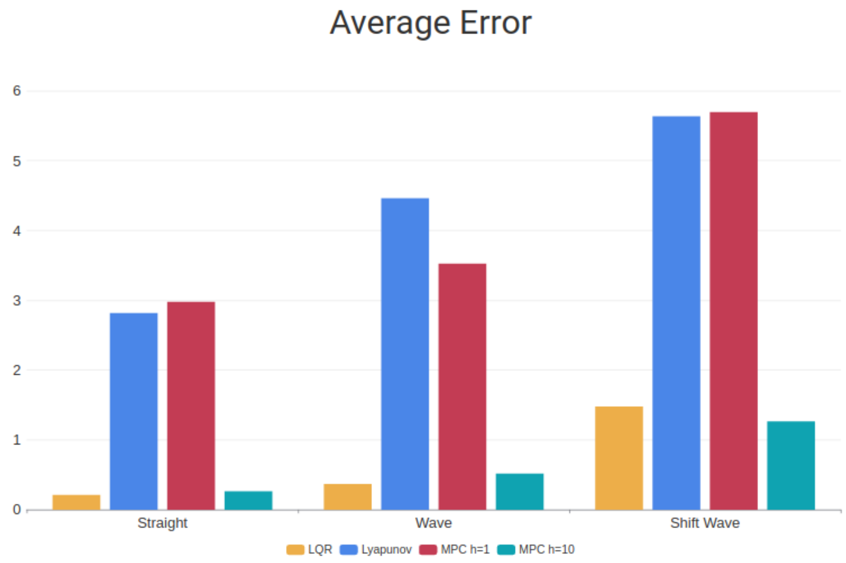


Figure 4.14: Chart of error for each computation

The noticeable data from this chart shows the advantage of using MPC with a high horizon. The average error is much lower than the low horizon version and the Lyapunov Function method.

It should also be noted how close the average error is between the Lyapunov method and the MPC with $h = 1$ one is. This can be explained by both methods arriving at the same stable local minimum, making their execution very similar.

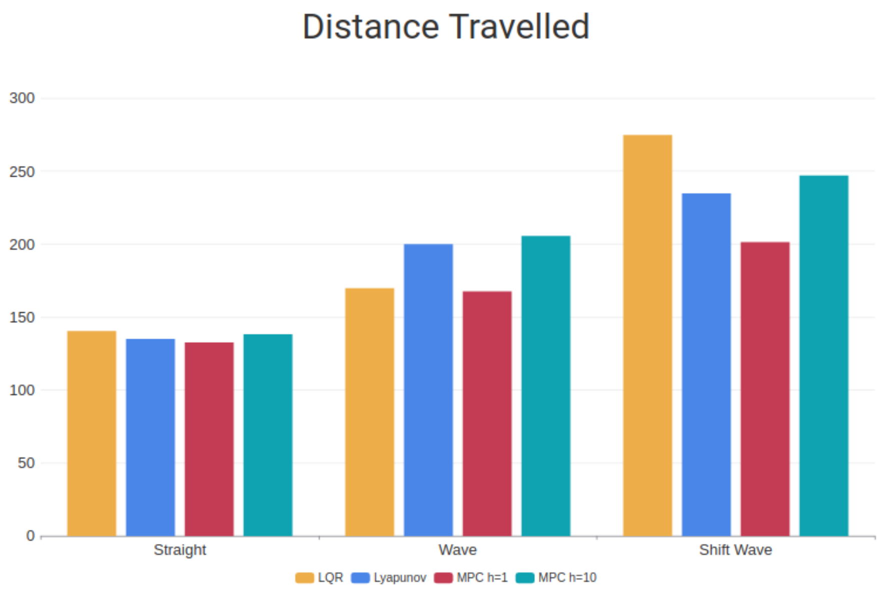


Figure 4.15: Chart of distance travelled for each computation

The distance travelled between the different methods is not very varied, but it should be noted how the distance travelled with MPC with $h = 1$ is lower on each formation. Both the Lyapunov and the MPC with low horizon methods arrive at a local minimum stable point. This point is notably lagging behind the formation, especially notable on the straight trajectory. However, there is a notable overshoot of the initial trajectory with the Lyapunov method, while the MPC with low horizon method smoothly arrives at this point without overshooting. This can explain the extra distance from the Lyapunov method.

The extra distance travelled from the MPC with $h = 10$ is simple to explain, it does not lag behind the formation, and therefore it travels further than the others.

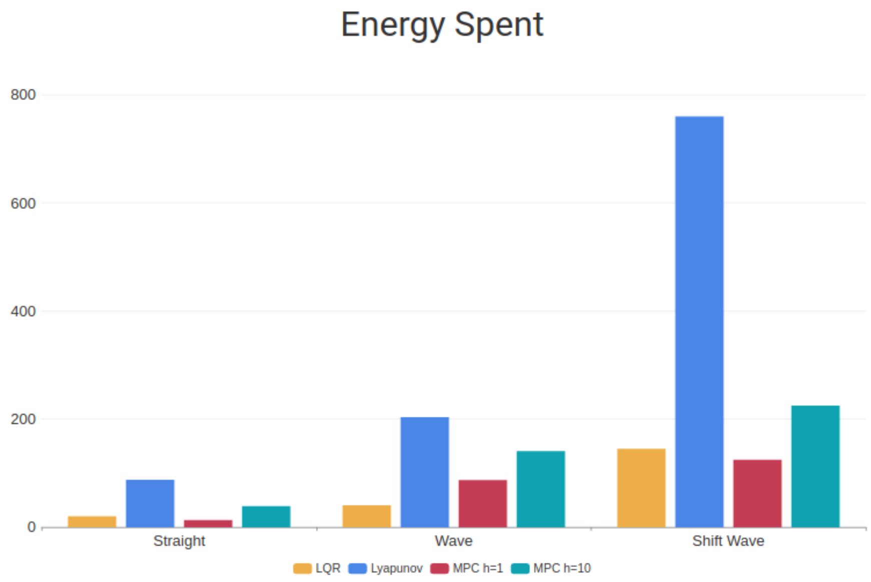


Figure 4.16: Chart of distance travelled for each computation

It is easy to notice the much higher energy spent by the Lyapunov method. Despite both MPC with $h = 1$ and the Lyapunov method arriving at very close local minimums, there is a much higher energy usage by the Lyapunov method. The Lyapunov method promotes a high actuation with causes the agents to quickly arrive at their destination, and they notably overshoot their trajectory, which causes a high energy actuation to break and stabilize their positions.

4.2 1 Obstacle

The following chapter will illustrate simulations for both the wave and shifting wave formations, with an added obstacle in the formation's path. Notably, the unconstrained LQR is omitted from this chapter. The obstacle used is a circle, centered at position $(15, 15)$ with a radius $r = 7$.

4.2.1 Control Lyapunov Functions

4.2.1.A Wave Movement

In this simulation, the formation follows a sine wave movement.

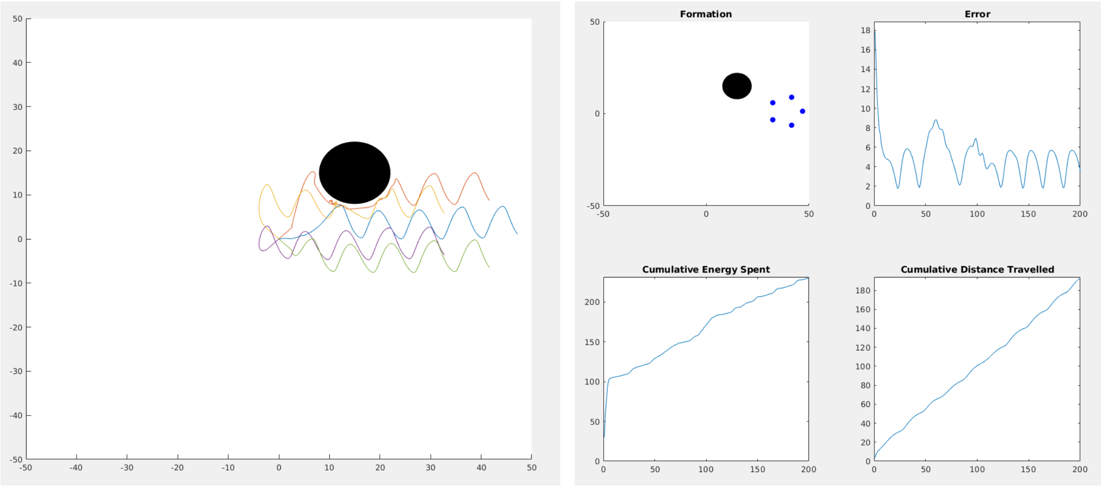


Figure 4.17: The final results of the CLF simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $8.6595s$

Units Travelled: 193.12

Energy Spent: 230.62

The behaviour of the Control Barrier Function (CBF) is noticeable in this simulation. While the safety of the agents is assured, this restriction causes some erratic movement at the edge of the safe set of positions.

4.2.1.B Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

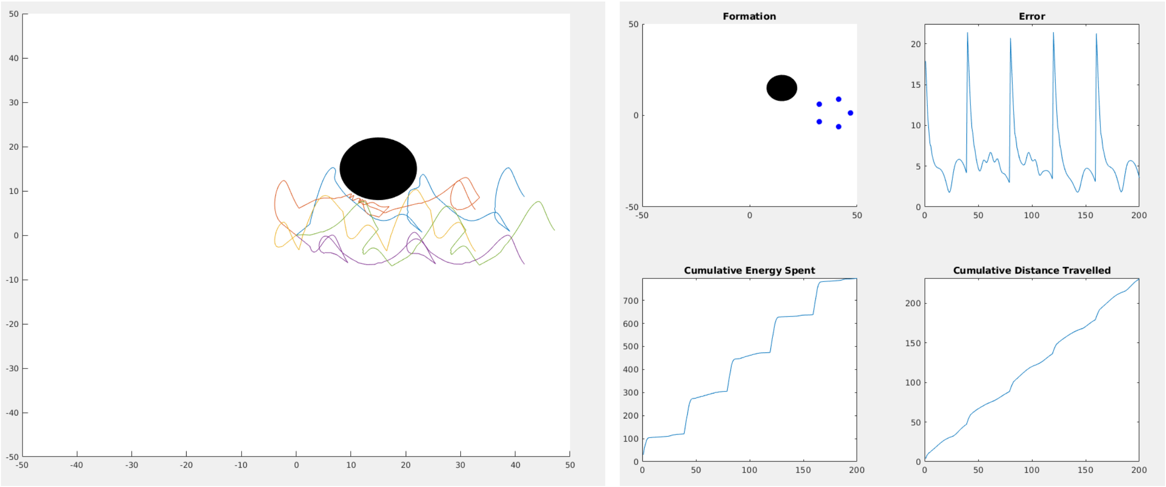


Figure 4.18: The final results of the CLF simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $8.5774s$

Units Travelled: 230.69

Energy Spent: 795.49

This simulation again illustrates the erratic movement at the edge of the safe set of positions.

4.2.2 MPC with horizon = 1

4.2.2.A Wave Movement

In this simulation, the formation follows a sine wave movement.

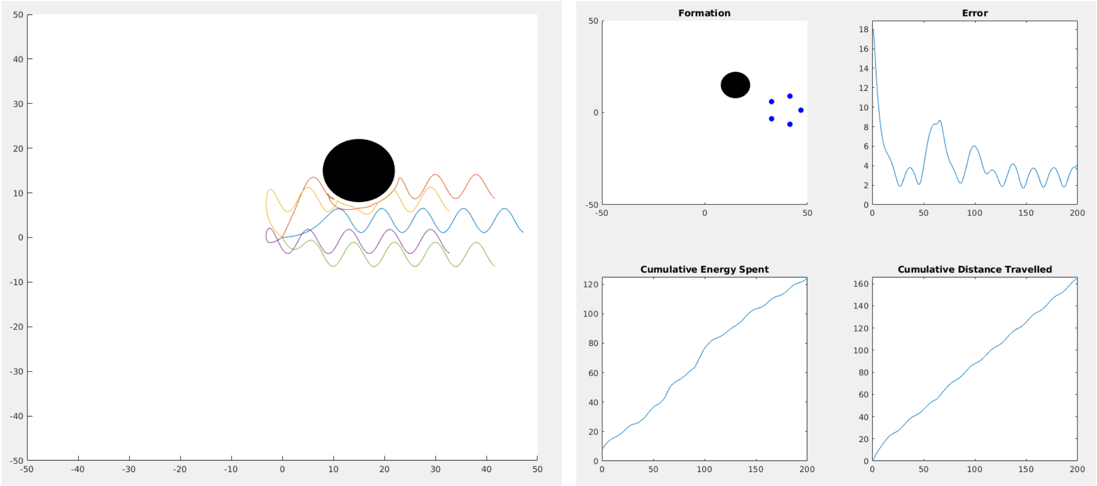


Figure 4.19: The final results of the MPC simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $12.1193s$

Units Travelled: 205.49

Energy Spent: 140.99

This simulation illustrates the different behaviour at the edge of the safe set from the MPC constraints. Movement at the edge is much smoother, supporting the system safety.

4.2.2.B Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

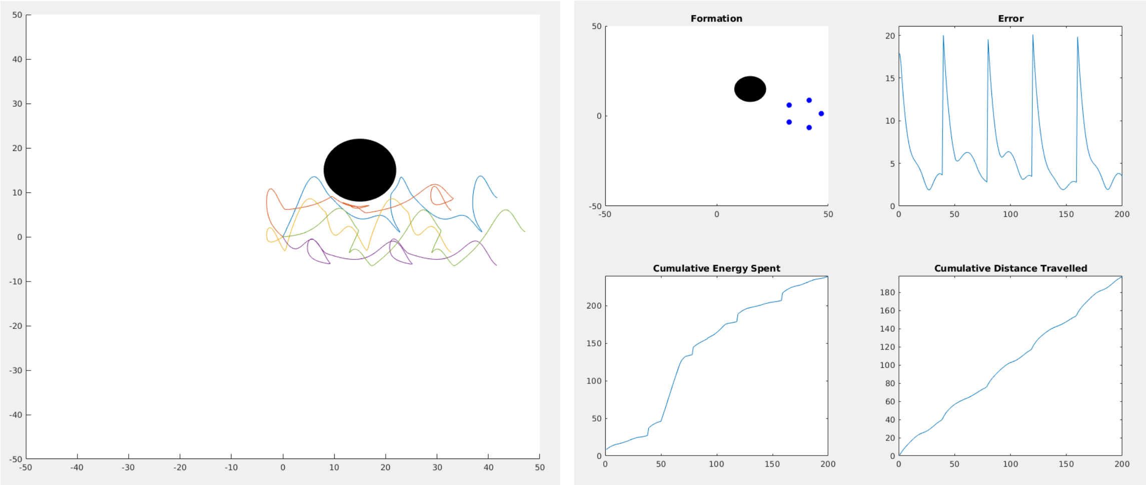


Figure 4.20: The final results of the MPC simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $17.5975s$

Units Travelled: 197.33

Energy Spent: 238.64

4.2.3 MPC with horizon = 10

4.2.3.A Wave Movement

In this simulation, the formation follows a sine wave movement.

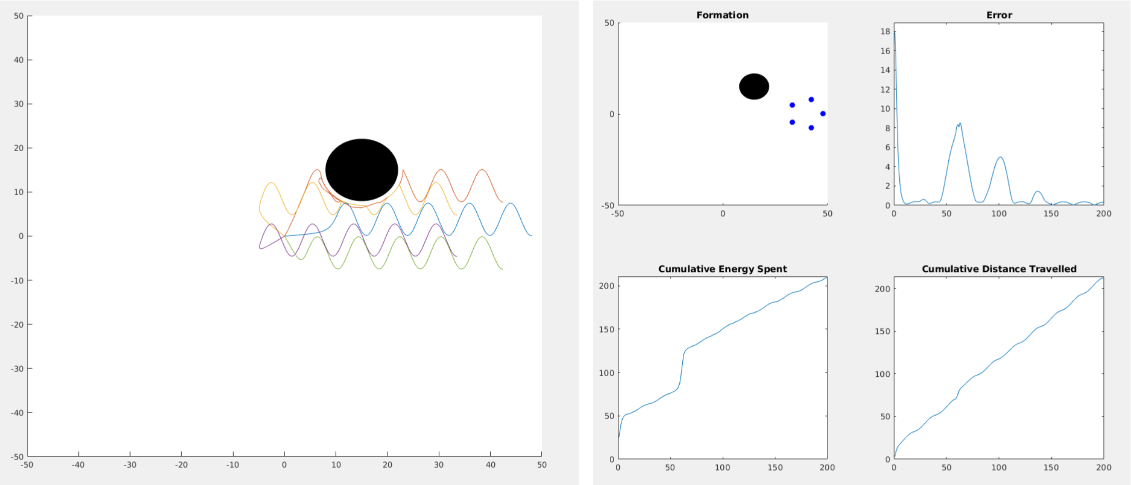


Figure 4.21: The final results of the MPC simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $48.3834s$

Units Travelled: 213.17

Energy Spent: 209.71

4.2.3.B Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

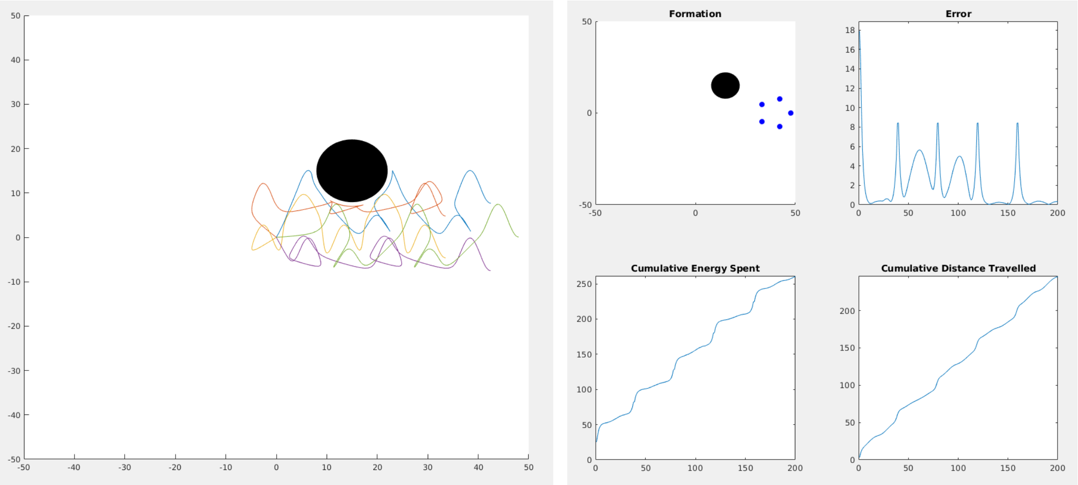


Figure 4.22: The final results of the MPC simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $61.1805s$

Units Travelled: 245.51

Energy Spent: 260.34

4.2.4 Results

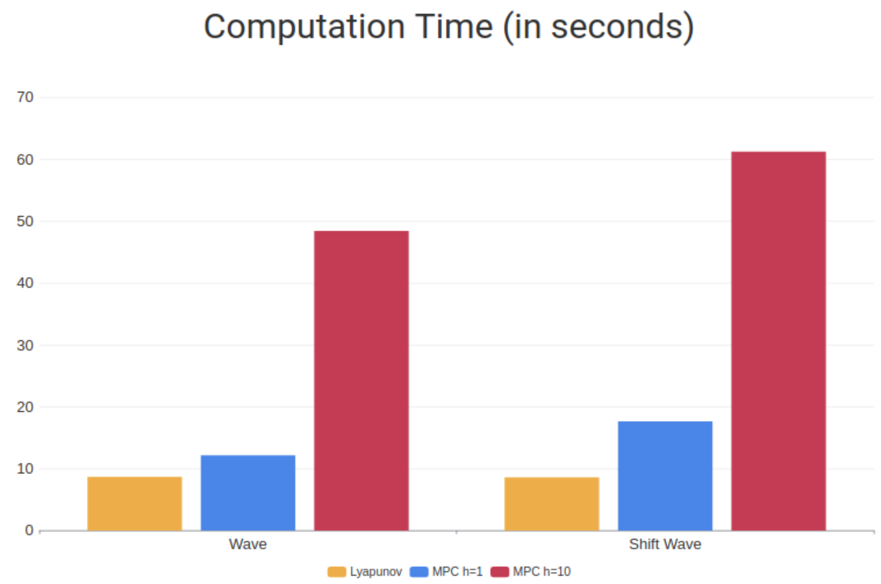


Figure 4.23: Chart of time for each computation

MPC with $h = 10$ already shows a much higher computation time with the added obstacle.

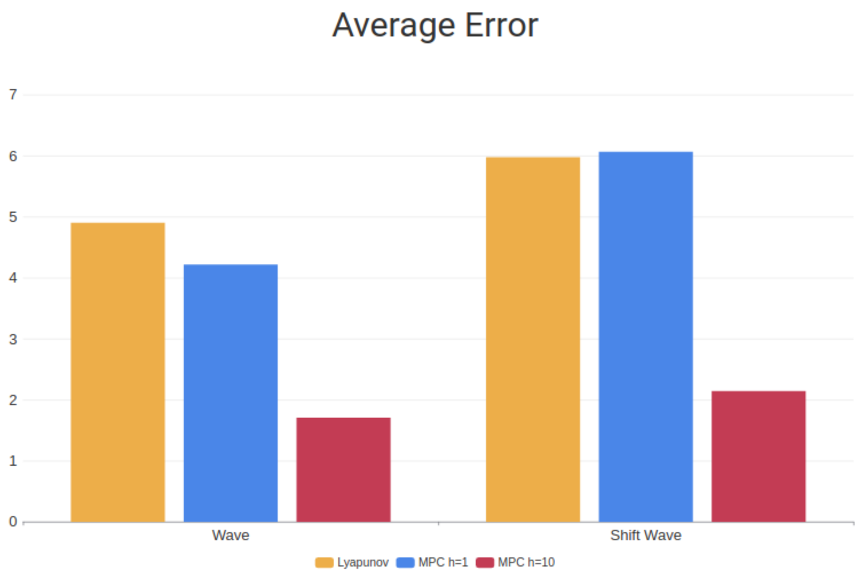


Figure 4.24: Chart of error for each computation

It should be noted that the MPC with $h = 10$ method has a higher error than in the version without obstacles. This is expected in this simulation given that the agents can't arrive at their optimal position if their optimal position is inside an obstacle.

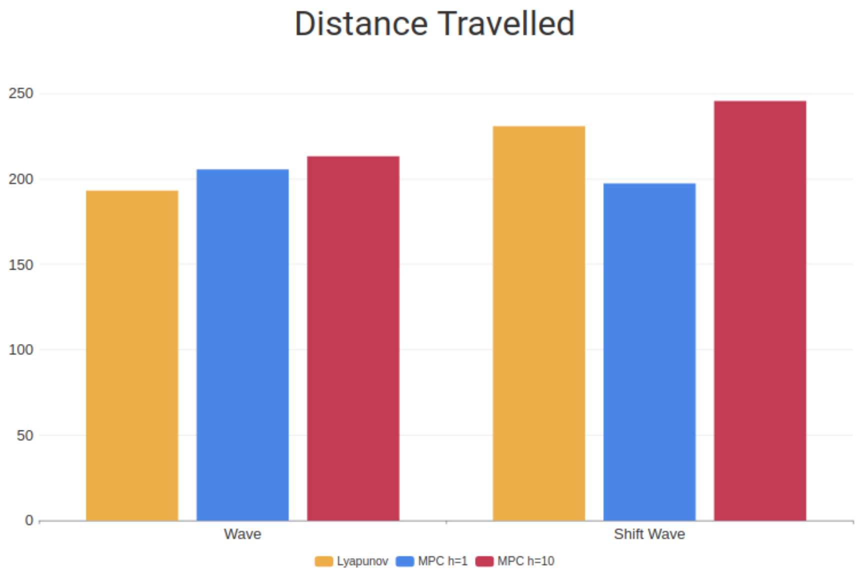


Figure 4.25: Chart of distance travelled for each computation

The notable smaller travel distance of the MPC with $h = 1$ should be noted. This can be explained by both the smoother behaviour at the edge of the safe set by both the MPC methods, causing the

Lyapunov function to have a higher travel distance due to the erratic movement at the edge of the safe set. Meanwhile, they both have lower travel distance than the MPC with $h = 10$ due to them lagging behind the formation.

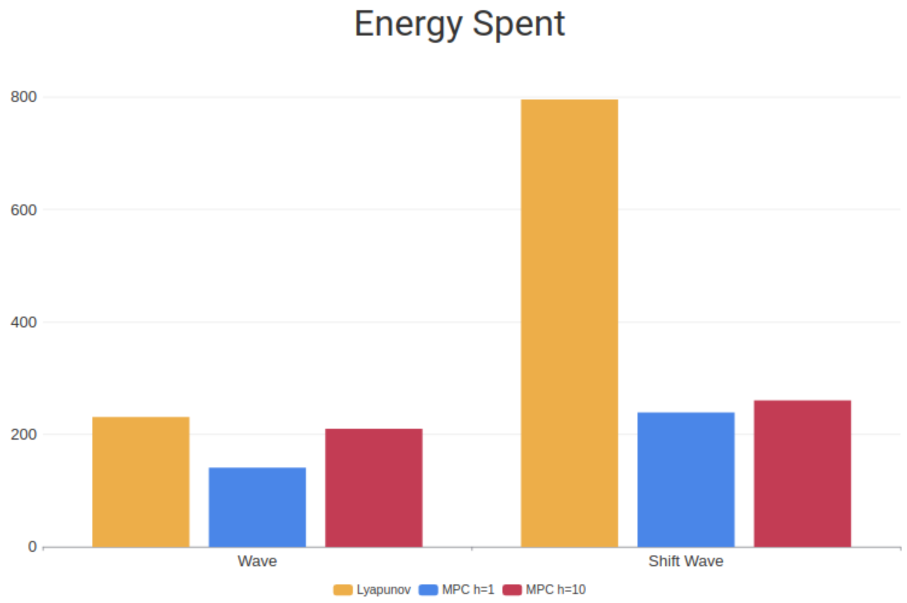


Figure 4.26: Chart of distance travelled for each computation

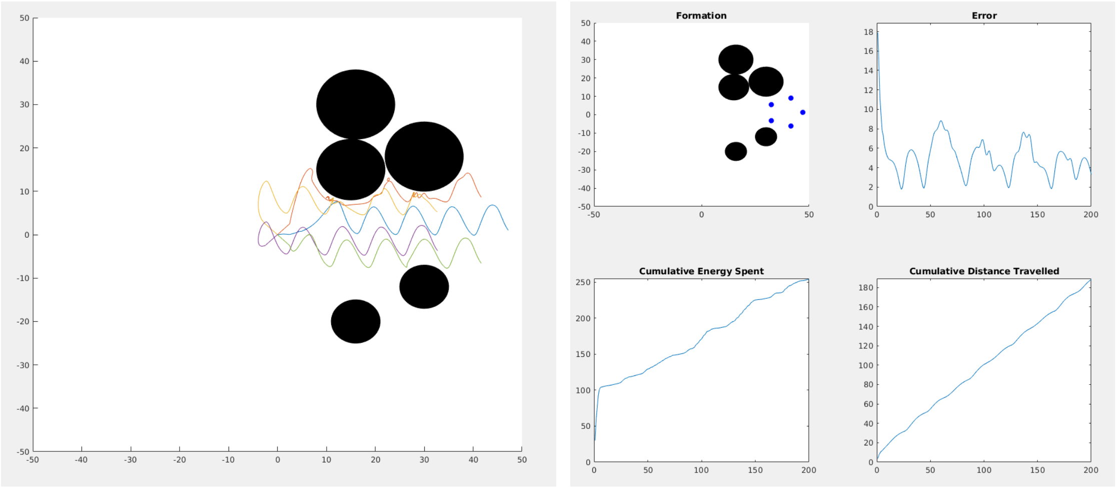
The Lyapunov method still spends more energy to maintain the formation than the others in the Shifting Wave formation. However, it should be noted that in the Wave formation, the energy spent is much closer to the other methods, which can lead us to a conclusion on the energy required to maintain the safe set by the MPC methods. It can be concluded that the energy they require to maneuver around the obstacles is higher than the one spent by the Lyapunov method.

4.3 Many Obstacles

4.3.1 Control Lyapunov Functions

4.3.1.A Wave Movement

In this simulation, the formation follows a sine wave movement.



Formation

Error

Cumulative Energy Spent

Cumulative Distance Travelled

Figure 4.27: The final results of the CLF simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $17.3169s$

Units Travelled: 188.34

Energy Spent: 254.08

4.3.1.B Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

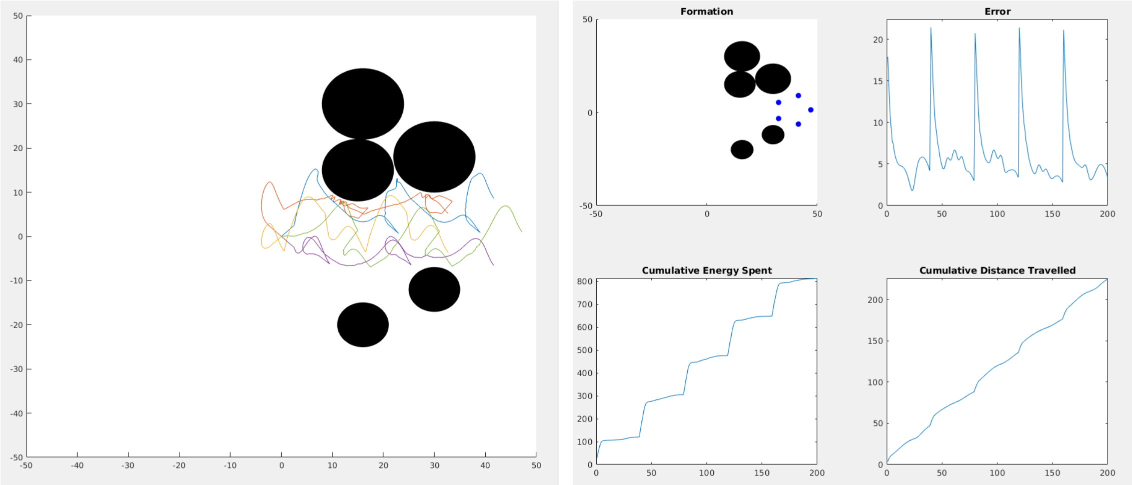


Figure 4.28: The final results of the CLF simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $17.0234s$

Units Travelled: 225.30

Energy Spent: 812.97

4.3.2 MPC with horizon = 1

4.3.2.A Wave Movement

In this simulation, the formation follows a sine wave movement.

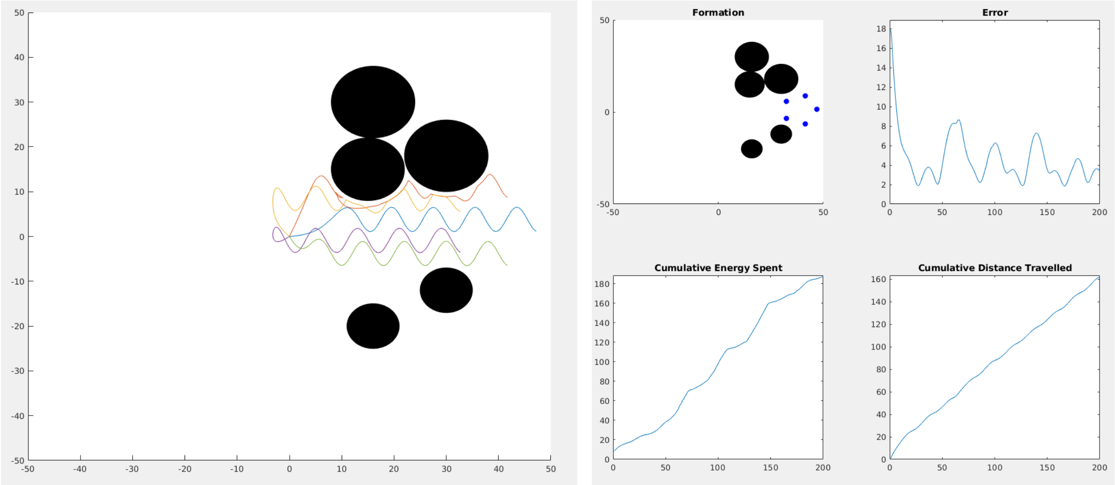


Figure 4.29: The final results of the MPC simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $42.0137s$

Units Travelled: 162.34

Energy Spent: 187.48

4.3.2.B Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

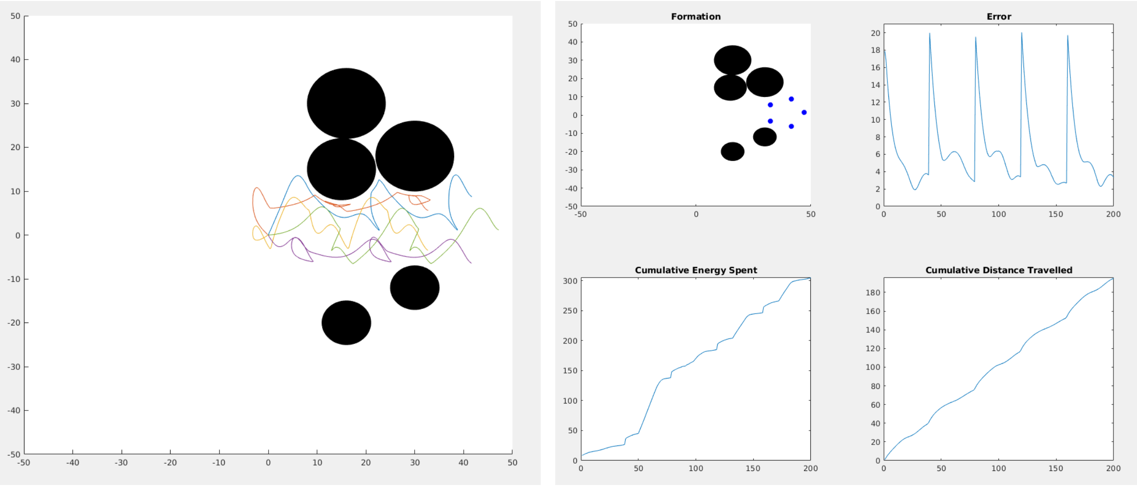


Figure 4.30: The final results of the MPC simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $50.4328s$

Units Travelled: 162.32

Energy Spent: 209.53

4.3.3 MPC with horizon = 10

4.3.3.A Wave Movement

In this simulation, the formation follows a sine wave movement.

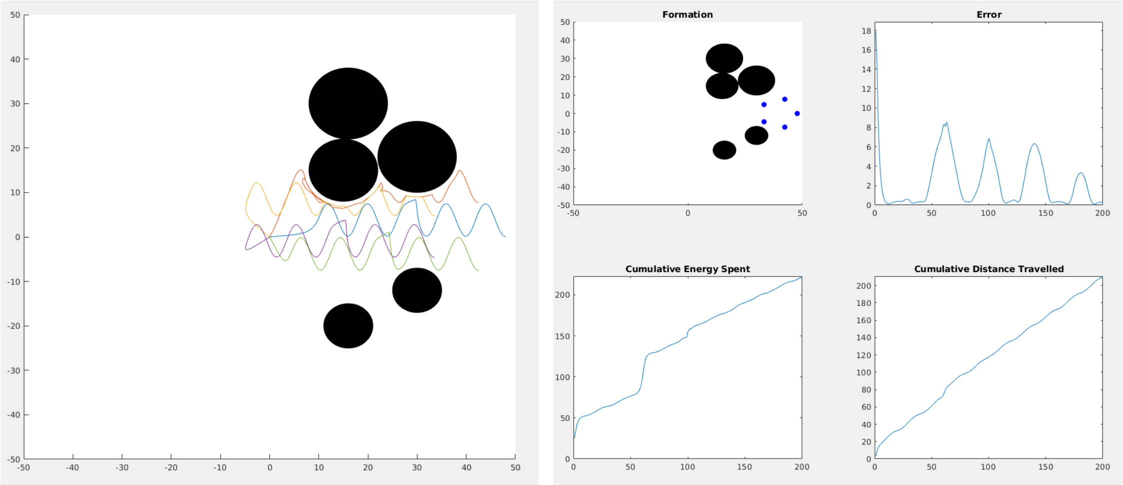


Figure 4.31: The final results of the MPC simulation moving in a sine wave pattern.

Used Formation: `wave_5_200_8.mat`

Elapsed Time: $215.9923s$

Units Travelled: 211.82

Energy Spent: 234.98

4.3.3.B Wave Shifting Movement

In this simulation, the formation follows a sine wave movement. Every 40 iterations, however, the optimal positions of the agents shift, adding more complexity.

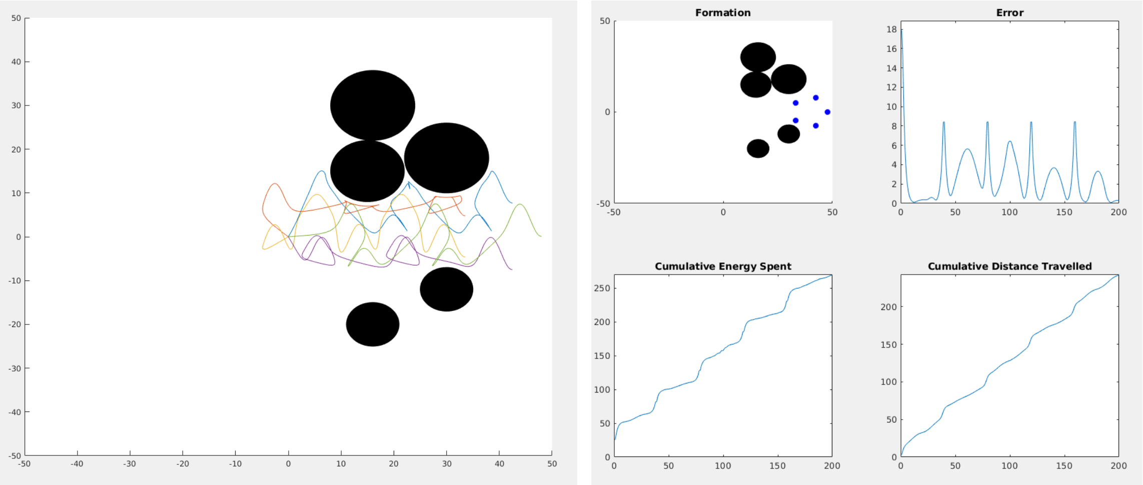


Figure 4.32: The final results of the MPC simulation moving in a sine wave pattern with shifting positions.

Used Formation: `s_wave_5_200_8.mat`

Elapsed Time: $201.0858s$

Units Travelled: 242.14

Energy Spent: 269.20

4.3.4 Results

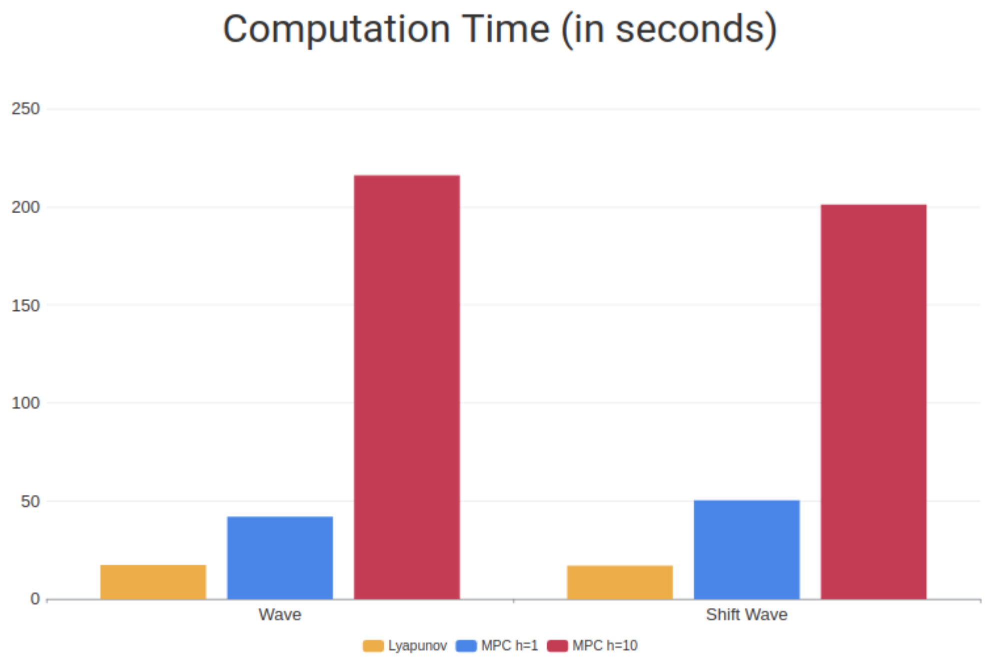


Figure 4.33: Chart of time for each computation

The increase in computation time required by the MPC with $h = 10$ method is noticeable. An increase in the number of obstacles leads to a much more computationally heavy optimization. To compare, it took 48.38s to complete the Wave formation simulation with one obstacle, while the time has increased to 215.99s with 5 obstacles, which is $4.47\times$ the time required before. The increase of time with the Lyapunov method was $2\times$, in comparison.

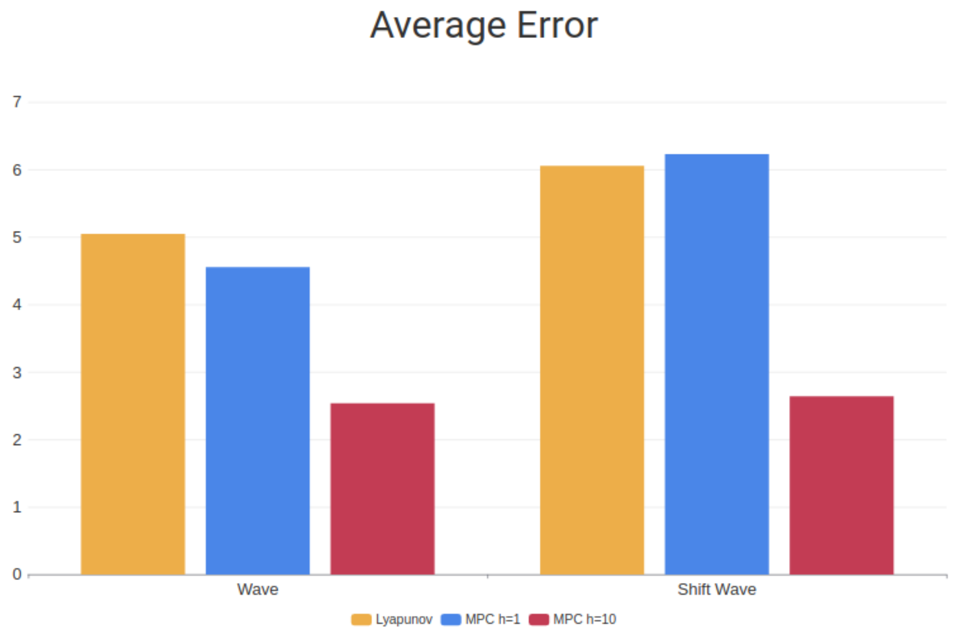


Figure 4.34: Chart of error for each computation

As before, the high average error from the MPC with $h = 10$ method can be explained by the optimal position being inside an obstacle.

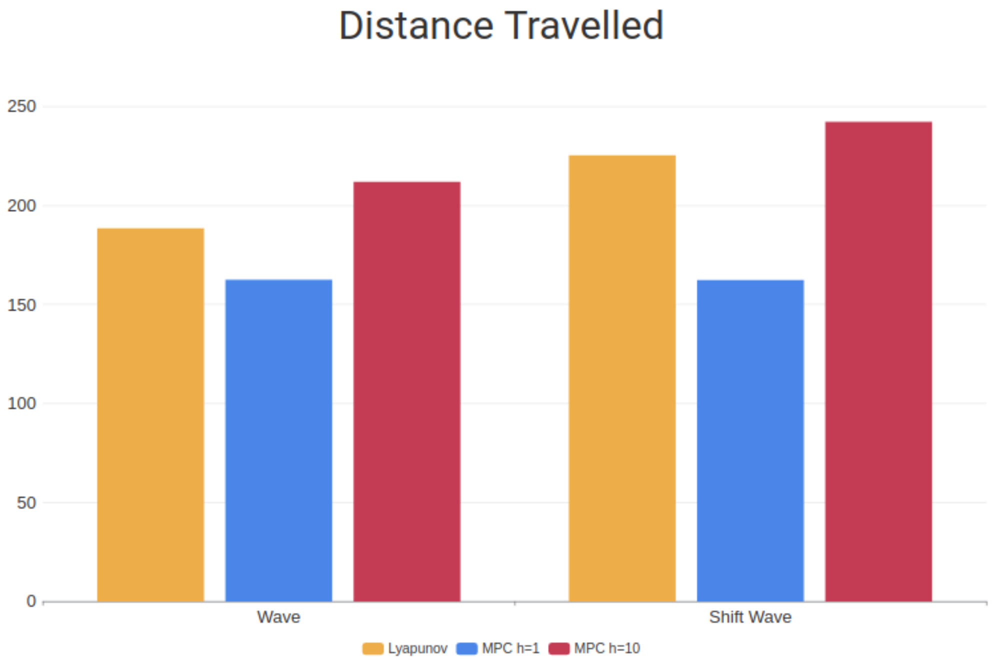


Figure 4.35: Chart of distance travelled for each computation

Both the distance and energy result conclusions are the same as in 4.2.4.

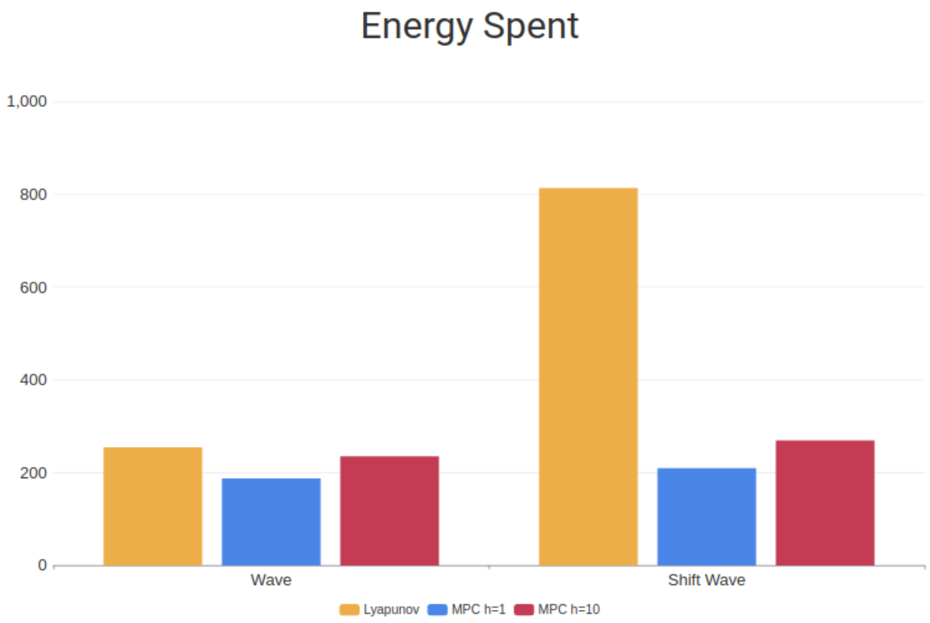


Figure 4.36: Chart of distance travelled for each computation

5

Conclusion

Contents

5.1 Findings

In the previous section, we found some noticeable similarities as well as differences between the Lyapunov Function system and the MPC with $h = 1$. Both are prone to stabilize in local minimums instead of achieving a global minimum, which is expected for Lyapunov Functions, given that they have a greedy approach to optimization, unlike MPC with a high horizon which achieves a global minimum error quickly.

However, using MPC with a large horizon also drastically increases computation time, especially with the introduction of obstacles. Care should be taken when applying large horizons to MPC computations, given that they increase accuracy and allow the system to avoid local minimums, however, that may not be worth the increase in computation time.

The obstacle avoidance methods for Lyapunov Functions and MPC are different and as such produce different behaviours on the edge of the safe set of positions. CBFs produce somewhat erratic behaviour when agents are in the edge of the safe set of positions, while the MPC linear constraint method produces a much smoother trajectory, maintaining distance from the edge of the unsafe set.

It should be noted that despite being the one with the highest error on average, the CLF and CBF is consistently faster than the other methods.

It should also be noted that Lyapunov Functions have a much higher energy usage on average, especially when the formation requires a sudden movement, expending a lot of energy to quickly correct the position.

Bibliography

[1] Y. Kuriki and T. Namerikawa, “Formation control with collision avoidance for a multi-uav system using decentralized mpc and consensus-based control,” *SICE Journal of Control, Measurement, and System Integration*, vol. 8, no. 4, pp. 285–294, 2015.

[2] Z. Chao, L. Ming, Z. Shaolei, and Z. Wenguang, “Collision-free uav formation flight control based on nonlinear mpc,” in *2011 international conference on electronics, communications and control (ICECC)*. IEEE, 2011, pp. 1951–1956.

[3] W. Zhao and T. H. Go, “Quadcopter formation flight control combining mpc and robust feedback linearization,” *Journal of the Franklin Institute*, vol. 351, no. 3, pp. 1335–1355, 2014.

[4] Z. Wu, F. Albalawi, Z. Zhang, J. Zhang, H. Durand, and P. D. Christofides, “Control lyapunov-barrier function-based model predictive control of nonlinear systems,” *Automatica*, vol. 109, p. 108508, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109819303693

[5] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[6] U. Rosolia and A. D. Ames, “Multi-rwu2019108508, 9137248, 9137248ate control design leveraging control barrier functions and model predictive control policies,” *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1007–1012, 2021.

[7] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control,” *Automatica*, vol. 53, pp. 424–440, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109814004038

[8] W. Ren and E. Atkins, “Distributed multi-vehicle coordinated control via local information exchange,” *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10[U+2010]11, pp. 1002–1033, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.1147

[9] W. Dong and J. A. Farrell, “Cooperative control of multiple nonholonomic mobile agents,” *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1434–1448, 2008.

[10] ——, “Decentralized cooperative control of multiple nonholonomic dynamic systems with uncertainty,” *Automatica*, vol. 45, no. 3, pp. 706–710, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109808004780

[11] T. H. van den Broek, N. van de Wouw, and H. Nijmeijer, “Formation control of unicycle mobile robots: a virtual structure approach,” in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 8328–8333.

[12] A. Sadowska, T. van den Broek, H. Huijberts, N. van de Wouw, D. Kostić, and H. Nijmeijer, “A virtual structure approach to formation control of unicycle mobile robots using mutual coupling,” *International Journal of Control*, vol. 84, no. 11, pp. 1886–1902, 2011. [Online]. Available: https://doi.org/10.1080/00207179.2011.627686

[13] T. Lewis, “High precision formation control of mobile robots using virtual structures,” *Autonomous Robots*, vol. 4, p. 387–403, 1997.

[14] K.-H. Tan and M. Lewis, “Virtual structures for high-precision cooperative mobile robotic control,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS ’96*, vol. 1, 1996, pp. 132–139 vol.1.

[15] D. V. Dimarogonas and K. J. Kyriakopoulos, “Connectedness preserving distributed swarm aggregation for multiple kinematic robots,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1213–1223, 2008.

[16] T. Liu and Z.-P. Jiang, “Distributed formation control of nonholonomic mobile robots without global position measurements,” *Automatica*, vol. 49, no. 2, pp. 592–600, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109812005675

[17] J. Marshall, M. Brouke, and B. Francis, “A pursuit strategy for wheeled-vehicle formations,” in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 3, 2003, pp. 2555–2560 Vol.3.

[18] Z. Lin, M. Broucke, and B. Francis, “Local control strategies for groups of mobile autonomous agents,” *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

[19] Z. Lin, B. Francis, and M. Maggiore, “Necessary and sufficient graphical conditions for formation control of unicycles,” *IEEE Transactions on Automatic Control*, vol. 50, no. 1, pp. 121–127, 2005.

[20] D. V. Dimarogonas and K. H. Johansson, “On the stability of distance-based formation control,” in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 1200–1205.

[21] D. Dimarogonas and K. Kyriakopoulos, “Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 84–89.

[22] ——, “A connection between formation control and flocking behavior in nonholonomic multiagent systems,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 940–945.

[23] B. Lei, W. Li, and F. Zhang, “Flocking algorithm for multi-robots formation control with a target steering agent,” in *2008 IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 3536–3541.

[24] K.-K. Oh and H.-S. Ahn, “Formation control of mobile agents based on distributed position estimation,” *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 737–742, 2013.

[25] B. Fidan, V. Gazi, S. Zhai, N. Cen, and E. Karataş, “Single-view distance-estimation-based formation control of robotic swarms,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5781–5791, 2013.

[26] G. Jing and L. Wang, “Multiagent flocking with angle-based formation shape control,” *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 817–823, 2020.

[27] T. Balch and R. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[28] S. Monteiro and E. Bicho, “A dynamical systems approach to behavior-based formation control,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, 2002, pp. 2606–2611 vol.3.

[29] P. Ogren, M. Egerstedt, and X. Hu, “A control lyapunov function approach to multi-agent coordination,” in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, vol. 2, 2001, pp. 1150–1155 vol.2.

[30] S. Panimadai Ramaswamy and S. Balakrishnan, “Formation control of car-like mobile robots: A lyapunov function based approach,” in *2008 American Control Conference*, 2008, pp. 657–662.

[31] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, “Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs,” *IEEE Access*, vol. 3, pp. 323–332, 2015.

[32] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[33] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.

[34] R. Takano and M. Yamakita, “Robust constrained stabilization control using control lyapunov and control barrier function in the presence of measurement noises,” in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 300–305.

[35] L. Acar, “Some examples for the decentralized receding horizon control,” in *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, 1992, pp. 1356–1359 vol.2.

[36] H. Fawal, D. Georges, and G. Bornard, “Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented lagrangian,” in *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, vol. 4, 1998, pp. 3874–3879 vol.4.

[37] M. Gomez, J. Rodellar, F. Vea, J. Mantecon, and J. Cardona, “Decentralized predictive control of multi-reach canals,” in *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, vol. 4, 1998, pp. 3885–3890 vol.4.

[38] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar, “Distributed model predictive control,” *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[39] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, “Cooperative distributed model predictive control,” *Systems & Control Letters*, vol. 59, no. 8, pp. 460–469, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167691110000691

[40] A. Venkat, J. Rawlings, and S. Wright, “Stability and optimality of distributed model predictive control,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 6680–6685.

[41] W. Dunbar and R. Murray, “Model predictive control of coordinated multi-vehicle formations,” in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 4, 2002, pp. 4631–4636 vol.4.

[42] H. Fukushima, K. Kon, and F. Matsuno, “Model predictive formation control using branch-and-bound compatible with collision avoidance problems,” *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1308–1317, 2013.

[43] J. Lavaei, A. Momeni, and A. G. Aghdam, “A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 268–278, 2008.

[44] C. E. García, D. M. Prett, and M. Morari, “Model predictive control: Theory and practice—a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109889900022

[45] A. Botelho, B. Parreira, P. N. Rosa, and J. M. Lemos, *Predictive Control for Spacecraft Rendezvous*. Springer, 2021.