MANCHESTER 1824
The University of Manchester

Formation Control of Nonholonomic Wheeled Mobile Robots Using Consensus-Based Distributed Model Predictive Control and Input-Output Linearisation

Third Year Individual Project - Final Report

Year of submission
2025

Student ID
10949254

School of Engineering

Contents

Word count: 10761

List of figures

List of tables

List of Acronyms

Acronym	Definition
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
AGV	Automated Guided Vehicle
AUV	Autonomous Underwater Vehicles
MRS	Multi-robot Systems
WMRS	Wheeled Mobile Robots
NMPC	Nonlinear Model Predictive Control
DLPC	Distributed Learning-based Predictive Control
CLF	Control Lyapunov Function
DTCBF	Discrete-time Control Barrier Function
NSB	Null-Space-based Behavioural
APF	Artificial Potential Fields
MPC	Model Predictive Control
MPOMDPs	Multi-agent Partially Observable Markov Decision Processes
MHE	Moving Horizon Estimation
DEKF	Delayed Extended Kalman Filter
QP	Quadratic Programming
ECBF	Exponential Control Barrier Function
CAGR	Compound Annual Growth Rate
IAE	Integral Absolute Error

Abstract

This dissertation addresses distributed formation control of nonholonomic wheeled mobile robots (WMRs), with applications in autonomous vehicles, swarm robotics, and industrial automation. A consensus-based nonlinear model predictive control (NMPC) framework is proposed, enhanced through input-output linearisation and integrated with discrete-time control barrier functions (DTCBFs) for safety. The framework ensures real-time trajectory tracking, obstacle avoidance, and collision-free operation while maintaining formation geometry.

Mathematical modelling captures the nonholonomic dynamics of individual WMRs. A distributed consensus algorithm enables coordination without central control, while DTCBFs enforce safety constraints throughout motion. The NMPC controller generates optimal control inputs over a predictive horizon, balancing tracking precision and constraint satisfaction.

The proposed approach is evaluated through simulations across linear and curved trajectories in dynamic environments. Results demonstrate strong formation-keeping performance, responsiveness to disturbances, and reliable obstacle clearance. Comparative analysis with a distributed learning-based predictive control (DLPC) method highlights that while DLPC offers lower computational load, the consensus-based NMPC delivers superior formation accuracy and formal safety guarantees.

This work contributes to the theoretical foundation of distributed multi-robot control by unifying consensus theory, NMPC, and barrier functions in a single framework. It also provides practical insights for deploying autonomous robot teams in complex, real-world environments.

Declaration of originality

I hereby confirm that this dissertation is my own original work unless referenced clearly to the contrary, and that no portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual property statement

i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights (the “Copyright”), and he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* under the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, per licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example, graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Dissertation restriction declarations deposited in the University Library, and The University Library’s regulations (see `http://www.library.manchester.ac.uk/about/regulations/_files/Library-regulations.pdf`).

Acknowledgements

I would like to thank my mother—without her unwavering support and belief in me, I would not have reached this stage.

I am deeply grateful to my supervisor, Dr. Joaquin Carrasco, for his outstanding guidance and support throughout the past year.

I also want to thank Dr. Wei Pan for generously granting me access to his lab's resources, which were invaluable to my research.

Finally, I would like to thank Weishu Zhan for the insightful discussions on advanced control techniques, which greatly enriched my work.

1 Introduction

1.1 Background and motivation

The formation control of multiple robots has garnered significant research attention due to its wide-ranging applications in autonomous vehicles, swarm robotics, and industrial automation [1], [2]. Multi-robot systems are being increasingly deployed in complex, dynamic environments where cooperation and coordination are essential. Achieving and maintaining a desired formation while ensuring global system stability is critical in applications such as unmanned aerial vehicle (UAV) swarms, unmanned ground vehicle (UGV) formations, and industrial robotic fleets [3], [4].

According to [5], ”Global factory automation & industrial controls market size was USD 124.3 Billion in 2024, and the market is projected to touch 178.84 Billion by 2033, exhibiting a CAGR (Compound Annual Growth Rate) of 4.1% during the forecast period.” This growth highlights the increasing reliance on autonomous robotic systems for industrial automation.

Various industries, including logistics, agriculture, and defence, leverage formation control for tasks such as autonomous convoys, drone swarms, and robotic surveillance. For example, Amazon warehouses utilise fleets of Automated Guided Vehicles (AGVs) to optimise inventory movement and reduce human workload [6]. Precise formation maintenance is essential for coordinated sensing and communication in satellite constellations [7]. Similarly, Autonomous Underwater Vehicles (AUVs) rely on formation control for efficient deep-sea exploration and environmental monitoring [8]. In industrial settings, consensus-based control strategies enable AGV fleets to synchronise warehouse logistics and streamline production lines, significantly improving operational efficiency.

However, real-time formation control presents several challenges that must be addressed to enable robust and scalable deployment. A significant difficulty arises from time delays in wireless communication networks, which can degrade system performance and stability [9]. Existing consensus algorithms often struggle with scalability due to increasing communication overhead and computational complexity [8]. Additionally, conventional delay-compensation strategies may not be well-suited for real-time applications with unpredictable network latencies. Furthermore, real-world disturbances such as sensor noise, actuator failures, and dynamic obstacles necessitate robust and adaptive control strategies [10]. Safe coordination in uncertain environments is critical, particularly in human-robot interactions and hazardous operational conditions.

To address these limitations, advanced control frameworks incorporating predictive modelling, distributed optimisation, and robustness to uncertainties are required [11]. By developing scalable and delay-tolerant formation control strategies, multi-robot systems can achieve higher efficiency, adaptability, and resilience in real-world applications.

1.2 Aims and objectives

This project aims to contribute to the development of safe and adaptable navigation in multi-robot systems capable of handling unforeseen obstacles and dynamically changing conditions. The goal is to ensure the safety and reliability of future autonomous vehicles in challenging environments. Specifically, the project will develop a consensus-based, nonlinear model predictive formation

controller implementation using input-output linearisation, evaluate its suitability for the desired application, and compare its performance to a recently developed, computationally fast and efficient distributed learning-based predictive control framework that integrates a receding horizon optimisation strategy into policy updates [12]. The main objectives of the project are summarised below.

- To develop a mathematical model of the wheeled mobile robot, considering its kinematics using nonholonomic constraints and dynamics using the Lagrangian formulation.
- To develop a consensus algorithm based on nonlinear model predictive control (NMPC) for path planning and formation control using input-output linearisation.
- To enforce safety formally in the control problem using Discrete-time Control Barrier Functions.
- To evaluate the performance of the consensus algorithm by comparing its performance with a distributed NMPC implementation without the consensus algorithm.
- To compare the developed consensus-based NMPC implementation using numerical solvers with the DLPC framework [12] for WMRs.
- To assess the trade-offs between safety, computational efficiency, and formation control in the integrated framework.

In this project, the collective goal is to navigate a terrain of obstacles while maintaining an arbitrary geometric formation.

1.3 Report structure

This report is organised as follows. Section 2 reviews existing literature on the formation control of multi-robot systems (MRS). Section 3 discusses preliminary concepts fundamental to the report. The project's methodology is considered in Section 4. The simulation results and a discussion on the performance evaluation are presented in Section 5. Finally, the conclusion and a consideration of future research avenues can be found in Section 6.

2 Literature review

The formation control of multiple mobile robot systems has wide applications in both military and civilian fields [13], [14], including target tracking, cooperative search and rescue, exploration, distributed manipulation, and patrol [15]. In such missions, a group of robots offers several advantages over a single robot, including increased efficiency, adaptability, and robustness [14], ultimately saving time and reducing operational costs [13].

Formation control involves multiple agents moving toward a specific target while maintaining a predetermined geometric shape relative to each other and adapting to environmental constraints such as obstacles and narrow passages [16]. Formation control methods can be categorised in various ways. One categorisation, based on sensing capabilities, includes position-based, displacement-based, and distance-based control [4]. Additionally, modern formation control strategies can be broadly grouped into four categories: the leader-follower paradigm [8], virtual

structure methodology [17], consensus-based strategies [18], and behaviour-based methods [19], each with its strengths and limitations.

The leader-follower method, one of the earliest and most straightforward approaches to formation control, establishes hierarchical relationships where followers track the leader robots' positions. This method is advantageous due to its simplicity, lower computational load, and ease of real-time implementation. However, the leader represents a single point of failure; if it fails or deviates from the expected trajectory, the formation will break. The method also suffers from accumulated tracking errors in multi-robot chains, and it is difficult to maintain precise inter-robot spacing. Recent improvements, such as the use of an embedded control technique for leader-follower formation control in wheeled mobile robots (WMRs) [8], ensure robustness despite disturbances. Furthermore, an extended state observer has been applied to leader-following formation control to compensate for inaccuracies in global positioning systems [14], improving real-time tracking for nonholonomic robots. Despite these improvements, a significant drawback remains: if the leader fails, the entire formation collapses. Some recent work, such as the integrated controller combining sliding mode control and a force function for obstacle avoidance proposed by [16], has attempted to mitigate these issues by incorporating additional controllers.

The second method of formation control, virtual structure control, is based on the concept of a virtual rigid body whose centre of mass is tracked to ensure precise formation maintenance. This approach does not require the selection of a leader and offers an intuitive way to define the formation. Robots maintain fixed positions relative to a moving reference frame, which is ideal for assembly tasks requiring rigid formations. However, when one of the robots experiences a disturbance, the formation can break apart, and the method often requires global information, reducing scalability and increasing communication demands. A high-precision formation control method for mobile robots using the virtual structure framework has been proposed [17], where robots track predefined formation trajectories with minimal deviation. This approach has demonstrated fault-tolerant movement with graceful performance degradation under disturbances.

Behaviour-based control focuses on creating complex behaviours from simpler, modular actions, drawing inspiration from biological systems to promote adaptability and robustness in dynamic environments. One example is the Null-Space-based Behavioural (NSB) control method [20], which integrates multiple prioritised behaviours using a projection mechanism, ensuring that lower-priority behaviours do not interfere with higher-priority ones. This strategy facilitates coordinated control in multi-robot systems, especially in complex environments. Another example is a behaviour-based formation control strategy that enables multi-robot teams to maintain their formations while navigating and avoiding obstacles [21].

Another popular method is consensus-based control, a distributed approach that allows robots to agree on their velocities and positions through local communication without requiring centralised control. It is a subset of graph-theoretical approaches where robot interactions are modelled through directed/undirected graphs, which enable formal stability analysis. One of the key advantages of consensus-based methods is their ability to handle disturbances and input constraints effectively. For example, a consensus formation control strategy for nonholonomic mobile robots ensures robust tracking performance despite mixed disturbances and input constraints [9]. Consensus-based control has been extended by designing a control law specifically for nonholonomic robots using Artificial Potential Fields (APFs) and providing a stability analysis

to guarantee convergence [22]. Furthermore, a consensus-based formation control approach incorporates time synchronisation to address the challenges posed by communication delays, ensuring coordinated motion in distributed systems [18].

Learning-based control is a more recent strategy for formation control. It integrates advanced algorithms like reinforcement learning and actor-critic networks with model predictive control (MPC) to overcome multi-robot systems' scalability, safety, and uncertainty challenges. A recent learning-based approach is to use a DLPC framework that leverages fast policy learning to manage scalability in multi-robot formations, enabling real-time coordination with minimal computational effort [12]. In a different approach, the policy synthesis process for multi-agent partially observable Markov decision processes (MPOMDPs) embeds discrete-time barrier functions to ensure that safety constraints are met despite system uncertainties [23]. Additionally, a method enhances sample efficiency and mitigates uncertainty within the context of learning-based MPC for aerial robots, improving the robustness and performance of the system [24].

The development of formation control methods for multi-robot systems has made significant progress over the years, with each approach offering unique advantages and addressing different challenges. While leader-follower and virtual structure methods are well-established, more recent approaches, such as consensus-based and behaviour-based control, provide decentralised, scalable, and adaptable solutions particularly suited for dynamic and uncertain environments.

Despite the advances in these methods, challenges remain, especially in environments with communication delays or unpredictable disturbances. For example, a moving horizon estimation (MHE) method using a delayed extended Kalman filter (DEKF)-based update law has been developed to accurately estimate the positions of unmanned underwater vehicles in the presence of delays [25], highlighting the need for robust estimation and control techniques when implementing network topologies on real systems.

In this report, the focus is placed on the problem of trajectory tracking for multiple WMRs while maintaining the desired geometric formation and avoiding obstacles. The consensus-based control strategy helps to guarantee formation stability, while the trajectory tracking problem is addressed using an NMPC-β controller [11]. NMPC-β is chosen because it ensures stability by incorporating a terminal cost derived from a CLF, eliminating the need for explicit terminal constraints while maintaining recursive feasibility and real-time implementability.

3 Preliminaries

This section briefly reviews formal definitions fundamental to the consensus and nonlinear control strategies discussed in Section 4, namely Control Lyapunov Functions (CLFs), Discrete-time Control Barrier Functions (DTCBFs), Input-Output Linearisation, and Algebraic Graph Theory. The proposed control architecture employs a Nonlinear Model Predictive Control (NMPC) scheme, which inherently involves discrete-time predictions and decision-making, even when controlling systems that evolve in continuous time. This dual nature of NMPC motivates the careful selection of both continuous and discrete-time theoretical tools for different aspects of the controller design.

Specifically, Control Lyapunov Functions (CLFs), which are typically developed for continuous-time systems, are utilised in this work to construct the terminal cost in the NMPC-β formulation

[11]. The inclusion of a CLF-based terminal cost is a well-established technique [11] to help ensure the stability of the NMPC by approximating the infinite-horizon cost and guiding the predicted state towards a stable region beyond the finite prediction horizon. It allows leveraging continuous-time stability theory to inform the long-term behaviour encouraged by the discrete-time optimiser.

Conversely, Discrete-Time Control Barrier Functions (DTCBFs) are employed to enforce safety constraints directly within the discrete-time predictive control loop [11]. As the NMPC calculates optimal control actions by predicting system states over a sequence of discrete future time steps, DTCBFs provide a mathematically rigorous and natural framework for ensuring that each planned state transition within this horizon adheres to safety requirements, such as collision avoidance. This ensures that safety is considered explicitly at each discrete decision point of the controller. The subsequent subsections will detail these concepts.

3.1 Mathematical Notation

This section defines the key mathematical symbols, sets, and functions used throughout the report for clarity and consistency. The notation is designed to provide a rigorous framework for understanding the mathematical formulations and algorithms presented.

3.1.1 General Symbols

- $\mathbb{R}$: The set of real numbers.
- $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x > 0\}$: The set of positive real numbers.
- $\mathbb{Z}$: The set of integers.
- $\mathbb{N}$: The set of natural numbers.
- $\mathbb{R}^n$: The n-dimensional Euclidean space.
- $\mathbb{R}^{n \times m}$: The space of $n \times m$ real matrices.
- $\mathbb{S}^n_{++} \subset \mathbb{R}^{n \times n}$: the set of positive-definite $n \times n$ real matrices.

3.1.2 Vectors and Matrices

- **Boldface symbols**: Represent vectors or matrices.
  - Example: $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ is a vector in $\mathbb{R}^n$.
- $\mathbf{I}_n$: the $n \times n$ identity matrix.
- $\mathbf{0}_{n \times m}$: the $n \times m$ zero matrix.

3.1.3 Functions and Operators

- diag($\mathbf{a}$): A diagonal matrix with the elements of vector $\mathbf{a} = [a_1, a_2, \ldots, a_n]^\top$ as its diagonal entries.

  - Example: For $\mathbf{a} = [1, 2, 3]^\top$, $\text{diag}(\mathbf{a}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$.
- $\|\cdot\|_p$: The p-norm of a vector or matrix.
  - Example: for a vector $(\mathbf{x}) = [x_1, x_2, \ldots, x_n]^\top$, the Euclidean norm is $\|\mathbf{x}\|_2 = (\sum_{i=1}^{n} x_i^2)^{1/2}$.
- $\text{Int}(\mathcal{S})$: The *interior* of the set $\mathcal{S} \subseteq \mathbb{R}^n$, defined as the set of all points $x \in \mathcal{S}$ for which there exists an open ball $B_\epsilon(x) = \{y \in \mathbb{R}^n \mid \|y - x\|_2 < \epsilon\}$ entirely contained in $\mathcal{S}$.
- $\partial\mathcal{S}$: The *boundary* of the set $\mathcal{S} \subseteq \mathbb{R}^n$, defined as the set of points $x \in \mathbb{R}^n$ such that every open ball $B_\epsilon(x)$ contains both points in $\mathcal{S}$ and in its complement $\mathbb{R}^n \setminus \mathcal{S}$.

3.2 Control Lyapunov Functions

Consider a state space $\mathbb{X} \subset \mathbb{R}^n$ and a control input space $\mathbb{U} \subset \mathbb{R}^m$, where $\mathbb{X}$ is assumed to be path-connected and contains the origin in its interior. The dynamics of a nonlinear control-affine system are given by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{1}$$

where $\mathbf{x} \in \mathbb{X}$, $\mathbf{u} \in \mathbb{U}$, and $\mathbf{f} : \mathbb{X} \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{X} \to \mathbb{R}^{n \times m}$ are Lipschitz continuous on $\mathbb{X}$.

Assumption 1. *Assume that the origin, i.e., $\mathbf{f}(0) = 0$, is an equilibrium point of the system.*

To analyse system stability, the concept of class $\mathcal{K}$ functions is introduced, which are commonly used to express stability conditions.

Definition 1. *(Class $\mathcal{K}$ Function [11]). A continuous function $\alpha : [0, a) \to \mathbb{R}_+$, with $a > 0$, belongs to class $\mathcal{K}$ if it satisfies:*

1. $\alpha(0) = 0$,
2. $\alpha(r)$ *is strictly monotonically increasing for all* $r \in [0, a)$.

If $a = \infty$ *and* $\lim_{r\to\infty} \alpha(r) = \infty$, *then* α *belongs to the class* $\mathcal{K}_\infty$.

A class $\mathcal{K}$ function can be interpreted as a nonlinear gain function, where larger values of r result in larger function values. A simple example is the linear function $\alpha(r) = \gamma r$, with $\gamma > 0$, which satisfies the definition.

Given this, CLFs, which belong to $\mathcal{K}_\infty$, are defined as introduced in [26].

Definition 2. *(Control Lyapunov Functions [26]). A continuously differentiable function $V : \mathbb{X} \to \mathbb{R}_+$ is a CLF for system* (1) *if there exist class $\mathcal{K}_\infty$ functions $\alpha_1, \alpha_2, \alpha_3$ such that for all $\mathbf{x} \in \mathbb{X}$:*

$$\begin{aligned} \alpha_1(\|\mathbf{x}\|) \leq V(\mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|), \\ \inf_{\mathbf{u}\in\mathbb{U}} \dot{V}(\mathbf{x}, \mathbf{u}) \leq -\alpha_3(\|\mathbf{x}\|). \end{aligned} \tag{2}$$

The existence of a CLF ensures the possibility of designing a continuous (except potentially at $\mathbf{x} = 0$) state-feedback controller $\mathbf{k} : \mathbb{X} \to \mathbb{U}$ that guarantees global asymptotic stability of the origin [26].

If the functions $\alpha_1, \alpha_2, \alpha_3$ take the form $\alpha_i(r) = c_i r^2$, with $c_i > 0$, then the system exhibits global exponential stability, with the state norm satisfying:

$$\|\mathbf{x}(t)\| \leq M\|\mathbf{x}(0)\|e^{-\gamma t}, \tag{3}$$

for some constants $M, \gamma \in \mathbb{R}_+$.

Furthermore, given a CLF, the set of admissible control inputs that stabilise the system is:

$$\mathcal{U}_{\text{CLF}}(\mathbf{x}) = \{\mathbf{u} \in \mathbb{U} \mid \dot{V}(\mathbf{x}, \mathbf{u}) \leq -\alpha_3(\|\mathbf{x}\|)\}. \tag{4}$$

A feedback controller $\mathbf{k}(\mathbf{x})$ is then defined such that $\mathbf{k}(\mathbf{x}) \in \mathcal{U}_{\text{CLF}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{X}$, ensuring stability.

Theorem 1. *(Stability via CLFs). Let $V : \mathbb{X} \to \mathbb{R}_+$ be a continuously differentiable CLF satisfying the conditions in Definition 2. If there exists a locally Lipschitz continuous control law $\mathbf{k} : \mathbb{X} \to \mathbb{U}$ such that:*

$$\dot{V}(\mathbf{x}, \mathbf{k}(\mathbf{x})) \leq -\alpha_3(\|\mathbf{x}\|), \tag{5}$$

for all $\mathbf{x} \in \mathbb{X}$, then:

1. *The origin $\mathbf{x} = 0$ is globally asymptotically stable,*
2. *$\alpha_1,\ \alpha_2,\ \alpha_3$ are quadratic functions, then the system is globally exponentially stable.*

Importantly, the set $\mathcal{U}_{\text{CLF}}(\mathbf{x})$ is described by an affine inequality in $\mathbf{u}$ due to the control-affine structure of the system dynamics:

$$\dot{V}(\mathbf{x}, \mathbf{u}) = (\nabla_{\mathbf{x}} V)^\top \dot{\mathbf{x}} = \frac{\partial V}{\partial \mathbf{x}}(\mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}). \tag{6}$$

This structure allows for the formulation of an optimisation-based control strategy using quadratic programming [11]. A standard quadratic program for determining the control input $\mathbf{k}(\mathbf{x})$ is given by:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \arg\min_{\mathbf{u} \in \mathbb{U}} \ & \frac{1}{2}\mathbf{u}^\top \mathbf{H}\mathbf{u} + \mathbf{f}^\top \mathbf{u} + C \\ \text{s.t. } & \frac{\partial V}{\partial \mathbf{x}}(\mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) \leq -\alpha_3(\|\mathbf{x}\|), \end{aligned} \tag{7}$$

where $\mathbf{H}$ is positive definite, C is a constant independent of $\mathbf{u}$, and $\mathbb{U}$ is assumed to be a convex polytope. The feasibility of this optimisation problem is directly guaranteed by the CLF condition (2).

3.3 Discrete-time Control Barrier Functions

Consider the discrete-time system:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \tag{8}$$

where $\mathbf{x}_k \in \mathbb{X} \subset \mathbb{R}^n$ is the system state at time step $k \in \mathbb{N}$, $\mathbf{u}_k \in \mathbb{U} \subset \mathbb{R}^m$ is the control input, and $\mathbf{f} : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ is a continuous function representing the system dynamics.

A key consideration in control design is enforcing safety, ensuring that the state $\mathbf{x}_k$ remains within a safe set $\mathcal{C} \subset \mathbb{X}$ defined as:

$$\begin{aligned} \mathcal{C} &:= \{\mathbf{x} \in \mathbb{X} \mid h(\mathbf{x}) \geq 0\}, \\ \text{Int}(\mathcal{C}) &:= \{\mathbf{x} \in \mathbb{X} \mid h(\mathbf{x}) > 0\}, \\ \partial\mathcal{C} &:= \{\mathbf{x} \in \mathbb{X} \mid h(\mathbf{x}) = 0\}, \end{aligned} \tag{9}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is a continuous function defining the boundary and interior of $\mathcal{C}$.

Definition 3. *The set $\mathcal{C}$ is forward invariant under* (8) *if, for any initial condition $\mathbf{x}_0 \in \mathcal{C}$, the trajectory satisfies $\mathbf{x}_k \in \mathcal{C}$ for all $k \in \mathbb{N}$.*

Definition 4. *(Discrete-Time Barrier Function [23]). A function $h : \mathbb{R}^n \to \mathbb{R}$ is a discrete-time barrier function (DTBF) for system* (8) *if there exists a class $\mathcal{K}$ function α satisfying $\alpha(r) < r$ for all $r > 0$ such that:*

$$\Delta h(\mathbf{x}_k, \mathbf{u}_k) \geq -\alpha(h(\mathbf{x}_k)), \quad \forall\, \mathbf{x}_k \in \mathbb{X}, \tag{10}$$

where the difference operator is defined as:

$$\Delta h(\mathbf{x}_k, \mathbf{u}_k) = h(\mathbf{x}_{k+1}) - h(\mathbf{x}_k). \tag{11}$$

As shown in [27], if h satisfies (10), then the safe set $\mathcal{C}$ remains forward invariant. A controller can then be designed to satisfy (10), ensuring safety via discrete-time Control Barrier Functions.

Definition 5. *(Discrete-time CBF [23]). A function $h : \mathbb{R}^n \to \mathbb{R}$ is a DTCBF if it is a discrete-time barrier function and can be used to enforce safety constraints via control design.*

Theorem 2. *([23]) Let $\mathcal{C} \subseteq \mathbb{X} \subset \mathbb{R}^n$ be defined as in* (9)*. If h is a discrete-time CBF, then any control input $\mathbf{u}_k$ satisfying* (10) *ensures that $\mathcal{C}$ remains forward invariant, thus guaranteeing safety.*

The function α in (10) can be chosen as a linear function $\alpha(h) = \gamma h$ with $0 < \gamma \leq 1$. Substituting this into (10), the following is obtained:

$$h(\mathbf{x}_k) \geq (1-\gamma)^k h(\mathbf{x}_0), \quad \forall\, k \in \mathbb{N}. \tag{12}$$

For $0 < \gamma \leq 1$, this ensures exponential convergence of $h(\mathbf{x}_k)$ towards zero, guaranteeing that the system remains within the safe set $\mathcal{C}$. This formulation corresponds to the discrete-time exponential Control Barrier Function (ECBF), as discussed in [28].

3.4 Input-Output Linearisation

Input-output linearisation is a nonlinear control technique that transforms the dynamics relating the system's input to its output into a linear form, typically a chain of integrators [29]. The following formalism is based on standard results in nonlinear control theory, adapted from sources such as [29], [30], [31].

Consider a configuration space $\mathcal{Q} \subseteq \mathbb{R}^n$ and an input space $\mathcal{U} \subseteq \mathbb{R}^m$.

Assumption 2. *Assume that $\mathcal{Q}$ is a non-empty, path-connected manifold.*

A general n-degree-of-freedom mechanical system can be described by the following Euler-Lagrange equations:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \underbrace{\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})}_{\mathbf{h}(\mathbf{q},\dot{\mathbf{q}})} = \mathbf{B}_{sys}\mathbf{u}, \tag{13}$$

where $\mathbf{q} \in \mathcal{Q}$ represents the generalised coordinates, $\dot{\mathbf{q}} \in T\mathcal{Q}$ (the tangent space of $\mathcal{Q}$) are the generalised velocities, and $\mathbf{u} \in \mathcal{U}$ is the control input vector. The inertia matrix $\mathbf{M} : \mathcal{Q} \to \mathbb{S}^n_{++}$ is symmetric and positive definite, while $\mathbf{C} : \mathcal{Q} \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ represents Coriolis and centrifugal effects. The gravitational forces are denoted by $\mathbf{G} : \mathcal{Q} \to \mathbb{R}^n$, and the static input matrix $\mathbf{B}_{sys} \in \mathbb{R}^{n\times m}$ is the input matrix that maps the control inputs to generalised forces/torques.

For a system output defined as $\mathbf{y}(\mathbf{q}) \in \mathbb{R}^k$, with $k \leq m$ (typically $k = m$ for full input-output linearisation of square systems, or $k < m$ for non-square systems where $m - k$ inputs might be used for other objectives), the following assumption is imposed:

Assumption 3. *Each component y_j of the output vector $\mathbf{y}$ has a relative degree of 2 with respect to the input $\mathbf{u}$ in a domain of interest $\mathcal{R} \subseteq \mathcal{Q}$ [30]. This implies that the second time derivative of $\mathbf{y}$ explicitly depends on the input $\mathbf{u}$, and no lower derivative does.*

Given a desired trajectory $\mathbf{y}_d : \mathcal{I} \to \mathbb{R}^k$ over a time interval $\mathcal{I} = [t_0, t_f]$, where $t_f > t_0$, the tracking error is $\mathbf{e}(t) = \mathbf{y}(\mathbf{q}(t)) - \mathbf{y}_d(t)$. The error dynamics are often analysed using an auxiliary error state:

$$\boldsymbol{\eta}(\mathbf{q},\dot{\mathbf{q}},t) = \begin{bmatrix}\mathbf{e}(t)\\ \dot{\mathbf{e}}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{y}(\mathbf{q}) - \mathbf{y}_d(t)\\ \dot{\mathbf{y}}(\mathbf{q},\dot{\mathbf{q}}) - \dot{\mathbf{y}}_d(t)\end{bmatrix}. \tag{14}$$

with $\mathbf{r}(t) = \begin{bmatrix}\mathbf{y}_d(t)^\top & \dot{\mathbf{y}}_d(t)^\top\end{bmatrix}^\top$.
Differentiating Eq. (14) and substituting Eq. (13):

$$\dot{\boldsymbol{\eta}}(\mathbf{q},\dot{\mathbf{q}},t) = \mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) - \dot{\mathbf{r}}(t) + \mathbf{g}(\mathbf{q})\mathbf{u}, \tag{15}$$

where the drift and input matrices are given by:

$$\mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix}\frac{\partial \mathbf{y}}{\partial \mathbf{q}}\dot{\mathbf{q}}\\ \frac{d}{dt}\left(\frac{\partial \mathbf{y}}{\partial \mathbf{q}}\right)\dot{\mathbf{q}} - \frac{\partial \mathbf{y}}{\partial \mathbf{q}}\mathbf{M}(\mathbf{q})^{-1}\mathbf{h}(\mathbf{q},\dot{\mathbf{q}})\end{bmatrix}, \quad \mathbf{g}(\mathbf{q}) = \begin{bmatrix}\mathbf{0}_{k\times m}\\ \frac{\partial \mathbf{y}}{\partial \mathbf{q}}\mathbf{M}(\mathbf{q})^{-1}\mathbf{B}_{sys}\end{bmatrix}. \tag{16}$$

For all $\mathbf{q} \in \mathcal{Q}$, Assumption 3 ensures that $\mathbf{g}(\mathbf{q})$ (known as the decoupling matrix) is full rank. Defining the feedback transformation:

$$\mathbf{u}(\mathbf{q},\dot{\mathbf{q}},t) = \widetilde{\mathbf{g}}(\mathbf{q})^\dagger\big(-\widetilde{\mathbf{f}}(\mathbf{q},\dot{\mathbf{q}}) + \ddot{\mathbf{y}}_d(t) + \boldsymbol{\nu}(\mathbf{q},\dot{\mathbf{q}},t)\big), \tag{17}$$

$$\widetilde{\mathbf{f}}(\mathbf{q},\dot{\mathbf{q}}) = \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{q}}\dot{\mathbf{q}} - \frac{\partial \mathbf{y}}{\partial \mathbf{q}}(\mathbf{M}^{-1}\mathbf{H}), \tag{18}$$

$$\widetilde{\mathbf{g}}(\mathbf{q}) = \frac{\partial \mathbf{y}}{\partial \mathbf{q}}\mathbf{M}^{-1}\mathbf{B}_{sys}, \tag{19}$$

where $\boldsymbol{\nu} \in \mathbb{R}^k$ is an auxiliary stabilising input, generates the following input-output linear system:

$$\dot{\boldsymbol{\eta}} = \mathbf{F}\boldsymbol{\eta} + \mathbf{G}\boldsymbol{\nu}, \tag{20}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{k\times k} & \mathbf{I}_k \\ \mathbf{0}_{k\times k} & \mathbf{0}_{k\times k} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{0}_{k\times k} \\ \mathbf{I}_k \end{bmatrix}. \tag{21}$$

Since $(\mathbf{F}, \mathbf{G})$ form a controllable pair, a feedback gain $\mathbf{K} = \begin{bmatrix} \mathbf{K}_p & \mathbf{K}_d \end{bmatrix}$ can be selected, where $\mathbf{K}_p,\ \mathbf{K}_d \in \mathbb{S}^k_{++}$, to define:

$$\boldsymbol{\nu} = -\mathbf{K}\boldsymbol{\eta}. \tag{22}$$

The resulting closed-loop system is given by:

$$\dot{\boldsymbol{\eta}} = \mathbf{A}_{cl}\boldsymbol{\eta}, \quad \mathbf{A}_{cl} = \mathbf{F} - \mathbf{G}\mathbf{K}, \tag{23}$$

where $\mathbf{A}_{cl}$ is Hurwitz, ensuring exponential stability of $\mathbf{y}_d$. Consequently, a Control Lyapunov Function (CLF) can be constructed using the Continuous Algebraic Riccati Equation (CARE) [29], guaranteeing stability in the control design.

3.5 Algebraic Graph Theory

The following formalism follows from [9]. Interactions among robots are represented using graphs. Let $G = (V, E, \mathbf{A})$ be a directed graph, where $V = 1, 2, \ldots, n$ is the set of nodes, $E \subseteq V \times V$ is the set of directed edges, and $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the adjacency matrix with non-negative elements. An edge (j, i) indicates that a node i can access node j. The element of the adjacency matrix $\mathbf{A} = [a_{ij}]_{n\times n}$ is defined as follows: $a_{ij} > 0 \iff (j, i) \in E$, otherwise $a_{ij} = 0$.

Definition 6. *([9]) The robot i intake is defined as $d_i = \sum_{j=1}^{n} a_{ij}$, and $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix made up of d_i. $\mathbf{D}$ can also be referred to as the degree matrix of the nodes in the graphs. Thus, the Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{L} \in \mathbb{R}^{n\times n}$.*

The virtual leader leading the formation is denoted by R_L, while the follower i is denoted by R_i. A leader-follower adjacency vector $\mathbf{b}$ describes the interaction between the leader and the followers—if the leader is in the neighbourhood of the follower i, then $b_i > 0$, otherwise $b_i = 0$; that is, the follower's access to the virtual leader's information can be expressed by a vector of weights, i.e., $\mathbf{b} = [b_1, b_2, \ldots, b_n]^\top$, and the modified Laplacian matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$ can be defined as $\boldsymbol{\Gamma} = \mathbf{L} + \mathbf{B}$, where $\mathbf{B} = \text{diag}(\mathbf{b})$ is the leader-follower adjacency matrix.

4 Methodology

4.1 Wheeled Mobile Robot Modelling

This section presents the kinematic and dynamic equations of a single differential drive mobile robot.

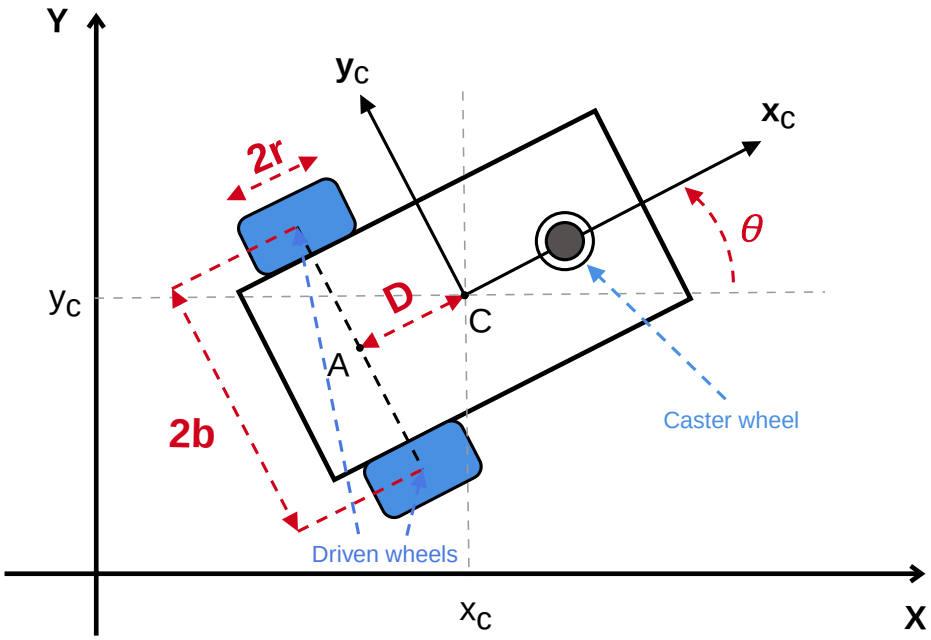

Fig. 1. Nonholonomic WMR

Assumption 4. *The WMR has two active wheels mounted on one axis and a front passive wheel.*

The vehicle configuration is described by three generalised coordinates: the Cartesian coordinates $(x_c,\ y_c)$ of the centre of mass measured in the fixed reference frame and the angle θ describing the platform's orientation relative to the x-axis (see Fig. 1). The state vector is thus $q = [x_c,\ y_c,\ \theta]^\top$.

4.1.1 Kinematics Model

This section is based on the work from [15].

Assumption 5. *The robot can only move along the vehicle's longitudinal axis, and the wheels roll without slipping in the horizontal plane.*

Assumption 5 describes a kinematic constraint called a nonholonomic constraint. Let the mobile robot be in the configuration described by $q(k) = [x_c(k),\ y_c(k),\ \theta(k)]^\top$. The pure rolling and non-slipping constraint can be expressed in the Pfaffian form [32] as:

$$\dot{x}_c \sin(\theta) - \dot{y}_c \cos(\theta) + D\dot{\theta} = 0, \tag{24}$$

where $\dot{x}_c$ and $\dot{y}_c$ are the robot velocity components in the Cartesian plane, $\dot{\theta}$ is the angular velocity, and D is the distance between A (the midpoint of the segment joining the centres of the two wheels) and C (the robot's centre of mass) (see Fig. 1).

The above equations can be written in the form:

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = 0, \tag{25}$$

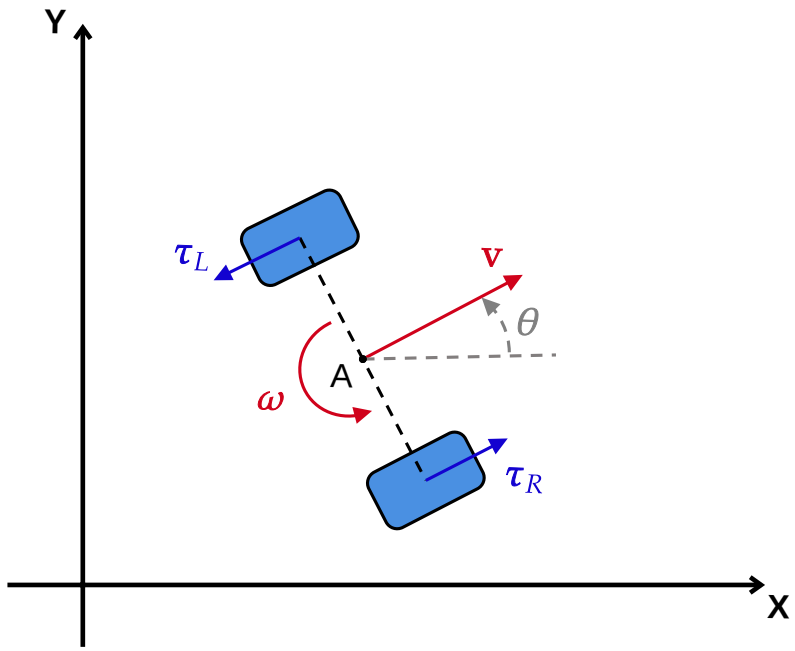

Fig. 2. Kinematic model of a differential drive robot

where $\mathbf{A}(\mathbf{q}) = \begin{bmatrix} \sin(\theta) & -\cos(\theta) & D \end{bmatrix}$ and $\dot{\mathbf{q}} = \begin{bmatrix} \dot{x}_c & \dot{y}_c & \dot{\theta} \end{bmatrix}^\top$ denotes the velocity vector. Thus, the kinematic model of the system can be written as follows:

$$\dot{\mathbf{q}}(k) = \mathbf{S}(\mathbf{q})\Phi, \tag{26}$$

where $\Phi = \begin{bmatrix} \dot{\phi}_r & \dot{\phi}_l \end{bmatrix}^\top$ is the vector of the angular velocities of the right and left wheels (see Fig. 2). To derive the matrix $\mathbf{S}(\mathbf{q})$, the position of the centre of mass relative to A can be written as follows:

$$\begin{aligned} x_c(k) &= x_a + D\cos(\theta), \\ y_c(k) &= y_a + D\sin(\theta), \\ \theta(k) &= \theta(k). \end{aligned} \tag{27}$$

Differentiating Eq. (27) relative to time yields:

$$\dot{\mathbf{q}}(k) = \begin{bmatrix} \cos(\theta) & -D\sin(\theta) \\ \sin(\theta) & D\cos(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{28}$$

where v and ω are the linear and angular velocities of the robot, respectively. According to [32], the linear and angular velocities of the robot in its local frame can be obtained as:

$$v = r\frac{\dot{\phi}_r + \dot{\phi}_l}{2}, \tag{29}$$

$$\omega = r\frac{\dot{\phi}_r - \dot{\phi}_l}{2b}, \tag{30}$$

where r is the radius of the driving wheels, and b is half the distance between the two wheel centres.

Substituting equations (29) in (28), the kinematics model of the WMR is expressed by the following:

$$\dot{\mathbf{q}}(k) = \frac{r}{2b}\begin{bmatrix} b\cos(\theta) - D\sin(\theta) & b\cos(\theta) + D\sin(\theta) \\ b\sin(\theta) + D\cos(\theta) & b\sin(\theta) - D\cos(\theta) \\ 1 & -1 \end{bmatrix}\mathbf{\Phi}. \tag{31}$$

4.1.2 Dynamics Model

This section is based on the work from [9], [15].

The Lagrange formulation is used to obtain the dynamic model of the WMR system subject to the kinematic constraints in Eq. (25). A mechanical system's Lagrangian $\mathcal{L}$ is the difference between its kinetic and potential energy. The Lagrange equations, in this case, are:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{B}_{sys}(\mathbf{q})\boldsymbol{\tau} + \mathbf{A}^{\top}(\mathbf{q})\lambda, \tag{32}$$

where $\mathbf{B}_{sys}(\mathbf{q})$ is the input transformation matrix, $\boldsymbol{\tau} = [\tau_r,\ \tau_l]^{\top}$ represents the actuating torques of the right and left wheels, and λ is a vector of Lagrangian multipliers. The term $\mathbf{A}^{\top}(\mathbf{q})\lambda$ represents the vector of reaction forces at the generalised coordinate level. The kinetic energy of the system can be written as:

$$T = \frac{1}{2}\dot{\mathbf{q}}^{\top}\mathbf{M}(\mathbf{q})\dot{\mathbf{q}} \tag{33}$$

where

$$\mathbf{M}(\mathbf{q}) = \text{diag}([m, m, J]),$$

$$m = (m_p + 2m_w),$$

$$J = J_{zp} + 2(J_{zw} + m_w(D^2 + b^2)),$$

where m is the total mass of the robot, m_p is the mass of the platform without the driven wheels and actuators, m_w is the mass of the wheels and their actuators, J is the moment of inertia of the robot about the centre of mass, J_{zp} is the moment of inertia of the platform, and J_{zw} is the moment of inertia of the wheels.

The robot cannot move along the z-axis; thus, the system's potential energy is defined to be zero, and the Lagrangian function is equal to the kinetic energy.

By solving the Lagrangian, the following matrix equation of motion of the WMR is obtained:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}_{sys}(\mathbf{q})\boldsymbol{\tau} + \mathbf{A}^{\top}(\mathbf{q})\lambda, \tag{34}$$

where

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{0}_{3\times 3}, \quad \mathbf{B}_{sys}(\mathbf{q}) = \frac{1}{r}\begin{bmatrix}\cos(\theta) & \cos(\theta)\\ \sin(\theta) & \sin(\theta)\\ b & -b\end{bmatrix}, \quad \mathbf{A}^{\top}(\mathbf{q}) = \begin{bmatrix}\sin(\theta)\\ -\cos(\theta)\\ D\end{bmatrix}.$$

Since the columns of $\mathbf{S}(\mathbf{q})$ span the null space of $\mathbf{A}(\mathbf{q})$, it follows that it follows that $\mathbf{S}^{\top}(\mathbf{q})\mathbf{A}^{\top}(\mathbf{q}) = \mathbf{0}$; thus, the velocity vector $\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\Phi$ inherently satisfies the nonholonomic constraint $\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = 0$ for any Φ. Hence, the Lagrange multipliers can be systematically eliminated by projecting Eq. (34) onto the constraint-consistent subspace spanned by $\mathbf{S}(\mathbf{q})$ using $\mathbf{S}^{\top}(\mathbf{q})$:

$$\overline{\mathbf{M}}(\mathbf{q})\dot{\Phi} + \overline{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\Phi = \overline{\mathbf{B}}_{sys}(\mathbf{q})\boldsymbol{\tau}, \tag{35}$$

where

$$\begin{aligned}
\overline{\mathbf{M}}(\mathbf{q}) &= \mathbf{S}^{\top}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{S}(\mathbf{q}) = \left(\frac{r}{2b}\right)^2\begin{bmatrix} m(b^2+D^2)+J & m(b^2-D^2)-J\\ m(b^2-D^2)-J & m(b^2+D^2)+J\end{bmatrix},\\
\overline{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}}) &= \mathbf{S}^{\top}(\mathbf{q})\left(\mathbf{M}(\mathbf{q})\dot{\mathbf{S}}(\mathbf{q}) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\mathbf{S}(\mathbf{q})\right) = -2mbD\left(\frac{r}{2b}\right)^2\mathbf{S}_z(\omega),\\
\overline{\mathbf{B}}_{sys}(\mathbf{q}) &= \mathbf{S}^{\top}(\mathbf{q})\mathbf{B}_{sys}(\mathbf{q}) = \mathbf{I}_2,
\end{aligned}$$

where $\mathbf{S}_z(\omega)$ is a skew-symmetric matrix representing the cross-product operation.

Therefore, Eq. (35) captures the dynamics projected onto the admissible subspace defined by the constraint distribution. The motion governed by this model is inherently restricted to directions that satisfy the nonholonomic constraint, ensuring that the constraint is respected implicitly throughout the system's evolution.

By considering the state variable,

$$x = \begin{bmatrix} x_c\\ y_c\\ \theta\\ \dot{\phi}_r\\ \dot{\phi}_l\end{bmatrix} = \begin{bmatrix}\mathbf{q}\\ \boldsymbol{\Phi}\end{bmatrix},$$

the equation of motion in the control-affine, nonlinear state-space form can be represented as follows:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{36}$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix}\mathbf{S}(\mathbf{q})\\ -\overline{\mathbf{M}}(\mathbf{q})^{-1}\overline{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\end{bmatrix}\Phi, \quad \mathbf{g}(\mathbf{x}) = \begin{bmatrix}\mathbf{0}_{3\times 2}\\ \overline{\mathbf{M}}(\mathbf{q})^{-1}\overline{\mathbf{B}}_{sys}(\mathbf{q})\end{bmatrix}, \quad \mathbf{u} = \boldsymbol{\tau},$$

for all $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^5$ and $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^2$.

4.2 Formation Control

Formation control ensures that a group of mobile robots maintains a desired geometric configuration relative to a virtual leader while accounting for communication constraints, collision avoidance, and dynamic feasibility. The proposed consensus-based approach integrates input-output linearisation with constrained NMPC to achieve robust formation tracking.

The virtual leader in the mobile robot system follows a kinematic model analogous to Eq. (26), given by:

$$\dot{\mathbf{q}}_L(k) = \mathbf{R}(\mathbf{q}_L)\mathbf{v}_L, \tag{37}$$

where $\mathbf{q}_L = [x_L, y_L, \theta_L]^\top$ represents its position and orientation in Cartesian coordinates, and $\mathbf{v}_L$ denotes the vector of its linear and angular velocities.

Assumption 6. *The communication topology forms a directed spanning tree, where the virtual leader is the root node.*

Assumption 7. *There is no communication delay between the robots.*

Assumption 8. *At least one follower robot receives information directly from the virtual leader.*

The objective of the formation control strategy is to regulate the positions of the mobile robots $\mathbf{P}_i$ so that they conform to a predefined geometric configuration relative to the virtual leader. The desired formation shape is represented by the desired displacements $\Delta_i = [\Delta_{ix}, \Delta_{iy}]^\top$ of each robot $i = 1, 2, \ldots, n$ relative to the virtual leader.

The proposed control scheme assumes a hierarchical structure, where a subset of followers has direct access to leader information, while others rely on local communication with neighbouring robots. A formation reference system is established, which incorporates collision avoidance constraints to prevent deadlock when combined with the dynamics controller. Let $\tilde{\mathbf{P}}_i = [\tilde{x}_i, \tilde{y}_i]^\top$ denote the optimal position of the i-th robot in the formation, computed by the solver. Similarly, $\mathbf{P}_L = [x_L, y_L]^\top$ represents the virtual leader's position and $\mathbf{P}_j = [x_j, y_j]^\top$ encapsulates the positions of all neighbouring robots distributed through the communication network.

Assumption 9. *Robots within the same communication neighbourhood share their state information and optimal control inputs as computed by the consensus algorithm at the first node.*

Over a given time interval $\mathcal{I} = [t_0, t_f], t_f > t_0$, the distributed formation tracking error for robot i is formulated as the following consensus algorithm to ensure the positions of the robots converge to the desired formation:

$$\mathbf{e}_i(t) = \sum_{j=1}^{n} \{a_{ij}([\tilde{\mathbf{P}}_i(t) - \mathbf{P}_j(t)] - \Delta_{ij}(t))\} + b_i([\tilde{\mathbf{P}}_i(t) - \mathbf{P}_L(t)] - \Delta_i(t)), \tag{38}$$

where $\Delta_{ij}(t) = \Delta_i(t) - \Delta_j(t)$ represents the desired inter-robot relative displacements in the formation.

It is important to note that $\tilde{\mathbf{P}}_i(t) = \mathbf{P}_j(t) \iff i = j$, otherwise they are not equal.

Using graph theory, as discussed in Section 3.5, the formation error can be expressed in terms of the adjacency matrix **A**, the leader-follower adjacency matrix **B**, and the modified Laplacian matrix

$\mathbf{\Gamma}$ for interrobot and leader-follower interactions, respectively. The formation error $\mathbf{e}_i(t)$ captures the deviation of the robot's position from its desired formation configuration relative to both the virtual leader and its neighbouring robots. Thus, the consensus formation errors of the robots in the communication network can be written as:

$$\mathbf{e}(t) = (\mathbf{P}(t) - \mathbf{\Delta}(t))\mathbf{\Gamma} - \overline{\mathbf{P}}_L(t)\mathbf{B}, \tag{39}$$

where $\mathbf{P}(t) \in \mathbb{R}^{2\times n} \triangleq \begin{bmatrix}\mathbf{P}_1(t) & \mathbf{P}_2(t) & \dots & \mathbf{P}_n(t)\end{bmatrix}$, $\mathbf{\Delta}(t) \in \mathbb{R}^{2\times n} \triangleq \begin{bmatrix}\Delta_1(t) & \Delta_2(t) & \dots & \Delta_n(t)\end{bmatrix}$, $\overline{\mathbf{P}}_L(t) \in \mathbb{R}^{2\times n} \triangleq \mathbf{P}_L(t)\mathbf{1}_{1\times n}$, and $\mathbf{e}(t) \triangleq \begin{bmatrix}\mathbf{e}_1(t) & \mathbf{e}_2(t) & \dots & \mathbf{e}_n(t)\end{bmatrix}$.

Following Eq. (39), we can rewrite Eq. (38) as the following matrix equation:

$$\mathbf{e}_i(t) = -(\mathbf{P}(t)\mathbf{A} + \mathbf{\Delta}(t)\mathbf{\Gamma} + \overline{\mathbf{P}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\tilde{\mathbf{P}}_i(t), \tag{40}$$

where $\hat{b}_i \in \mathbb{R}^n$ is the i-th standard basis vector of the n-dimensional Euclidean space, $\mathbf{g}^{dis} \in \mathbb{R}^{n\times n} \triangleq \mathbf{D} + \mathbf{B}$, and $g_i^{dis} \in \mathbb{R} \triangleq \mathbf{g}^{dis}(i,i)$.

Assumption 10. *Assuming that the virtual leader's trajectory and those of neighbouring robots are sufficiently smooth, the formation error $\mathbf{e}_i(t)$ is at least twice continuously differentiable in some domain $\mathcal{E} \subseteq \mathbb{R}^2$.*

Under assumption 10, the time derivative of the formation error can be expressed as follows, representing the consensus-based formation error dynamics:

$$\dot{\mathbf{e}}_i(t) = -\left(\dot{\mathbf{P}}(t)\mathbf{A} + \dot{\mathbf{\Delta}}(t)\mathbf{\Gamma} + \dot{\overline{\mathbf{P}}}_L(t)\mathbf{B}\right)\hat{b}_i + g_i^{dis}\dot{\tilde{\mathbf{P}}}_i(t). \tag{41}$$

From Eq. (35), the wheel dynamics of the i-th robot is formulated as follows:

$$\dot{\mathbf{\Phi}}_i = -\overline{\mathbf{M}}(\mathbf{q}_i)^{-1}\overline{\mathbf{C}}(\mathbf{q}_i, \dot{\mathbf{q}}_i)\mathbf{\Phi}_{\boldsymbol{i}} + \overline{\mathbf{M}}(\mathbf{q}_i)^{-1}\overline{\mathbf{B}}_{sys}(\mathbf{q}_i)\boldsymbol{\tau}_i. \tag{42}$$

Eq. (42) satisfies the conditions for input-output linearisation [29], thus it is input-output linearisable. To linearise the wheel dynamics, we define the control torque as follows, following the formulation described in Section 3.4:

$$\boldsymbol{\tau}_i(\mathbf{x}_i, t) = \overline{\mathbf{B}}_{sys}(\mathbf{q}_i)\overline{\mathbf{M}}(\mathbf{q}_i)\left(\overline{\mathbf{M}}(\mathbf{q}_i)^{-1}\overline{\mathbf{C}}(\mathbf{q}_i, \dot{\mathbf{q}}_i)\mathbf{\Phi}_{\boldsymbol{i}} + \boldsymbol{\nu}_i(\mathbf{x}_i, t)\right), \tag{43}$$

where $\boldsymbol{\nu}_i(\mathbf{x}_i, t)$ is an auxiliary control input that will be designed to stabilise the wheel dynamics.

Substituting the control torque into Eq. (42), we obtain the following first-order input-output linear form of the wheel dynamics:

$$\dot{\mathbf{\Phi}}_i = \boldsymbol{\nu}_i(\mathbf{x}_i, t). \tag{44}$$

The system's dynamic model presented in Eq. (36) takes the wheel torques as the control input, thus the formation error $\mathbf{e}_i$ has a relative degree of 2, and the second derivative of the robot's position is defined as follows:

$$\ddot{\tilde{\mathbf{P}}}_i(t) = \mathbf{S}_z(\omega_i)\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_{\boldsymbol{i}} + \mathbf{S}_y(\mathbf{q}_i)\boldsymbol{\nu}_i(\mathbf{x}_i, t), \tag{45}$$

where $\mathbf{S}_y(\mathbf{q}_i) = \begin{bmatrix}\mathbf{I}_2 & \mathbf{0}_{2\times 1}\end{bmatrix}\mathbf{S}(\mathbf{q_i})$.

From Eq. (41), we can define the second derivative of the consensus-based formation error as follows:

$$\ddot{\mathbf{e}}_i(t) = -(\ddot{\mathbf{P}}(t)\mathbf{A} + \ddot{\mathbf{\Delta}}(t)\mathbf{\Gamma} + \ddot{\overline{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\dddot{\tilde{\mathbf{P}}}_i(t). \tag{46}$$

Remark 1. *Eq.* (36) *establishes a relationship between the wheel torques and body frame velocities of the robot in the global frame. Thus, the formation tracking errors do not need to be mapped to the robot's local frame.*

Let's define an auxiliary state space for the consensus problem using the auxiliary state $\boldsymbol{\eta}_i(\mathbf{x}_i, t) = \begin{bmatrix}\mathbf{e}_i(t) & \dot{\mathbf{e}}_i(t)\end{bmatrix}^\top$ as follows:

$$\begin{aligned}\dot{\boldsymbol{\eta}}_i &= \frac{\partial}{\partial t}\begin{bmatrix}\mathbf{e}_i(t)\\ \dot{\mathbf{e}}_i(t)\end{bmatrix}\\ &= \underbrace{-\left(\begin{bmatrix}\dot{\mathbf{P}}(t)\\ \ddot{\mathbf{P}}(t)\end{bmatrix}\mathbf{A} + \begin{bmatrix}\dot{\mathbf{\Delta}}(t)\\ \ddot{\mathbf{\Delta}}(t)\end{bmatrix}\mathbf{\Gamma} + \begin{bmatrix}\dot{\overline{\mathbf{P}}}_L(t)\\ \ddot{\overline{\mathbf{P}}}_L(t)\end{bmatrix}\mathbf{B}\right)\hat{b}_i}_{\mathbf{l}(\mathbf{x}_i,t)} + \underbrace{g_i^{dis}\begin{bmatrix}\mathbf{I}_2\\ \mathbf{S}_z(\omega_i)\end{bmatrix}\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi_i}}_{\mathbf{m}(\mathbf{x},t)}\\ &\quad + \underbrace{g_i^{dis}\begin{bmatrix}\mathbf{0}_{2\times 2}\\ \mathbf{S}_y(\mathbf{q}_i)\end{bmatrix}}_{\mathbf{n}(\mathbf{x}_i,t)}\boldsymbol{\nu}(\mathbf{x}_i,t).\end{aligned} \tag{47}$$

The complete derivation of the above dynamics is presented in Appendix A.

The auxiliary system dynamics in Eq. (47) are in a control-affine form with respect to $\boldsymbol{\nu}_i(\mathbf{x}_i, t)$. Specifically, the second block-row, which defines $\ddot{\mathbf{e}}_i(t)$, is affine in $\boldsymbol{\nu}_i$. To achieve input-output linearisation for the error dynamics $\boldsymbol{\eta}_i$, such that $\ddot{\mathbf{e}}_i(t) = \boldsymbol{\xi}_i(\mathbf{x}_i, t)$ (where $\boldsymbol{\xi}_i$ is a new, higher-level auxiliary input), we define the components affecting $\ddot{\mathbf{e}}_i(t)$ from Eq. (47) as:

$$\begin{aligned}\widetilde{\mathbf{l}}(\mathbf{x}_i,t) &= -(\ddot{\mathbf{P}}(t)\mathbf{A} + \ddot{\mathbf{\Delta}}(t)\mathbf{\Gamma} + \ddot{\overline{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i,\\ \widetilde{\mathbf{m}}(\mathbf{x}_i,t) &= g_i^{dis}\mathbf{S}_z(\omega_i)\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi_i},\\ \widetilde{\mathbf{n}}(\mathbf{x}_i,t) &= g_i^{dis}\mathbf{S}_y(\mathbf{q}_i).\end{aligned}$$

The input-output linearising control law for $\boldsymbol{\nu}_i(\mathbf{x}_i, t)$ is then chosen as:

$$\boldsymbol{\nu}(\mathbf{x}_i,t) = \widetilde{\mathbf{n}}(\mathbf{x}_i,t)^\dagger\big(-\widetilde{\mathbf{l}}(\mathbf{x}_i,t) - \widetilde{\mathbf{m}}(\mathbf{x}_i,t) + \boldsymbol{\xi}(\mathbf{x}_i,t)\big), \tag{48}$$

with auxiliary control input $\boldsymbol{\xi}(\mathbf{x}_i, t) \in \mathbb{R}^2$ for all $\mathbf{x}_i \in \mathcal{X}$, $t \in \mathcal{I}$.

Substituting Eq. (48) into Eq. (47) yields the linearised second-order system for the error dynamics:

$$\dot{\boldsymbol{\eta}}_i(\mathbf{x}_i,t) = \begin{bmatrix}\dot{\mathbf{e}}_i(t)\\ \boldsymbol{\xi}_i(\mathbf{x}_i,t)\end{bmatrix}. \tag{49}$$

To express this in the standard form $\dot{\boldsymbol{\eta}}_i = \overline{\mathbf{H}}\boldsymbol{\eta}_i + \overline{\mathbf{G}}\boldsymbol{\xi}_i$, we note that $\dot{\mathbf{e}}_i(t)$ is the second component of $\boldsymbol{\eta}_i$. Thus,

$$\dot{\boldsymbol{\eta}}(\mathbf{x}_i, t) = \underbrace{\begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_2 \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \end{bmatrix}}_{\overline{\mathbf{H}}} \boldsymbol{\eta}(\mathbf{x}_i, t) + \underbrace{\begin{bmatrix} \mathbf{0}_{2\times2} \\ \mathbf{I}_2 \end{bmatrix}}_{\overline{\mathbf{G}}} \boldsymbol{\xi}(\mathbf{x}_i, t), \tag{50}$$

where $(\overline{\mathbf{H}}, \overline{\mathbf{G}})$ form a controllable pair.

To generate stable dynamics, define a feedback-stabilising input as $\boldsymbol{\xi}(\mathbf{x}_i, t) = -\mathbf{K}\boldsymbol{\eta}(\mathbf{x}_i, t)$. $\mathbf{K}$ is then obtained using standard LQR control techniques by defining the following objective function:

$$J_i^c = \int_0^\infty [\boldsymbol{\eta}_i^\top \mathbf{Q}\boldsymbol{\eta}_i + \boldsymbol{\xi}_i^\top \mathbf{R}\boldsymbol{\xi}_i]\, dt, \tag{51}$$

where $\mathbf{Q} \in \mathbb{S}^4_{++}$, and $\mathbf{R} \in \mathbb{S}^2_{++}$, which allows us to obtain a positive definite matrix $\mathbf{P} \in \mathbb{S}^4_{++}$ by defining the following CARE:

$$\mathbf{Q} + \overline{\mathbf{H}}^\top \mathbf{P} + \mathbf{P}\overline{\mathbf{H}} - \mathbf{P}\overline{\mathbf{G}}^\top \mathbf{R}^{-1}\overline{\mathbf{G}}\mathbf{P} = 0. \tag{52}$$

Now, $\mathbf{P}$ can be used to define the quadratic Lyapunov function $V : \mathcal{X} \times \mathcal{I} \to \mathbb{R}$ given by $V(\boldsymbol{\eta}_i, t) = \boldsymbol{\eta}_i^\top \mathbf{P}\boldsymbol{\eta}_i$ with a negative-definite time derivative $\dot{V}(\boldsymbol{\eta}_i, t) = -\boldsymbol{\eta}_i^\top \mathbf{Q}\boldsymbol{\eta}_i$.

Theorem 3. *(Implicit Stability of Zero Dynamics in Consensus-based Formation Control of Nonholonomic WMRs) Consider the input–output linearised system under the proposed consensus-based formation control law:*

$$\dot{\boldsymbol{\eta}}(\mathbf{x}_i, t) = \overline{\mathbf{H}}\boldsymbol{\eta}(\mathbf{x}_i, t) + \overline{\mathbf{G}}\boldsymbol{\xi}(\mathbf{x}_i, t), \tag{53}$$

where $\boldsymbol{\xi}(\mathbf{x}_i, t)$ is the auxiliary control input. Suppose there exists a state feedback control law of the form:

$$\boldsymbol{\xi}(\mathbf{x}_i, t) = -\mathbf{K}\boldsymbol{\eta}(\mathbf{x}_i, t), \tag{54}$$

such that the closed-loop system matrix $\mathbf{A}_{cl} = \overline{\mathbf{H}} - \overline{\mathbf{G}}\mathbf{K}$ is Hurwitz. Then the following conclusions hold:

1. ***Stability of the Zero Dynamics:*** *If the system satisfies the minimum-phase condition (i.e., the zero dynamics are internally stable), stabilising the auxiliary system implicitly ensures stability of the wheel velocity and heading direction dynamics.*
2. ***Propagation of Stability through the Control Mapping:*** *The diffeomorphic transformation*

$$\boldsymbol{\nu} = \widetilde{\mathbf{n}}(\mathbf{x}_i, t)^\dagger \big(-\widetilde{\mathbf{l}}(\mathbf{x}_i, t) - \widetilde{\mathbf{m}}(\mathbf{x}_i, t) + \boldsymbol{\xi}(\mathbf{x}_i, t) \big)$$

   preserves stability, ensuring that $\boldsymbol{\nu}$ is also a stabilising input. Since $\boldsymbol{\nu}$ determines the torque input $\boldsymbol{\tau}$ through another diffeomorphic transformation, stability propagates to the wheel velocity dynamics.
3. ***Convergence of Tracking Errors:*** *The auxiliary state $\boldsymbol{\eta}(\mathbf{x}_i, t)$ asymptotically converges to zero, ensuring that the tracking errors vanish over time.*

4. ***Stability in the Global Frame:*** *Since the tracking error is defined in the global frame, the asymptotic convergence of $\boldsymbol{\eta}(\mathbf{x}_i, t)$ implies that the heading error also asymptotically converges to zero, guaranteeing overall trajectory tracking stability.*

Proof Sketch. By construction, the auxiliary system is a fully actuated linear system controlled via state feedback. If the closed-loop system matrix $\mathbf{A}_{cl}$ is Hurwitz, then $\boldsymbol{\eta}(\mathbf{x}_i, t)$ asymptotically converges to zero. The diffeomorphic transformation ensures that stability propagates through the coordinate change, meaning that $\boldsymbol{\nu}$ and ultimately $\boldsymbol{\tau}$ remain stabilising inputs [29], [30], [31]. Since the displacement error converges to zero, the heading error also converges asymptotically, completing the proof. □

Remark 2. *The validity of this theorem is based on the assumption that the system satisfies the **minimum-phase condition**, meaning that the zero dynamics (which represent the internal states not directly controlled by the output feedback) are asymptotically stable or at least not unstable. Additionally, the diffeomorphic transformations between the control inputs must be well-defined, continuously differentiable, and invertible to ensure that stability propagates correctly through the hierarchy of control laws.*

As we are considering a discrete system, a constrained NMPC-β approach [11] is used to transform this continuous optimal control problem into a convex optimisation problem. The continuous control signal $\boldsymbol{\xi}(\mathbf{x}_i, t)$ in the auxiliary state space is parametrised over subintervals of the prediction horizon $[0,\ T]$. This parameterisation creates a fixed grid of nodes $k \in \{0, \ldots, N\}$ defining control times t_k separated by uniform duration intervals $\delta t = T/(N-1)$. In this report, a zero-order hold discretisation is considered. Denoting $\mathbf{e}_i^k = \mathbf{e}_i(t_k), \boldsymbol{\xi}_i^k = \boldsymbol{\xi}_i(\mathbf{e}_i^k, t_k)$ and integrating the continuous linearised dynamics over an interval leads to the following discrete-time representation of the dynamics:

$$\boldsymbol{\eta}_i^{k+1} = \widehat{\mathbf{H}}\boldsymbol{\eta}_i^k + \widehat{\mathbf{G}}\boldsymbol{\xi}_i^k, \tag{55}$$

where $\widehat{\mathbf{H}}$ is the discrete-time state transition matrix, and $\widehat{\mathbf{G}}$ is the discrete-time input matrix.

Let us define the robot coordinate selection matrix $\mathbf{C}_{CBF} = \begin{bmatrix} I_{2\times 2} & 0_{2\times 3} \end{bmatrix}$ for the robot i at time k. To guarantee safety, a DTCBF can be designed in quadratic form as:

$$h(\mathbf{q}_i(k)) = \|\mathbf{C}_{CBF}\mathbf{x}_i(k) - \mathbf{x}_{obj}^j(k)\|_2 - d_{safe}, \tag{56}$$

where $\mathbf{x}_{obj}^j(k)$ is the global position of the j-th obstacle at a time k, $d_{safe} = R + r_{obs}$ is the safe distance, R is the radius of the minimum bounding circle of the robot, and r_{obs} is the obstacle radius. This DTCBF certifies the system's safety if Eq. (10) is satisfied.

The forward invariance safety condition is enforced through a DTCBF as the following constraint:

$$h_{CBF}(\mathbf{x}_i(k)) = \Delta h(\mathbf{x}_i(k)) + \gamma_i h(\mathbf{x}_i(k)) \geq 0, \tag{57}$$

where $\gamma_i \in [0, 1)$ is the CBF parameter.

The overall NMPC-β controller for the consensus formation control problem is presented below:

$$[\boldsymbol{\Xi}_i^*,\ \gamma_i^*] = \arg\min_{\boldsymbol{\Xi}_i,\ \gamma_i} \quad \beta V(\boldsymbol{\eta}_i^N, t_N) + \zeta(\gamma_i - \overline{\gamma})^2 + \sum_{k=0}^{N-1} (\boldsymbol{\eta}_i^k)^\top \mathbf{Q}\boldsymbol{\eta}_i^k + (\boldsymbol{\xi}_i^k)^\top \mathbf{R}(\boldsymbol{\xi}_i^k) + \rho(\mathbf{v}_i^k - \overline{\mathbf{v}}_i)^2,$$

$$
\begin{aligned}
s.t.\quad & \boldsymbol{\eta}_i^0 = \hat{\boldsymbol{\eta}}_i, \\
& \boldsymbol{\eta}_i^{k+1} = \widehat{\mathbf{H}}\boldsymbol{\eta}_i^k + \widehat{\mathbf{G}}\boldsymbol{\xi}_i^k, \quad k = 0, \ldots, N-1, \\
& \boldsymbol{\nu}_i^k = \widetilde{\mathbf{n}}(\mathbf{x}_i^k, k)^\dagger \big[-\widetilde{\mathbf{l}}(\mathbf{x}_i^k, k) - \widetilde{\mathbf{m}}(\mathbf{x}_i^k, k) + \boldsymbol{\xi}_i^k \big], \quad k = 0, \ldots, N-1, \\
& \boldsymbol{\tau}_i^k = \overline{\mathbf{C}}(\mathbf{q}_i^k, \dot{\mathbf{q}}_i^k)\boldsymbol{\Phi}_i^k + \overline{\mathbf{B}}_{sys}(\mathbf{q}_i^k)\boldsymbol{\nu}_i^k, \quad k = 0, \ldots, N-1, \\
& \boldsymbol{\Phi}_i^{k+1} = \boldsymbol{\Phi}_i^k + \boldsymbol{\nu}_i^k \delta t, \quad k = 0, \ldots, N-1, \\
& \mathbf{v}_i^k = \| \mathbf{S}_y(\mathbf{q}_i^k)\boldsymbol{\Phi}_i^k \|_2, \quad k = 0, \ldots, N-1, \\
& \overline{\mathbf{v}}_i = \| \left(\left(\mathbf{g}^{dis} \right)^{-1} \left(\dot{\mathbf{P}}(t)\mathbf{A} + \dot{\overline{\mathbf{P}}}_L(t)\mathbf{B} \right) \right) \hat{b}_i \|_2 \\
& h_{CBF}(\mathbf{x}_i^k) \geq 0, \quad k = 0, \\
& 0 \leq \gamma \leq 1, \\
& \boldsymbol{\Phi}_i^{lb} \leq \boldsymbol{\Phi}_i^k \leq \boldsymbol{\Phi}_i^{ub}, \quad k = 0, \ldots, N-1, \\
& \boldsymbol{\tau}_i^{lb} \leq \boldsymbol{\tau}_i^k \leq \boldsymbol{\tau}_i^{ub}, \quad k = 0, \ldots, N-1,
\end{aligned} \tag{58}
$$

where $\boldsymbol{\Xi}_i = [(\boldsymbol{\xi}_i^0)^\top, \ldots, (\boldsymbol{\xi}_i^{N-1})^\top]^\top$ is the sequence of auxiliary control inputs for robot i over the prediction horizon. The parameter $\zeta \in \mathbb{R}_+$ is a weighting term that penalises deviations of the CBF parameter γ_i from a desired reference value $\overline{\gamma}$. Further, $\mathbf{v}_i^k$ is the magnitude of the velocity of the i-th robot at the k-th node, $\overline{\mathbf{v}}_i$ is the target consensus velocity magnitude for robot i, and $\rho \in \mathbb{R}_+$ is a weighting term that penalises deviations of the i-th robot's velocity from this target. The terms $\boldsymbol{\Phi}_i^{lb}, \boldsymbol{\Phi}_i^{ub}$ represent the lower and upper bounds on the robot's wheel angular velocities, respectively, while $\boldsymbol{\tau}_i^{lb}, \boldsymbol{\tau}_i^{ub}$ are the lower and upper bounds on the robot's wheel torques. Finally, $\beta \in \mathbb{R}_+$ is a parameter that scales the terminal cost $V(\boldsymbol{\eta}_i^N, t_N)$. As noted in [11], if β is selected large enough, stability can be achieved without specifying a terminal state constraint.

The DTCBF term is applied only at the first step of the prediction horizon ($k = 0$). This simplification is adopted to reduce the computational burden of the optimisation problem solved at each time step. While this ensures that the initial action planned is safe according to the CBF condition, it is a less stringent approach than enforcing the constraint across the entire horizon ($k = 0, \ldots, N-1$). Consequently, the long-term forward invariance of the safe set $\mathcal{C}$ relies on the receding horizon nature of the NMPC, where the safety condition for the immediate next step is re-evaluated and enforced at each control cycle, rather than being guaranteed over the full prediction horizon from a single optimisation.

Interior-point methods (IP) and Sequential Quadratic Programming (SQP) are two popular families of algorithms used to solve general optimisation problems [33]. Additionally, the sparsity of Eq. 58 can be exploited to obtain solutions in real time and at a high sampling rate. The IPOPT solver provided in the MATLAB CasADi library is used to solve the Quadratic Programming problem to get a practical numerical implementation in this report.

This approach extends previous formation control methods [9] by incorporating explicit safety guarantees through a DTCBF [27] while maintaining computational efficiency.

4.3 Algorithmic Implementation of Consensus-Based Formation Control for WMRs

This section provides a high-level overview of the algorithmic implementation of the unified consensus-based formation control framework proposed in this dissertation. The approach integrates formation objectives, consensus principles, and robot dynamics into a single Nonlinear Model Predictive Control (NMPC) scheme. This NMPC controller directly computes the optimal torques required for each robot to achieve and maintain the desired formation while adhering to safety and operational constraints, as mathematically detailed in Section 4.2.

4.3.1 Unified NMPC for Direct Torque Computation

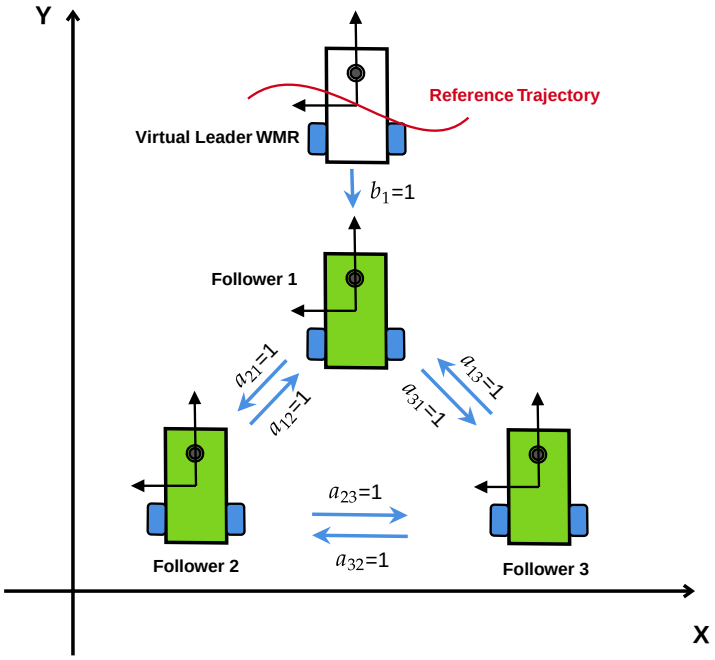

Fig. 3. Communication interaction between robots

The core of the proposed framework is a unified NMPC controller that directly calculates the necessary control torques τ_i for each robot i. Unlike traditional hierarchical approaches that might separate formation planning (e.g., generating reference velocities) from low-level trajectory tracking, this integrated controller considers the consensus-based formation objectives (quantified by the error state $\eta_i(t)$, which is derived from $\mathbf{e}_i(t)$ in Eq. 39) and the complete nonholonomic dynamics of the WMRs (Eq. 36) simultaneously within a single optimisation problem (formulated in Eq. 58). This direct synthesis of control torques eliminates the need for intermediate reference trajectories that a separate controller would then track. Such decoupling can introduce issues like oscillations or performance degradation due to mismatched controller bandwidths or unmodeled interactions, which the unified approach aims to avoid.

By directly optimising for the control torques τ_i based on the current states of all relevant robots in the formation (obtained via the communication network, Fig. 3) and the desired formation geometry, the NMPC inherently accounts for the robots' dynamic responses. The controller, detailed in Eq. 58, optimises a cost function that includes terms for formation error minimisation (via η_i^k), control effort (via the auxiliary input ξ_i^k, which maps to τ_i^k), velocity consensus among

neighbours (weighted by ρ), and a terminal cost $V(\boldsymbol{\eta}_i^N, t_N)$ to promote stability. Safety is explicitly incorporated through DTCBFs as constraints ($h_{CBF}(\mathbf{x}_i^k) \geq 0$), ensuring obstacle avoidance and adherence to other safety-critical limits.

This integrated approach ensures that the computed torques are dynamically feasible and directly contribute to maintaining the formation and satisfying all constraints. The predictive nature of NMPC allows the controller to anticipate future state evolution over a horizon N and proactively adjust torques, leading to smooth and robust formation control even in the presence of disturbances or when navigating complex paths.

A diagram illustrating the mobile communication topology between the robots is provided in Fig. 3.

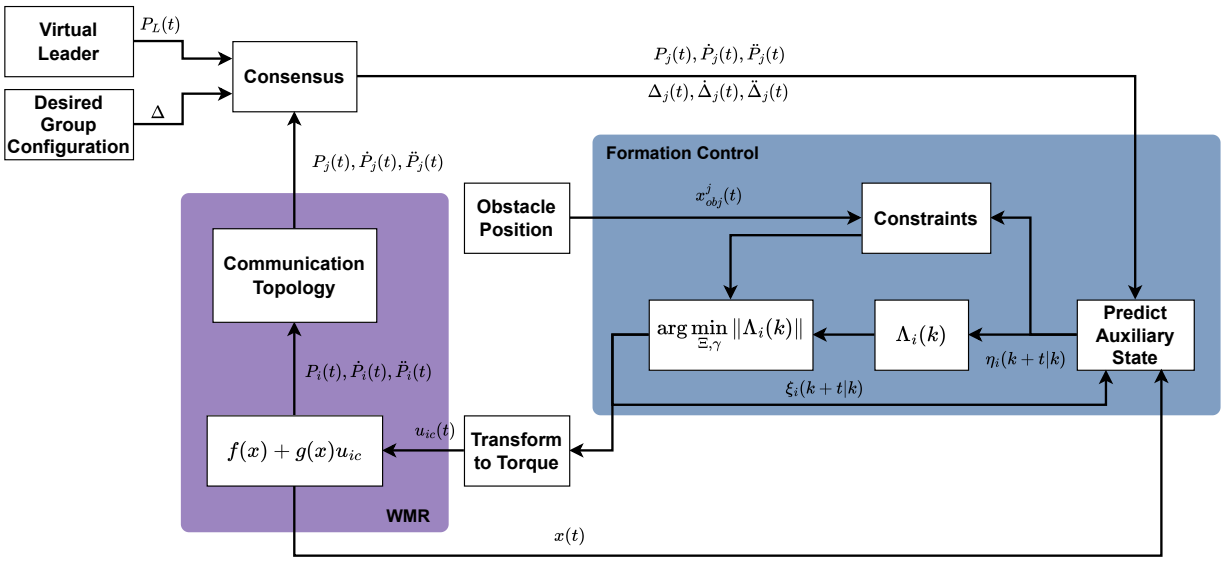

Fig. 4. Consensus NMPC Structure

A conceptual diagram illustrating the overall structure of the unified consensus NMPC is notionally represented in Fig. 4. This figure should depict how each robot's NMPC controller (as per Eq. 58) takes its own estimated state, information from neighbouring robots (via the communication network), and the global formation objectives to compute the optimal control torques directly. This unified structure streamlines the control process compared to multi-layered architectures.

The following algorithm outlines the sequential execution of the formation and trajectory tracking controllers, incorporating the virtual layer.

Algorithm 1 Unified Consensus NMPC Algorithm for Direct Torque Computation

Require: Initial robot states $\mathbf{x}_i(0)$ for $i = 1, \ldots, n$.

Require: Control parameters (e.g., $\mathbf{Q}, \mathbf{R}, \beta, \zeta, \rho, \overline{\gamma}, \hat{\gamma}$ from Eq. 58).

Require: Prediction horizon N, sampling time δt.

Require: Constraints (e.g., $\mathbf{\Phi}_i^{lb}, \mathbf{\Phi}_i^{ub}, \boldsymbol{\tau}_i^{lb}, \boldsymbol{\tau}_i^{ub}$).

Require: Desired formation displacement vectors $\Delta_i(t)$.

Require: Communication topology (adjacency matrices $\mathbf{A}, \mathbf{B}$).

$t_k \leftarrow 0$ ▷ Discrete time step for simulation/operation

while mission not completed **do**

  for each robot $i = 1, \ldots, n$ **do**

    $\hat{\mathbf{x}}_i(t_k) \leftarrow$ **EstimateCurrentState()** ▷ e.g., from sensors

    $(\mathbf{P}_j(t_k), \dot{\mathbf{P}}_j(t_k), \ddot{\mathbf{P}}_j(t_k)$ for $j \in \mathcal{N}_i) \leftarrow$ **ReceiveNeighbourStates()** ▷ $\mathcal{N}_i$ is set of neighbours

    $\mathbf{P}_L(t_k), \dot{\mathbf{P}}_L(t_k), \ddot{\mathbf{P}}_L(t_k) \leftarrow$ **GetLeaderState()** ▷ If $b_i > 0$

    Construct initial state $\boldsymbol{\eta}_i^0$ for NMPC problem

    $\mathbf{e}_i(t_k) \leftarrow$ Compute current formation error $\left(\widetilde{\mathbf{P}}_i(t) = \mathbf{C}_{CBF}\hat{\mathbf{x}}_i(t_k)\right)$

    $\dot{\mathbf{e}}_i(t_k) \leftarrow$ Compute current formation error rate

    $\boldsymbol{\eta}_i^0 \leftarrow [\mathbf{e}_i(t_k)^\top, \dot{\mathbf{e}}_i(t_k)^\top]^\top$

    Solve the unified NMPC problem (Eq. 58)

    $[\mathbf{\Xi}_i^*, \gamma_i^*] \leftarrow$ **SolveNMPCProblem**(*using* $\boldsymbol{\eta}_i^0$, *current states, and parameters*)

    Extract the first optimal auxiliary control input

    $\boldsymbol{\xi}_i^{0*} \leftarrow$ First element of $\mathbf{\Xi}_i^*$

    Compute the corresponding actual control torque $\boldsymbol{\tau}_i^{0*}$

    $\boldsymbol{\nu}_i^{0*} \leftarrow \widetilde{\mathbf{n}}(\hat{\mathbf{x}}_i(t_k), t_k)^\dagger \left[-\widetilde{\mathbf{l}}(\hat{\mathbf{x}}_i(t_k), t_k) - \widetilde{\mathbf{m}}(\hat{\mathbf{x}}_i(t_k), t_k) + \boldsymbol{\xi}_i^{0*} \right]$

    $\boldsymbol{\tau}_i^{0*} \leftarrow \overline{\mathbf{C}}(\mathbf{q}_i(t_k), \dot{\mathbf{q}}_i(t_k))\mathbf{\Phi}_i(t_k) + \overline{\mathbf{M}}(\mathbf{q}_i(t_k))\boldsymbol{\nu}_i^{0*}$ ▷ Derived from input-output linearisation

    ApplyTorqueToRobot($\boldsymbol{\tau}_i^{0*}$)

  end for

  $t_k \leftarrow t_k + 1$

  BroadcastOwnState() ▷ Each robot makes its new state available

end while

5 Results and discussion

This section presents the simulation results and discusses their implications. The simulation was conducted in MATLAB, where a group of WMRs navigates from an initial position, following a trajectory defined by a virtual leader. The proposed method enables the robots to track their target paths, avoid obstacles, and maintain the desired formation. Three case studies are presented below. Results are also compared against the DLPC framework introduced in Section 1.2 [12].

5.1 Formation Evaluation Criteria

Formation stability is assessed using a modified version of the consensus error defined in Eq. (38). Here, the leader-follower component is omitted to isolate formation behaviour. The total absolute formation error is calculated as:

$$\mathbf{e}_i(t) = \left| \sum_{j=1}^{n} \{a_{ij}([\mathbf{P}_i(t) - \mathbf{P}_j(t)] - \Delta_{ij}(t))\} \right|. \tag{59}$$

The Integral Absolute Error (IAE) is a performance metric used to quantify the total accumulated error of a system over time. In discrete-time systems, the IAE is calculated as the sum of the absolute values of the error at each time step:

$$\text{IAE}_k = \delta t \sum_{k=0}^{N} |e_i(k)|, \tag{60}$$

where δt is the sampling period, and e_i is the previously discussed absolute formation error. In this report, the IAE is used as the performance metric for analysing the performance of the control systems in the case studies. It provides a clear indication of how well the system is able to minimise deviations from the desired trajectories over time.

5.2 Case study 1: illustration of the proposed method for a linear trajectory

The system consists of three nonholonomic differential drive WMRs. The three robots are identical, each with parameters shown in Table 1.

Table 1. Definition of parameters

Variable	Value
m_p	$4.385\ kg$
m_w	$0.0575\ kg$
J_{zp}	$0.05\ kg \cdot m^2$
J_{zw}	$0.002\ kg \cdot m^2$
b	$0.125\ m$
D	$0.11\ m$
r	$0.05\ m$
N_g	1

Table 2. Obstacle Specifications 1

Obstacle	Position	Radius
Obs1	$[3.975;\ 1.1]$	0.25

Table 3. Constraints on inputs and states

Parameter	Lower Limit	Upper Limit
$\tau\ (N \cdot m)$	-0.25	0.25
$\mathbf{v}$ (rad/s)	-90	90

The virtual leader and follower initial values are set as $\mathbf{q}_L(0) = [1.5, 1.2, 0]^\top$, $\mathbf{q}_1(0) = [0, 0.7, 0]^\top$, $\mathbf{q}_2(0) = [-2.5, 2, 0]^\top$, and $\mathbf{q}_3(0) = [-1.8, -1.2, 0]^\top$, respectively.

The communication interactions between the virtual leader and the followers are implemented as in Fig. 3. The corresponding adjacency matrix and leader-follower adjacency vector for the communication network are shown below:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{61}$$

The desired geometrical formation that the followers maintain relative to the virtual leader is set as follows:

$$\mathbf{\Delta} = \begin{bmatrix} \Delta_1 & \Delta_2 & \Delta_3 \end{bmatrix} = \begin{bmatrix} -1.5 & -3.0 & -3.0 \\ 0 & 1.5 & -1.5 \end{bmatrix}.$$

To enforce the linear trajectory, the velocities of the leader are set to $v_L(k) = 0.25$ m/s and $\omega_L(k) = 0$ rad/s.

The specifications for the obstacles are shown in Table 2, and the torque and velocity constraints of the robots are shown in Table 3.

The three robots are simulated using the defined parameters to maintain a triangle formation. Each robot should maintain the formation while following the trajectory prescribed by the virtual leader. Two cases are discussed in this section to demonstrate the effectiveness of the proposed control method:

- **Case 1.1:** The robots use the control method without formation control by implementing the following communication network:

$$\mathbf{A} = \mathbf{0}_{3\times 3}, \quad \mathbf{b} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

- **Case 1.2:** The robots use the control method with the consensus formation control by implementing the communication network described in Equation (61).

The total simulation time for this case study is $40\ s$, and the sampling time is $\delta t = 0.1$ s. To have a fair comparison between the two scenarios, the control parameters for each case are given as follows:

- **Case 1.1 and 1.2:** with/without formation control:

$$\mathbf{Q} = \text{diag}([5, 5, 5, 5]), \quad \mathbf{R} = \text{diag}([2.5, 2.5]), \quad \mathbf{P} = \begin{bmatrix} 7.7689\,\mathbf{I}_2 & 3.5355\,\mathbf{I}_2 \\ 3.5355\,\mathbf{I}_2 & 5.4934\,\mathbf{I}_2 \end{bmatrix}$$
$$\mathbf{N} = 10, \quad \beta = 10, \quad \overline{\gamma} = 0.2, \quad \rho = 50, \quad \zeta = 100$$

5.2.1 Case 1.1: Without Formation Control

In the first case, when the consensus algorithm is not considered, the robots do not receive formation feedback from each other. This case is essentially a leader-follower system as discussed in Section 2. Fig. 5 illustrates the simulation results for the first case. In the absence of any obstacles, the three robots can track the trajectories prescribed by their neighbours and

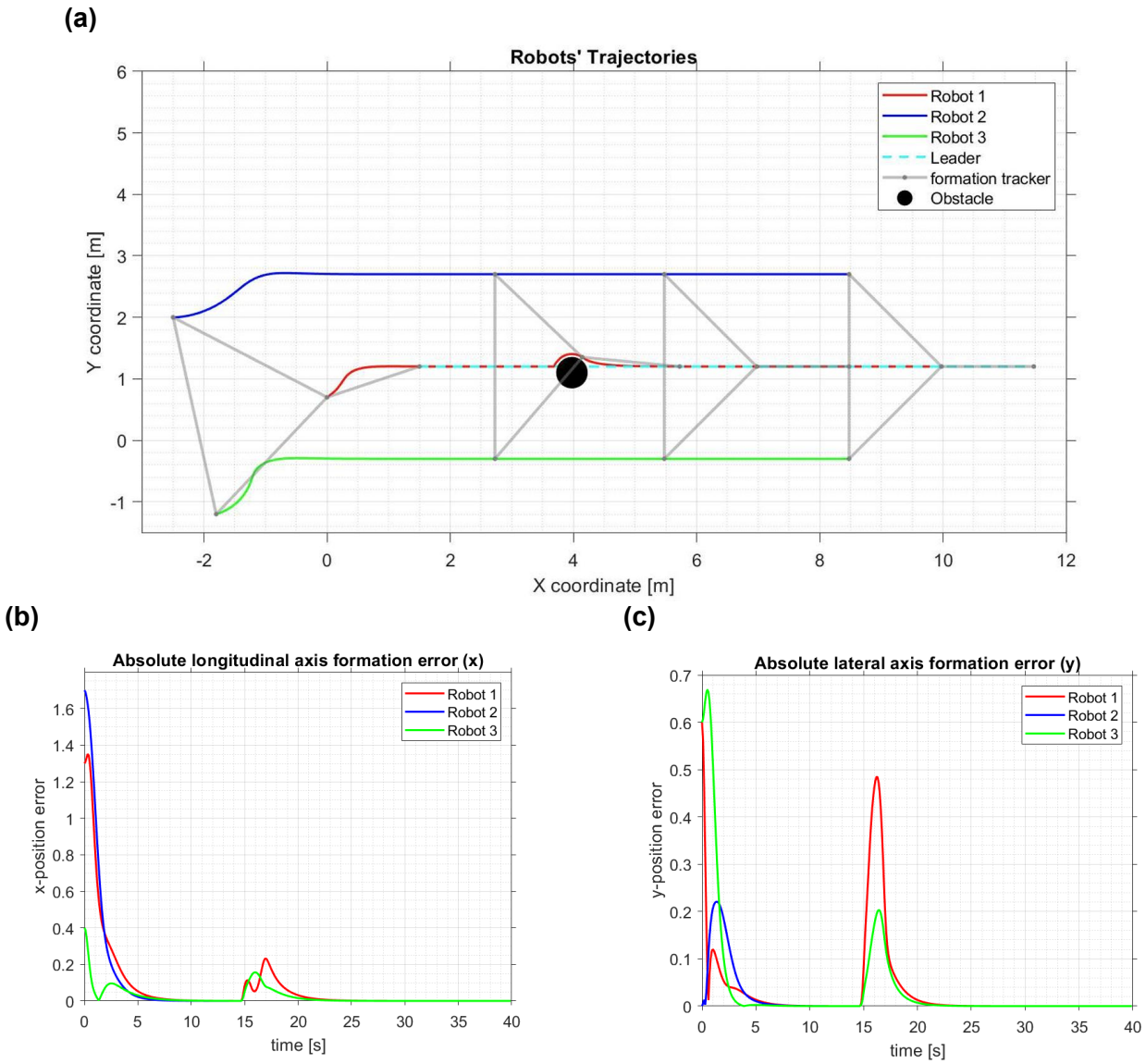

Fig. 5. Proposed method without formation control. (a) Virtual Leader and robot's trajectories. (b) The absolute longitudinal axis formation error of the robots. (c) The absolute lateral axis formation error of the robots.

asymptotically converge and maintain the correct formation. However, the simulation results indicate that the formation error of the three robots is relatively large in the interval when robot 1 encounters an external obstacle. In this interval, as there is no feedback information between the robots, the formation error (both longitudinal and lateral) increases, and the formation cannot keep the desired shape in the presence of the obstacle, which acts as an external disturbance. Nevertheless, outside the influence of the obstacle (disturbance), the formation errors converge to zero, demonstrating the system's stability following a linear trajectory with no formation feedback.

5.2.2 Case 1.2: With Formation Control

In the second case, when the consensus algorithm is considered, the robots receive formation feedback from each other. This has similar elements to leader-follower and virtual-structure systems, as discussed in Section 2. Fig. 6 illustrates the simulation results for the second case. Without any obstacles, the three robots can track the trajectories prescribed by their neighbours and asymptotically converge and maintain the correct formation. However, unlike the first case, the

simulation results indicate that the formation error of the three robots is relatively tiny in the interval when robot 1 encounters an external obstacle. In this interval, as there is feedback information between the robots, the formation error (both longitudinal and lateral) increases slightly, and the formation keeps the desired shape in the presence of the obstacle while deviating somewhat from the intended trajectory. Both controllers' integrated absolute errors are compared and reported in Table 4. It can be observed that the formation tracking errors of the proposed method were smaller with formation control than without formation control, as expected.

Table 4. Integral Absolute Error of x, y, case study 1.

	Robot 1		**Robot 2**		**Robot 3**	
	No Consensus	**Consensus**	**No Consensus**	**Consensus**	**No Consensus**	**Consensus**
e_x	2.98	**1.92**	2.94	**1.97**	0.91	**0.49**
e_y	1.34	**0.67**	0.92	**0.48**	1.40	**0.76**

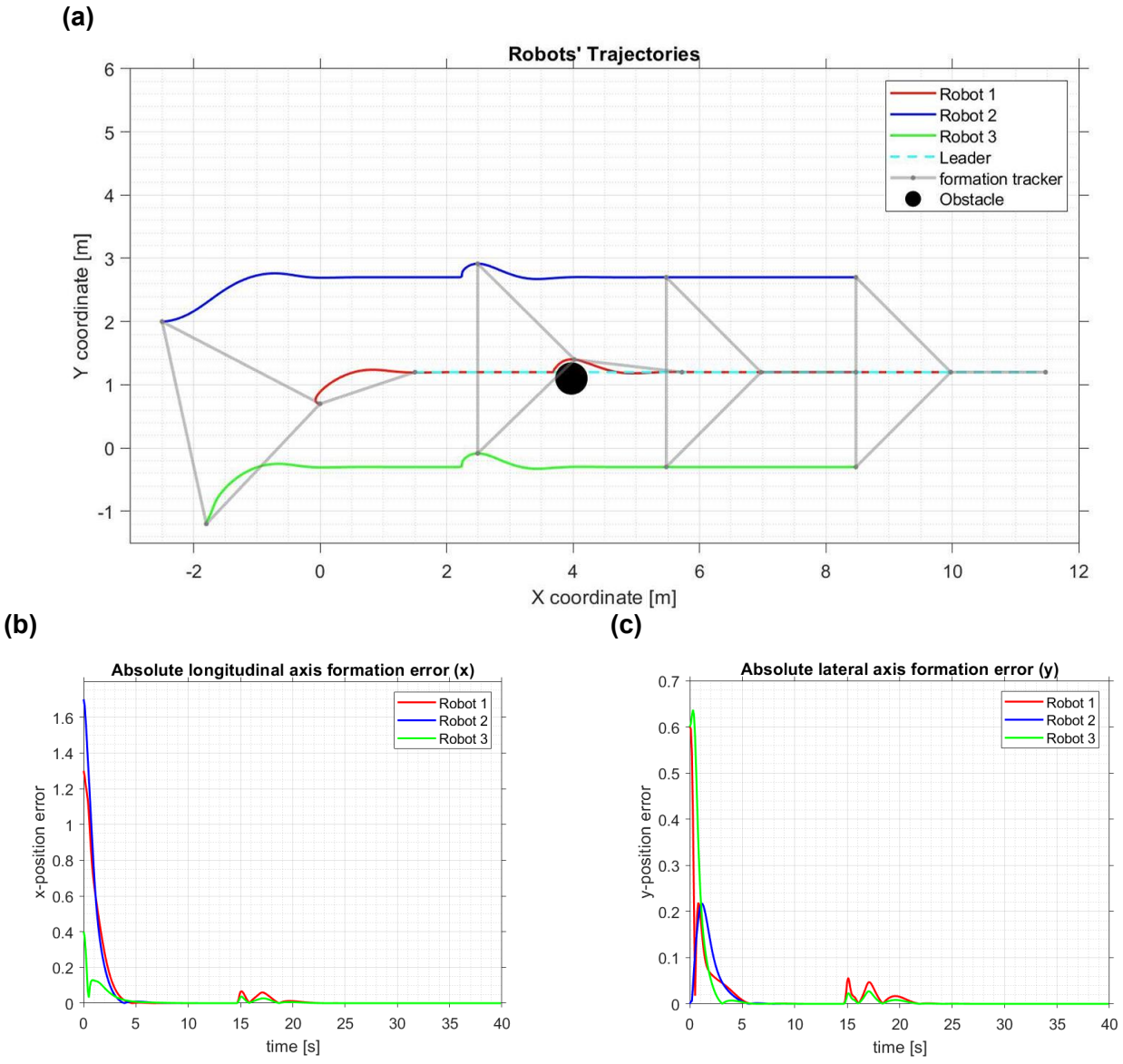

Fig. 6. Proposed method with formation control. (a) Virtual Leader and robot's trajectories. (b) The absolute longitudinal axis formation error of the robots. (c) The absolute lateral axis formation error of the robots.

5.3 Case study 2: illustration of the proposed method for a curved trajectory

In this case study, the simulation is run for a curved virtual leader trajectory and the same formation geometry as the previous section, in the presence of obstacles. The performance of the proposed method is again shown with and without formation control feedback. The same controller parameters as in the previous case study were used in this scenario. However, the obstacle positions were changed. The position and radius for the new obstacles are presented in Table 5.

Table 5. Obstacle Specifications 2

Obstacle	Position	Radius
Obs1	$[3.975;\ 2.0]$	0.25

5.3.1 Case 2.1: Without Formation Control

In this case, when the consensus algorithm is not considered, as before, the robots do not receive formation feedback from each other. Fig. 7 illustrates the simulation results for the first case. In the absence of any obstacles, the three robots can track the curved trajectories prescribed by their neighbours and asymptotically converge and maintain the correct formation. However, the simulation results indicate that the formation error of the three robots is relatively large in the interval when robot 1 encounters an external obstacle. In this interval, as there is no feedback information between the robots, the formation errors (longitudinal and lateral) increase, and the formation cannot keep the desired shape in the presence of the obstacle, corroborating the results of Case 1.1.

5.3.2 Case 2.2: With Formation Control

In this case, when the consensus algorithm is considered, the robots receive formation feedback from each other, as before. Fig. 8 illustrates the simulation results for this case. Without any obstacles, the three robots can track the trajectories prescribed by their neighbours and asymptotically converge and maintain the correct formation. Similar to Case 1.2, the simulation results indicate that the formation error of the three robots is relatively tiny in the interval when robot 1 encounters an external obstacle. In this interval, as there is feedback information between the robots, the formation errors (longitudinal and lateral) increase slightly, and the formation keeps the desired shape in the presence of the obstacle while deviating somewhat from the intended trajectory. Both controllers' integrated absolute errors are compared and reported in Table 6, where the case with formation control consistently had a smaller IAE than the case without the formation control. It can be observed that this case corroborates the findings of Case 1.2, confirming that the proposed formation controller is capable of rejecting external disturbances to maintain the formation.

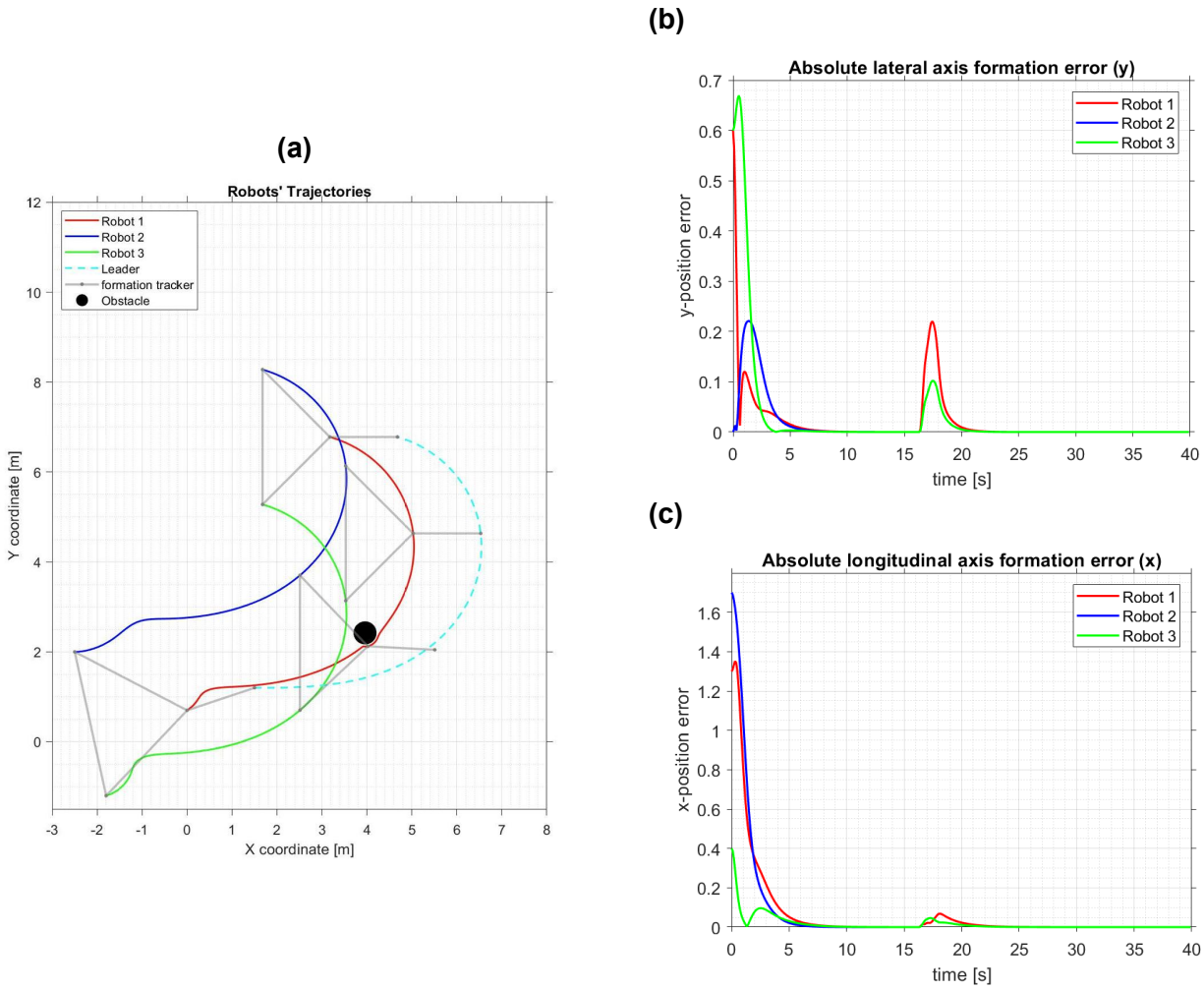

Fig. 7. Proposed method without formation control. (a) Virtual Leader and robot's trajectories. (b) The absolute longitudinal axis formation error of the robots. (c) The absolute lateral axis formation error of the robots.

Table 6. Integral Absolute Error of x, y, case study 2.

	Robot 1		**Robot 2**		**Robot 3**	
	No Consensus	**Consensus**	**No Consensus**	**Consensus**	**No Consensus**	**Consensus**
e_x	2.55	**1.81**	2.65	**1.92**	0.62	**0.44**
e_y	0.81	**0.81**	0.68	**0.56**	1.16	**0.83**

5.4 Case study 3: comparison between the proposed controller and the DLPC framework

This section compares the DLPC framework developed in [12] and the proposed control method. The main advantages of the DLPC method are that it is computationally fast, efficient, and highly scalable, which are all desirable in a scenario with multiple robots. It has also been demonstrated that the DLPC method scales up to 10000 units [12]. The DLPC framework and the proposed controller are applied to a scenario where two robots form a ring communication network with the virtual leader. The corresponding adjacency matrix and leader-follower adjacency vector for the

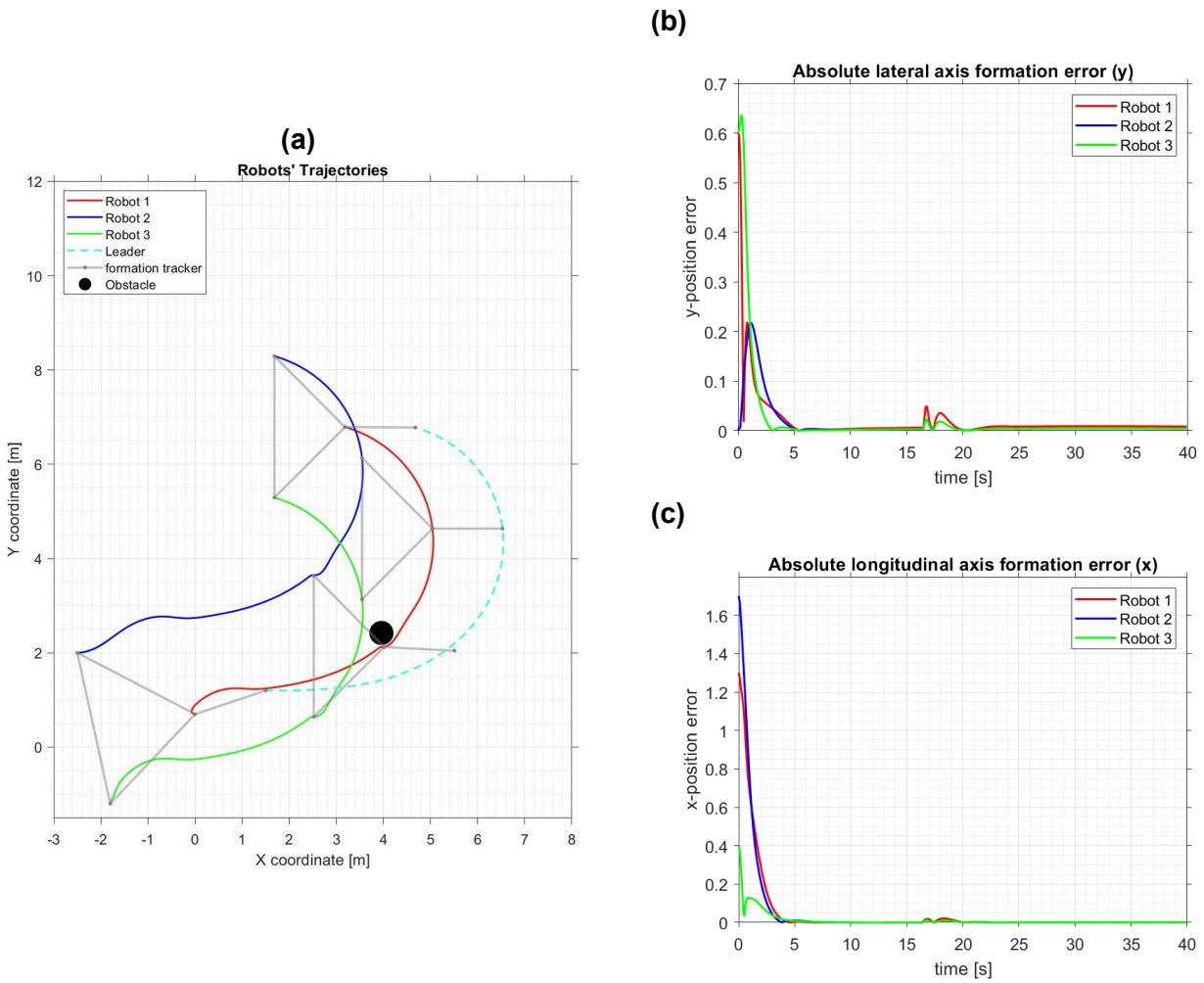

Fig. 8. Proposed method with formation control. (a) Virtual Leader and robot's trajectories. (b) The absolute longitudinal axis formation error of the robots. (c) The absolute lateral axis formation error of the robots.

communication network are shown below:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 & 1 \end{bmatrix}. \tag{62}$$

To make a fair comparison, only the formation controller is considered in this case, hence the trajectory tracker is not used. The virtual robot moves in a straight line parallel to the global x-axis at a constant velocity. The virtual leader and robot initial positions are set as $\mathbf{q}_L(0) = [0, 1, 0]^\top$, $\mathbf{q}_1(0) = [0, 1, 0]^\top$, and $\mathbf{q}_2(0) = [0, 0, 0]^\top$, respectively. To enforce the linear trajectory, the velocities of the leader are set to $v_L(k) = 1.0$ m/s and $\omega_L(k) = 0$ rad/s.

The desired geometrical formation that the followers maintain relative to the virtual leader is set as follows:

$$\mathbf{\Delta} = \begin{bmatrix} \Delta_1 & \Delta_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}.$$

The specifications for the obstacles are shown in Table 7.

Table 7. Obstacle Specifications 3

Obstacle	Position	Radius
Obs1	$[3;\ 1]$	0.1

The total simulation time for this case study is 9 s, and the sampling time is $\delta t = 0.05$ s. To have a fair comparison between the two scenarios, the control parameters for each case are given as follows:

- **Case 3.1:** Proposed Controller:

$$\mathbf{Q} = \text{diag}([5,5,5,5]), \quad \mathbf{R} = \text{diag}([2.5,2.5]), \quad \mathbf{P} = \begin{bmatrix} 7.7689\,\mathbf{I}_2 & 3.5355\,\mathbf{I}_2 \\ 3.5355\,\mathbf{I}_2 & 5.4934\,\mathbf{I}_2 \end{bmatrix}$$

$$\mathbf{N} = 10, \quad \beta = 10, \quad \overline{\gamma} = 0.2, \quad \rho = 50, \quad \zeta = 100$$

- **Case 3.2:** DLPC framework:

  This case uses the same parameters described in [12].

5.4.1 Case 3.1: Proposed Controller

Fig. 9 illustrates the simulation results for this case. As observed, the formation error between the robots is initially zero, and the controller can maintain the formation when no external disturbances are present. At about $t = 3\ s$, robot 1 encounters an obstacle, and the controller navigates around the obstacle while maintaining minimal deviation. However, a relatively large spike in the formation error can be observed at that instant. This spike is because robot 2 also has access to the leader trajectory; as such, it is not prioritising maintaining the formation only, leading to a delay in synchronising robot 1 and robot 2 until robot 1 is no longer under the influence of the obstacle. After robot 1 has navigated around the obstacle, it is observed that the formation error asymptotically converges to zero within 2 seconds. The formation error can further be minimised using the following communication network, instead, allowing the robots to prioritise maintaining the formation over following the virtual leader:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{63}$$

5.4.2 Case 3.2: DLPC Framework

Fig. 10 illustrates the simulation results for this case. The DLPC framework can also maintain the formation in the presence of external disturbances. However, unlike the proposed method, the DLPC framework begins avoiding the obstacle much earlier. This is because the DLPC framework uses a force field-inspired barrier function framework for collision avoidance instead of the DTCBF

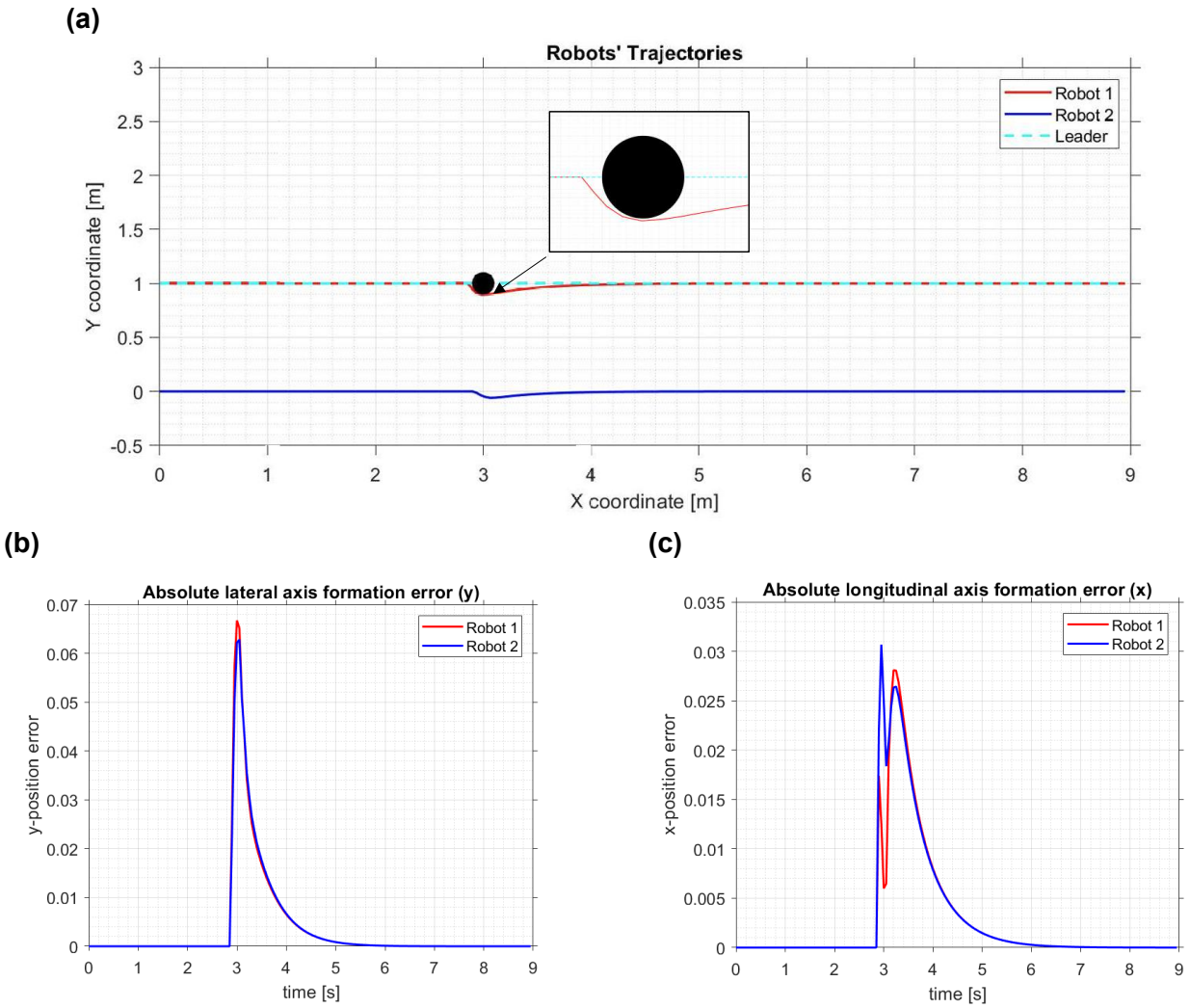

Fig. 9. Proposed method. (a) Virtual Leader and robot's trajectories. (b) The absolute longitudinal axis formation error of the robots. (c) The absolute lateral axis formation error of the robots.

Table 8. Integral Absolute Error of x, y, case study 3.

	Robot 1		**Robot 2**	
	Consensus	**DLPC**	**Consensus**	**DLPC**
e_x	**0.024**	0.067	**0.026**	0.067
e_y	**0.032**	0.092	**0.032**	0.092

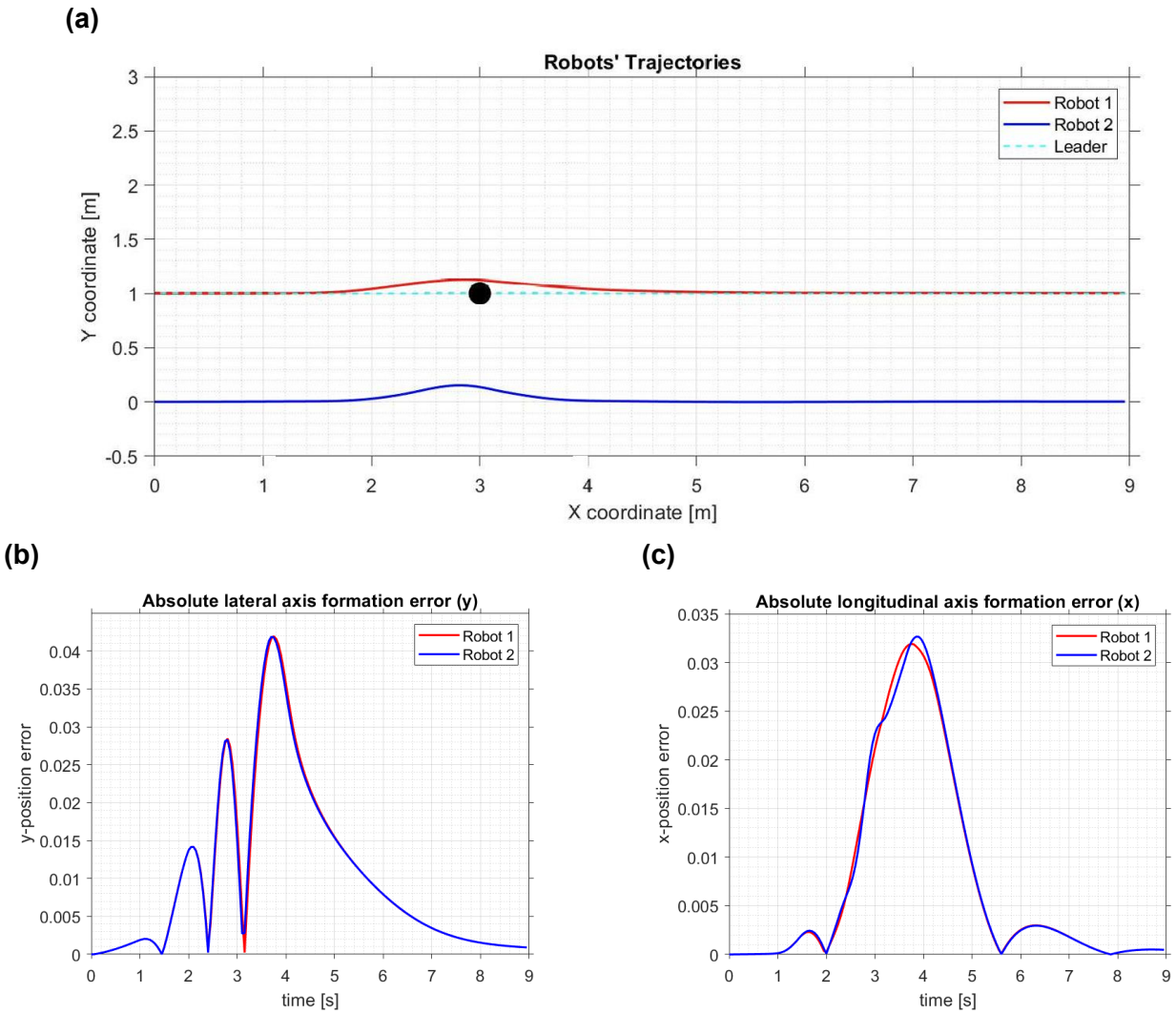

Fig. 10. DLPC Framework. (a) Virtual Leader and robot's trajectories. (b) The absolute longitudinal axis formation error of the robots. (c) The absolute lateral axis formation error of the robots.

used in this report. Thus, leading to the obstacle influencing the robot's trajectory much earlier, and the influence increases as the robot gets closer to the obstacle, contrasted by the DTCBF method, which has a minimal effect until the robot is on a direct collision course with the obstacle. Thus, the proposed method outperforms the DLPC method for this particular scenario. This is further demonstrated by the IAE of both methods shown in Table 8. The IAE is smaller for the proposed method in all robots than the DLPC method.

6 Conclusions and future work

This report has developed and evaluated a novel consensus-based model predictive formation control framework for nonholonomic WMRs using input-output linearisation. This report has successfully addressed critical challenges in multi-robot formation control, particularly focusing on distributed coordination, safety guarantees, and computational efficiency.

6.1 Conclusions

This report has accomplished the aims of contributing to the development of safe and adaptable navigation in multi-robot systems. The mathematical modelling of wheeled mobile robots incorporating nonholonomic constraints was thoroughly developed, providing a robust foundation for the controller design. The proposed consensus-based NMPC framework, enhanced with input-output linearisation techniques, demonstrated exceptional performance in maintaining precise formations while navigating complex trajectories.

The integration of DTCBFs into the control architecture was found to be highly effective in formally ensuring safety during operation. This approach successfully avoided obstacles while maintaining formation integrity, even in challenging scenarios.

The comparative analysis between the proposed consensus-based NMPC and the standard distributed NMPC without consensus mechanisms revealed substantial improvements in formation stability, convergence rates, and disturbance rejection capabilities. The consensus algorithm enabled robots to rapidly agree on formation parameters while adapting to environmental changes, demonstrating superior coordination compared to non-consensus approaches.

6.1.1 Performance Evaluation and Significance

The case studies presented in this report comprehensively validated the proposed methodology in various operational scenarios. The controller exhibited excellent tracking performance for linear trajectories with minimal steady-state errors and rapid convergence to the desired formation. The formation error metrics remained consistently small, demonstrating the effectiveness of the consensus mechanism in maintaining geometric constraints.

For curved trajectories, which present significantly greater challenges due to the nonholonomic constraints of WMRs, the controller demonstrated remarkable adaptability and precision. The dynamic reconfiguration capabilities allowed the formation to navigate tight turns while preserving the relative positions between robots, showcasing the controller's ability to handle complex manoeuvres.

The comparison with the DLPC framework revealed interesting trade-offs. Although the proposed approach demonstrated superior performance in terms of formation accuracy and safety guarantees, the DLPC framework exhibited advantages in computational efficiency. This finding is still inconclusive and requires further research to confirm.

From a practical standpoint, the controller implementation demonstrated feasibility for real-time applications, with computational requirements compatible with embedded systems typically found in mobile robots. The distributed nature of the algorithm makes it particularly suitable for large-scale deployments where centralised coordination is impractical or undesirable.

6.1.2 Limitations and Challenges

Despite the promising results, several limitations were identified during this research. The assumption of perfect state information presents a simplification that may not hold in real-world

scenarios with sensor noise and measurement uncertainties. Furthermore, the communication topology was assumed to be fixed, whereas practical deployments could experience intermittent connectivity and varying network topologies.

The computational complexity of solving the nonlinear optimisation problem in real-time remains a challenge, particularly for systems with limited processing capabilities. While the input-output linearisation technique and the hierarchical structure significantly reduced complexity compared to full nonlinear optimisation, further efficiency improvements are necessary for very large robot groups.

6.2 Future work

Several promising directions for future research emerge from this report. First, extending the framework to incorporate state estimation techniques would enhance robustness against sensor noise and measurement uncertainties, making the system more suitable for real-world deployment. Developing adaptive consensus gains that adjust based on formation errors and environmental conditions could further improve performance in dynamic environments.

Incorporating learning-based approaches to predict disturbances and optimise control parameters would leverage the strengths of both model-based and data-driven methods. This hybrid approach could potentially reduce computational demands while maintaining performance guarantees. In addition, exploring time-varying communication topologies and developing resilience against communication failures would address critical practical considerations for robust deployment.

Testing the framework on heterogeneous multi-robot systems with varying dynamic capabilities would expand its applicability across diverse robotic platforms. Implementing the approach on physical robotic systems would provide critical validation of the simulation results and identify practical implementation challenges.

Finally, extending the methodology to three-dimensional formations would enable applications in aerial and underwater robotics, significantly broadening the impact of this research. The theoretical foundations established in this report provide a solid platform for these future developments, paving the way for increasingly autonomous and coordinated multi-robot systems capable of operating safely and efficiently in complex environments.

In conclusion, this report has made significant contributions to the multi-robot formation control field by developing a comprehensive consensus-based model predictive control framework that effectively balances formation accuracy, safety guarantees, and computational efficiency. The methodologies and results presented provide valuable insights for theoretical advancement and practical implementation of coordinated control strategies for autonomous robot teams.

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007, Conference Name: Proceedings of the IEEE, ISSN: 1558-2256. DOI: 10.1109/JPROC.2006.887293. Accessed: Mar. 24, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/4118472 (cited on p. 1).

[2] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the 2005, American Control Conference, 2005.*, ISSN: 2378-5861, Jun. 2005, 1859–1864 vol. 3. DOI: 10.1109/ACC.2005.1470239. Accessed: Mar. 24, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/1470239 (cited on p. 1).

[3] Y. Cao, W. Yu, W. Ren, and G. Chen, "An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, Feb. 2013, Conference Name: IEEE Transactions on Industrial Informatics, ISSN: 1941-0050. DOI: 10.1109/TII.2012.2219061. Accessed: Mar. 24, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6303906 (cited on p. 1).

[4] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015, ISSN: 0005-1098. DOI: 10.1016/j.automatica.2014.10.022. Accessed: Mar. 5, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109814004038 (cited on pp. 1, 2).

[5] *Factory Automation & Industrial Controls Market Size [2033] Report*. Accessed: Mar. 24, 2025. [Online]. Available: https://www.businessresearchinsights.com/market-reports/factory-automation-industrial-controls-market-107084 (cited on p. 1).

[6] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses," en, *AI Magazine*, vol. 29, no. 1, pp. 9–19, 2008, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1609/aimag.v29i1.2082, ISSN: 2371-9621. DOI: 10.1609/aimag.v29i1.2082. Accessed: Mar. 24, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1609/aimag.v29i1.2082 (cited on p. 1).

[7] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized Cooperative Aerial Surveillance Using Fixed-Wing Miniature UAVs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, Jul. 2006, Conference Name: Proceedings of the IEEE, ISSN: 1558-2256. DOI: 10.1109/JPROC.2006.876930. Accessed: Mar. 25, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1677946 (cited on p. 1).

[8] W. Liu, X. Wang, and S. Li, "Formation Control for Leader–Follower Wheeled Mobile Robots Based on Embedded Control Technique," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 1, pp. 265–280, Jan. 2023, ISSN: 1558-0865. DOI: 10.1109/TCST.2022.3173887. Accessed: Feb. 5, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9775002/?arnumber=9775002 (cited on pp. 1–3).

[9] J. Peng, H. Xiao, G. Lai, and C. Philip Chen, "Consensus formation control of wheeled mobile robots with mixed disturbances under input constraints," en, *Journal of the Franklin Institute*, vol. 361, no. 17, p. 107 300, Nov. 2024, ISSN: 00160032. DOI: 10.1016/j.jfranklin.2024.107300. Accessed: Feb. 3,

2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S001600322400721X (cited on pp. 1, 3, 10, 13, 20).

[10] X. Tan and D. V. Dimarogonas, "Distributed Implementation of Control Barrier Functions for Multi-agent Systems," *IEEE Control Systems Letters*, vol. 6, pp. 1879–1884, 2022, ISSN: 2475-1456. DOI: 10.1109/LCSYS.2021.3133802. Accessed: Dec. 16, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9642050 (cited on p. 1).

[11] R. Grandia, A. J. Taylor, A. Singletary, M. Hutter, and A. D. Ames, "Nonlinear Model Predictive Control of Robotic Systems with Control Lyapunov Functions," in *Robotics: Science and Systems XVI*, Jul. 2020. DOI: 10.15607/RSS.2020.XVI.098. Accessed: Dec. 22, 2024. [Online]. Available: http://arxiv.org/abs/2006.01229 (cited on pp. 1, 4–7, 19, 20).

[12] X. Zhang et al., "Toward Scalable Multirobot Control: Fast Policy Learning in Distributed MPC," *IEEE Transactions on Robotics*, vol. 41, pp. 1491–1512, 2025, Conference Name: IEEE Transactions on Robotics, ISSN: 1941-0468. DOI: 10.1109/TRO.2025.3531818. Accessed: Mar. 5, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10847886 (cited on pp. 2, 4, 23, 29, 31).

[13] Z. Lin and H. H. Liu, "Topology-based distributed optimization for multi-UAV cooperative wildfire monitoring," en, *Optimal Control Applications and Methods*, vol. 39, no. 4, pp. 1530–1548, 2018, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.2424, ISSN: 1099-1514. DOI: 10.1002/oca.2424. Accessed: Mar. 5, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.2424 (cited on p. 2).

[14] L. Li, C. Kuang, Y. Xia, and J. Qiang, "Formation control of nonholonomic mobile robots with inaccurate global positions and velocities," en, *International Journal of Robust and Nonlinear Control*, vol. 32, no. 18, pp. 9776–9790, 2022, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rnc.6392, ISSN: 1099-1239. DOI: 10.1002/rnc.6392. Accessed: Mar. 5, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.6392 (cited on pp. 2, 3).

[15] N. Nfaileh, K. Alipour, B. Tarvirdizadeh, and A. Hadi, "Formation control of multiple wheeled mobile robots based on model predictive control," en, *Robotica*, vol. 40, no. 9, pp. 3178–3213, Sep. 2022, ISSN: 0263-5747, 1469-8668. DOI: 10.1017/S0263574722000121. Accessed: Mar. 1, 2025. [Online]. Available: https://www.cambridge.org/core/journals/robotica/article/formation-control-of-multiple-wheeled-mobile-robots-based-on-model-predictive-control/AC68F68C04C5F9E6FE193254BF902478 (cited on pp. 2, 11, 13).

[16] J. Wang, Y. Peng, S. Wen, H. Wang, and Y. Wan, "Formation control for multiple nonholonomic wheeled mobile robots based on integrated sliding mode controller and force function," en, *Transactions of the Institute of Measurement and Control*, p. 01 423 312 241 291 093, Dec. 2024, ISSN: 0142-3312. DOI: 10.1177/01423312241291093. Accessed: Feb. 5, 2025. [Online]. Available: https://doi.org/10.1177/01423312241291093 (cited on pp. 2, 3).

[17] M. A. Lewis and K.-H. Tan, "High Precision Formation Control of Mobile Robots Using Virtual Structures," en, *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, Oct. 1997, ISSN: 1573-7527. DOI: 10.1023/A:1008814708459. Accessed: Mar. 5, 2025. [Online]. Available: https://doi.org/10.1023/A:1008814708459 (cited on p. 3).

[18] M. Siwek, “Consensus-Based Formation Control with Time Synchronization for a Decentralized Group of Mobile Robots,” *Sensors (Basel, Switzerland)*, vol. 24, no. 12, p. 3717, Jun. 2024, ISSN: 1424-8220. DOI: 10.3390/s24123717. Accessed: Feb. 18, 2025. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11207745/ (cited on pp. 3, 4).

[19] F. Chen and W. Ren, *On the Control of Multi-Agent Systems: A Survey*. Now Foundations and Trends, 2019, vol. 1. DOI: 10.1561/2600000019 (cited on p. 3).

[20] G. Antonelli, F. Arrichiello, and S. Chiaverini, “The NSB control: A behavior-based approach for multi-robot systems,” en, *Paladyn, Journal of Behavioral Robotics*, vol. 1, no. 1, pp. 48–56, Mar. 2010, Publisher: De Gruyter Open Access Section: Paladyn, ISSN: 2081-4836. DOI: 10.2478/s13230-010-0006-0. Accessed: Mar. 6, 2025. [Online]. Available: https://www.degruyter.com/document/doi/10.2478/s13230-010-0006-0/html?utm_source=chatgpt.com (cited on p. 3).

[21] 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, Dec. 1998, Conference Name: IEEE Transactions on Robotics and Automation, ISSN: 2374-958X. DOI: 10.1109/70.736776. Accessed: Mar. 6, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/736776 (cited on p. 3).

[22] K. D. Listmann, M. V. Masalawala, and J. Adamy, “Consensus for formation control of nonholonomic mobile robots,” in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3886–3891. DOI: 10.1109/ROBOT.2009.5152865. Accessed: Feb. 5, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/5152865 (cited on p. 4).

[23] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, “Safe Policy Synthesis in Multi-Agent POMDPs via Discrete-Time Barrier Functions,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Dec. 2019, pp. 4797–4803. DOI: 10.1109/CDC40024.2019.9030241. Accessed: Dec. 23, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9030241 (cited on pp. 4, 8).

[24] K. Y. Chee, T. C. Silva, M. A. Hsieh, and G. J. Pappas, *Enhancing Sample Efficiency and Uncertainty Compensation in Learning-based Model Predictive Control for Aerial Robots*, Aug. 2023. DOI: 10.48550/arXiv.2308.00570. Accessed: Dec. 16, 2024. [Online]. Available: http://arxiv.org/abs/2308.00570 (cited on p. 4).

[25] W. Gao, J. Yang, J. Liu, H. Shi, and B. Xu, “Moving Horizon Estimation for Cooperative Localisation with Communication Delay,” en, *The Journal of Navigation*, vol. 68, no. 3, pp. 493–510, May 2015, ISSN: 0373-4633, 1469-7785. DOI: 10.1017/S037346331400085X. Accessed: Dec. 16, 2024. [Online]. Available: https://www.cambridge.org/core/journals/journal-of-navigation/article/moving-horizon-estimation-for-cooperative-localisation-with-communication-delay/55872A43B94BF061AAA195ABBDCC4FB3# (cited on p. 4).

[26] Z. Artstein, “Stabilization with relaxed controls,” en, *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, Jan. 1983, ISSN: 0362546X. DOI: 10.1016/0362-546X(83)90049-4. Accessed: Dec. 22, 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0362546X83900494 (cited on p. 6).

[27] Y. Xiong, D.-H. Zhai, M. Tavakoli, and Y. Xia, “Discrete-Time Control Barrier Function: High-Order Case and Adaptive Case,” *IEEE Transactions on Cybernetics*, vol. 53, pp. 1–9, May 2022. DOI: 10.1109/TCYB.2022.3170607 (cited on pp. 8, 20).

[28] A. Agrawal and K. Sreenath, “Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation,” en, in *Robotics: Science and Systems XIII*, Robotics: Science and Systems Foundation, Jul. 2017, ISBN: 978-0-9923747-3-0. DOI: 10.15607/RSS.2017.XIII.073. Accessed: Dec. 23, 2024. [Online]. Available: http://www.roboticsproceedings.org/rss13/p73.pdf (cited on p. 8).

[29] H. K. Khalil, *Nonlinear Systems*, en. Prentice Hall, 2002, ISBN: 978-0-13-067389-3 (cited on pp. 8, 10, 16, 19).

[30] S. Sastry, *Nonlinear Systems* (Interdisciplinary Applied Mathematics), J. E. Marsden, L. Sirovich, and S. Wiggins, Eds. New York, NY: Springer, 1999, vol. 10, ISBN: 978-1-4757-3108-8. DOI: 10.1007/978-1-4757-3108-8. Accessed: Mar. 22, 2025. [Online]. Available: http://link.springer.com/10.1007/978-1-4757-3108-8 (cited on pp. 8, 9, 19).

[31] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, en. Prentice-Hall, 1991, Google-Books-ID: HddxQgAACAAJ, ISBN: 978-0-13-040049-9 (cited on pp. 8, 19).

[32] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, en. MIT Press, 2004, Google-Books-ID: gUbQ9_weg88C, ISBN: 978-0-262-19502-7 (cited on pp. 11, 12).

[33] J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering), en. Springer New York, 2006, ISBN: 978-0-387-30303-1. DOI: 10.1007/978-0-387-40065-5. Accessed: Mar. 10, 2025. [Online]. Available: http://link.springer.com/10.1007/978-0-387-40065-5 (cited on p. 20).

Appendices

A Full Derivation of Auxiliary Tracking System Dynamics

This appendix provides a detailed derivation of the auxiliary tracking system dynamics, as presented in Eq. (47) in the main text. The goal is to express the dynamics of the formation error in a state-space form suitable for control design using input-output linearisation.

The consensus-based formation tracking error for robot i is defined as:

$$\mathbf{e}_i(t) = -(\mathbf{P}(t)\mathbf{A} + \boldsymbol{\Delta}(t)\boldsymbol{\Gamma} + \overline{\mathbf{P}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\tilde{\mathbf{P}}_i(t), \tag{A.64}$$

where $\hat{b}_i \in \mathbb{R}^n$ is the i-th standard basis vector of the n-dimensional Euclidean space, $\mathbf{P}(t)$ is the matrix of actual positions of all robots, $\boldsymbol{\Delta}(t)$ contains the desired relative displacements, $\overline{\mathbf{P}}_L(t)$ is the leader's position replicated, $\mathbf{A}$ is the adjacency matrix, $\boldsymbol{\Gamma}$ is the modified Laplacian, $\mathbf{B}$ is the leader adjacency matrix, $g_i^{dis} \in \mathbb{R}$ is the (i,i)-th element of $\mathbf{g}^{dis} \triangleq \mathbf{D} + \mathbf{B}$, and $\tilde{\mathbf{P}}_i(t)$ is the position of robot i to be controlled.

Under Assumption 10 (ensuring sufficient smoothness), we define the auxiliary state for the consensus problem as:

$$\boldsymbol{\eta}_i(\mathbf{x}_i, t) = \begin{bmatrix} \mathbf{e}_i(t) \\ \dot{\mathbf{e}}_i(t) \end{bmatrix}. \tag{A.65}$$

Our objective is to find the time derivative $\dot{\boldsymbol{\eta}}_i(\mathbf{x}_i, t) = \frac{d}{dt}\boldsymbol{\eta}_i(\mathbf{x}_i, t) = \begin{bmatrix} \dot{\mathbf{e}}_i(t) \\ \ddot{\mathbf{e}}_i(t) \end{bmatrix}$.

First, let us find $\dot{\mathbf{e}}_i(t)$ by differentiating Eq. (A.64) with respect to time:

$$\dot{\mathbf{e}}_i(t) = -(\dot{\mathbf{P}}(t)\mathbf{A} + \dot{\boldsymbol{\Delta}}(t)\boldsymbol{\Gamma} + \dot{\overline{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\dot{\tilde{\mathbf{P}}}_i(t). \tag{A.66}$$

From the kinematic model of the i-th robot given in Eq. (26), its velocity $\dot{\tilde{\mathbf{P}}}_i(t)$ (which is the Cartesian velocity part of $\dot{\mathbf{q}}_i$) is given by:

$$\dot{\tilde{\mathbf{P}}}_i(t) = \mathbf{S}_y(\mathbf{q}_i)\boldsymbol{\Phi}_i, \tag{A.67}$$

where $\mathbf{S}_y(\mathbf{q}_i) = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times 1} \end{bmatrix} \mathbf{S}(\mathbf{q}_i)$ extracts the x, y velocity components. Substituting Eq. (A.67) into Eq. (A.66) yields:

$$\dot{\mathbf{e}}_i(t) = -(\dot{\mathbf{P}}(t)\mathbf{A} + \dot{\boldsymbol{\Delta}}(t)\boldsymbol{\Gamma} + \dot{\overline{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\mathbf{S}_y(\mathbf{q}_i)\boldsymbol{\Phi}_i. \tag{A.68}$$

Next, we find $\ddot{\mathbf{e}}_i(t)$ by differentiating Eq. (A.68) with respect to time:

$$\ddot{\mathbf{e}}_i(t) = -(\ddot{\mathbf{P}}(t)\mathbf{A} + \ddot{\boldsymbol{\Delta}}(t)\boldsymbol{\Gamma} + \ddot{\overline{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\ddot{\tilde{\mathbf{P}}}_i(t). \tag{A.69}$$

The acceleration of the i-th robot, $\ddot{\bar{\mathbf{P}}}_i(t)$, after applying input-output linearisation at the wheel dynamics level described in Eq. (44), is given by Eq. (45) as:

$$\ddot{\bar{\mathbf{P}}}_i(t) = \mathbf{S}_z(\omega_i)\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_i + \mathbf{S}_y(\mathbf{q}_i)\boldsymbol{\nu}_i(\mathbf{x}_i, t), \tag{A.70}$$

where $\boldsymbol{\nu}_i(\mathbf{x}_i, t)$ is the auxiliary input from the linearised wheel dynamics. Substituting Eq. (A.70) into Eq. (A.69) gives:

$$\ddot{\mathbf{e}}_i(t) = -(\ddot{\mathbf{P}}(t)\mathbf{A} + \ddot{\mathbf{\Delta}}(t)\mathbf{\Gamma} + \ddot{\bar{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\left(\mathbf{S}_z(\omega_i)\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_i + \mathbf{S}_y(\mathbf{q}_i)\boldsymbol{\nu}_i(\mathbf{x}_i, t)\right). \tag{A.71}$$

Finally, we assemble $\dot{\boldsymbol{\eta}}_i$ using $\dot{\mathbf{e}}_i(t)$ from Eq. (A.68) as its first component and $\ddot{\mathbf{e}}_i(t)$ from Eq. (A.71) as its second component:

$$\begin{aligned}
\dot{\boldsymbol{\eta}}_i(\mathbf{x}_i, t) &= \begin{bmatrix} \dot{\mathbf{e}}_i(t) \\ \ddot{\mathbf{e}}_i(t) \end{bmatrix} \\
&= \begin{bmatrix} -(\dot{\mathbf{P}}(t)\mathbf{A} + \dot{\mathbf{\Delta}}(t)\mathbf{\Gamma} + \dot{\bar{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_i \\ -(\ddot{\mathbf{P}}(t)\mathbf{A} + \ddot{\mathbf{\Delta}}(t)\mathbf{\Gamma} + \ddot{\bar{\mathbf{P}}}_L(t)\mathbf{B})\hat{b}_i + g_i^{dis}\left(\mathbf{S}_z(\omega_i)\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_i + \mathbf{S}_y(\mathbf{q}_i)\boldsymbol{\nu}_i(\mathbf{x}_i, t)\right) \end{bmatrix}.
\end{aligned} \tag{A.72}$$

This can be written in the control-affine form $\dot{\boldsymbol{\eta}}_i = \mathbf{l}(\mathbf{x}_i, t) + \mathbf{m}(\mathbf{x}_i, t) + \mathbf{n}(\mathbf{x}_i, t)\boldsymbol{\nu}_i(\mathbf{x}_i, t)$, which matches Eq. (47), by grouping terms as follows:

$$\begin{aligned}
\dot{\boldsymbol{\eta}}_i(\mathbf{x}_i, t) = &\underbrace{-\left(\begin{bmatrix} \dot{\mathbf{P}}(t) \\ \ddot{\mathbf{P}}(t) \end{bmatrix}\mathbf{A} + \begin{bmatrix} \dot{\mathbf{\Delta}}(t) \\ \ddot{\mathbf{\Delta}}(t) \end{bmatrix}\mathbf{\Gamma} + \begin{bmatrix} \dot{\bar{\mathbf{P}}}_L(t) \\ \ddot{\bar{\mathbf{P}}}_L(t) \end{bmatrix}\mathbf{B}\right)\hat{b}_i}_{\mathbf{l}(\mathbf{x}_i, t)} \\
&\underbrace{+ g_i^{dis}\begin{bmatrix} \mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_i \\ \mathbf{S}_z(\omega_i)\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_i \end{bmatrix}}_{\mathbf{m}(\mathbf{x}_i, t)} \\
&\underbrace{+ g_i^{dis}\begin{bmatrix} \mathbf{0}_{2\times 2} \\ \mathbf{S}_y(\mathbf{q}_i) \end{bmatrix}}_{\mathbf{n}(\mathbf{x}_i, t)}\boldsymbol{\nu}_i(\mathbf{x}_i, t).
\end{aligned} \tag{A.73}$$

The terms $\mathbf{l}(\mathbf{x}_i, t)$, $\mathbf{m}(\mathbf{x}_i, t)$, and $\mathbf{n}(\mathbf{x}_i, t)$ are thus explicitly defined by this grouping, consistent with their use in the main text for the input-output linearising control law design. Note that the definition of $\mathbf{m}(\mathbf{x}_i, t)$ can also be expressed as $g_i^{dis}\begin{bmatrix} \mathbf{I}_2 \\ \mathbf{S}_z(\omega_i) \end{bmatrix}\mathbf{S}_y(\mathbf{q}_i)\mathbf{\Phi}_i$, which is equivalent to the form above.

B Tuning

This section describes the procedure for tuning the parameters of the unified NMPC controller (Eq. 58) to replicate or adapt the presented results. Effective tuning is an iterative process, typically refined through simulation and empirical adjustments.

Central to tuning are the objective function weights, specifically the state weighting matrix **Q** for the auxiliary state $\boldsymbol{\eta}_i^k$ and the control input weighting matrix **R** for the auxiliary input $\boldsymbol{\xi}_i^k$. To prioritise tighter formation tracking and faster error correction, one would typically increase the diagonal elements of **Q** relative to those of **R**, making formation deviations more costly. Conversely, to achieve smoother control actions and lessen actuator effort, increasing **R**'s elements relative to **Q**'s is advisable, as this penalises large or rapid changes in control inputs. A common starting point is to use scaled identity matrices, then iteratively adjust: if robots are sluggish in forming up, **Q** might be increased; if control inputs appear overly aggressive or oscillatory, **R** should be increased.

The prediction horizon, N, dictates how far ahead the controller optimises. A longer N can improve performance and stability by allowing better anticipation of future events like obstacles or complex manoeuvres, but at the cost of significantly increased computation time. A shorter N is less computationally demanding but may result in more reactive, potentially less optimal, behaviour. The aim is to select the shortest N that yields satisfactory performance and stability for the target scenarios while respecting real-time execution constraints. For the results reported herein, N was chosen to balance these considerations (specific values used can be found in Section 5).

The velocity consensus weight, ρ, influences the term $(\mathbf{v}_i^k - \overline{\mathbf{v}}_i)^2$, balancing individual robot trajectory needs with overall formation cohesion via speed synchronisation. For tighter, more rigid formations where speed synchrony is paramount, ρ should be increased. However, if robots require more individual freedom for navigation, such as in dense obstacle fields, or if a high ρ leads to infeasibility, it should be decreased. Observing the collective movement provides cues: unsynchronised speeds suggest increasing ρ, while an overly stiff formation struggling with individual manoeuvres indicates a need to reduce it.

Stability in NMPC-β schemes heavily relies on the terminal cost weight, β, associated with $V(\boldsymbol{\eta}_i^N, t_N)$. This should generally be chosen "sufficiently large" to ensure the terminal cost effectively guides the system towards a stable region. Values for β that are significantly larger than stage cost weights (e.g., elements of **Q**) are often effective. The precise value might be determined through theoretical analysis if a Control Lyapunov Function (CLF) is explicitly known, or more commonly, through empirical tuning aimed at ensuring stability and convergence.

For the Control Barrier Function (CBF) parameters, ζ weights the penalty on deviations of the CBF parameter γ_i from its desired value $\overline{\gamma}$, with a larger ζ enforcing stricter adherence to $\overline{\gamma}$. The parameter $\overline{\gamma}$ (where $0 < \overline{\gamma} \leq 1$) itself dictates how conservatively the safety constraint $h_{CBF}(\mathbf{x}_i^k) \geq 0$ is met. A smaller $\overline{\gamma}$ permits $h(\mathbf{x}_k)$ to approach zero more slowly (less conservative), whereas a $\overline{\gamma}$ closer to 1 enforces a more rapid decay from the boundary (more conservative). A practical starting point for $\overline{\gamma}$ is a moderate value (e.g., 0.2 to 0.5), adjusted based on observed safety margins and proximity to obstacles. If safety enforcement seems overly aggressive, leading to excessively cautious manoeuvres, a slightly smaller $\overline{\gamma}$ or an adjustment to ζ might be warranted.

Accurate specification of physical constraints, namely the bounds on wheel speeds ($\mathbf{\Phi}_i^{lb/ub}$) and motor torques ($\boldsymbol{\tau}_i^{lb/ub}$), is essential. These must reflect the robot's actual physical limitations. Overly loose bounds can result in unrealistic commands, while excessively tight ones might unduly restrict performance or lead to infeasibility. These should always be based on verified hardware specifications. Similarly, the safe distance d_{safe} used in the DTCBF (Eq. 56) must be chosen carefully, considering robot and obstacle dimensions plus a desired safety margin.

In general, parameter tuning should be approached iteratively. Begin with a baseline set of parameters, perhaps derived from simpler scenarios or existing literature. Observe the system's behaviour in simulation. If performance is deficient in a particular aspect—such as tracking precision, control smoothness, or safety margins—identify the most relevant parameter(s) as discussed and make adjustments. It is often most effective to test one parameter, or a small group of closely related parameters, at a time to clearly understand the impact of each change.

C Software Code

Consensus-Based MPC for Multi-Robot Formation Control

Introduction

This repository contains the MATLAB implementation of a **Consensus-Based Model Predictive Control (MPC)** algorithm for multi-robot formation control. The code simulates a distributed multi-robot system where robots maintain a desired formation while following a leader robot's trajectory. The implementation incorporates **control barrier functions (CBFs)** for obstacle avoidance and ensures consensus among robots to achieve desired trajectories.

The code simulates a system of three robots following a virtual leader while avoiding obstacles. Each robot operates under constraints defined by its kinematics, dynamics, and inter-robot communication network.

System Overview

- **Leader-follower architecture:** The leader robot generates a trajectory, and the followers maintain a predefined formation relative to the leader. **Distributed control:** Each robot
- computes its control inputs locally using consensus-based MPC.
  Obstacle avoidance: Control barrier functions ensure safety by maintaining a minimum distance
- from obstacles.
  Simulation visualization: The code generates plots to visualize robot trajectories,
- velocities, formation errors, wheel torques, and other key parameters.

Installation Instructions

To run the simulation and visualize the results, follow these steps:

1. **Install MATLAB:** Ensure that MATLAB is installed on your system. This code was tested on MATLAB R2023b but should work on other versions.
2. **Install CasADi:**
   - Download the CasADi toolbox from CasADi official website
   - Add the CasADi toolbox to your MATLAB path.
3. **Clone the Repository :**

   ```
   bash
   git clone https://github.com/your-username/Consensus-MPC-Formation-Control.git
   cd Consensus-MPC-Formation-Control
   ```

4. **Run the code :**
   - Open MATLAB.
   - Navigate to the folder containing the code file, e.g., `Consensus_MPC_formation_control_straight.m`.
   - Run the script: `run('Consensus_MPC_formation_control_straight.m')`

How to Run the Code

1. Open the script, e.g., `Consensus_MPC_formation_control_straight.m`, in MATLAB.
2. Modify parameters if necessary:
   - `activate_formation_control`: Set to `true` to enable formation control with consensus.
   - **Obstacle Position:** Adjust `p_obs` to change the obstacle's position. **Simulation time:**
   - Modify `t0`, `tf`, and `dt` for the desired simulation duration and timestep.
3. Run the script and view the generated plots.

The script will produce the following visualisations:

- Trajectories of robots and the leader.
- Linear and angular velocities.
- Formation errors in x and y directions.
- Wheel angular velocities and torques.

Technical Details

System Dynamics and Kinematics

- **Kinematics**: The robots' motion is governed by differential drive kinematics, which includes states for position, orientation, and wheel velocities. **Dynamics**: The motor dynamics for each
- wheel are modelled, including resistances, inertias, and damping coefficients.
  Consensus Algorithm: The robots use a consensus-based approach to minimise formation errors
- relative to their neighbours and the leader.

Control Methodology

- **Model Predictive Control (MPC)**: The MPC formulation optimises the robots' trajectories over a finite time horizon.
- **Control Barrier Functions (CBF)**: CBFs ensure that robots avoid collisions with obstacles by enforcing safety constraints.

Parameters

The following key parameters are defined in the script:

- **Robot Parameters**
  - e.g., `R, L, J_m, m`: Physical properties of the robot and its wheels.
- **Formation Control**
  - e.g., `Delta`: Desired relative positions between robots in the formation.
- **Simulation Settings**
  - e.g., `tf, dt`: Final simulation time and timestep.

Key Equations

Control inputs are derived by solving optimisation problems formulated as nonlinear programs (NLPs) using CasADi. The objective function includes:

- Formation error minimisation.
- Penalisation of control input rates.
- Obstacle avoidance constraints.

Known Issues and Future Improvements

Known Issues

1. **Obstacle position**: The obstacle position is hardcoded and must be manually modified in the script.
2. **Leader trajectory**: The leader's trajectory is predefined and does not dynamically adapt to the environment.
3. **Computation time**: The simulation can be slow for large numbers of robots due to the computational complexity of solving NLPs for each robot.

Future Improvements

1. **Dynamic obstacles**: Incorporate dynamic obstacle avoidance using real-time updates.
2. **Scalability**: Optimize the code for larger robot teams by reducing computational overhead.
3. **Real-world implementation**: Extend the simulation to work with hardware robots.

License

MIT

[0]You can access the repository at: `https://github.com/FHL-08/Consensus_based_formation_NMPC`.