Omnidirectional Mobile Robots in Agricultural Automation

Data-Efficient Modelling and Robust Control

Elysia Bonello

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Omnidirectional Mobile Robots in Agricultural Automation

Data-Efficient Modelling and Robust Control

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Elysia Bonello

September 22, 2025

Supervisors: Dr. Meichen Guo, Delft Center for Systems and Control, TU Delft
Yixuan Liu, Delft Center for Systems and Control, TU Delft

Faculty of Mechanical Engineering (ME) · Delft University of Technology

GOVERNMENT
OF MALTA

FONDI.eu

The research work disclosed in this publication is partially funded by the Endeavour II Scholarships Scheme. The project is co-funded by the ESF+ 2021-2027

Co-funded by
the European Union

ENDEAVOUR
SCHOLARSHIPS SCHEME

Abstract

Food production in agriculture is facing significant challenges from population growth and labour shortages, increasing the need to automate labour-intensive tasks. Omnidirectional mobile robots (OMRs) with three planar degrees of freedom are well suited to navigating narrow greenhouse aisles for harvesting, pruning, pest detection, and related applications. This thesis identifies a nonlinear model of such an OMR and designs a trajectory-tracking controller for the MIRTE Master platform developed in the Cognitive Robotics department at TU Delft. Limited data availability and onboard computation make this problem especially challenging.

Sparse identification of nonlinear dynamics with control (SINDYc) is used to learn a sparse, physics-informed model of the MIRTE Master from data. On top of this model, robust tube-based nonlinear model predictive control (NMPC) is implemented for real-time trajectory tracking. The controller uses a two-layer structure that combines nominal planning with an invariant error tube to reject disturbances and model mismatch.

The proposed modelling and control pipeline is validated both in simulation and experimentally on the robot. The results demonstrate real-time feasibility and robust tracking performance, supporting the development of reliable and efficient control systems for agricultural OMRs.

Table of Contents

Acknowledgements

I would like to first thank my main supervisor, Dr. Meichen Guo, for her constant support, guidance, and patience throughout this project. I also wish to express my appreciation to my second supervisor, Yixuan Liu, whose insight and engagement during our weekly discussions were invaluable. Together, they greatly shaped the development of this work. I am also grateful to Dr. Chris Pek, who, together with Dr. Meichen, made it possible for me to work on the MIRTE Master robot.

My sincere thanks go to Arend-Jan and Jasper for their continued support with the MIRTE Master robot, always answering my questions promptly and resolving technical issues without delay.

I am deeply grateful to my parents for their unwavering support throughout my MSc journey and for their regular phone calls that kept me grounded. To Danjel, your encouragement from before I began this journey and throughout its most difficult moments has been indispensable, and I am truly thankful for your presence. To my siblings, thank you for the daily messages with updates of the little ones, and to Emma, for your voice notes that reminded me I was never far away. All of you encouraged me whenever I doubted myself, and for that I am profoundly grateful.

Finally, to my DCSC friends and to all those I have met in the Netherlands, thank you for the memorable moments and shared meals that made this experience all the more rewarding.

Delft, University of Technology — Elysia Bonello
September 22, 2025

Chapter 1

Introduction

The global demand for agricultural production continues to rise due to population growth. At the same time, the agricultural sector faces a declining labour force, further straining the supply of food products [8]. This challenge is compounded by the limited availability of arable land, constrained by climate conditions, soil quality, and other environmental factors.

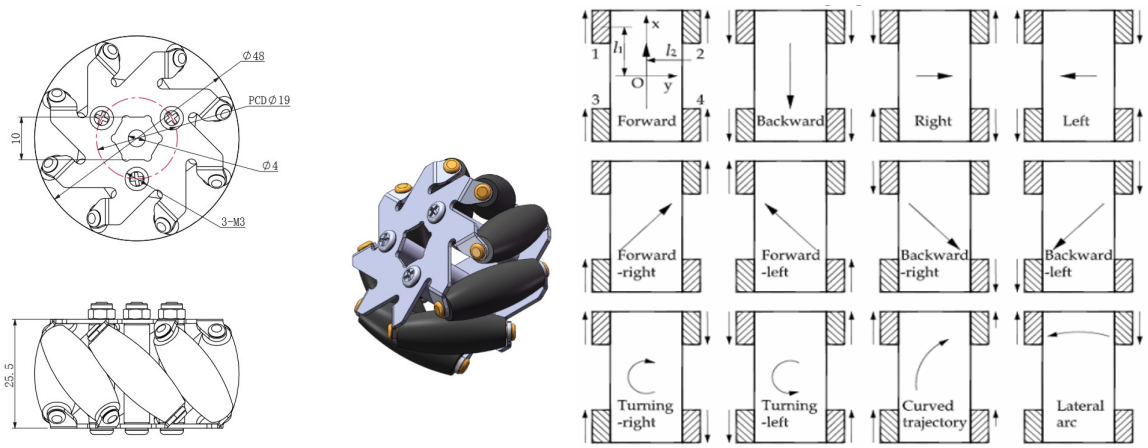
To meet this growing demand, agricultural productivity must improve in a sustainable manner. One promising solution is the automation of labour-intensive tasks. Over the past decade, ground-based robots have been developed for a range of agricultural applications, including automated planting, weeding, harvesting, and more [49], as shown in Figure 1-1. However, these autonomous systems face significant challenges in achieving safe, efficient, and reliable operation. This has led to the emergence of precision agriculture (PA), a multidisciplinary approach to farm management that aims to optimise yields through the efficient use of available resources [48].

Figure 1-1: An OMR deployed in a tomato crop greenhouse [33]

In this context, a reconfigurable omnidirectional mobile robot (OMR) designed for greenhouse environments is introduced in [42]. The robot improves manoeuvrability and adaptability in the confined spaces typical of such settings. The OMR considered in this work are equipped with Mecanum wheels which exhibit holonomic motion capabilities, allowing movement in any planar direction without changing orientation. A Mecanum wheel consists of multiple passive rollers, typically mounted at $\pm 45^\circ$ around the wheel circumference, as shown in Figure 1-2a [9]. The possible motion obtained by an OMR is illustrated in Figure1-2b. These robots can perform three degrees of freedom (DoF) motion in a two-dimensional space, including translation in both the x and y directions, and rotation about the vertical axis [51].

To function effectively in agricultural environments, OMRs require robust and precise control strategies. Consequently, suitable model identification is essential for achieving satisfactory model-based trajectory tracking, thus ensuring reliable performance and improved efficiency in precision farming applications. The work in this thesis focuses on two central challenges. The first concerns the identification of a sufficiently accurate nonlinear model of an OMR, in this case the MIRTE Master robot. The second involves the development of a control strategy that can guide the robot along precise trajectories for PA tasks.

The MIRTE Master robot, developed by the Cognitive Robotics department at the Mechanical Engineering Faculty of TU Delft, serves as the experimental platform for this work. It operates on a Robot Operating System (ROS) platform and is currently being transitioned to ROS 2. At present, the robot lacks a formally identified model and no control is implemented. Motion is executed in an open loop configuration, where pulse width modulation (PWM) signals are directly applied to the wheel motors without feedback correction. This absence of feedback control and accurate modelling presents clear limitations in achieving the performance required for PA.



(a) Mecanum wheel design [47] **(b)** OMR motion [52]

Figure 1-2: Mecanum wheel and possible OMR motion

1-1 Challenges in the MIRTE Master OMR

One of the main difficulties of the MIRTE Master robot lies in obtaining accurate state information. Body velocities are derived from wheel encoder measurements and integrated to estimate position and orientation. However, wheel encoders are subject to errors such as quantisation noise, electrical interference, and wheel slip, particularly on uneven terrain [39]. These factors introduce small velocity errors at each integration step, which accumulate over time and lead to drift in the pose estimate.

The onboard inertial measurement unit (IMU) provides additional motion information, but its affordability comes with reduced precision, increased biases, and susceptibility to drift over time [39]. Previous attempts to fuse encoder and IMU data using an extended Kalman filter (EKF) have, in this case, degraded estimation performance. As a result, body velocity estimates rely solely on wheel encoders and remain noisy.

The robot adopts a two-layer hardware architecture. The OrangePi single-board computer runs ROS 2 and hosts the high-level control algorithm, while a dedicated microcontroller performs low-level tasks such as reading sensor data and controlling actuators. While this arrangement ensures responsive timing for low-level operations, it introduces communication latency and variable update rates between the two devices. Real-time communication stacks can experience latency variation and jitter, particularly in multi-node ROS 2 systems [24], which limits the frequency at which control loops can reliably be executed.

These restrictions impose a dual constraint on controller design. The control algorithm running on the OrangePi must be computationally efficient and tolerant of limited update rates, while remaining robust to noisy state estimates and communication delays. Traditional proportional, integral, derivative (PID) control can face difficulties, as the integral term is prone to windup under delays and actuator saturation, while the derivative term amplifies measurement noise. PIDs also struggle with the strong state-dependent coupling present in the robot dynamics [61].

In this context, model predictive control (MPC) provides a structured way to handle such challenges. By embedding system dynamics and explicit constraints directly into the optimisation, it can use prediction to anticipate future states rather than reacting solely to the present. This allows improved trajectory tracking, reduces unnecessary actuation, and helps extend battery life. Furthermore, a sufficiently accurate model increases system interpretability and provides a foundation for integrating advanced control techniques in future agricultural tasks.

For these reasons, the objective of this thesis is to develop and verify suitable methods for modelling and trajectory control of the MIRTE Master robot, with particular attention to the limitations imposed by scarce noisy data and restricted computational resources.

1-2 Thesis Goals

The aim of this thesis is to establish a solution that balances model accuracy, control performance, and computational efficiency. The modelling and control methods are intended for deployment across multiple MIRTE Master robots without requiring extensive per-unit

tuning. This is essential because, although the robots share a common design, their physical properties vary slightly between units, and these differences can have significant consequences if not addressed.

The operating conditions and system dynamics of the MIRTE Master robot can change considerably due to factors such as variations in ground surface properties and the execution of different agricultural tasks, including harvesting and pest monitoring. For this reason, the methods considered in this thesis are required to perform reliably under low-data conditions and within the computational limits of the onboard controller. The broader objective is to identify approaches that are adaptable to the dynamic and constrained environments typical of PA.

Based on these considerations, the specific goals of this thesis are:

- To identify a sufficiently accurate model of the MIRTE Master OMR for use in agricultural environments, given limited data availability and constrained computational resources.
- To validate the identified model using different open-loop trajectory paths commonly encountered in greenhouse operations.
- To design, within the stated constraints, an effective control strategy that optimises the trajectory tracking performance of the OMR along greenhouse paths while ensuring robust performance.
- To verify the proposed control method both in simulation and experimental tests on the MIRTE Master robot for trajectory tracking.

Beyond these goals, a key practical contribution of this thesis is a transparent, step-by-step synthesis workflow that turns data and hardware constraints into a deployable controller. It makes the approximations, parameter choices, and design trade-offs explicit, which most reviewed literature presents only at a high level.

1-3 Outline

This work begins with the development of a mathematical model of the OMR dynamics in Chapter 2. The same chapter also presents a concise review of the existing literature on OMRs, covering modelling, identification, and trajectory control methods. Subsequently, Chapter 3 details the formal system identification method and the procedure used to identify and validate the dynamics, accompanied by a discussion of practical implementation challenges and possible strategies for improvement.

Chapter 4 formulates the trajectory control problem, synthesises the controller for the MIRTE Master robot, and reports simulation verification results. Chapter 5 reports on the real-time deployment of the control algorithm, presenting both simulation and experimental results, with the experimental results compared against a baseline controller. Finally, Chapter 6 concludes the thesis and outlines potential directions for future work.

Chapter 2

Modelling Dynamics of OMRs

This chapter reviews the principal strategies for modelling the dynamics of OMRs, forming the foundation for system identification and trajectory control design in later sections. Two modelling approaches are presented in this work, namely the kinematic model and the dynamic model. The term dynamics can be interpreted in two ways. In its broadest sense it refers to how the system evolves over time. In a more specific sense a dynamic model captures the underlying physical phenomena, including forces, torques, friction, and inertia.

2-1 Kinematic modelling

The pioneering work of Muir and Neuman [37] established the fundamental kinematic relationships that map body-frame velocities to individual wheel angular velocities. The standard kinematic model assumes ideal conditions such as no slip and perfect roller contact, thus neglecting frictional effects, wheel slip, and inertia. Despite these simplifications, the model remains nonlinear because the global motion depends on trigonometric functions of the heading angle, which varies over time.

The geometric parameters of an OMR are illustrated in Figure 2-1, where world-frame coordinates are indicated by the subscript w and body-frame coordinates by the subscript b. As shown in the figure, the MIRTE Master robot follows the ROS body-frame coordinates convention, meaning that the positive x-axis corresponds to forward motion, the positive y-axis corresponds to motion to the left, and a positive change in the yaw angle corresponds to counter-clockwise rotation [56]. The states of the system in the world frame are defined by the position and orientation of the robot, with derivatives $\dot{\mathbf{P}}_w = [\dot{x}_w \ \dot{y}_w \ \dot{\theta}]^T$, while the state derivatives in the body frame are denoted by $\dot{\mathbf{P}}_b = [v_x \ v_y \ \Omega]^T$.

The measurement data from the MIRTE Master robot are obtained through the `odom` package in ROS 2, which publishes odometry information on a dedicated topic. This package provides the robot pose expressed in the world frame and the linear and angular velocities expressed in the body frame.

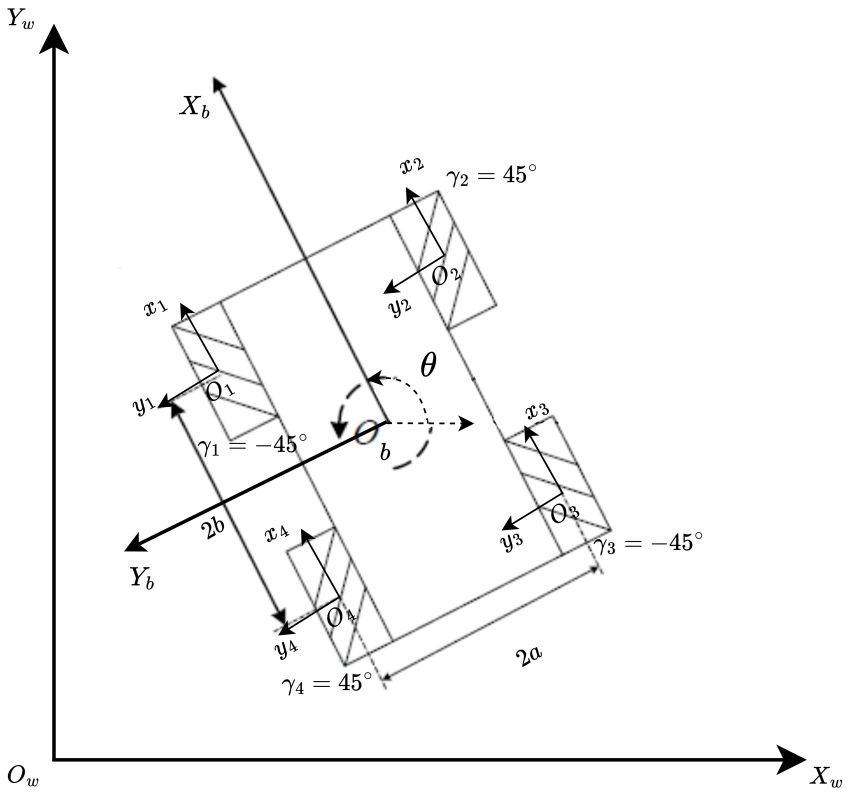


Figure 2-1: Geometric Parameters of the OMR adapted from [9]

A coordinate transformation matrix $\mathbf{R}(\theta) \in \mathbb{R}^{3\times3}$ relates the two coordinate systems, as given by

$$\underbrace{\begin{bmatrix}\dot{x}_w\\ \dot{y}_w\\ \dot{\theta}\end{bmatrix}}_{\dot{\mathbf{P}}_w} = \underbrace{\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}}_{\mathbf{R}(\theta)} \underbrace{\begin{bmatrix}v_x\\ v_y\\ \Omega\end{bmatrix}}_{\dot{\mathbf{P}}_b}. \tag{2-1}$$

Each wheel is actuated by an independent motor, allowing individual control of the wheel velocities. The relationship between the wheel angular velocities $\boldsymbol{\omega} \in \mathbb{R}^4$ and the body velocities of the robot is expressed as

$$\underbrace{\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix}}_{\boldsymbol{\omega}} = \frac{1}{R} \underbrace{\begin{bmatrix}1 & -1 & -(a+b)\\ 1 & 1 & (a+b)\\ 1 & -1 & (a+b)\\ 1 & 1 & -(a+b)\end{bmatrix}}_{\mathbf{J}} \begin{bmatrix}v_x\\ v_y\\ \Omega\end{bmatrix}, \tag{2-2}$$

where $\mathbf{J} \in \mathbb{R}^{4\times3}$ denotes the inverse kinematic Jacobian matrix, R is the wheel radius, and a and b are half the wheelbase dimensions, as illustrated in Figure 2-1.

The forward kinematic model, which expresses the vehicle velocities in terms of the wheel angular velocities, is obtained using the Moore–Penrose pseudoinverse $\mathbf{J}^+ \in \mathbb{R}^{3\times4}$, as given by

$$\begin{aligned}\dot{\mathbf{P}}_b &= \mathbf{J}^+\boldsymbol{\omega} = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\boldsymbol{\omega} \\ \begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} &= \frac{R}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ \frac{-1}{a+b} & \frac{1}{a+b} & \frac{1}{a+b} & \frac{-1}{a+b} \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}.\end{aligned} \tag{2-3}$$

The dynamic relation should reflect the available input and output data of the MIRTE Master OMR. Therefore, the existing ROS 2 packages were examined to determine the raw command applied to the direct current (DC) motors. The motors are driven by PWM duty-cycle commands, $\mathbf{u} \in \mathcal{U}$, where $\mathcal{U} = [-100, 100]^4 \subset \mathbb{R}^4$. Consequently, the kinematic model must be reformulated so that the world-frame state derivatives are parameterised by the PWM inputs of each wheel. From the `mirte_base_control` package [36], the wheel angular velocity to PWM mapping is implemented as

$$\mathbf{u} = \mathrm{sat}_{[-100,100]}\left(\frac{100}{6\pi}\boldsymbol{\omega}\right), \tag{2-4}$$

where 6π rad/s corresponds to the maximum command. Hence, the mapping is proportional with saturation. In the unsaturated region, the relation is reduced to

$$\boldsymbol{\omega} = K\mathbf{u}, \quad K = \frac{6\pi}{100}, \quad \mathbf{u} \in [-100, 100]^4 \subset \mathbb{R}^4. \tag{2-5}$$

Therefore, the final 3-DoF kinematic model of the OMR with PWM inputs is given by

$$\begin{aligned}\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta} \end{bmatrix} &= \frac{RK}{4}\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ \frac{-1}{a+b} & \frac{1}{a+b} & \frac{1}{a+b} & \frac{-1}{a+b} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \\ \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta} \end{bmatrix} &= \frac{RK}{4}\begin{bmatrix} \cos\theta + \sin\theta & \cos\theta - \sin\theta & \cos\theta + \sin\theta & \cos\theta - \sin\theta \\ \sin\theta - \cos\theta & \sin\theta + \cos\theta & \sin\theta - \cos\theta & \sin\theta + \cos\theta \\ \frac{-1}{a+b} & \frac{1}{a+b} & \frac{1}{a+b} & \frac{-1}{a+b} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}.\end{aligned} \tag{2-6}$$

While these simplifications are computationally efficient, they can introduce drift and tracking errors. In particular, errors in the heading angle affect the motion of the robot due to the state-dependent nature of the trigonometric terms. Accuracy can be improved by incorporating other mechanical effects into a dynamic model.

2-2 Dynamic modelling

Certain design characteristics of mecanum wheels introduce disadvantages for navigation. Although translational movement is enabled, vibrations that appear at high speeds reduce energy efficiency because of the contact forces between the rollers and the ground [1]. For this reason, the dynamic model is formulated so that the forces acting on the system are represented explicitly.

Four-wheeled OMR dynamics can be derived using Newton's Second Law of Motion [32, 7], Newton–Euler equations [20], or Lagrangian mechanics [54]. Different studies often present variations of the same system. Certain contributions are grouped into effective parameters, others are neglected as higher-order effects, and some are retained explicitly.

In this work, the dynamics of an OMR are derived from Newton's Second Law of Motion. By analogy with the kinematic model, the dynamic relations are expressed in terms of the available input and output data. This data consists of the world-frame coordinates, the body-frame velocities, and the wheel PWM control input signals. The dynamic model of an OMR, written in terms of the world-frame accelerations, is given as

$$\underbrace{\begin{bmatrix} F_{w_x} \\ F_{w_y} \\ \tau \end{bmatrix}}_{\mathbf{F}_w} = \underbrace{\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_w \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\theta} \end{bmatrix}}_{\ddot{\mathbf{P}}_w}, \tag{2-7}$$

where F_{w_x} and F_{w_y} are the total forces in the world x and y directions, τ is the torque about the centroid of the robot, m is the total mass of the robot and I_w is the moment of inertia about the rotation axis.

The forces $\mathbf{F}_w \in \mathbb{R}^3$ applied to the OMR are then analysed. The driving force at each wheel i is generated by its DC motor and is expressed as

$$F_{d_i} = \alpha V_i - \beta R \omega_i \qquad i \in \{1, 2, 3, 4\}, \tag{2-8}$$

where α and β are the DC motor coefficients and V_i is the input voltage applied to wheel i [16]. The second term represents velocity-dependent effects that model motor back-electromotive force (EMF). The input voltage vector $\mathbf{V} \in [-V_s, V_s]^4 \subset \mathbb{R}^4$ is proportional to the PWM control input signal $\mathbf{u}$ and is written as

$$\mathbf{V} = K_v \mathbf{u} = \frac{V_s}{100} \mathbf{u}, \tag{2-9}$$

where V_s is the motor supply voltage.

In this model, two types of friction are considered. Static friction, denoted by $\mathbf{F}_s \in \mathbb{R}^4$, prevents the rollers from slipping and counteracts the initiation of wheel motion during ground contact. Viscous friction, denoted by $\mathbf{F}_v = D_v \boldsymbol{\omega} \in \mathbb{R}^4$, resists the angular velocity of the wheels through internal damping of the shafts [29].

The total generalised force acting on the robot is then transformed into the world frame and is expressed as

$$\begin{aligned} \mathbf{F}_w &= \mathbf{R}(\theta)\mathbf{T}\left[\mathbf{F}_d - \mathbf{F}_s - \mathbf{F}_v\right] \\ \begin{bmatrix} F_{w_x} \\ F_{w_y} \\ \tau \end{bmatrix} &= \mathbf{R}(\theta)\mathbf{T}\left[\left(\alpha K_v \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} - \beta R \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}\right) - \begin{bmatrix} F_{s_1} \\ F_{s_2} \\ F_{s_3} \\ F_{s_4} \end{bmatrix} - D_v \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}\right], \end{aligned} \tag{2-10}$$

where $\mathbf{T} \in \mathbb{R}^{3\times 4}$ is the wheel-to-body transformation matrix, defined as

$$\mathbf{T} = \begin{bmatrix} \cos\gamma_1 & \cos\gamma_2 & \cos\gamma_3 & \cos\gamma_4 \\ \sin\gamma_1 & \sin\gamma_2 & \sin\gamma_3 & \sin\gamma_4 \\ a\sin\gamma_1 - b\cos\gamma_1 & a\sin\gamma_2 + b\cos\gamma_2 & b\cos\gamma_3 - a\sin\gamma_3 & -a\sin\gamma_4 - b\cos\gamma_4 \end{bmatrix} \tag{2-11}$$

and $\gamma_i \in \{-45°, 45°\}$ denotes the roller angle of wheel i, as illustrated in Figure 2-1.

Given that body-frame velocity measurements are available, Equation (2-2) is applied to substitute the angular velocity into Equation (2-10). In compact form, this yields

$$\mathbf{F}_w = \mathbf{R}(\theta)\mathbf{T}\left[\alpha K_v \mathbf{u} - \beta \mathbf{J}\dot{\mathbf{P}}_b - \mathbf{F}_s - \frac{1}{R} D_v \mathbf{J}\dot{\mathbf{P}}_b\right]. \tag{2-12}$$

This expression is substituted into Equation (2-7) to obtain the dynamic model of an OMR,

$$\ddot{\mathbf{P}}_w = \mathbf{M}^{-1}\mathbf{R}(\theta)\mathbf{T}\left[\alpha K_v \mathbf{u} - \mathbf{F}_s - \left(\frac{1}{R} D_v + \beta\right)\mathbf{J}\dot{\mathbf{P}}_b\right]. \tag{2-13}$$

The transformation matrices $\mathbf{T}$ and $\mathbf{R}(\theta)$ ensure that the wheel forces are mapped consistently to the body frame and then expressed in the world frame coordinates. In this way, the derived model provides a physically consistent description of how the PWM signals applied to the wheels result in translational and rotational accelerations of the OMR.

2-3 Overview of modelling & control approaches in the literature

The studies in Table 2-1 highlight the diversity of modelling and control approaches proposed for four-wheeled OMRs with the same chassis layout as the MIRTE Master robot. Existing work illustrates a trade-off between model complexity and fidelity. Kinematic models dominate due to their simplicity, but they fail to capture important effects such as slip and inertia. Dynamic models, by contrast, provide a richer description but require detailed parameter identification, which is rarely validated outside structured environments. Another approach explored in the literature review is data-based modelling, which constructs models directly from measurement data without relying on physical knowledge. Although promising, most of these methods rely on large datasets, and their limited interpretability poses challenges for deployment in agricultural settings.

These observations motivate the use of modelling approaches that combine physics-based formulations with measurement data, remaining parsimonious and interpretable while still

Table 2-1: Literature of modelling and control of four-wheeled rectangular chassis OMRs

Study	Modelling Dynamics	Control Approach	Simulation	Experiment	Stability Analysis
[50]	Kinematic model	Linear Time-Varying MPC with PID cascade control	Yes	Yes	No
[30]	Kinematic with adaptive slippage model	Online adaptive control	No	Yes	No
[29]	Kinematic and dynamic models	NN Adaptive SMC and PID cascade control	Yes	No	No
[5]	Dynamic model	-	Yes	No	No
[61]	Dynamic Model	SMC	Yes	No	Lyapunov-based asymptotic stability
[2]	Dynamic Model	Adaptive SMC	Yes	Yes	Lyapunov-based asymptotic stability
[14]	Differentiable physics-based model with online friction learning using NN	Gradient-based optimization using NN	Yes	Yes	No

capturing the dominant nonlinear behaviours relevant for control design. In particular, the upcoming chapters introduce the sparse identification of nonlinear dynamics with control (SINDYc) framework and its integration with tube-based nonlinear model predictive control (NMPC), which directly address these challenges.

Chapter 3

System Identification using SINDYc

Precise trajectory tracking of four-wheeled OMRs in greenhouse environments requires models that are sufficiently accurate and computationally efficient. The preceding review has shown that existing kinematic, dynamic, and data-based models each present limitations related to fidelity, parameter dependence, data requirements, or interpretability. To obtain a sufficiently accurate and tractable model for control, the present chapter develops a modelling framework based on SINDYc. This method combines the parsimony of physics-based structures with the flexibility of data-driven identification, allowing essential nonlinear dynamics to be captured from relatively limited and noisy experimental data while reducing the risk of overfitting [45].

An overview of the SINDYc framework is shown in Figure 3-1. A dataset is collected on the MIRTE Master robot, after which a library of candidate nonlinear functions is constructed. The library can be formed from generic nonlinear functions such as polynomial and trigonometric terms, and also from terms informed by known kinematic and dynamic relationships [12]. Control inputs are included in the library to represent actuation, which yields the controlled formulation SINDYc rather than the original sparse identification of nonlinear dynamics (SINDy) method [22]. Sparse regression is then applied to identify the dominant terms that describe the measured dynamics.

In this chapter, the theoretical background of SINDYc is presented together with its algorithmic implementation for system identification. The method is applied to the MIRTE Master robot to identify kinematic, dynamic, and extended hybrid models, with the aim of selecting the most suitable representation for control. Experimental validation is carried out to demonstrate that SINDYc produces interpretable models that are suitable for subsequent control design.

3-1 SINDYc framework

A continuous-time nonlinear system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ is considered with state $\mathbf{x} \in \mathbb{R}^n$ and input $\mathbf{u} \in \mathbb{R}^m$. N snapshots are collected at times t_k and arranged column-wise as

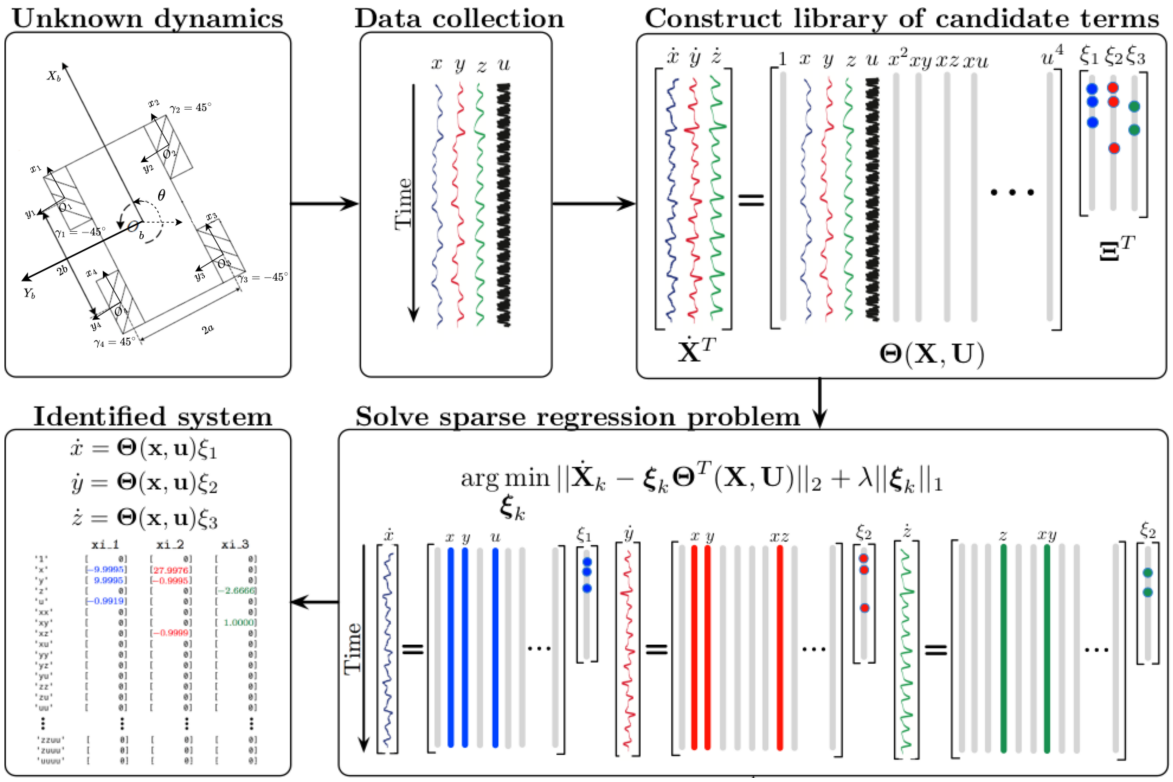


Figure 3-1: SINDYc algorithm adapted from [22]

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \end{bmatrix}. \tag{3-1}$$

A selected library of candidate functions is denoted by $\mathbf{\Theta}(\mathbf{X}, \mathbf{U}) \in \mathbb{R}^{N \times L}$, which contains L nonlinear functions of the state and input vectors evaluated over N sampled instants. The objective is to identify a sparse coefficient matrix $\mathbf{\Xi} \in \mathbb{R}^{n \times L}$ such that

$$\dot{\mathbf{X}} \approx \mathbf{\Xi}\mathbf{\Theta}^T(\mathbf{X}, \mathbf{U}), \tag{3-2}$$

where $\dot{\mathbf{X}} \in \mathbb{R}^{n \times N}$ contains the derivatives of the n states at the sampled instants. The sparse coefficient matrix is estimated efficiently using the sequentially thresholded least squares (STLSQ) algorithm [22],

$$\begin{aligned} \hat{\boldsymbol{\xi}}_k = \arg\min_{\hat{\boldsymbol{\xi}}_k} \frac{1}{2}\|\dot{\mathbf{X}}_k - \hat{\boldsymbol{\xi}}_k \mathbf{\Theta}^T(\mathbf{X}, \mathbf{U})\|_2^2 + \frac{\lambda_r}{2}\|\hat{\boldsymbol{\xi}}_k\|_2^2 \\ \text{subject to} \quad |\hat{\xi}_{k_l}| \geq \lambda_k \quad \text{or} \quad |\hat{\xi}_{k_l}| = 0, \end{aligned} \tag{3-3}$$

where $\dot{\mathbf{X}}_k \in \mathbb{R}^N$ is the trajectory of the k^{th} state derivative with $k \in \{1, \dots, n\}$, $\hat{\boldsymbol{\xi}}_k \in \mathbb{R}^L$ is the sparse coefficient vector that selects the active candidate terms for the k^{th} state, and $\hat{\xi}_{k,l}$ denotes the coefficient associated with the l^{th} candidate function in the library for the k^{th} state. The parameter $\lambda_r > 0$ is a small ridge term that ensures numerical conditioning

of the regression, while $\lambda_k > 0$ is the sparsity threshold that balances model parsimony with accuracy. Each stage of the algorithm is described in detail in the following subsections.

3-1-1 Data collection

The first step in any system identification method is to collect data from the system being studied. The systematic design of persistently exciting training data has received little attention in the SINDy literature and its extensions. The convergence rate of the identified model depends strongly on the excitation signal, which directly influences data richness [19]. By contrast, in the broader field of data-driven identification and control, persistency of excitation is recognised as a fundamental requirement for ensuring data richness and guaranteeing unique model recovery.

SINDYc has been shown to require substantially less training data than NN-based approaches. For example, a NN trained on the Lotka–Volterra system required 250 trajectories of 10^3 snapshots each to achieve satisfactory performance, whereas SINDYc achieved comparable accuracy using only a single trajectory of 10^4 snapshots. In addition, SINDYc identified the correct governing terms of the system from limited observations, which enabled accurate prediction of behaviours beyond the training trajectories [22].

For the MIRTE Master platform, world-frame states, body-frame state derivatives and PWM input trajectories are sampled every 0.1 s, and the dataset covers the complete range of relevant motions as illustrated in Figure 1-2b.

The yaw angle is extracted from the pose quaternion and unwrapped to remove the discontinuity between $-\pi$ rad and π rad. Without unwrapping, finite-difference derivatives and the subsequent sparse regression interpret $-\pi$ and π as different states, creating artificial jumps in the data. For example, a transition from 0.99π to π should correspond to a near-zero derivative, but when wrapped it appears as a large negative jump from π to $-\pi$. This discontinuity prevents the method from converging to the true system dynamics. By unwrapping, the angular trajectory remains continuous, allowing derivatives to be estimated correctly and ensuring consistent identification.

The algorithm used to generate the training data is presented in Algorithm 1. Excitation is generated by alternating short, randomised templates with randomised magnitudes during the first 75% of the run. In the remaining 25%, patterned segments are applied. Within each pattern, equal PWM commands u_i with $i \in \{1, 2, 3, 4\}$, are applied to the active motors while the inactive motors are held at zero. This design promotes excitation across all motion modes while maintaining relevance to practical robot operation. Unlike benchmark systems such as Lotka–Volterra, where the governing equations consist of simple polynomial interactions that can be recovered from limited trajectories, the OMR requires targeted excitation of structured motion patterns to ensure that slip, friction, and coupling effects are sufficiently represented during identification. The templates are updated approximately every $\Delta t_u = 1$ s, so that translation in x, translation in y, and yaw motion are all excited in comparable proportions, which supports the correct identification of each state derivative $\dot{\mathbf{x}}$.

A slew-rate limit $|\Delta u_{\max}| = 20$ is imposed on the PWM commands to protect the DC motors and to align state and input dynamics in time. This constraint produces smooth trajectories that are representative of nominal operation and also improves numerical differentiation. The

Algorithm 1: PWM excitation for SINDYc data collection

Input: duration $T = 120$s, control interval $\Delta t = 0.3$s, template update interval $\Delta t_u = 1$s, maximum PWM $|u_{\max}| = 100$, maximum PWM change $|\Delta u_{\max}| = 20$

Output: command sequence $\mathbf{u}(t) \in [-u_{\max}, u_{\max}]^4$

Initialise time $t \leftarrow 0$, $t_u \leftarrow \Delta t_u$, command $\mathbf{u} \leftarrow \mathbf{0}$

while $t < T$ **do**
- **if** $t \leq 60$ **then**
  - **if** $t_u \geq \Delta t_u$ **then**
    - Select a random template that targets x, y, or yaw so that coverage remains balanced
    - Select a random magnitude $u_i \sim \mathcal{U}(30, 100)$ for each wheel i to form $\mathbf{u}$
    - $t_u \leftarrow 0$
  - **end**
- **else**
  - **if** $t_u \geq \Delta t_u$ **then**
    - Select a random pattern from a fixed list with equal magnitudes to form $\mathbf{u}$
    - $t_u \leftarrow 0$
  - **end**
- **end**
- **foreach** *wheel* i **do**
  - $d_i \leftarrow \bar{u}_i - u_i$
  - $\Delta u_i \leftarrow \text{clamp}(d_i,\ -\Delta u_{\max},\ \Delta u_{\max})$
  - $u_i \leftarrow \text{clamp}(u_i + \Delta u_i,\ -u_{\max},\ u_{\max})$
- **end**
- Send $\mathbf{u}$ to the motors and record $(\mathbf{u}(t), \mathbf{x}(t), \dot{\mathbf{x}}(t))$
- $t \leftarrow t + \Delta t$
- $t_u \leftarrow t_u + \Delta t$

end

absolute PWM magnitudes are drawn from the interval $[30, 100]$, with $|u_{\max}| = 100$ denoting the saturation level. The inherent deadband within $(-30, 30)$ is traversed during the ramps defined by Δu_i, which provides informative samples near zero and assists in identifying friction terms that appear in the dynamic model.

The actuation signals $\mathbf{u}$ are applied directly to the wheel motors and are therefore independent of the measured states, which avoids ambiguity between process dynamics and control action during identification.

3-1-2 Numerical differentiation using total variation regularised (TVR)

After obtaining the dataset containing the robot world-frame pose and body-frame velocities, the next step in the SINDYc framework is to estimate the state derivatives required for model identification. Specifically, the state derivative matrix $\dot{\mathbf{X}}$ must be expressed in the world frame, so the body-frame velocities are mapped to world-frame derivatives using the kinematic transformation in Equation (2-1).

However, since the body-frame velocities are computed from incremental encoder counts, they are affected by commutation ripple and brush bounce. Furthermore, broadband disturbances are injected into the estimated wheel angular velocities [58]. At low speeds, encoder quantisation becomes pronounced and a delay between samples is introduced [62]. Therefore, the estimated speed is held constant for several samples and short spikes appear when a pulse arrives, so the true smooth motion is not represented. Direct finite-difference differentiation is therefore noisy and biased. World-frame velocities are instead estimated using TVR applied to the world-frame pose trajectory. With appropriate tuning, high-frequency and impulsive noise components are attenuated while physically meaningful edges are preserved [6].

The TVR problem is formulated to estimate the derivative without amplifying the measurement noise. An integrated derivative is fitted to the data and the total variation of the derivative is penalised so that a piecewise-smooth estimate is produced that preserves genuine changes [6]. The approximated world-frame state derivative $\hat{\dot{\mathbf{X}}}$ is determined by solving the optimisation problem

$$\min_{\hat{\dot{\mathbf{X}}}} \tfrac{1}{2}||A_T\hat{\dot{\mathbf{X}}} - (\mathbf{X} - \mathbf{X}_0)||_2^2 + \alpha \sum_{i=1}^{N} \sqrt{(D\hat{\dot{\mathbf{X}}})_i^2 + \varepsilon}\ , \tag{3-4}$$

where for the MIRTE Master robot, $\mathbf{X}$ denotes the measured world-frame pose of the robot with initial pose $\mathbf{X}_0$. The operator A_T performs discrete integration on the sampling grid and reconstructs the approximated velocity along the pose trajectory relative to the initial pose. The second term penalises rapid oscillations in the estimated velocity across all time samples using the first difference operator D and the regularisation parameter α. Larger values of α suppress encoder and localisation noise more strongly but may oversmooth slope changes. The parameter ε ensures numerical stability and influences corner sharpness. Smaller values preserve sharper transitions but can reduce conditioning [6].

3-1-3 Library of nonlinear candidate functions

Once the state derivatives $\dot{\mathbf{X}}$ have been estimated, the next step in the SINDYc framework is to specify the library of candidate functions from which the governing equations are identified. This library $\boldsymbol{\Theta}(\mathbf{X}, \mathbf{U})$ provides the dictionary of possible nonlinear interactions between the states and inputs. It is constructed from terms motivated by the kinematic and dynamic models derived in Chapter 2. For the extended kinematic model, the library is further augmented with additional nonlinear terms to account for unmodelled effects, as described is more detail in Section 3-2. Each column is normalised so that all terms have a comparable scale. This procedure prevents numerical bias during regression [55].

The candidate terms included for each state derivative in the kinematic model are listed in Table 3-1. The coefficients are not shown, since they are identified during regression, and only the structure of the model is tabulated. These terms follow directly from Equation (2-6).

For the dynamic model, Equation (2-13) is expanded to obtain the candidate terms shown in Table 3-2. The terms $\sin(\theta)$ and $\cos(\theta)$ in the translational accelerations and the constant term in the angular acceleration arise from the static friction component. In this work, these terms are omitted from the library. A constant contribution would be integrated at every time step and cause the system state to drift even when the control inputs are zero.

State derivative	Candidate terms
$\dot{x}_w$	$\sin(\theta)\mathbf{u}$, $\cos(\theta)\mathbf{u}$
$\dot{y}_w$	$\sin(\theta)\mathbf{u}$, $\cos(\theta)\mathbf{u}$
$\dot{\theta}$	$\mathbf{u}$

Table 3-1: Candidate library terms for the kinematic model

State derivative	Candidate terms
$\ddot{x}_w$	$\sin(\theta)\mathbf{u}$, $\cos(\theta)\mathbf{u}$, $\sin(\theta)$, $\cos(\theta)$, $\sin(\theta)v_y$, $\cos(\theta)v_x$
$\ddot{y}_w$	$\sin(\theta)\mathbf{u}$, $\cos(\theta)\mathbf{u}$, $\sin(\theta)$, $\cos(\theta)$, $\sin(\theta)v_x$, $\cos(\theta)v_y$
$\ddot{\theta}$	$\mathbf{u}$, constant, Ω

Table 3-2: Candidate library terms for the dynamic model

Angular acceleration is determined by the net yaw torque generated by the wheels. The torque is represented as a linear function of the PWM input signals $\mathbf{u}$ with a damping term proportional to the angular velocity Ω. Rolling resistance, viscous losses, and floor interaction are reflected primarily through this damping term [38].

3-1-4 Sparse regression

With the state derivatives estimated and the nonlinear candidate library defined, the final step in the SINDYc identification procedure is to determine which terms best describe the system dynamics. This is achieved using sparse regression, which identifies a parsimonious model from the normalised library by solving Equation (3-3) for each state $k \in \{1, 2, 3\}$. The library is normalised before fitting so that the sparsity threshold acts fairly across all terms [45].

The STLSQ procedure begins with a least-squares fit. Coefficients with magnitude below the sparsity threshold λ_k are set to zero. Equation (3-3) is then solved on the remaining active set. Thresholding and refitting are repeated until a sparse representation is obtained and the active set no longer changes [63].

The threshold $0 < \lambda_k < 1$ is scaled to reflect the normalisation of the library and the magnitude of the numerical derivatives. Larger thresholds yield sparser models and improve interpretability but may remove physically relevant terms. Smaller thresholds lower the residuals on the identification data, but may admit noise and reduce predictive performance on new data [45]. The threshold for each state derivative is set by selecting the least approximation error between the TVR-based derivatives, and the model constructed from the identified coefficients and the library. A very small ridge regularisation term λ_r is included to improve numerical conditioning. The bias introduced by this term is negligible and the stability of the fit is preserved.

3-2 Model validation results

Datasets were collected on the MIRTE Master robot, world-frame derivatives were estimated using TVR differentiation, and libraries of candidate functions were constructed from kinematic and dynamic models. Sparse regression with STLSQ was then applied to obtain parsimonious models that balanced accuracy, interpretability, and robustness to noise. This formulation ensured that physically meaningful terms were retained while redundant or noise-driven contributions were removed.

The SINDYc algorithm is implemented in MATLAB, and the identified models are evaluated by comparing predicted state trajectories with independent validation data and by assessing whether the identified dynamics generalise beyond the training set. This evaluation demonstrates that the coefficients obtained through sparse regression capture the essential behaviour of the robot and provide a reliable basis for subsequent control design.

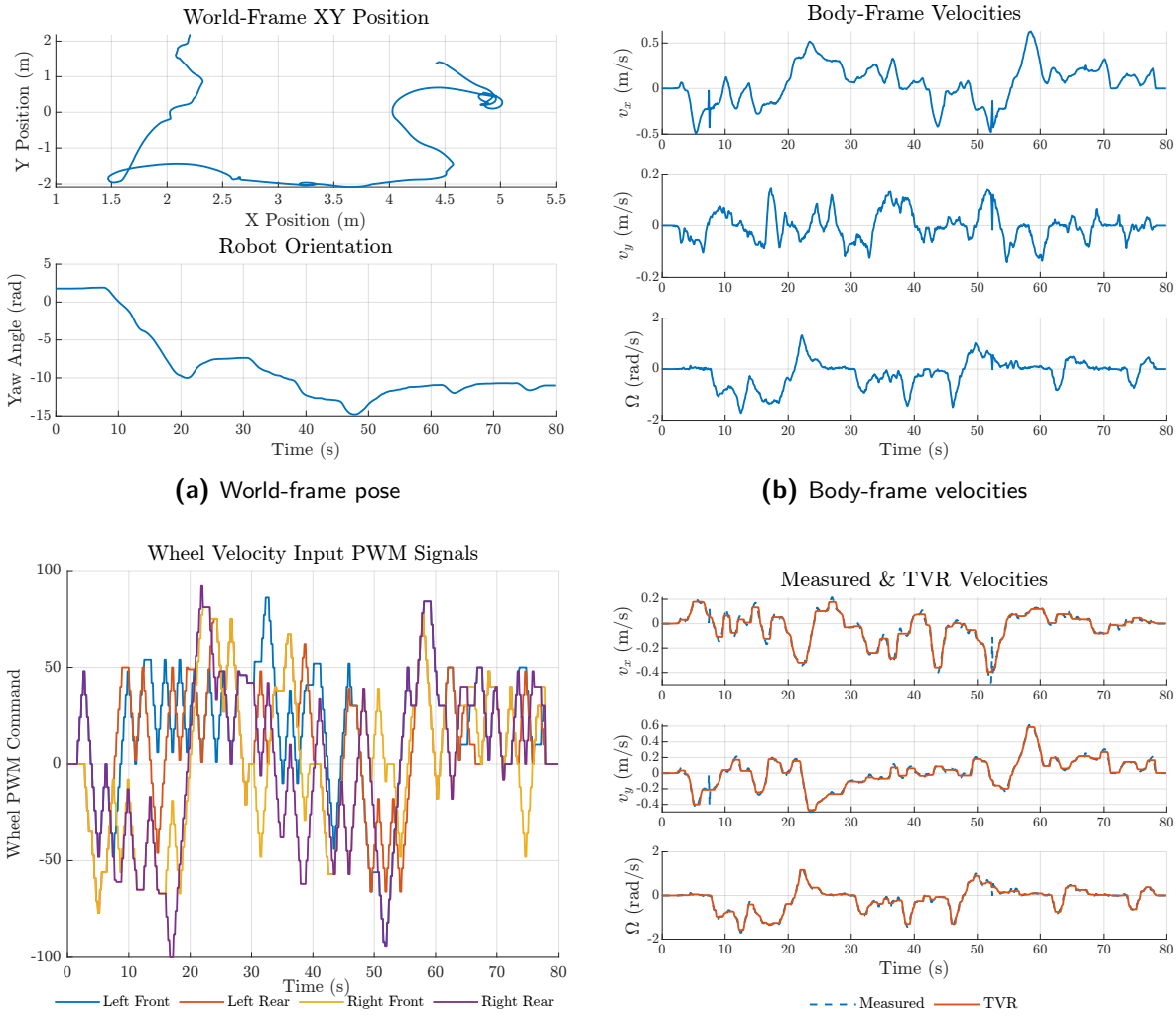
Algorithm 1 is used to generate the training data for identifying the SINDYc models, as shown in Figure 3-2a. In practice, the collection of such data faces several challenges. A sufficiently large dataset of randomised signals is difficult to obtain without risking collisions. Equal excitation of the three motion modes is also difficult to achieve. From the inverse kinematic Jacobian $\mathbf{J}$ in Equation (2-2) it follows that the forward velocity v_x is more likely to be excited, since contributions to v_y and Ω can cancel or diminish for some PWM signals. These effects are visible in Figure 3-2b. The forward velocity exhibits stronger and more persistent excitation than the lateral velocity and the angular velocity, even though the data generation algorithm is designed to excite all velocities equally.

A trade-off is therefore required in the design of the excitation signals. The chosen design ensures that the training data span representative operating conditions while safe operation is maintained. This choice is justified because the identified model is used within a robust trajectory tracking controller that compensates for residual modelling errors through feedback. The resulting training dataset contains 800 samples.

The TVR differentiation method is applied to the world-frame pose to obtain smooth and reliable state derivatives. The measured body-frame velocities are transformed to the world frame using Equation (2-1) and are compared with the corresponding TVR estimates in Figure 3-2d. The TVR estimates preserve the overall velocity trends while attenuating noise and impulsive disturbances. This behaviour limits bias in the regression due to measurement errors.

3-2-1 Model types

Three models are validated on different trajectories using SINDYc, namely a kinematic model, an extended kinematic model and a dynamic model. The kinematic and dynamic models are identified using the libraries shown in Table 3-1 and Table 3-2, respectively. The extended kinematic model uses an expanded candidate library with relaxed constraints, allowing sparse selection of additional terms. An extended dynamic model was also identified, but its behaviour closely matched that of the standard dynamic model and is therefore omitted from this work.



(a) World-frame pose

(b) Body-frame velocities

(c) Wheel PWM signals

(d) Comparison of measured and TVR estimated world-frame velocities

Figure 3-2: Training data used for SINDYc

Kinematic model

This model is identified to highlight the mismatch between ideal kinematic assumptions and the actual system behaviour. It also serves as a baseline for tuning the sparsity weight $\boldsymbol{\lambda}$. The library is fixed to the standard mecanum kinematics derived in Section 2-1. The coefficients are obtained by applying the same STLSQ procedure with the sparsity weight set sufficiently low to prevent any term from being eliminated. This is equivalent to performing an ordinary least-squares fit on the fixed library.

Extended kinematic model

The extended kinematic model removes constraints on the library and uses sparse regression to select coefficients from an augmented set of candidate functions. The extended library

targets slip and coupling effects that are absent from the original model. While such effects may be absorbed implicitly into the coefficients associated with the control inputs, they are not captured in a physically meaningful or reliable way. Since the state dimension remains three, the same variables are used, and nonlinear variants are added, including trigonometric functions of the yaw angle and functions of the control inputs. The library is refined and the sparsity weight is tuned iteratively. Terms that do not improve validation accuracy are removed. Moreover, candidate functions are restricted to be continuously differentiable to preserve linearisation accuracy and computational efficiency for control. The added terms are listed in Table 3-3.

Dynamic model

World-frame accelerations are identified with the dynamic model and are integrated twice in time to recover the pose. The state space is extended to six variables, including world-frame pose and velocities. Since direct acceleration measurements are unavailable, the TVR differentiation algorithm is applied a second time to estimate the world-frame acceleration used in the regression. In this case, tuning is more sensitive since small acceleration errors accumulate in position and can promote overfitting when the acceleration estimates are noisy.

3-2-2 Validation trajectories

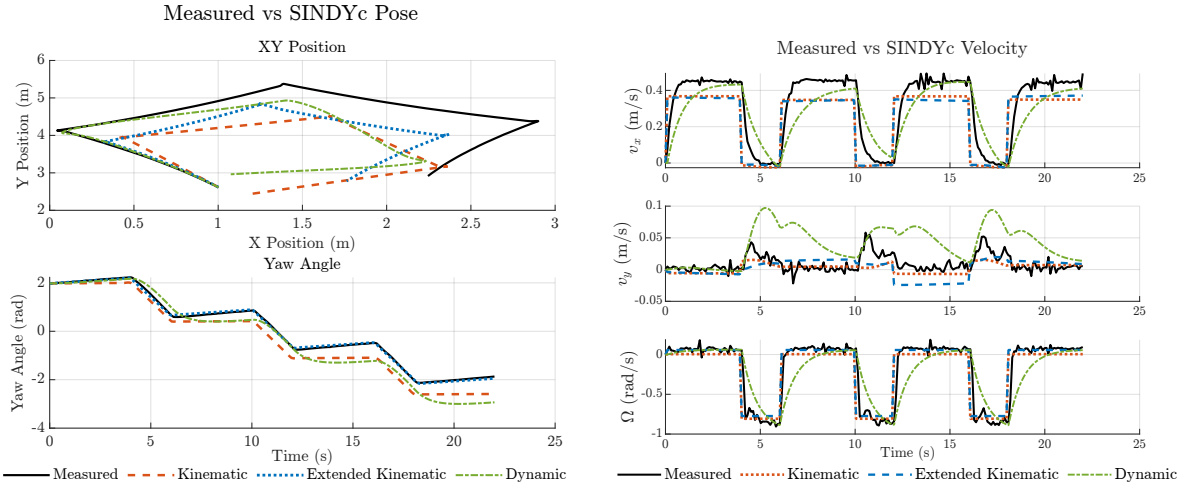
Several trajectories are validated in this project. Two simple trajectories are presented to highlight the behaviour of the MIRTE Master robot without feedback control. One is designed to assess yaw behaviour, and the other is representative of greenhouse operation. Significant deviations from ideal mecanum kinematics are revealed by the measured ground truth for both trajectories. In segments where equal PWM commands are applied to trace axis-aligned rectangles, a perfectly closed path is expected from an ideal platform. Instead, drift, curvature, and non-closure are observed. These deviations are due to slip, frictional asymmetries, and drive or encoder mismatch.

Rectangular path with changing heading

The forward velocity v_x dominates the motion and the angular rate Ω increases at the corners of the rectangular path. The ground truth is shown by the black curves in Figure 3-3a. In contrast to the ideal case, a gradual increase in the measured yaw angle is observed during the straight segments. After about four seconds of forward motion, the yaw angle has increased by approximately 0.2 rad.

State derivative	Candidate terms
$\dot{x}_w$	$\sin(\theta)\mathbf{u}$, $\cos(\theta)\mathbf{u}$, $\mathbf{u}$
$\dot{y}_w$	$\sin(\theta)\mathbf{u}$, $\cos(\theta)\mathbf{u}$, $\mathbf{u}$
$\dot{\theta}$	$\mathbf{u}$, $\{\ u_i u_j \mid 1 \leq i \leq j \leq 4\ \}$

Table 3-3: Candidate library terms for the extended kinematic model



(a) World-frame pose **(b)** Body-frame velocities

Figure 3-3: Measured vs SINDYc Validation Plots for the different models on a rectangular path with changing heading

The kinematic models track the time profile of v_x but remain below the measured magnitude by a nearly constant offset. This bias is explained by losses that change with the wheel PWM signals. Rolling resistance and viscous friction increase with speed, and the motor back-EMF reduces available torque as the PWM increases. The kinematic formulation maps the PWM inputs to velocity with fixed gains, so these effects cannot be represented. The extended kinematic model relaxes the library constraints and selects additional input and yaw terms, yet it still operates at the velocity level without states for acceleration, damping, or drag. Speed dependent slip and drive losses are therefore not captured, and the same bias remains.

The measured v_x levels are matched more closely by the dynamic model, since accelerations are identified and inertial and damping effects are represented. A pronounced first-order lag is observed, arising from the estimation and integration of acceleration to recover velocity and also from the TVR based acceleration estimates.

The lateral velocity in this validation data remains close to zero, so the signal-to-noise ratio (SNR) is low. The estimated velocities of the kinematic models remain near zero, while that of the dynamic model shows larger deviations that follow small measured peaks. This behaviour indicates sensitivity to measurement noise in the acceleration estimate.

On the other hand, the yaw rate is tracked well by both kinematic models. The dynamic model exhibits a smooth but lagging response similar to that observed in v_x. However, since the identified accelerations are integrated twice to obtain the pose, the trajectory drifts and lags more than for the other models.

Overall, the extended kinematic model provides the best balance between pose tracking and velocity fidelity on this trajectory, as shown by the root mean square error (RMSE) in Table 3-4. The heading angle RMSE of the extended model is significantly lower than for the others. Priority is given to heading accuracy because it affects the world-frame velocity mapping and couples into the evolution of x and y. A small discrepancy in the heading angle therefore produces larger errors in the integrated pose.

Comparable velocity RMSE values are obtained across models. In the dynamic model a slow

Table 3-4: RMSE for pose and velocities over the rectangular path with changing heading

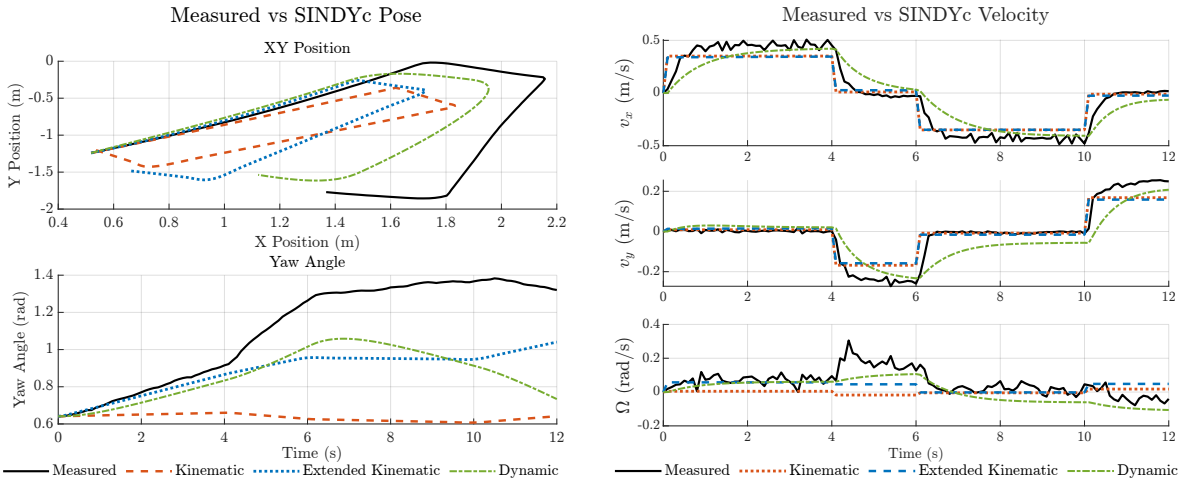
Model	Pose			Velocities		
	x [m]	y [m]	θ [rad]	v_x [m/s]	v_y [m/s]	Ω [rad/s]
Kinematic	0.45	0.78	0.43	0.12	0.01	0.17
Extended kinematic	0.31	0.36	0.05	0.12	0.02	0.17
Dynamic	0.45	0.51	0.51	0.10	0.04	0.22

transient is observed, while the kinematic models respond faster with steady-state offsets. The pose RMSE shows that a fast response is preferred, since position error grows during the transients of the velocity signals.

Rectangular path with constant heading

A constant-heading rectangular path is expected to produce motion in which one body-axis velocity dominates each straight segment, while the angular velocity remains zero throughout the trajectory. During longitudinal segments, $|v_x| > 0$ and $v_y = 0$, while during lateral segments, the converse holds. The ground-truth trajectory, shown as the black curve in Figure 3-4a, is observed to deviate from this ideal. A peak yaw rate of 0.3 rad/s is measured during lateral motion, which indicates slip of the mecanum wheels. The rectangle is not closed and the yaw angle increases by approximately 0.7 rad by the end of the trajectory.

Behaviour similar to that of the previous trajectory is observed in the identified models. The kinematic and extended kinematic models reproduce the time profiles of v_x and v_y, but a nearly constant magnitude offset is exhibited. The dynamic model attains the measured steady-state levels with smooth transients. A noticeable lag is present, introduced by the estimation and integration of acceleration to recover velocity.



(a) World-frame pose

(b) Body-frame velocities

Figure 3-4: Measured vs SINDYc Validation Plots for the different models on a rectangular ground truth with ideal constant heading

A yaw rate close to zero is predicted by the original kinematic model along the entire path, as the slip-induced rotation is missed. This is seen in the top plot of Figure 3-4, where the red dashed curve closes the rectangle, and in the bottom plot, where the yaw angle remains nearly constant. A non-zero yaw rate is captured by the extended kinematic model, and a closer match is obtained. Lag and amplified transients near zero yaw rate are observed in the dynamic model. The heading angle error of this model is accumulated due to double integration.

Similarly to the trajectory with changing heading, the velocity RMSE values are comparable across models, whereas the pose errors differ significantly, as shown in Table 3-5. The largest pose errors are obtained by the original kinematic model. The heading angle RMSE is high because ideal kinematics are assumed, and the resulting yaw error propagates into the other states. The lowest heading angle RMSE is achieved by the extended kinematic model, and the lateral position error is reduced. The steady-state offset in the forward velocity keeps the error in state x elevated. The lowest RMSE for state x is attained by the dynamic model, while its heading-angle RMSE is slightly higher than that of the extended model. The improvement in x arises from a reduced steady-state bias in the forward velocity provided by the additional dynamic terms in the library.

3-3 Discussion

This section discusses the validation results, highlighting limitations and factors that influence modelling accuracy. A PWM deadband of approximately $(-30, 30)$ is observed, particularly at start up and during direction changes. This nonlinearity is not represented in the candidate libraries. Static friction can not be modelled in the kinematic formulations and is neglected in the dynamic model, so behaviour near zero input is poorly represented. As a result, the learned models might absorb the deadband as bias or apparent delay.

The identified kinematic and dynamic models use restricted libraries of nonlinear candidate functions so that their structures match the models derived in Chapter 2. Both reproduce the expected polarities of the coefficients in the derived equations, which supports the SINDYc implementation and the analytical derivations. However, the RMSE values are not low. From the validation body-frame velocity plots in Figures 3-3b and 3-4b, offsets with fast transients are observed for the kinematic and extended kinematic models, whereas a pronounced first-order lag is observed for the dynamic model. In this context, RMSE may fail to distinguish between models whose behaviours differ despite similar error magnitudes.

Identification can be improved by introducing an explicit map from PWM to wheel angular

Table 3-5: RMSE for pose and velocities over the rectangular path with constant heading

Model	**Pose**			**Velocities**		
	x [m]	y [m]	θ [rad]	v_x [m/s]	v_y [m/s]	Ω [rad/s]
Kinematic	0.58	0.31	0.55	0.11	0.05	0.10
Extended kinematic	0.52	0.18	0.28	0.11	0.05	0.07
Dynamic	0.22	0.18	0.30	0.11	0.07	0.06

velocity that captures the deadband and the motor time constant so that the first-order lag of the ground truth is reproduced. Moreover, model accuracy depends strongly on the quality of the training data. A larger and more diverse dataset would further reduce modelling discrepancies [45]. The dataset is limited to about 800 samples due to collision risks with randomised inputs, which is small for system identification. Equal excitation of all motion modes is difficult to achieve, and insufficient excitation of a state derivative is associated with degraded estimation for that state. The available dataset is therefore not ideal, and broader coverage with balanced excitation would likely reduce the RMSE.

The kinematic formulations cannot represent friction forces explicitly, since forces are not included as states in a third-order description. At higher speeds, frictional nonlinearities behave differently than at low speeds, which complicates identification when using static coefficients [59]. This effect was evident in the identification runs, where the estimated coefficients varied when high-acceleration data were included. To mitigate this, a slew-rate limit is applied to restrict rapid changes in velocity and to focus the identification on the nominal operating region, thus capturing the dominant nonlinearities. This choice, however, may introduce an offset when the system operates at other accelerations.

A more exhaustive candidate library is expected to improve identification. In this work, candidate terms are added manually. An automated search over the sparsity thresholds $\lambda \in \mathbb{R}^3$ is performed and the combination with the lowest training error is selected. The resulting model is validated on separate datasets and terms that do not improve validation error are removed for model simplicity. This process is time-consuming and non-comprehensive. Consequently, some higher-order effects may be omitted, which can contribute to residual modelling error.

Priority is given to minimising the heading angle RMSE, since the heading angle is present in the other state derivatives. The kinematic model provides a simple baseline and reproduces the general velocity profiles but does not capture slip-induced coupling, which yields the largest pose errors. The extended kinematic model retains a compact structure, introduces a small set of physically meaningful nonlinearities, and delivers the lowest heading-angle RMSE with reduced lateral error. The lowest forward-position error is achieved by the dynamic model through reduced steady-state bias. However, noticeable lag and amplified transients are observed due to acceleration estimation and integration, which reduces pose accuracy. The state dimension is increased to six in the dynamic formulation, so the computational complexity for control design would increase accordingly.

Therefore, the extended kinematic model offers a practical foundation for subsequent control design. It provides a useful balance between the simplicity of the kinematic model and the accuracy of the dynamic model.

3-4 Summary

The SINDYc framework is presented for the MIRTE Master robot, and each step is detailed from data collection to model identification. Three dynamical systems are identified, namely a kinematic, an extended kinematic, and a dynamic model. The models are validated against independent ground-truth data and evaluated using the RMSE metric to assess their suitability for subsequent control design.

Although the time profiles of the modelled velocities track the measurements, the resulting RMSE values are not small. Several factors are identified as contributors to these discrepancies as discussed in Section 3-3.

For control design, the extended kinematic model provides the best balance between fidelity and simplicity. A robust trajectory control approach is adopted to accommodate the remaining unmodelled dynamics. Overall, the identified coefficients are shown to generalise beyond the training data and to capture the dominant behaviours of the robot.

Chapter 4

Robust Trajectory Control using Tube-based NMPC

Model-based control for OMR trajectory tracking is designed to be robust, as the identified model does not fully represent the true system dynamics. Robot behaviour and stability are influenced by parameter uncertainties and external disturbances [61]. During greenhouse operation, disturbances arise from variations in wheel to ground interaction, floor moisture, loose debris, floor joints, and mild slopes. Slip and actuation bias are introduced by these effects.

Common robust control strategies include SMC [21, 61, 15], H_∞ optimal control [25, 46], and robust MPC [41, 44, 10, 64, 60]. Stability guarantees and bounded tracking errors are typically ensured under defined uncertainty bounds, with potential conservatism as a trade-off.

Tube-based NMPC is chosen for this work for its computational efficiency, explicit constraint handling, and limited conservatism when disturbance bounds are estimated accurately. A two-layer structure is used, in which reference trajectory planning is separated from disturbance rejection. The extended kinematic model identified in the previous chapter is used as the nominal prediction model. A nominal controller uses this model to compute a nominal trajectory compatible with system constraints based on a reference trajectory. A stabilising feedback law is then applied so that the actual system is maintained within a robust invariant tube around the nominal trajectory, as illustrated in Figure 4-1. Constraint satisfaction is ensured by set tightening for all disturbances within the estimated bounds.

The tube-based NMPC framework is presented, and each stage is described in detail applied for the MIRTE Master robot. Simulation results are provided in `MATLAB` before the control method is deployed on the robot.

4-1 Tube-based NMPC framework

The MIRTE Master robot is represented as a nonlinear continuous-time system

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) + \mathbf{R}\big(\theta(t)\big)\, \mathbf{w}(t), \tag{4-1}$$

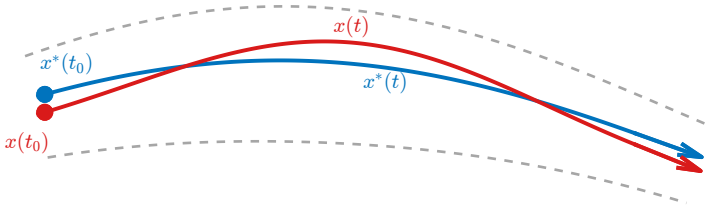


Figure 4-1: Tube-based MPC illustration with nominal (blue) and actual (red) trajectories within an error tube (grey)

where $\mathbf{f}$ is locally Lipschitz, since it consists of a finite sum of polynomials in $\mathbf{u}$ and the smooth functions $\sin\theta$ and $\cos\theta$. The input set is

$$\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^m \mid |u_i| \leq u_{\max,i},\ i = 1, \ldots, m\}. \tag{4-2}$$

Unmodelled dynamics and external effects are captured by a disturbance $\mathbf{w}(t)$ that belongs to a compact and convex set

$$\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^3 \mid |w_i| \leq w_{\max,i},\ i = 1, 2, 3\}. \tag{4-3}$$

The maximum disturbance is estimated from measured body-frame velocity data and is mapped to the world-frame state derivatives through the transformation matrix $\mathbf{R}(\theta)$ defined in (2-1). The disturbance-free nominal model identified by SINDYc is

$$\dot{\bar{\mathbf{x}}}(t) = \mathbf{f}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)). \tag{4-4}$$

The applied input of the tube-based NMPC controller is composed of a nominal component and an ancillary feedback control law,

$$\mathbf{u}(t) = \bar{\mathbf{u}}(t) + \mathbf{k}(\bar{\mathbf{x}}(t), \mathbf{x}(t)). \tag{4-5}$$

The tracking error is defined as

$$\mathbf{e}(t) = \mathbf{x}(t) - \bar{\mathbf{x}}(t). \tag{4-6}$$

Under the ancillary feedback control law, the error dynamics become an autonomous disturbed system [44]. A set $\mathcal{E} \subset \mathbb{R}^n$ is called a robust positive invariant (RPI) set for the error dynamics under Equation (4-5) if

$$\mathbf{e}(t_0) \in \mathcal{E} \ \Rightarrow\ \mathbf{e}(t) \in \mathcal{E} \quad \forall\, t \geq t_0,\ \forall\, \mathbf{w}(t) \in \mathcal{W}. \tag{4-7}$$

Equivalently, the true state remains in the tube

$$\mathbf{x}(t) \in \bar{\mathbf{x}}(t) \oplus \mathcal{E} \text{ for all } t \geq t_0, \tag{4-8}$$

where $\oplus$ is the Minkowski sum with definition $A \oplus B := \{\, a + b \mid a \in A, b \in B \,\}$ [18].

4-1-1 Nominal Controller

The first step in the tube-based formulation is the design of the nominal controller, which uses the extended kinematic model as the nominal model. This controller is implemented as an NMPC that optimises the disturbance-free trajectory to track a reference $\mathbf{x}_{\text{ref}}$. More accurate nominal predictions lead to smaller tubes and less constraint tightening [43]. The nominal trajectory is obtained by solving a finite-horizon optimal control problem (OCP) on the interval $[t_k, t_k + T]$,

$$\begin{aligned}
\min_{\hat{\mathbf{x}}_0,\ \bar{\mathbf{u}}(\cdot)} \quad & \int_{t_k}^{t_k+T} \ell\big(\bar{\mathbf{x}}(t) - \mathbf{x}_{\text{ref}}(t),\, \bar{\mathbf{u}}(t),\, \dot{\bar{\mathbf{u}}}(t)\big)\, \mathrm{d}t + V_f\big(\bar{\mathbf{x}}(t_k+T) - \mathbf{x}_{\text{ref}}(t_k+T)\big) \\
\text{s.t.} \quad & \dot{\bar{\mathbf{x}}}(t) = \mathbf{f}\big(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)\big) \quad \forall t \in [t_k, t_k+T] \\
& \bar{\mathbf{x}}(t_k) = \hat{\mathbf{x}}_0 \quad \mathbf{x}(t_k) \in \hat{\mathbf{x}}_0 \oplus \mathcal{E} \\
& \bar{\mathbf{u}}(t) \in \bar{\mathcal{U}} \quad \forall t \in [t_k, t_k+T] \\
& \|\dot{\bar{\mathbf{u}}}(t)\|_\infty \le \dot{\mathbf{u}}_{\max} \quad \forall t \in [t_k, t_k+T],
\end{aligned} \tag{4-9}$$

$$\begin{aligned}
\text{where} \quad & \ell(\cdot) = \tfrac{1}{2}\|\bar{\mathbf{x}}(t) - \mathbf{x}_{\text{ref}}(t)\|^2_{\mathbf{Q}} + \tfrac{1}{2}\|\bar{\mathbf{u}}(t)\|^2_{\mathbf{R}} + \tfrac{1}{2}\|\dot{\bar{\mathbf{u}}}(t)\|^2_{\mathbf{R_d}} \\
\text{and} \quad & V_f(\cdot) = \tfrac{1}{2}\|\bar{\mathbf{x}}(t_k+T) - \mathbf{x}_{\text{ref}}(t_k+T)\|^2_{\mathbf{P}}.
\end{aligned}$$

In this work, a quadratic stage cost $\ell(\cdot)$ is used to penalise tracking error, actuation, and input rate with positive definite weights Q, R, and R_d. A quadratic terminal cost $V_f(\cdot)$ with positive definite weight P is used to promote convergence. The tightened input set $\bar{\mathcal{U}}$ ensures that the applied input in Equation (4-5) respects $\mathcal{U}$ for all $\mathbf{e}(t) \in \mathcal{E}$ and all $\mathbf{w}(t) \in \mathcal{W}$ [43].

Terminal constraints are widely used in MPC to promote stability. However, schemes without terminal constraints are attractive because they avoid conservative terminal ingredients and are simpler to implement in practice [23]. In many applications, enforcing a terminal constraint increases computational load and may be impractical for real-time control [3]. Consequently, many implemented NMPC designs adopt a terminal-cost formulation, which removes the terminal set and retains a terminal cost to encourage convergence [4]. A comparison of the two approaches in a tube-based NMPC setting reported similar performance in simulation [4]. In hardware experiments, the terminal set formulation was not used because its real-time implementation was too slow.

Therefore, a terminal cost is used in this formulation as a soft mechanism to promote recursive feasibility and convergence. Under suitable assumptions, a sufficiently long prediction horizon combined with an appropriately weighted terminal cost and a locally stabilising ancillary controller can ensure robust recursive feasibility and convergence to an RPI set, even without a terminal constraint [27]. Formal proofs of recursive feasibility are beyond the scope of this work.

The same framework is followed in this work, by tightening inputs using a tube on the error dynamics and by keeping the terminal cost. The full set of theoretical assumptions is not verified on the robot, so the design is presented as a practically robust tube-based NMPC rather than a fully proven stability guarantee. This mirrors a common observation in the literature that robustness properties for terminal-constrained and terminal-free MPC are established in theory, yet the formal bounds often remain conceptual and do not transfer

directly to practical applications [23]. This motivates the use of simple tube-based tightenings for the OMR.

Instead of reusing the previous nominal state as the next initial state, as is common in standard NMPC schemes, the nominal initial state $\hat{\mathbf{x}}_0$ is treated as a decision variable of the OCP. Given the measured state and the RPI tube, the optimiser selects $\hat{\mathbf{x}}_0$ such that the tube is centred at this state while enclosing the measured state [53]. At time t_k this requirement is enforced as

$$\hat{\mathbf{x}}_0 \in \{\, \mathbf{x}(t_k) \,\} \ominus \mathcal{E}, \tag{4-10}$$

where $\ominus$ denotes the Pontryagin difference defined as $A \ominus B := \{\, z \in \mathbb{R}^n \mid z \oplus B \subseteq A \,\}$.

4-1-2 Ancillary time-varying discrete linear–quadratic regulator (DLQR)

The next component of the tube-based NMPC framework is the ancillary feedback law. Its role is to counteract disturbances and model mismatch, thus ensuring that the true state remains within the invariant tube around the nominal trajectory. To achieve this, a time-varying DLQR is used as the ancillary feedback controller to stabilise the error dynamics and keep the true state inside the RPI tube under all admissible disturbances [11]. In the neighbourhood of the nominal trajectory, local linearisation captures the dynamics sufficiently well, so a linear scheme is used. Computational limits on the MIRTE Master platform also favour this choice over a second nonlinear controller. Since the locally optimal DLQR solution varies in time, the tube is treated as a time-varying RPI sequence.

The nominal trajectory $(\bar{\mathbf{x}}^*(t),\, \bar{\mathbf{u}}^*(t))$, computed by the NMPC in the previous subsection, is linearised over time,

$$\mathbf{A}_{\mathrm{ct}}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{(\bar{\mathbf{x}}^\star(t),\, \bar{\mathbf{u}}^\star(t))} \qquad \mathbf{B}_{\mathrm{ct}}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{(\bar{\mathbf{x}}^\star(t),\, \bar{\mathbf{u}}^\star(t))}. \tag{4-11}$$

The linearised model is discretised at each sampling instant t_k with period T_s by a first-order forward-Euler approximation to reduce computational effort,

$$\mathbf{A}_k = \mathbf{I} + T_s\, \mathbf{A}_{\mathrm{ct}}(t_k) \qquad \mathbf{B}_k = T_s\, \mathbf{B}_{\mathrm{ct}}(t_k). \tag{4-12}$$

The DLQR problem is solved to obtain the infinite-horizon solution $\mathbf{P}_k$ with weighting matrices $\mathbf{Q}_{\mathrm{lqr}} \succeq 0$ and $\mathbf{R}_{\mathrm{lqr}} \succ 0$,

$$\mathbf{P}_k = \mathbf{A}_k^\top \mathbf{P}_k \mathbf{A}_k - (\mathbf{A}_k^\top \mathbf{P}_k \mathbf{B}_k)(\mathbf{B}_k^\top \mathbf{P}_k \mathbf{B}_k + \mathbf{R}_{\mathrm{lqr}})^{-1}(\mathbf{B}_k^\top \mathbf{P}_k \mathbf{A}_k) + \mathbf{Q}_{\mathrm{lqr}}. \tag{4-13}$$

The time-varying gain $\mathbf{K}_k$ is obtained from the discrete algebraic Riccati equation (DARE) solution,

$$K_k = (\mathbf{B}_k^\top \mathbf{P}_k \mathbf{B}_k + \mathbf{R}_{\mathrm{lqr}})^{-1} \mathbf{B}_k^\top \mathbf{P}_k \mathbf{A}_k. \tag{4-14}$$

Thus, the total control input applied to the MIRTE Master robot is

$$\mathbf{u}_k = \bar{\mathbf{u}}_k^* \underbrace{-\mathbf{K}_k(\mathbf{x}_k - \bar{\mathbf{x}}_k^*)}_{\text{Ancillary DLQR}}. \tag{4-15}$$

4-1-3 Time-varying RPI tube approximation and constraint tightening

Following the design of the nominal controller and ancillary feedback law, the next step in the tube-based NMPC framework is to approximate a time-varying RPI tube. This reduces conservatism compared to a static tube formulation [28]. At each sampling instant k, a discrete-time error–prediction model is formed by linearising the nominal dynamics along $(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k)$ and closing the loop with the ancillary gain $\mathbf{K}_k$. With the ancillary control law, the closed-loop system is defined as,

$$\mathbf{A}_{\mathrm{cl},k} \;=\; \mathbf{A}_k - \mathbf{B}_k\,\mathbf{K}_k, \tag{4-16}$$

and $\mathbf{K}_k$ ensures that $\mathbf{A}_{\mathrm{cl},k}$ is Schur stable.

Under a bounded disturbance $\mathbf{w}_k \in \mathcal{W}$, the error dynamics satisfy the set recursion [26]

$$\mathcal{E}_{k+1} \;\supseteq\; \mathbf{A}_{\mathrm{cl},k}\,\mathcal{E}_k \;\oplus\; \mathcal{W}. \tag{4-17}$$

In error coordinates, the tube is initialised as $\mathcal{E}_0 = \{\mathbf{0}\}$ because the measured tracking error at the start is zero. For each subsequent step, the tube cross-sections $\mathcal{E}_{k+1}$ are updated according to (4-17). In the state space, the tube is centred at the optimised nominal initial state $\hat{\mathbf{x}}_0$ at each step. The tube size adapts over time and is typically less conservative than a stationary tube [44, 34].

Ellipsoidal tubes approximate (4-17) well [17]. However, the optimiser deployed on the robot does not handle ellipsoidal constraints efficiently, since they appear as coupled quadratic inequalities and increase computational load [57]. A box representation is therefore used as a trade-off between conservatism and computational effort. Box constraints map to simple bounds that the solver processes efficiently. The box tube is defined by the component-wise half-width $\mathbf{d}_k \in \mathbb{R}^n_{\geq 0}$ as

$$\mathcal{E}_k := \{\mathbf{e} \in \mathbb{R}^n \mid |\mathbf{e}| \leq \mathbf{d}_k\}, \tag{4-18}$$

with the robust update

$$\mathbf{d}_{k+1} \;\geq\; |\mathbf{A}_{\mathrm{cl},k}|\,\mathbf{d}_k \;+\; \mathbf{w}_{\max}, \tag{4-19}$$

where $\mathbf{w}_{\max} \in \mathbb{R}^n$ denotes the vector of worst-case absolute disturbances in the world-frame state derivatives, that is, the element-wise bounds of $\mathcal{W}$. These bounds are approximated from experimental data, as described in Subsection 4-1-4.

Constraint tightening ensures the applied input remains in $\mathcal{U}$ for all admissible errors. From (4-15), the nominal input is restricted to

$$\bar{\mathcal{U}}_k := \mathcal{U} \ominus \mathbf{K}_k\mathcal{E}_k = \Big\{\bar{\mathbf{u}} \in \mathbb{R}^m \mid \mathrm{LB}_i + \big(|\mathbf{K}_k|\,\mathbf{d}_k\big)_i \leq \bar{u}_i \leq \mathrm{UB}_i - \big(|\mathbf{K}_k|\,\mathbf{d}_k\big)_i,\; i = 1, \ldots, m\Big\}, \tag{4-20}$$

so that if $\bar{\mathbf{u}}_k \in \bar{\mathcal{U}}_k$ and $|\mathbf{e}_k| \leq \mathbf{d}_k$, then $\mathbf{u}_k \in \mathcal{U}$ holds for all $\mathbf{w}_k \in \mathcal{W}$. State constraints are not imposed in this work, since the states represent the robot pose.

To reduce computation, the tube half-width $\mathbf{d}_k$ and the tightening (4-20) are held fixed across stages within each NMPC solve. This avoids stage-dependent constraints while preserving robustness to a worst-case disturbance over the horizon. A more accurate alternative propagates (4-19) across the horizon with an online estimated $\mathbf{w}_k$ and applies stage-wise half-widths and tightenings, reducing conservatism at additional computational cost. The deployed controller adopts the simpler option as a practical trade-off.

4-1-4 Approximation of a maximum disturbance bound

The maximum disturbance bound must be approximated in order to determine the RPI set introduced in the previous subsection. This bound is inferred from the body-velocity training data used to identify the SINDYc models so that the tube update uses realistic worst–case values which reduces conservatism. A residual-based bound is estimated between the measured velocities and the nominal model velocities. The validation results in Figures 3-3b and 3-4b show small delays, first–order lag, and offset bias between the extended kinematic model and the ground truth. These effects are predictable actuator dynamics that are handled by the NMPC. The disturbance bound is intended to reflect unmodelled effects such as slip and uneven terrain. The velocity measurements are therefore corrected to remove the deterministic actuator dynamics so that the estimated disturbance is not inflated.

To estimate the deterministic effects, the dataset is split into a training set and a validation set. For each velocity state i, the identified velocity $v_i^{\text{ext}}(t)$ is delayed by $\tau_{d,i} \geq 0$

$$v_i^{\text{corr}_\text{d}}(t) = v_i^{\text{ext}}(t - \tau_{d,i}), \tag{4-21}$$

and passed through a first–order lag filter $\xi_i(t)$ with time constant $\tau_{\ell,i} > 0$

$$\tau_{\ell,i}\,\dot{\xi}_i(t) \;=\; -\,\xi_i(t) \;+\; v_i^{\text{corr}_\text{d}}(t). \tag{4-22}$$

A state-wise gain g_i and bias b_i complete the corrected map,

$$v_i^{\text{corr}}(t) \;=\; g_i\,\xi_i(t) \;+\; b_i. \tag{4-23}$$

The parameters $\tau_{d,i}$, $\tau_{\ell,i}$, g_i, and b_i are fitted on the training set by least squares. Delay, dominant first–order dynamics, steady–state gain, and bias are thus separated from the disturbance.

The fitted correction is applied to the whole dataset to obtain $v_i^{\text{corr}}(t)$. Residuals on the validation set are then computed as

$$w_i(t) \;=\; v_i^{\text{meas}}(t) \;-\; v_i^{\text{corr}}(t), \tag{4-24}$$

and are interpreted as disturbance samples. An affine regression is fitted to capture their dependence on the input slew-rate used in the NMPC formulation (4-9). The time derivative of the input is denoted by $\dot{u}(t)$, and its component-wise infinity norm by $\|\dot{u}(t)\|_\infty$. For each state i, non-negative coefficients are estimated on the validation set so that

$$|w_i(t)| \;\approx\; a_i \;+\; b_i \,\|\dot{u}(t)\|_\infty, \qquad a_i \geq 0,\; b_i \geq 0. \tag{4-25}$$

The resulting symmetric continuous-time maximum disturbance bound is given by

$$\hat{w}_{\max,i} = a_i \;+\; b_i \,\dot{u}_{\max}, \qquad i = 1, \ldots n. \tag{4-26}$$

The worst-case disturbance bound is used in discrete-time in (4-19). Therefore, this bound is discretised as $\mathbf{w}_{\max} = \hat{\mathbf{w}}_{\max} T_s$.

4-2 MATLAB simulation results

The tube-based NMPC framework is first prototyped in MATLAB to verify that the identified model tracks different reference trajectories and tolerates additive disturbances within the estimated bounds. The nominal OCP is solved with `fmincon` using the sequential quadratic programming (SQP) algorithm. At each iteration a local quadratic model of the Lagrangian is formed and the constraints are linearised, then a convex quadratic programming (QP) is solved while a Hessian approximation is updated [40].

A single sampling time of 0.1 s is used throughout identification, discretisation, and control. This value is selected during system identification where sensor latency is observed. The choice keeps finite differences with suitable SNR and the model dynamics is captured well at this rate.

The prediction and control horizons are set to $N_p = N_c = 10$. The tightened input bounds are initialised to $\pm[90]^4$. The NMPC and DLQR weights are tuned iteratively. A straight-line trajectory is simulated to compute the DLQR solution and to inspect the Riccati matrix P. During receding-horizon operation, the terminal region varies little. The end of the horizon typically lies near steady conditions with small tracking error and non-extreme inputs. The nominal state is steered toward a similar region along the trajectory, so a constant P remains an appropriate local quadratic approximation from solve to solve. Therefore, the DLQR matrix P is used as the terminal weight in the quadratic terminal cost, as it serves as a local control Lyapunov function and represents the infinite-horizon cost near the terminal region [35].

A sequence of fictitious random disturbances is drawn independent and identically distributed (iid) over time from the axis-aligned box $[-\mathbf{w}_{\max},\, \mathbf{w}_{\max}]$. Each sample is mapped to the world-frame state derivatives and injected while the SINDYc model is propagated to emulate the real system. This setting verifies disturbance handling under the approximated RPI tube and the associated constraint tightening. The tube-based NMPC procedure is summarised in Algorithm 2, where the main steps are stated and the terminal weight is held fixed. The notation used in the algorithm is defined in Table 4-1.

4-2-1 Rectangular path with constant heading

The first trajectory matches one of the validation cases in Subsection 3-2-2. The nominal controller computes the input sequence that tracks a rectangular path with constant heading

Algorithm 2: Tube-based NMPC with DLQR ancillary feedback

Input: N_p, N_t, T_s, $\mathbf{x}_{\text{ref}}$, $\mathbf{w}_{\max}$, $\mathbf{LB}$, $\mathbf{UB}$, $\mathbf{LB}_d$, $\mathbf{UB}_d$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{R}_d$, $\mathbf{P}$, $\mathbf{Q}_{\text{LQR}}$, $\mathbf{R}_{\text{LQR}}$
Output: $\{\mathbf{u}_k\}_{0:N_t-1}$, $\{\bar{\mathbf{x}}_k\}_{0:N_t}$, $\{\mathbf{x}_k\}_{0:N_t}$
$\mathbf{x}_0 \leftarrow \mathbf{x}_{\text{ref},0}$, $\bar{\mathbf{x}}_0 \leftarrow \mathbf{x}_0$, $d_0 \leftarrow \mathbf{0}$, $\bar{\mathcal{U}}_0 : \bar{\mathbf{u}} \in [-90, 90]^4$
for $k = 0$ **to** $N_t - 1$ **do**
  Window $\mathbf{x}_{\text{ref}}[k{:}k{+}N_p{-}1]$
  $(\hat{\mathbf{x}}_0^\star,\ \bar{\mathbf{u}}_k) \leftarrow$ Solve OCP (SQP): $z = [\hat{\mathbf{x}}_0;\ \text{vec}(\bar{\mathbf{U}})]$
    cost: $\mathbf{Q}, \mathbf{R}, \mathbf{R}_d$ + terminal $\mathbf{P}$
    cons: $|\mathbf{x}_k - \hat{\mathbf{x}}_0| \le \mathbf{d}_k$, $\bar{\mathbf{u}} \in \bar{\mathcal{U}}_k$, $\mathbf{LB}_d \le \Delta\bar{\mathbf{u}} \le \mathbf{UB}_d$
  $\bar{\mathbf{x}}_{k+1} \leftarrow \text{Propagate}(\hat{\mathbf{x}}_0^\star, \bar{\mathbf{u}}_k, T_s, \mathbf{0})$
  $(\mathbf{A}_d,\ \mathbf{B}_d) \leftarrow$ linearise then discretise at $(\hat{\mathbf{x}}_0^\star, \bar{\mathbf{u}}_k)$
  $\mathbf{K}_k \leftarrow \text{dlqr}(\mathbf{A}_d, \mathbf{B}_d, \mathbf{Q}_{\text{LQR}}, \mathbf{R}_{\text{LQR}})$
  $\mathbf{u}_k \leftarrow \bar{\mathbf{u}}_k - \mathbf{K}_k(\mathbf{x}_k - \hat{\mathbf{x}}_0^\star)$
  $\mathbf{x}_{k+1} \leftarrow \text{Propagate}(\mathbf{x}_k, \mathbf{u}_k, T_s, \text{rand}_{[-1,1]} \odot \mathbf{w}_{\max})$
  $\mathbf{d}_{k+1} \leftarrow |\mathbf{A}_d - \mathbf{B}_d\mathbf{K}_k|\,\mathbf{d}_k + \mathbf{w}_{\max}$
  $\bar{\mathcal{U}}_{k+1} :\ \bar{\mathbf{u}} \in \left[\mathbf{LB} + |\mathbf{K}_k|\,\mathbf{d}_{k+1},\ \mathbf{UB} - |\mathbf{K}_k|\,\mathbf{d}_{k+1}\right]$
end

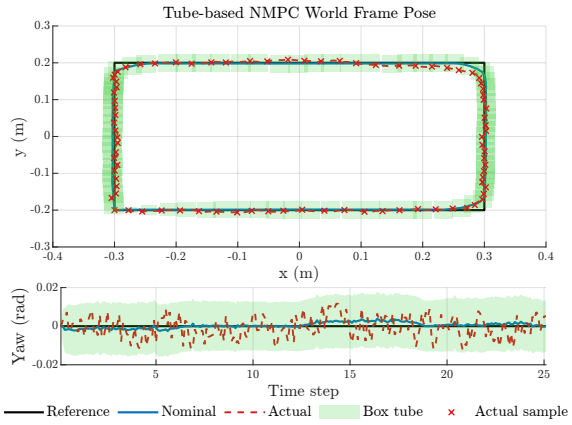
Symbol	Meaning	Value
N_p	Prediction horizon steps	10
N_t	Total simulation steps	Set per experiment
T_s	Sampling time	0.1 s
$\mathbf{x}_{\text{ref}}$	Reference trajectory in $\mathbb{R}^3$	Rectangular or Circular path with constant heading
$\mathbf{x}_k$	Actual state $[x_w, y_w, \theta]^\top$	$\mathbf{x}_0 = \mathbf{x}_{\text{ref},0}$
$\bar{\mathbf{x}}_k$	Nominal state	$\bar{\mathbf{x}}_0 = \mathbf{x}_0$
$\bar{\mathbf{u}}_k$	Nominal input (first element of $\bar{\mathbf{U}}^\star$) $\in \mathbb{R}^4$	-
$\mathbf{u}_k$	Applied input	-
$\mathbf{w}_{\max}$	Maximum discrete disturbance bound	$[0.004,\ 0.003,\ 0.009]^\top$
$\text{rand}_{[-1,1]}$	iid random vector	Uniform on [-1,1] sampled each step
$\odot$	Hadamard (element-wise) product	-
$\mathbf{LB}, \mathbf{UB}$	Input bounds	$[-100,\ 100]^4$
$\mathbf{LB}_d, \mathbf{UB}_d$	Input-rate bounds	$[-30,\ 30]^4$
$\mathbf{Q}$	NMPC stage state weight	$1000\,\mathbf{I}_3$
$\mathbf{R}$	NMPC stage input weight	$0.001\,\mathbf{I}_4$
$\mathbf{R}_d$	NMPC input-rate weight	$0.001\,\mathbf{I}_4$
$\mathbf{P}$	NMPC terminal weight	DLQR solution P
$\mathbf{Q}_{\text{LQR}}$	linear–quadratic regulator (LQR) state weight	$1000\,\mathbf{I}_3$
$\mathbf{R}_{\text{LQR}}$	LQR input weight	$0.001\,\mathbf{I}_4$
$\mathbf{d}_k$	Box tube half-width $\in \mathbb{R}^3$	$\mathbf{d}_0 = \mathbf{0}$
$\bar{\mathcal{U}}_k$	Tightened input set	$[\mathbf{LB} + \lvert\mathbf{K}_k\rvert\mathbf{d}_k,\ \mathbf{UB} - \lvert\mathbf{K}_k\rvert\mathbf{d}_k]$
$\mathbf{A}_d, \mathbf{B}_d$	Linearised and discretised model	at $(\hat{\mathbf{x}}_0^\star, \bar{\mathbf{u}}_k)$
$\mathbf{K}_k$	DLQR gain	$\text{dlqr}(\mathbf{A}_d, \mathbf{B}_d, \mathbf{Q}_{\text{LQR}}, \mathbf{R}_{\text{LQR}})$

Table 4-1: Notation used in Algorithm 2

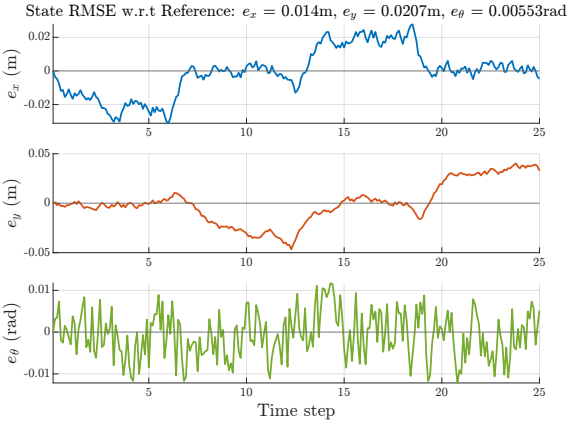
under the nonlinear dynamics, the input-rate limits, and the tightened input constraints. At each step the initial nominal state is optimised at the centre of the RPI tube, which yields a better local prediction while the actual state remains inside the tube, as shown in Figure 4-2a.

A short dwell of 0.3 s is inserted at the end of each straight segment to reduce curvature of the nominal path at the corners. The executed path smooths the nominal 90° change into a curve due to input slew-rate limits. Some curvature is acceptable, since exact corner tracking would require the motion to come almost to a stop and would lengthen the run. The tube grows from zero at the start and then settles to a steady size, consistent with the update in (4-19). This behaviour is consistent with a contractive closed-loop error map and Schur-stable discrete-time linearised dynamics $\mathbf{A}_{\mathrm{cl}}$.

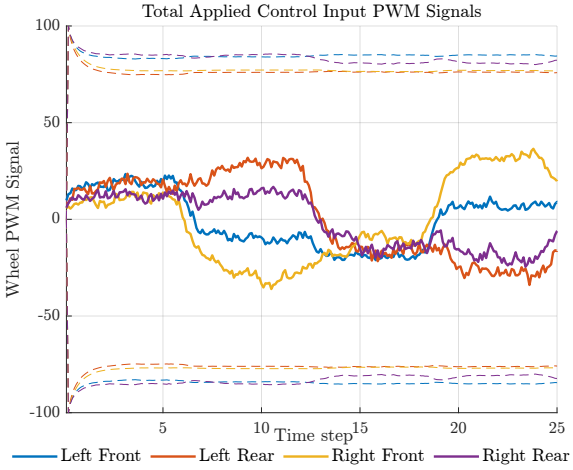
Figure 4-2b presents the tracking errors between the actual and reference states. The x and y positions exhibit a slight lag during motion and a rapid correction when the reference holds, consistent with the input-rate penalty. The yaw error remains bounded within approximately



(a) World-frame pose



(b) RMSE between actual and reference trajectories



(c) PWM control input signals (solid) and tightened sets (dashed)

Figure 4-2: Tube-based NMPC for a rectangular path with constant heading

± 0.02 rad under the injected disturbance. RMSE values are reported in the title and remain modest in magnitude despite transient deviations at the corners, which temporarily increase the tracking errors. Even with these three cornering episodes, the RMSE remains low.

Figure 4-2c shows the applied PWM signals together with the tightened admissible sets for the nominal controller. The tightened set is included to illustrate the effect of the tube-based implementation. It is important to note that the total applied control input can exceed the nominal tightened set, since the ancillary control law is added to the nominal input. The control signals remain within the $[-100, 100]$ constraints and transition smoothly between levels. This confirms that the ancillary feedback and tube-based tightening maintain the applied inputs admissible while the nominal controller delivers the required manoeuvres.

4-2-2 Circular path with constant heading

The second trajectory consists of a circular path with constant heading, as illustrated in Figure 4-3a. The resulting response is observed to behave similarly to the rectangular reference path. The tube is grown from zero at the start and gradually settles into a steady size along the orbit.

Tracking performance is illustrated in Figure 4-3b. The x and y position states exhibit a slight, consistent lag relative to the reference, reflecting the effect of input constraints and slew-rate limits in the NMPC. The yaw error remains within approximately ± 0.01 rad under the injected disturbance. The RMSE values across all states are comparable to those for the rectangular path and remain low.

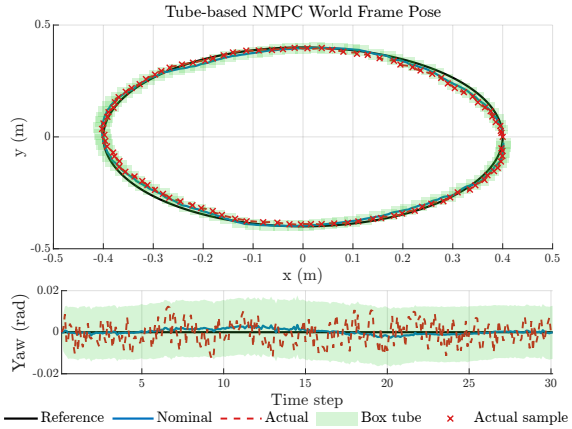
The total applied PWM signals and the tightened input set for the nominal controller are shown in Figure 4-3c. The inputs stay well within the admissible limits and transition smoothly, confirming that the ancillary feedback ensures robustness while the nominal controller enforces constraint satisfaction.

4-3 Summary

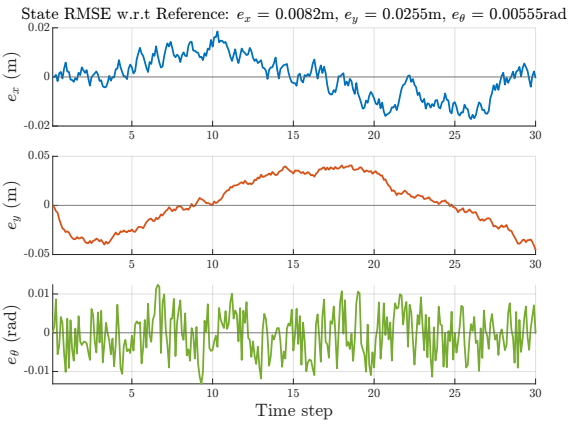
A tube-based NMPC framework is developed for the identified SINDYc model. The receding-horizon NMPC computes a nominal trajectory that tracks a time-varying reference while satisfying input constraints that are tightened to account for uncertainty. The nominal system is linearised and discretised along the trajectory. A time-varying DLQR law is used as ancillary feedback to maintain the real state within a time-varying RPI set under bounded disturbances. This tube-based structure ensures that constraint satisfaction and robust tracking are preserved in closed-loop operation.

The controller is first verified in simulation using `MATLAB`. Bounded additive disturbances are applied to emulate real conditions. Two distinct reference trajectories are simulated and it is demonstrated that the closed-loop system satisfies the input constraints while maintaining small tracking errors, with the real trajectory remaining within the tightened tube throughout. The control input evolves smoothly and remains admissible at all times, confirming the effectiveness of the constraint tightening strategy.

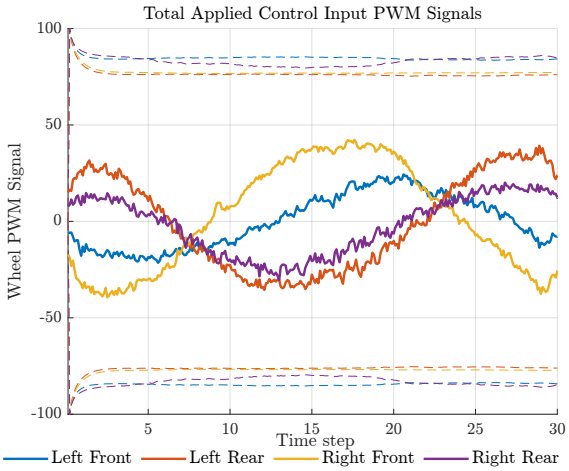
This implementation shows that robust performance can be achieved without a hard terminal constraint, provided that appropriate terminal cost and constraint tightening are used. The



(a) World-frame pose



(b) RMSE between actual and reference trajectories



(c) PWM control input signals (solid) and tightened sets (dashed)

Figure 4-3: Tube-based NMPC for a circular path with constant heading

approach balances theoretical design and practical feasibility, making it suitable for real-time deployment.

In the next chapter, the control algorithm is deployed on the MIRTE Master robot. A real-time feasible solver for the NMPC is required. Therefore, the algorithm is first simulated offline with the candidate solver to reproduce the results obtained in `MATLAB` and to assess suitability in terms of tracking performance and execution time.

Chapter 5

Real-Time Implementation & Experimental Validation

The proposed tube-based NMPC is deployed on the MIRTE Master robot. Real-time implementation requires a fast, reliable solver capable of handling nonlinear systems. `acados`, an open-source framework for embedded optimisation [57], is used. Its solver structure is tailored to real-time hardware with low computational constraints. The framework is modular and flexible, so the nominal NMPC can be combined with the ancillary feedback law. `acados` is widely used in embedded robotic systems that demand robustness and rapid convergence [13].

Conceptually similar to the `fmincon` function in `MATLAB`, `acados` is configured to solve non-convex OCPs via SQP. In this approach the nonlinear OCP is locally approximated at each iteration by a convex QP, which is then solved efficiently.

5-1 Control software implementation

The software architecture used for deployment is shown in Fig. 5-1. The tube-based NMPC is integrated into a ROS 2 package. State feedback is obtained by subscribing to an odometry topic, and PWM wheel signals are published to dedicated command topics.

In `acados`, the OCP is defined in a `Python` generation script that uses `acados_template` together with the `casadi` symbolic library. In this script the OCP defined in (4-9) is declared. Therefore, the continuous-time extended kinematic model is defined symbolically. The stage and terminal costs with their weighting matrices are set. The input and input-rate constraints are declared. The prediction horizon and the discretisation method are selected. Solver options are also configured.

When the script is executed, the generated `C` code and Makefile are written to `c_generated_code/` in the working directory. The code is then compiled to produce the solver artifacts.

At runtime, the generated solver artifacts are used by a `C++` ROS 2 node. `C++` is chosen because it integrates naturally with ROS 2 and avoids `Python` interpreter overhead in the

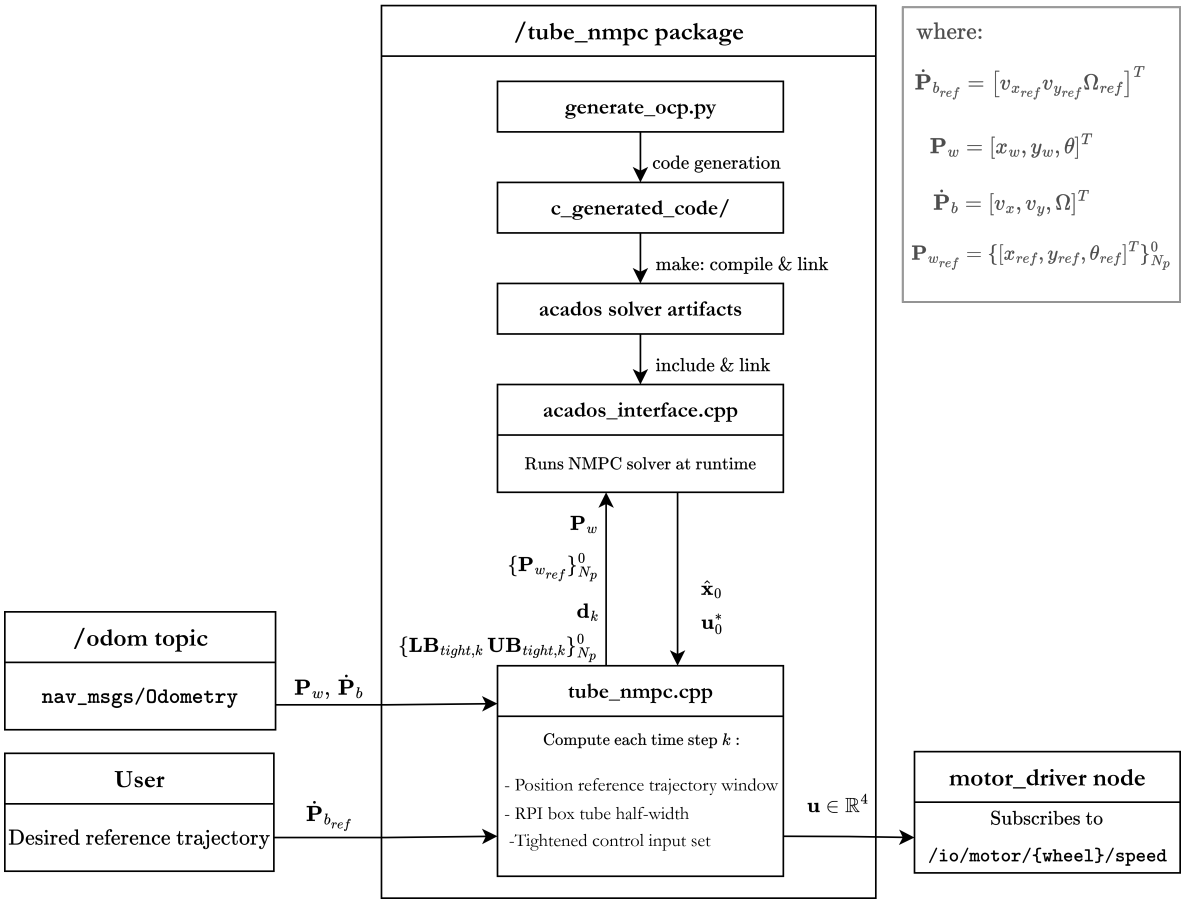


Figure 5-1: ROS 2 tube-based NMPC architecture with `acados` solver

real-time control loop [31]. The generated headers are included and the solver library is linked at build time. At each sampling instant, the problem data are set as shown by the `tube_nmpc.cpp` block in Figure 5-1. The solver is called and returns the initial nominal state and the first optimal input to the tube formulation. The ancillary controller then computes the total applied control input and publishes it as motor commands. The RPI set and the tightened control input set are updated, and the process repeats until the end of the trajectory.

As noted in Section 4-2, the control algorithm runs at 10Hz. This rate provides sufficient margin for the communication stack and for the response of the DC motors when updated wheel speeds are received. Hence, increasing the rate at any stage of the implementation is unnecessary.

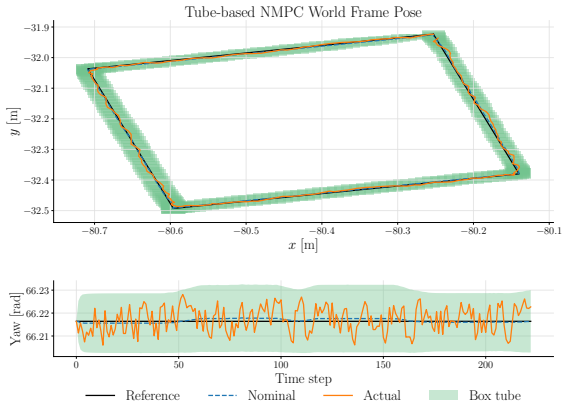
To preserve a similar user interface, body-frame velocity setpoints are used. Internally, a pose reference trajectory is generated from the initial pose and the commanded velocities, and a slew-rate limiter is applied so that the reference varies smoothly. This choice enables trajectory tracking while keeping a similar external command interface.

Before hardware deployment, the full pipeline is validated offline. Robot measurements are simulated consistently with the `MATLAB` setup, in which an additive iid disturbance is injected along the trajectory to test robust performance.

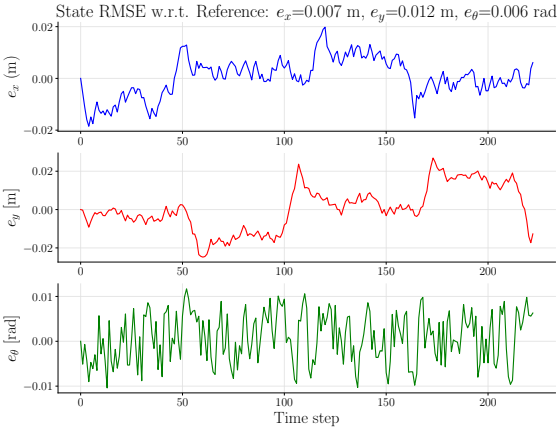
5-2 `acados` Simulation Results

The tube-based NMPC implemented in `acados` is validated on rectangular and circular trajectories at constant heading, mirroring the `MATLAB` tests. The `acados` runs complete in 2.1 s and 2.6 s for the rectangular and circular paths, respectively, which is significantly faster than the `fmincon` runs that take 42.8 s and 46.4 s for the same paths. The maximum number of iterations in `acados` is set to 1000, whereas in `fmincon` it is limited to 50 to keep execution time minimal. The resulting trajectories are shown in Figures 5-2 and 5-3.

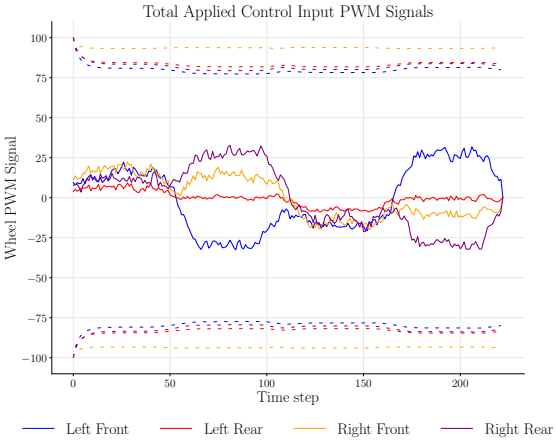
In both cases the nominal trajectory closely tracks the reference and the simulated disturbed state remains within the RPI set. The RMSE values are reported in the titles of the error plots in Figures 5-2b and 5-3b and are of comparable magnitude across the two trajectories. For the rectangular path the `acados` implementation yields position RMSE in x and y that is roughly 1 cm lower than in `MATLAB`, while the yaw RMSE is approximately equivalent. For



(a) World-frame pose

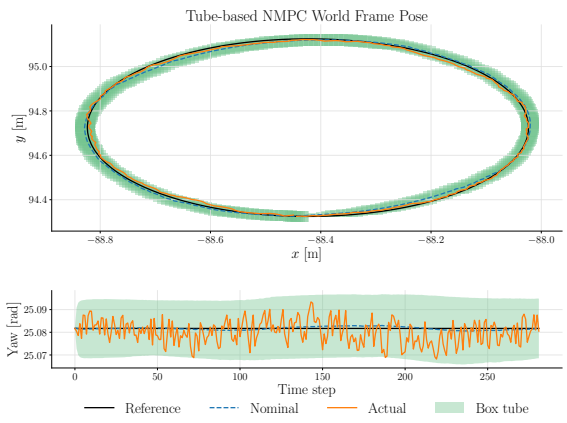


(b) RMSE between actual and reference trajectories

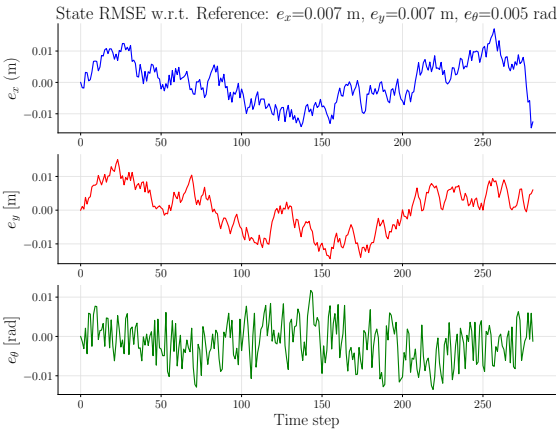


(c) PWM control input signals (solid) and tightened sets (dashed)

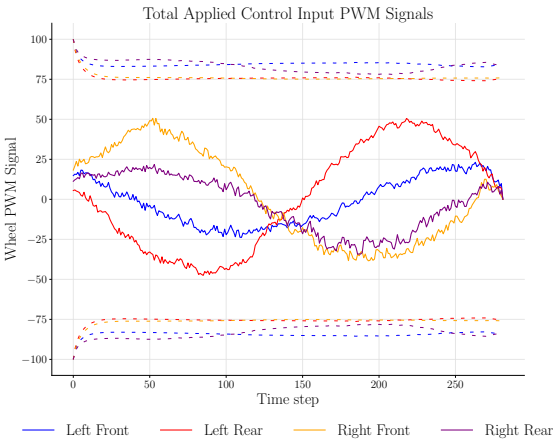
Figure 5-2: Tube-based NMPC for a rectangular path with constant heading



(a) World-frame pose



(b) RMSE between actual and reference trajectories



(c) PWM control input signals (solid) and tightened sets (dashed)

Figure 5-3: Tube-based NMPC for a circular path with constant heading

the circular path the y position shows an improvement of about 2.5 cm with `acados`, while the remaining states are similar between implementations. This improvement may reflect the different solver configurations, including the higher iteration limit, which can allow a better suboptimal solution.

The total applied PWM inputs are illustrated in Figures 5-2c and 5-3c, together with the tightened admissible set for the nominal controller. The inputs have smooth transitions throughout.

The `acados` controller tracks the reference under bounded disturbances comparably to the `MATLAB` version while achieving lower runtimes. Given the nonlinearity of the OMR, the underlying OCP is non-convex and solutions are generally suboptimal [27]. Consequently, the two implementations are not expected to produce identical results, but both consistently demonstrate robust tracking performance.

5-3 Experimental Results

The `acados` implementation is deployed on the MIRTE Master robot. The robot pose is obtained by subscribing to the odometry node, and PWM wheel-speed commands are published to the individual motor nodes, as indicated in Figure 5-1. All experiments were conducted indoors in the Mobile Robotics Lab of the Cognitive Robotics department at TU Delft. The robot operated on an even, matte vinyl floor.

A baseline NMPC controller is used with a definition similar to the nominal controller to compare robust performance. The key differences are that the control set is not tightened but remains bounded at $[-100, 100]$ and the initial optimisation state is set to the current robot pose. The NMPC weighting matrices are tuned for the MIRTE Master robot, but are set the same for both controllers. Thus, this comparison isolates the effect of the ancillary control law. For a clear comparison, the reference, NMPC, and tube-based trajectories are plotted superimposed. The measured data were shifted to share the same initial coordinates, as shown in Figures 5-4 and 5-5.

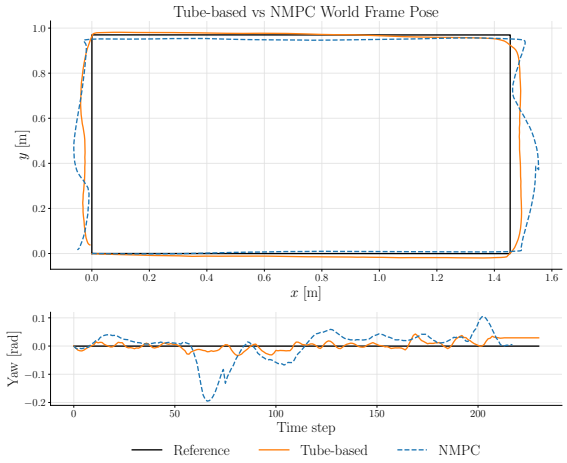
On the rectangular path, tighter tracking is obtained with the tube-based controller, with reduced yaw fluctuations and position error. This is reflected in lower RMSE values. Under the NMPC, larger deviations at the corners with greater yaw drift are observed, reducing overall accuracy. Wheel slippage is corrected more promptly by the tube-based controller, while a slower correction is observed for the NMPC. The benefit arises from how the tube formulation reshapes the nominal problem and from the ancillary feedback that acts at each sample. The tightened sets reduce the feasible search region, so the nominal optimiser selects inputs that anticipate curvature in the trajectory. The ancillary law then corrects slippage and heading error immediately, which prevents the growth of corner transients. This is evident in the plot of PWM input signals in Figure 5-4c, where the general trend is similar but the corrections introduced by the ancillary control law are noticeable in the discrepancies along the straight segments.

On the circular path the curvature and velocity vary smoothly, so both controllers achieve satisfactory performance and the discrepancies are smaller. Slightly lower errors in the longitudinal direction and yaw are produced by the tube-based controller, whereas lower lateral error is obtained with the NMPC. The NMPC already tracks the trajectory well, so there are fewer opportunities for the tube approach to deliver a significant advantage. This is also evident in the plot of PWM control input signals in Figure 5-5c, where both controllers exhibit very similar trends.

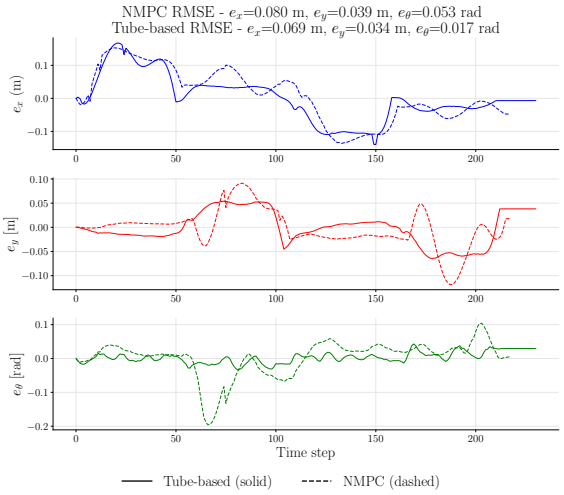
5-4 Summary

This chapter defines the implementation pipeline for real-time deployment on the MIRTE Master robot. `acados` is used as the optimisation solver and is first validated in simulation with two trajectories. The rectangular path is typical of greenhouse operation, whereas the circular path is included to assess performance on a more curved trajectory. The RMSE levels are similar to the `MATLAB` results, with improvements in position for both paths.

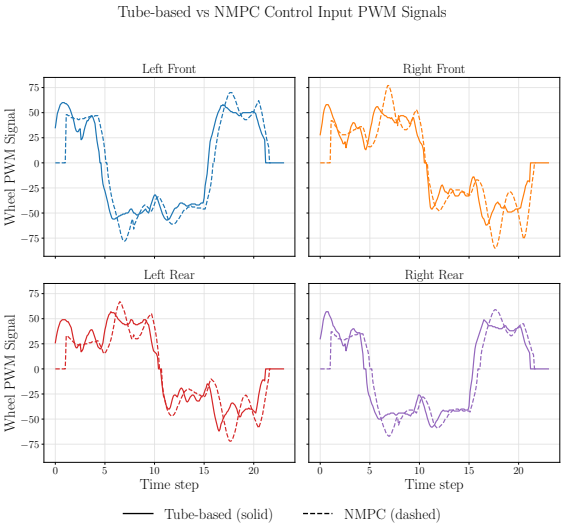
On hardware, a baseline NMPC without tightening is used to compare the robust performance of the tube-based controller. Real-time control is delivered reliably and efficiently



(a) World-frame pose



(b) RMSE between actual and reference trajectories

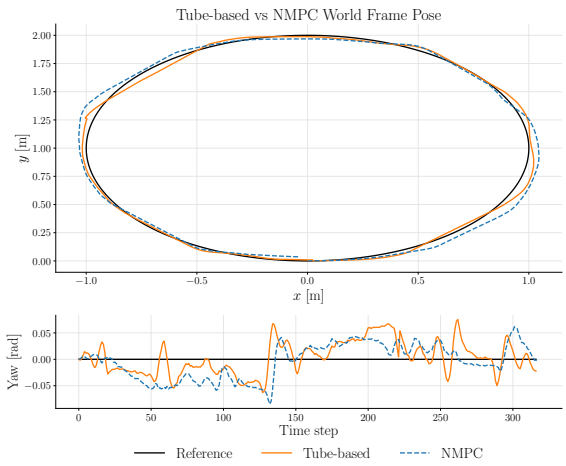


(c) PWM tube-based (solid) and NMPC (dashed) control input signals

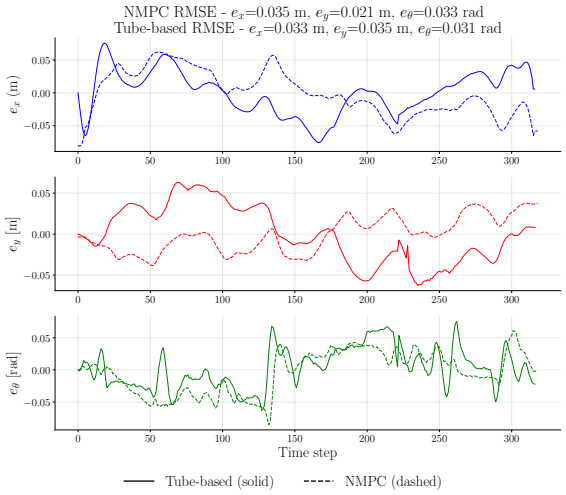
Figure 5-4: Tube-based vs NMPC for a rectangular path with constant heading

with `acados`, while staying within the available computational resources. For the rectangular path, better tracking performance is obtained with the tube-based controller, particularly in lower yaw RMSE. For the circular path, both controllers perform well with minor discrepancies in the reported RMSEs.

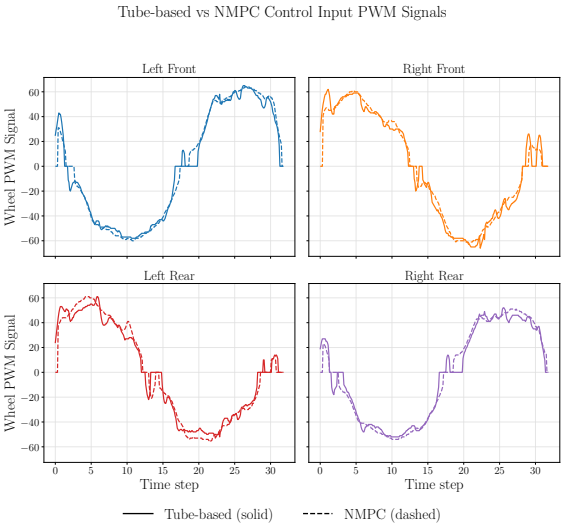
Consequently, cornering is seen to excite modelling errors and to be more prone to slippage. This is corrected more effectively by the tube-based approach, which yields tighter tracking without actuator saturation. This finding is relevant for OMRs in greenhouse applications, since common trajectories include cornering at a constant heading.



(a) World-frame pose



(b) RMSE between actual and reference trajectories



(c) PWM tube-based (solid) and NMPC (dashed) control input signals

Figure 5-5: Tube-based vs NMPC for a circular path with constant heading

Chapter 6

Conclusions & Future Work

This thesis has addressed the modelling and control of a four-wheeled OMR using the MIRTE Master as the experimental platform. Kinematic and dynamic models were derived analytically to highlight their structural differences and to illustrate the respective advantages and limitations of each approach. These models were then identified from data using SINDYc, together with an extended kinematic formulation. While the kinematic and dynamic models were restricted to the structures derived in Chapter 2, the extended kinematic model relaxed these restrictions, enabling additional nonlinearities to be recovered from data while remaining physically interpretable.

The extended kinematic model offered the best balance between complexity and accuracy and was therefore selected as the nominal model for control. A tube-based NMPC framework was developed, with the nominal trajectory generated by NMPC and a time-varying DLQR ancillary law enforcing invariance within an approximated time-varying RPI set. To enable this construction, disturbance bounds representing uncertainties such as wheel slip, surface irregularities, and other unmodelled effects were estimated from residuals of filtered training data. These bounds were used to define tightened constraints and tube half-widths. The resulting controller was validated in simulation, where both `MATLAB` and `acados` implementations produced similar tracking results. Importantly, `acados` achieved significantly faster runtimes, confirming its suitability for real-time control.

The experiments compared the tube-based controller with a nominal NMPC to demonstrate the added value of the ancillary structure. On a rectangular path with constant heading, the tube-based controller reduced cornering error without saturating the actuators, as the ancillary law corrected slip and heading deviation at each sample and limited the growth of transients. On a smoother circular path, where curvature and velocity varied gradually, both controllers performed comparably and the benefit of ancillary feedback was less pronounced. These results confirm that the tube-based formulation enhances robustness particularly under manoeuvres with large heading changes or when slip is present.

Overall, the thesis goals defined in Section 1-2 have been achieved. A sufficiently accurate model was identified under limited data and computational constraints, and it was validated on representative greenhouse trajectories. A robust trajectory-tracking controller was

designed and implemented within these constraints, and the proposed control scheme was verified both in simulation and experiments. In addition, an explicit workflow has been established that makes the approximations, parameter choices, and design trade-offs transparent, whereas these are often implicit in the literature.

The complete source code can be found in the repository.

6-1 Future Work

Although the main goals of the thesis have been achieved, several directions for further research remain open. Model identification could be improved by collecting longer and more diverse datasets that better excite all motion modes, since the current training set is limited in both size and coverage. Ensuring balanced excitation across translation and yaw would likely reduce the modelling discrepancies observed in validation. The actuation dynamics could also be identified separately. Introducing an explicit mapping from PWM to wheel angular velocity, including deadband and motor time constants, would allow the first-order lag observed in the measurements to be reproduced more accurately. Frictional effects are also only partially captured. Their velocity dependence, particularly at higher accelerations, suggests that explicit friction models may further improve accuracy.

Future work should also consider more systematic approaches to constructing candidate libraries. In this thesis, additional nonlinear terms were included manually, and sparsity thresholds were tuned through search and validation. Automated generation of candidate functions, combined with structured selection criteria that balance fit and parsimony, would reduce the time required for identification and uncover higher-order interactions that may remain hidden in the current formulation. An alternative approach is to explore identification in the body frame, which removes trigonometric dependencies and simplifies the model structure, though at the cost of interpretability of the resulting states.

On the control side, the treatment of disturbances can be refined. The current approach relies on deterministic worst-case bounds, which are conservative by design. Extending the tube-based framework with adaptive or probabilistic disturbance sets would reduce conservatism and allow the controller to adapt to changing environments or wear effects over time.

Evaluating the controller under concurrent robot tasks, such as pruning or harvesting, would further test its robustness under realistic operational conditions and computational load. Direct data-driven predictive control methods provide another interesting benchmark for comparison with tube-based NMPC.

Finally, the scalability of the proposed methods should be assessed across multiple MIRTE Master robots and environments. While the robot served as a representative platform, validating the approach on additional units would help quantify robot-to-robot differences and determine the extent of calibration required for each system. Testing on varied surfaces and greenhouse layouts, particularly under conditions of slip and uneven terrain, would further confirm the robustness of the identified models and the proposed controller. Such studies would strengthen the case for deployment in PA, which was the overarching aim of this thesis.

Bibliography

[1] Boris I. Adamov and Gasan R. Saypulaev. A Study of the Dynamics of an Omnidirectional Platform, Taking into Account the Design of Mecanum Wheels and Multicomponent Contact Friction. In *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, pages 1–6, Innopolis, Russia, December 2020. IEEE.

[2] Veer Alakshendra and Shital S. Chiddarwar. A robust adaptive control of mecanum wheel mobile robot: simulation and experimental validation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5606–5611. IEEE.

[3] Sh. Ameri, N. Yousefi Lademakhi, and A. H. Korayem. Artificial steady-state-based nonlinear MPC without terminal ingredients for wheeled mobile robot. In *2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pages 939–944. IEEE.

[4] Xuefeng Bao, Zhiyu Sheng, Brad E. Dicianno, and Nitin Sharma. A tube-based model predictive control method to regulate a knee joint with functional electrical stimulation and electric motor assist. *IEEE Transactions on Control Systems Technology*, 29(5):2180–2191.

[5] Chawannat Chaichumporn, Patinya Ketthong, Nghia Thi Mai, Kotaro Hashikura, Md Abdus Samad Kamal, Iwanori Murakami, and Kou Yamada. The Dynamical Modeling of Four Mecanum Wheel Mobile Robot on Typical Unstructured Terrain. In *2024 21st International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–5, Khon Kaen, Thailand, May 2024. IEEE.

[6] Rick Chartrand. Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011:1–11, May 2011.

[7] A.S. Conceicao, A.P. Moreira, and P.J. Costa. Practical approach of modeling and parameters estimation for omnidirectional mobile robots. *IEEE/ASME Transactions on Mechatronics*, 14(3):377–381.

[8] Khalid K. Dandago, Ameer Mohammed, Jamilu Y. Umar, Abdulbasit Hassan, and Mahmud S. Zango. Trajectory tracking of a ground agricultural robot using sliding mode control. In *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, pages 1–5. IEEE.

[9] Catalin Dosoftei, Vasile Horga, Ioan Doroftei, Tudor Popovici, and Stefan Custura. Simplified Mecanum Wheel Modelling using a Reduced Omni Wheel Model for Dynamic Simulation of an Omnidirectional Mobile Robot. In *2020 International Conference and Exposition on Electrical and Power Engineering (EPE)*, pages 721–726, Iasi, Romania, October 2020. IEEE.

[10] Taher Ebrahim, Sankaranarayanan Subramanian, and Sebastian Engell. Robust tube-based NMPC for dynamic systems with discrete degrees of freedom. *Automatica*, 160:111395.

[11] Willem Esterhuizen, Tim Aschenbruck, and Stefan Streif. On maximal robust positively invariant sets in constrained nonlinear systems. *Automatica*, 119:109044.

[12] Urban Fasel, Eurika Kaiser, J. Nathan Kutz, Bingni W. Brunton, and Steven L. Brunton. SINDy with control: A tutorial. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 16–21. IEEE.

[13] Jonathan Frey, Yunfan Gao, Florian Messerer, Amon Lahr, Melanie Zeilinger, and Moritz Diehl. Efficient zero-order robust optimization for real-time model predictive control with acados. In *2024 European Control Conference (ECC)*, pages 3470–3475. IEEE.

[14] Edgar Granados, Abdeslam Boularias, Kostas Bekris, and Mridul Aanjaneya. Model identification and control of a low-cost mobile robot with omnidirectional wheels using differentiable physics. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1358–1364. IEEE.

[15] Ariana Gutierrez, Manuel Mera, and Hector Rios. An integral sliding–mode robust regulation for constrained three–wheeled omnidirectional mobile robots. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 3637–3642. IEEE.

[16] Ehsan Hashemi, Maani Ghaffari Jadidi, and Omid Bakhshandeh Babarsad. Trajectory planning optimization with dynamic modeling of four wheeled omni-directional mobile robots. In *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)*, pages 272–277. IEEE.

[17] Haimin Hu, Xuhui Feng, Rien Quirynen, Mario Eduardo Villanueva, and Boris Houska. Real-time tube MPC applied to a 10-state quadrotor model. In *2018 Annual American Control Conference (ACC)*, pages 3135–3140. IEEE.

[18] Yonghao Hu, Linfeng Gou, Yingzhi Huang, and Yingxue Chen. Tube-based model predictive control for aero-engine transient-state tracking problems. In *2023 14th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pages 225–229. IEEE.

[19] Chin-Yang Jen and Cheng-Wei Chen. Online Optimal Input Excitation Signal for Sparse Identification of Euler-Lagrangian Systems. In *2025 American Control Conference (ACC)*, pages 1814–1820, Denver, CO, USA, July 2025. IEEE.

[20] Sangyoon Jeong and Dongkyoung Chwa. Sliding-mode-disturbance-observer-based robust tracking control for omnidirectional mobile robots with kinematic and dynamic uncertainties. *IEEE/ASME Transactions on Mechatronics*, 26(2):741–752.

[21] Jong-Min Yang and Jong-Hwan Kim. Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3):578–587.

[22] Eurika Kaiser, J. Nathan Kutz, and Steven L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2219):20180335.

[23] Johannes Kohler, Matthias A. Muller, and Frank Allgower. A novel constraint tightening approach for nonlinear robust model predictive control. In *2018 Annual American Control Conference (ACC)*, pages 728–734. IEEE.

[24] Tobias Kronauer, Joshwa Pohlmann, Maximilian Matthe, Till Smejkal, and Gerhard Fettweis. Latency Analysis of ROS2 Multi-Node Systems. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–7, Karlsruhe, Germany, September 2021. IEEE.

[25] Amir Salimi Lafmejani, Hamed Farivarnejad, and Spring Berman. H -optimal tracking controller for three-wheeled omnidirectional mobile robots with uncertain dynamics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7587–7594. IEEE.

[26] W. Langson, I. Chryssochoos, S.V. Raković, and D.Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133.

[27] D. Limon, T. Alamo, and E.F. Camacho. Stable constrained MPC without terminal constraint. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 6, pages 4893–4898. IEEE.

[28] Brett T. Lopez, Jean-Jacques E. Slotine, and Jonathan P. How. Dynamic tube mpc for nonlinear systems. In *2019 American Control Conference (ACC)*, pages 1655–1662, 2019.

[29] Xingyang Lu, Xiangying Zhang, Guoliang Zhang, and Songmin Jia. Design of Adaptive Sliding Mode Controller for Four-Mecanum Wheel Mobile Robot. In *2018 37th Chinese Control Conference (CCC)*, pages 3983–3987, Wuhan, July 2018. IEEE.

[30] Ren C. Luo and Yueh-Shiuan Tsai. On-line adaptive control for minimizing slippage error while mobile platform and manipulator operate simultaneously for robotics mobile manipulation. In *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pages 002679–002684. IEEE.

[31] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66), May 2022.

[32] Sandro Costa Magalhães, António Paulo Moreira, and Paulo Costa. Omnidirectional robot modeling and simulation. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 251–256. IEEE.

[33] Jon Martin, Ander Ansuategi, Inaki Maurtua, Aitor Gutierrez, David Obregon, Oskar Casquero, and Marga Marcos. A Generic ROS-Based Control Architecture for Pest Inspection and Treatment in Greenhouses Using a Mobile Manipulator. *IEEE Access*, 9:94981–94995, 2021.

[34] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, May 2011.

[35] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[36] MIRTE. mirte-robot/mirte-ros-packages. https://github.com/mirte-robot/mirte-ros-packages.

[37] P. Muir and C. Neuman. Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1772–1778, Raleigh, NC, USA, 1987. Institute of Electrical and Electronics Engineers.

[38] Jonathan J. Plumpton, M. John D. Hayes, Robert G. Langlois, and Bruce V. Burlton. Atlas Motion Platform Mecanum Wheel Jacobian in the velocity and static force domains. *Transactions of the Canadian Society for Mechanical Engineering*, 38(2):251–261.

[39] Easton R. Potokar, Daniel McGann, and Michael Kaess. Robust Preintegrated Wheel Odometry for Off-Road Autonomous Ground Vehicles. *IEEE Robotics and Automation Letters*, 9(12):11649–11656, December 2024.

[40] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson, editor, *Numerical Analysis*, pages 144–157, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

[41] Tianqi Qie, Weida Wang, Chao Yang, Ying Li, Yuhang Zhang, Wenjie Liu, and Changle Xiang. An improved model predictive control-based trajectory planning method for automated driving vehicles under uncertainty environments. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):3999–4015.

[42] Giuseppe Quaglia, Giovanni Colucci, Andrea Botta, and Luigi Tagaliavini. Development of a novel reconfigurable omnidirectional robot for precision agriculture in greenhouses. In Lotfi Romdhane, Abdelfattah Mlika, Saïd Zeghloul, Abdelbadia Chaker, and Med Amine Laribi, editors, *Robotics and Mechatronics*, pages 77–85, Cham, 2024. Springer Nature Switzerland.

[43] Vignesh Raghuraman and Justin P. Koeln. Tube-based robust MPC with adjustable uncertainty sets using zonotopes. In *2021 American Control Conference (ACC)*, pages 462–469. IEEE.

[44] S. V. Rakovic, A. R. Teel, D. Q. Mayne, and A. Astolfi. Simple robust control invariant tubes for some classes of nonlinear discrete time systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6397–6402. IEEE.

[45] Abdullah Rashad, Hashem Ghorab, Mostafa Othman, and Ayman El-Badawy. Data-Driven MPC for Quadrotor UAVs Using SINDy. In *2024 34th International Conference on Computer Theory and Applications (ICCTA)*, pages 165–170, Alexandria, Egypt, December 2024. IEEE.

[46] Gerasimos Rigatos and Pierluigi Siano. An h-infinity feedback control approach to autonomous robot navigation. In *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pages 2689–2694. IEEE.

[47] Oz Robotics. 48mm steel mecanum wheels. https://ozrobotics.com/shop/48mm-steel-mecanum-wheels-set-2-left-and-2-right-14209/.

[48] Anil Kumar Saini, Anshul Kumar Yadav, and Dhiraj. A comprehensive review on technological breakthroughs in precision agriculture: Iot and emerging data analytics. *European Journal of Agronomy*, 163:127440, 2025.

[49] Luís C. Santos, Filipe N. Santos, E. J. Solteiro Pires, António Valente, Pedro Costa, and Sandro Magalhães. Path planning for ground robots in agriculture: a short review. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 61–66. IEEE, 2020.

[50] Seyed Ali Shahzadeh Fazeli, Sohrab Allahyari, Arman Gholibeikian, and S. Ali A. Moosavian. Kinematic Control of a Mecanum Mobile Robot using Time-Varying Model Predictive Control. In *2024 12th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pages 449–454, Tehran, Iran, Islamic Republic of, December 2024. IEEE.

[51] Jae-Bok Song and Kyung-Seok Byun. Design and control of a four-wheeled omnidirectional mobile robot with steerable omnidirectional wheels. *Journal of Robotic Systems*, 21(4):193–208, 2004.

[52] Sanket Soni, Trilok Mistry, and Jayesh Hanath. Experimental analysis of mecanum wheel and omni wheel. *International Journal of Innovative Science, Engineering Technology*, 1(3).

[53] Marco Pavone Sumeet Singh and Jean-Jacques Slotine. Tube-based mpc: a contraction theory approach. In *Decision and Control, 2016. CDC. 55th IEEE Conference on*, 2016.

[54] Mihai Olimpiu Tatar, Catalin Popovici, Dan Mandru, Ioan Ardelean, and Alin Plesa. Design and development of an autonomous omni-directional mobile robot with Mecanum wheels. In *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pages 1–6, Cluj-Napoca, Romania, May 2014. IEEE.

[55] Gregor Thiele, Arne Fey, David Sommer, and Jorg Kruger. System identification of a hysteresis-controlled pump system using SINDy. In *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 457–464, Sinaia, Romania, October 2020. IEEE.

[56] Mike Purvis Tully Foote. REP 103 – Standard Units of Measure and Coordinate Conventions. https://www.ros.org/reps/rep-0103.html.

[57] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 2021.

[58] Mark Wendler and József Kopják. Low power brushed direct current motor commutation noise analysis. In *2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000553–000558. IEEE.

[59] R.L. Williams, B.E. Carter, P. Gallina, and G. Rosati. Dynamic model with slip for wheeled omnidirectional robots. *IEEE Transactions on Robotics and Automation*, 18(3):285–293.

[60] Xitao Wu, Chao Wei, Hanqing Tian, Weida Wang, and Chaoyang Jiang. Fault-tolerant control for path-following of independently actuated autonomous vehicles using tube-based model predictive control. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20282–20297.

[61] Hao Xu, Dengxiu Yu, Qiuyue Wang, Pan Qi, and Gang Lu. Current research status of omnidirectional mobile robots with four mecanum wheels tracking based on sliding mode control. In *2019 IEEE International Conference on Unmanned Systems and Artificial Intelligence (ICUSAI)*, pages 13–18, 2019.

[62] Can Yang, Ying Fan, and Qiushi Zhang. Speed estimation from a low-resolution encoder using adaptative extend kalman observer in low-speed range. In *2021 IEEE 2nd China International Youth Conference on Electrical Engineering (CIYCEE)*, pages 1–7. IEEE.

[63] Linan Zhang and Hayden Schaeffer. On the convergence of the sindy algorithm. *Multiscale Modeling amp; Simulation*, 17(3):948–972, January 2019.

[64] Peiyu Zhang, Daxin Tian, Jianshan Zhou, Xuting Duan, Zhengguo Sheng, Dezong Zhao, and Dongpu Cao. Efficient robust control of mixed platoon for improving fuel economy and ride comfort. *IEEE Transactions on Vehicular Technology*, 73(9):12542–12555.

Glossary

List of Acronyms

DARE	discrete algebraic Riccati equation
DC	direct current
DLQR	discrete linear–quadratic regulator
DoF	degrees of freedom
EKF	extended Kalman filter
EMF	electromotive force
iid	independent and identically distributed
IMU	inertial measurement unit
LQR	linear–quadratic regulator
MPC	model predictive control
NMPC	nonlinear model predictive control
NN	neural network
OCP	optimal control problem
OMR	omnidirectional mobile robot
PA	precision agriculture
PID	proportional, integral, derivative
PWM	pulse width modulation
QP	quadratic programming
RMSE	root mean square error
ROS	Robot Operating System
RPI	robust positive invariant
SINDy	sparse identification of nonlinear dynamics
SINDYc	sparse identification of nonlinear dynamics with control
SMC	sliding mode control

SNR	signal-to-noise ratio
SQP	sequential quadratic programming
STLSQ	sequentially thresholded least squares
TVR	total variation regularised