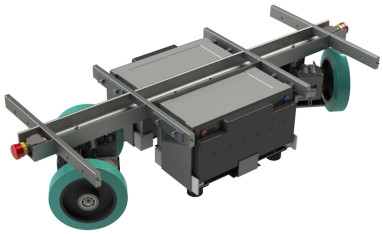
TÉCNICO LISBOA



# Modelling and Control for a Rhombic-Like Vehicle: A case study

**Chenshan Xu**

Thesis to obtain the Master of Science Degree in

## Mechanical Engineering

Supervisors: Dr. Alberto Manuel Martinho Vale
Prof. Alexandra Bento Moutinho

### Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Dr. Alberto Manuel Martinho Vale
Members of the Committee: Prof. Rita Maria Mendes de Almeida Correia da Cunha
Prof. José Raul Carreira Azinheira

**January 2021**

# Acknowledgments

My first word of appreciation goes to my supervisors, Doctor Alberto Vale and Professor Alexandra Moutinho, for their intuitive suggestions during this work, for all the thesis proofreading, for always been there listening even during the most challenging time of Covid-19, and for all their patience and caring.
My appreciation also goes to technical support from Chris Dickens, Application Engineering Manager from Applied Motion Products, Inc., for always being available and for all the technical support.
I would also like to thank the Institute for Plasmas and Nuclear Fusion department (IPFN), for their hospitality and material support.
Last but not least, I want to thank my family and my friends for being there.

# Resumo

O trabalho desenvolvido nesta tese foca-se na conceção, validação e comparação de diferentes soluções de controlo que permitam um robô móvel particular - veículo tipo romba seguir uma dada referência. O veículo tipo romba está equipado com dois módulos dirigíveis e direcionáveis, cada módulo com duas rodas nas laterais. A modelação do veículo é baseado nas equações cinemáticas e dinâmicas. O simulador do veículo é desenvolvido em *Simulink /Matlab*. O modelo cinemático do veículo aderente a restrições não-holonómicas é projetada para o projeto de controlo, é também realizada uma linearização do modelo em torno do ponto de funcionamento, obtendo assim um sistema linear em espaço de estados.

O controlo de movimento é desenvolvido com a seguinte abordagem hierárquica: i) lei de baixo nível para o controlo de velocidade do motor e ii) lei de controlo de alto nível para o controlo de posição. A lei de controlo de baixo nível é baseado na teoria de controlo Proporcional-Integral-Derivativo (PID). A modelação do sistema do atuador específico, e ainda uma implementação do controlo de velocidade é realizada. As metodologias de controlo desenvolvidas para lei de controlo de alto nível são métodos de controlo geométrico e de controlo ótimo. Os resultados da simulação demonstram que com as soluções apresentadas, o veículo tipo romba segue a referência desejada com uma margem de erro aceitável, e as vantagens e desvantagens de diferentes soluções de controlo são estudadas.

A tese permitiu uma contribuição significativa no âmbito do projeto de investigação e desenvolvimento FORMULAfusion do Instituto de Plasmas Fusão Nuclear no Instituto Superior Técnico da Universidade de Lisboa.

**Palavras-chave:** Robô móvel, Veículo tipo romba, restrições não-holonómicas, Controle de movimento, Controlo linear, Controlo geométrico

# Abstract

The work developed in this thesis focuses on the design, validation, and comparison of different control solutions allowing a particular wheeled mobile robot - rhombic like vehicle to follow a reference path. The rhombic like vehicle equips with two drivable and steerable modules, and each module has two wheels on the sides. The vehicle's modeling is based on kinematic and dynamic equations. A simulator is built in *Simulink/Matlab*. Furthermore, the vehicle's kinematic model adherence to a nonholonomic constraint is studied for the control design purpose. The linearizing of the kinematic model around the operating point is performed to obtain the linear state-space system.

The motion control is developed with the hierarchical approach: i) low-level law for the motor speed control and ii) high-level control law for the position control. The low-level control is based on the Proportional–Integral–Derivative (PID) control theory. The modeling of the specific drive system followed by implementing the speed control is performed. The high-level control methodologies developed for position control are geometric control and linear optimal control methods. The desired performance criteria are defined to allow proposed solutions to be compared. Simulation results demonstrate the presented solutions enable the vehicle to follow the desired reference with an acceptable error margin, and the advantages and disadvantages of different control solutions are studied.

The work presented in this thesis played an essential contribution to the project FORMULAfusion of the Instituto de Plasmas Fusão Nuclear in Instituto Superior Técnico, University of Lisbon.



**Keywords:** Wheeled mobile robot, Rhombic like vehicle, Nonholonomic constraints, Motion control, Linear control, Geometric control

# Contents

# List of Tables

# List of Figures

# Nomenclature

The next list describes several symbols that will be later used within the body of the document. The time-dependent variable is omitted from the time law due to simplicity

**Greek Symbols**

| | | |
|---|---|---|
| $\beta$ | slip angle of the vehicle | rad |
| $\dot{\psi}$ | angular rate of the vehicle in $z$ direction | rad/s |
| $\omega$ | angular velocity of the module | rad/s |
| $\omega_m$ | rotor angular velocity of the motor | rad/s |
| $\omega_v$ | angular velocity of the vehicle | rad/s |
| $\omega_{ij}$ | angular velocity of the $i$-$th$ module $j$-$th$ wheel | rad/s |
| $\tau_m$ | torque generated by motor | N · m |
| $\tau_{ij}$ | torque generated $ij$-$th$ wheel | N · m |
| $\theta$ | vehicle/rigid body orientation | rad |
| $\theta_f$ | steering angle of the front module | rad |
| $\theta_i$ | steering angle of the $i$-$th$ module | rad |
| $\theta_r$ | steering angle of the rear module | rad |

**Roman Symbols**

| | | |
|---|---|---|
| $c$ | motor friction coefficient | N · m/s |
| $D$ | distance between the left wheel and right wheel of the module | m |
| $e_c$ | cross tracking error | |
| $F_p$ | resulting force in the center of the vehicle | N |
| $F_{d_{ij}}$ | driving force acting on the $i$-$th$ module $j$-$th$ wheel | N |
| $F_{d_{jy}}$ | lateral force acting on the $j$-$th$ wheel | N |

| | | |
|---|---|---|
| $F_{di}$ | driving force acting on the $i$-$th$ wheel | N |
| $F_{fi}$ | force acting on the front module | N |
| $F_{jx}$ | longitudinal force acting on the $j$-$th$ wheel | N |
| $F_{p_{fj}}$ | force of the $j$-$th$ wheel of front module | N |
| $F_{p_{rj}}$ | force of the $j$-$th$ wheel of rear module | N |
| $F_{ri}$ | force acting on the rear module | N |
| $I$ | motor current | A |
| $I_z$ | inertia moment around the $CG$ of the rigid body | $\mathsf{kg} \cdot \mathsf{m}^2$ |
| $I_{ij}$ | current of a $ij$-$th$ motor | A |
| $J$ | inertial of the wheel | $\mathsf{kg} \cdot \mathsf{m}^2$ |
| $K_b$ | motor voltage constant | $\mathsf{V(rad/s)}$ |
| $K_t$ | motor torque constant | $\mathsf{N} \cdot \mathsf{m/A}$ |
| $L$ | vehicle wheel base | m |
| $l_d$ | look ahead distance | m |
| $L_e$ | inductance | H |
| $L_f$ | distance between the front module and the vehicle CG | m |
| $L_r$ | distance between the rear module and the vehicle CG | m |
| $M$ | mass of the vehicle | kg |
| $M_p$ | resulting moment in the center of the vehicle | N |
| $n$ | gear ratio | |
| $R$ | wheel radius | m |
| $R_e$ | resistance | $\Omega$ |
| $v$ | vehicle linear velocity | $\mathsf{m/s}$ |
| $v_f$ | linear velocity of the front module | $\mathsf{m/s}$ |
| $V_{ij}$ | voltage of $ij$-$th$ motor | V |
| $v_{ix}$ | $i$-$th$ module velocity in $x$ direction | $\mathsf{m/s}$ |
| $v_i$ | linear velocity of the $i$-$th$ module | $\mathsf{m/s}$ |
| $v_r$ | linear velocity of the rear module | $\mathsf{m/s}$ |

x state vector

**Superscripts**

$B$ body frame

$c$ control variables

$I$ Inertial frame

# Acronyms

**2D** Two Dimensional

**AGVs** Automated Guided Vehicles

**AMRs** Autonomous Mobile Robots

**AKM** Alonzo Kelly Modified Controller

**ARE** Algebraic Riccati Equation

**CG** Center of Gravity

**DP** Dynamic Programming

**HCB** Hot Cell Building

**ICR** Instantaneous Center of Rotation

**ITER** International thermonuclear experimental reactor

**LQR** Linear Quadratic Regulator

**PID** Proportional–Integral–Derivative

**RH** Remote Handling

**RMS** Root Mean Square

**RLVs** Rhombic Like Vehicle System

**RWS** Rolling Without Slip

**WMR** wheeled mobile robot

**TB** Tokamak Building

# Chapter 1

# Introduction

International thermonuclear experimental reactor (ITER) project is one of the most aspiring energy projects today; it aims to explore nuclear fusion as a viable energy resource for the future. In order to achieve the ITER's main objective, the Remote Handling (RH) technique was needed due to the restrictive presence of humans in inactivated areas during the maintenance operations. A particular configuration of a wheeled mobile robot (WMR), the rhombic like vehicle system (RLVs), is expected to operate in the ITER to transport the heavy load between two buildings: the Tokamak Building (TB) and the Hot Cell Building (HCB). RLVs is a type of vehicle equips with two drivable and steerable modules, the front and rear modules, each module with two wheels on the sides, thus enables higher maneuverability. From now on, the RLVs is used to refer to the rhombic like vehicle system for simplicity.
The dissertation's theme is integrated into the FORMULAfusion project. It aims to demonstrate a proof-of-concept with a scale robot of RLVs with $250\,\mathrm{kg}$ of payload, transport necessary equipment, and move along a highly confined space in complex paths.
This chapter introduces the mobile robot history, the significant challenges in the autonomous navigation problem, and the dissertation's main objectives and contributions.

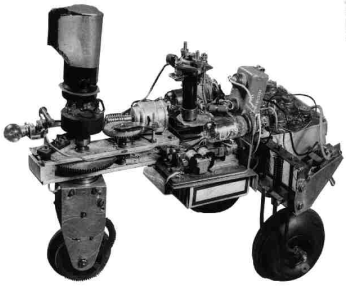
## 1.1 Historical overview of mobile robots

The first unique WMR was built in the 1950s by Grey Walter (the UK, University of Bristol), a three-wheeled, turtle-like mobile robotic vehicles. These vehicles are equipped with a light sensor, touch sensor, propulsion motor, steering motor, and a two vacuum tube analog computer. He named the Adam and Eve of his robots Elmer and Elsie (Electro-Mechanical Robots, Light Sensitive). Figure 1.1(a)) shows the Elsie without her shell. Even with the simple design, his robot was unique because they do not have fixed behavior by the different responses to their environment.
Also, in the 1950s, the first AGV was invented by Barrett Electronics, USA. It was only a tow truck that followed a wire on the floor instead of a rail (see Figure 1.1(b)). In 1969s, Shakey (Stanford University) invented the first mobile robot controlled by vision(see Figure 1.1(c)). It was recognized as "the first general-purpose mobile robot to be able to reason about its actions." because it recognizes the environ-

ment by the vision and creates plans to recover the error from the plan execution.

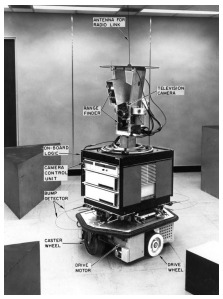
Later in 1979s, the first computer controlled autonomous vehicle was built by Hans Moravec - Stanford Cart(see Figure. 1.1(d)), it successfully traverses a chair-filled room and circumnavigates the Stanford AI Lab without any human intervention.

The research on solving the problem of making robots intelligent became popular in the 1980s and 90s. The Hilare robot (see Figure 1.1(e)) was designed in 1977 at the LAAS (Laboratory of Analysis and Architecture of Systems) in Toulouse. It is recognized as the first French mobile robot capable of autonomously negotiating an unknown environment. It has an ultrasonic sensor to detect the nearby object and a laser sensor to perceive the environment, and an odometer to get the distance measurement and make appropriate decisions with all the in formations. The successive versions of this research project were performed during the 1977s to 1992s; it led to significant progress in modeling the environment and the autonomy, reactivity, and programming of mobile robots. Later on legged robot has been studied; roboticist RodneyBrooks created Genghis, a six-legged insect-like robot at MIT in 1989s. He intended to create robots that displayed intelligence and react to the world around them. The Genghis design was inspired by the insect, which has limited brain function but still has tremendous functionality. In order to do so, Brooks' main idea is to" removed all cognition processors from Genghis and left only the sensors and the code/hardware to allow it to walk." It does not have any pre-defined path; the robot took actions when the sensor detects the obstacles. In Portugal, the first Automatically Guided Vehicles system



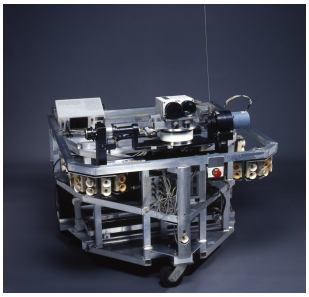
(a) Elsie without her shell in 1950s

(b) First AGV in 1950s



(c) First mobile robot to be controlled by vision in 1969s

(d) First computer controlled autonomous vehicle in 1979s

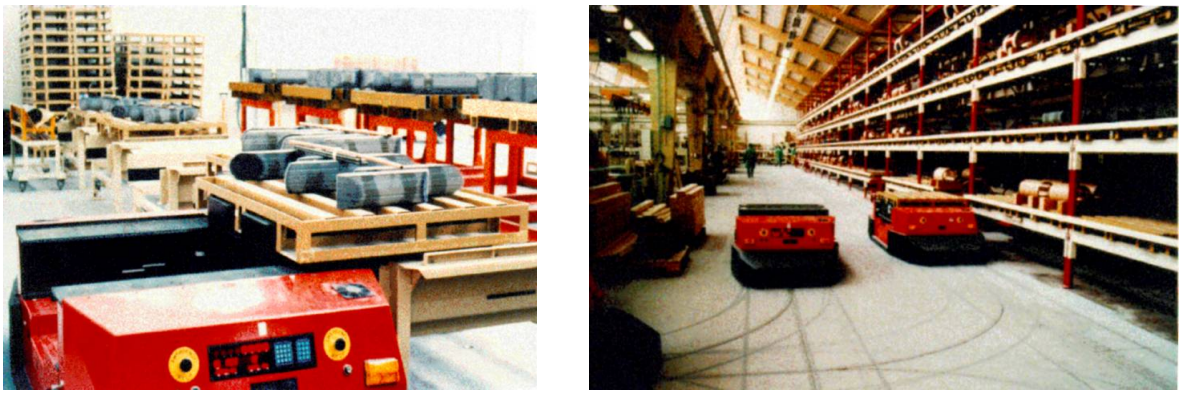


(e) The Hilare I in France in 1977s

(f) Genghis in 1989s

Figure 1.1: The history of mobile robot [1]

for the automatic transport of materials in industrial units was built in 1991 for the joint project with a EFACEC by Maria Isabel Ribeiro, IST Professor and currently part of the Intelligent Robotic Systems group (ISR) in IST. A particular mobile robot is now developed by IST on the field of remote handling



(a) Material loading (b) Operation in industrial environment

Figure 1.2: Instituto de Sistemas e Robótica (ISR) - EFACEC project: first AGVs in industrial units in Portugal

operations in the ITER related with a transfer casks system. Gonzalez Gutierrez et al. [2] give a brief induction about the transfer cask system in ITER, and [3] presents an overview of the work developed in IST. A ongoing FORMULAfusion project aims create a scale model of the vehicle to be used at ITER, is developed in IST, an automated vehicle with two steerable modules for cargo transportation using 3D CAD software is already developed.

## 1.2 Motivation

The interest in using WMR for several different applications is increasing worldwide nowadays [4], such as transportation in factories, warehouses, and places where there is a need for material delivery. In warehouse robotics, two types of mobile robot are widely used for automating internal transportation tasks: Automated Guided Vehicles (AGVs) [1] and Autonomous Mobile Robots (AMRs) [2]. The most significant differences between the AGVs and AMRs, are AMRs equips with sophisticated sensors that enable them to understand and interpret their environment, which helps them to perform their task most efficiently. Thus on most occasions, an AMRs is preferable to AGVs because it has higher maneuverability and it is more flexible.

The WMR is defined as a robot capable of moving on a surface through the actuation of wheel assemblies mounted on the robot and in contact with the surface. A wheel assembly is a device which provides or allows relative motion between its mount and a surface on which it is intended to have a single point of rolling contact. Each wheel (conventional, omnidirectional or ball wheel) and all links between the robot body and the wheel constitute a wheel assembly. Wheeled mobile robot can be classified according to the drive system:

- Differential drive

[1] AGVs is a robot with minimal on-board intelligence and can only obey simple programming instructions, it is restricted to follow the fixed routes, which require additional cost and disruption of the facilities infrastructure.

[2] AMRs is any robot that can understand and move through its environment without being overseen directly by an operator or on a fixed predetermined path.

- Car type
- Omnidirectional
- Synchro drive

The type of WMR as differential drive mobile robot is subject to the nonholonomic constraints. On the other hand, the omnidirectional robots is subject to the holonomic constraints. General speaking, a holonomic constraint is a constraint on space configuration and a nonholonomic constraint is a constraint on velocity. Each type of mobile robot system has its own advantages and disadvantages. The selection of the system for a certain task should take into account simultaneously maneuverability, controllability, and stability. A particular rhombic like vehicle RLVs with nonholonomic constraint is developed that allows transportation of equipment inside of the ITER.

ITER project is officially announced in 1997 as part of a seven-party consortium (six countries and the EU). It aims to produce a fusion plasma ten times more fusion power than the power put into the plasma and is currently under construction in Cadarache, France. The two main buildings of ITER are: the TB were lodging the tokamak reactor and the HCB as a technical support area (see Figure 1.3). The ex-

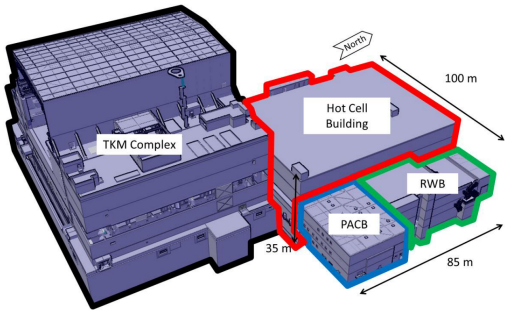



Figure 1.3: Overview of the two main buildings of ITER: the tokamak reactor building (left) and the hot cell complex (tight) [5]

treme work conditions of the reactor did not authorize the human's presence in activated areas. Remote handling (RH) operations are required to perform maintenance operations. Therefore, RH equipment will play an important role in the success of ITER. The main operation of RH equipment in ITER includes manipulating and exchanging components with weight up to 100 tons. The mission mentioned above required the WMR moves autonomously along an optimized path or trajectory. Due to a complexity of the ITER work conditions (see Figure 1.4), a particular configuration of a WMR, the RLVs, is expected to perform the missions. This type of vehicle equips with two drivable and steerable modules, the front and rear modules, each module with two wheels on the sides. This configuration of mobile robot is preferred than others, because it enables higher maneuverability, which is a crucial attribute to consider ITER's cluttered environment. The advantage of the using the RLVs becomes more evident with a case study mission in Figure 1.5. The possibles maneuvers of the RLVs system are: turned with the path as showed in Figure 1.5(a) or sliding to the desired position with all wheels oriented in the desired orientation (see Figure 1.5(b) and 1.5(c) ). The variety of the RLVs maneuver is possible because of high degree of freedom on the modules steering configuration. Depending on the work condition and space

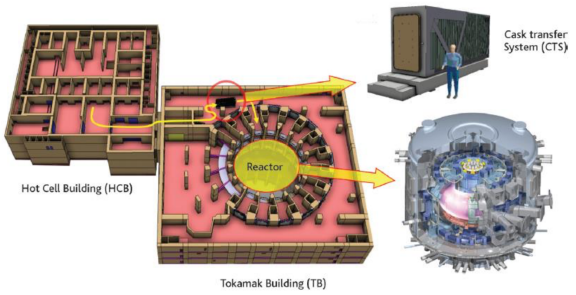



Figure 1.4: Simplified versions of two levels in the main buildings of ITER: storage and maintenance areas in the hot cell building (left) and divertor level of the tokamak reactor building (right) [5]

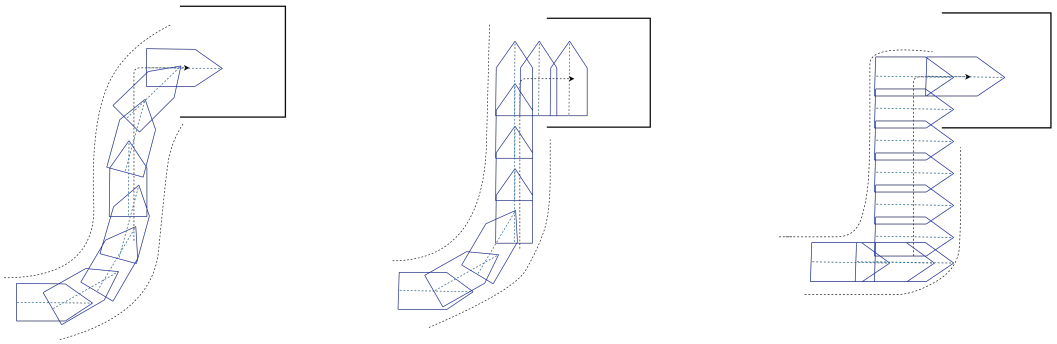



Figure 1.5: Maneuverability advantage of the RLVs: three scenarios

constraints, some of the driving configurations are preferable to others, and it should be take in account in the motion planning stage. This dissertation only concerns if the designed controller can ensure all the possible motion of the RLVs system.

There are three critical challenges in the mobile robot autonomous navigation: localization, motion planning, and motion control. This thesis studied the mobile robot motion control problem, which the system receives as the input the motion plan in the previous stage - motion planning, and develop the feedback control law that allows the vehicle to track the desired reference correctly.

## 1.3 Objectives and Contributions

The dissertation focus on the motion control problem of the scale prototype of a rhombic like vehicle, developing the feedback control law that allows the RLVs to track the desired reference. In order to reach the goal, the dissertation has two major objectives:

- Vehicle Modeling Analysis. This first part of the dissertation is the mathematical modeling of the RLVs scale prototype. In order to reach this objective, different steps have been reached:
  - A mathematical modelling of the vehicle based on the kinematic and dynamic equations is performed, which the vehicle is divided into three systems: the drive system, the modules

system, and the vehicle system

  - The vehicle's kinematic model adherence to a nonholonomic constraint is studied for the control design purpose
  - A vehicle simulator is developed in *Simulink/Matlab* for the validation of the control system

- Development of control solution for the vehicle motion control problem. This part aims to develop motion controller that is able to allow the vehicle to follow the desired reference. The proposed solutions are the following:

  - A speed controller for the DC motor based on the PID control theory
  - Geometric controller
  - Optimal controller

  Analysis of case study missions is performed to evaluated the proposed controllers performance.

## 1.4 Thesis Outline

Chapter 2 presents a general view of the mobile robot's autonomous navigation problem, different system constraints, and the existing solution. Chapter 3 introduces an analysis of rhombic like vehicle motion behavior, developing the simulator for the validation of the control system. In this chapter, a mathematical modelling of the vehicle is based on the kinematic and dynamic equations. The vehicle is divided into three systems: the drive subsystem, the modules system, and the vehicle system. the motor system modeling, followed by an analysis of the modules system, is provided. The dynamic and kinematic analysis of the RLVs and an evaluation of the building simulator is presented. The chapter closed with an introduction of a control-oriented kinematic model and general consideration of vehicle motion capabilities. Chapter 4 explained a control theory, including PID control, geometric motion control, and optimal control methodology. With the model presented in Chapter 3 and theoretical background in Chapter 4, the different control strategy is presented in Chapter 5 to track the predefined reference trajectory. The simulation of the designed controller with several challenging paths shows the control design can follow the desired reference with a reasonable error. The comparison between the proposed controller is presented in Chapter 6, and the assessment is made considering parameters such as tracking errors and actuators' request for a case study. In Chapter 7, the conclusions of this work summarize the knowledge gained and point the directions of our forthcoming investigation.

# Chapter 2

# State of the Art

Autonomous navigation is an essential issue in robotics research. Three general problems are involved in autonomous navigation: localization, path planning and motion control. Since in this work only consider the nonholonimic mobile robots, the nonholonomic system is introduced in section 2.1. The system constraints are studied to identify the RLVs system and design a controller for it. Section 2.2 described the importance of the localization and path planning problem in the autonomous navigation problem. The existing solution of mobile robot motion control problem is presented in Section 2.3.

## 2.1 Nonholonomic Mechanical System

The concept of nonholonomic and holonomic systems is introduced to understand the mobile robot system constraints. The system is holonomic if it withholds holonomic constraints; otherwise, it is nonholonomic. The term constraints in the mechanical system refer to constraints of position and velocity, which means that the relation between the position and velocity must be satisfied. The conception of holonomic constraints in wheeled mobile robots refers to the constraints on the position (configuration). There are two type of holonomic contrains[6]:

- Constraints in which time explicitly enters into the constraint equation are called rheonomic.

$$f(x_1, x_2, x_3, t) = 0 \tag{2.1}$$

  with $t$ time as an explicit variable.

- Constraints in which time is not explicitly present are called scleronomic.

$$f(x_1, x_2, x_3) = 0 \tag{2.2}$$

where the $x_1$,$x_2$,$x_3$ are the system coordinate.
On the other hand, the nonholonomic constraints also know as velocity constraints or kinematic constraints. These constraints limit the space of possible velocities, and the relation is non-integrable to

position constraints [7]:

$$f(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3) = 0 \tag{2.3}$$

where the $x_1$,$x_2$,$x_3$ are the system coordinate, $\dot{x}_1, \dot{x}_2, \dot{x}_3$ respect velocities.

Since the studied system is a type of nonholonomic mobile robot, let us look at the nonholonomic mechanical system. The wheeled mobile robot, such as a RLVs, is subject to nonholonomic constraints because of the constraints of rolling wheels. As many WMR literature concerns methodologies for analyzing, designing, and controlling the mobility system due to rolling without slip constraints, the kinematic methodology is widely used. Muir and Neuman [8] describe nonholonomic kinematic constraints of a unicycle and several wheeled mobile robots' kinematic models, and the general kinematic methodology for a different type of WMRs is presented.

Moreover, Kelly and Seegmiller [9] applied the kinematic approach to several different examples, including differential steer, ackerman steer, a generalized bicycle model and the different cases of four steered and driven wheels. A nonholonomic mechanical system with symmetry is discussed using the Lagrangian mechanic approach to view the control application by Bloch et al. [10]. A dynamic model for three case studies is performed in [11] to provide more accurate studies of wheeled mobile robot motions, which can be later used in control system design and simulation.

In many studies, the kinematic model of the complex mobile robot are determined based on the kinematic model of the unicycle system (see Figure 2.1). Without a loss of generality, the kinematic modeling of a unicycle can be expressed as follows:

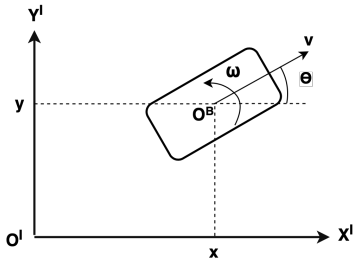



Figure 2.1: Relevant variables for the unicycle system

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} \tag{2.4}$$

with $(x, y, \theta)$ the pose in the global frame, $(\dot{x}, \dot{y}, \dot{\theta})$ the velocities in the global frame, $v$ the linear velocity, $\omega$ the angular velocity in the local frame, and $\theta$ is the angle of the wheel with respect $x$ axis. The unicycle system is under a Rolling Without Slip (RWS) condition, where the following relation must be obey:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{2.5}$$

Based on the previous equations, the kinematic modeling of rhombic like vehicle and analysis of motion capabilities is proposed in Fonte [5]. The type of thombic vehicle in study is a nonholonomic vehicle

with two driven wheels, and presents increasing maneuvering ability compared to the remaining nonholonomic vehicles because of the unique mechanical configuration. However, it does not model the kinematics in the modules level. Based on a formulation of kinematic modeling of a RLVs described by Fonte [5], this dissertation presents a kinematic analysis of each module's system and kinematic modeling of the vehicle.

## 2.2 Localization and Path Planning

The localization problem is a crucial key for the path planning and motion control problem. In short, the localization problem intent to estimated the mobile robot position and orientation in a certain instant of time. The precise localization of the robot ensure the accurate navigation. In general, there are two type of localization approach: absolute (global) localization and relative (local) localization [12]. The absolute localization method use the beacons, landmarks or satellite based signals to obtain the absolute positions, while the relative localization method evaluate the position using on-board sensors. Li et al. [13] propose a mobile robot localization method which combines relative positioning with absolute orientation. It use the Extended Kalman Filtering (EKF) to realize the precise localization of wheeled mobile robot. Doumbia and Cheng [14] introduce algorithm of state estimation and localization based on sensor fusion for four wheeled mobile robot, in indoor environment.

The motion planning problem is as much important as the localization problem, because it calculates the robot's feasible path for the robot in a particular environment. The path planning algorithms can also be divided into two broad categories: global path planning and local path planning. If the work condition of the robot is well known before the robot moves, then the robot path can be obtained by global path planning algorithms, take in consideration of the travelled distance, energy expended, time exposed to danger. Different kinds of approaches have been proposed, such as cell decomposition Glavaški et al. [15]. The cell decomposition is a type of the path planning algorithm that divides the space into connected regions, which called cells, and the desired path can be obtained by connecting the midpoints of the adjacent cells. Another approach is the visibility graph method in Dudek and Jenkin [16], and it has been used for path planning of mobile robots among polygonal obstacles for a long time. This approach has the advantage of calculating the shortest collision-free optimal trajectory quickly and easy implementing.

When the robot has partial knowledge about the environment before it starts, the robot has to plan the path locally. Defoort et al. [17] describes the path planning problem as a linear optimal problem with constraints, which guarantees the navigation of the robot in unknown environments.

Laumond et al. [18] presented a efficient motion planner for a car-like robot that involved the metric induced by the shortest viable path length in the absence of obstacles. Several approaches are presented by Triggs [19] that enable plan complex paths for vehicles that are subject to nonholonomic constraints. It proposed Dubin's theorem and Reeds-Shepp theorem to calculate the shortest paths for car-like vehicles. For the RLVs in studied, Fonte [5] proposed two different motion planning problem: the refinement planning strategy and the Rapidly-exploring Random Tree (RRT), and the simulation results show the

good proficiency of the presented solutions on handling feasible reliable paths in cluttered scenarios. A complete and feasible path planning for the RLVs that will operate in the buildings of the ITER provided a insight for the trajectory guidance and path following in the next stage.

## 2.3 Motion Control

Many WMR applications required the capacity of moving autonomously along a desired path or trajectory. Therefore, after the mobile robot's motion planning stage, how the system reaches the desired trajectory involves developing a control system. The possible motion tasks are classified as follows:

- **Path following**: The path following problem concerns the design of control laws that allows an object (robot manipulator, mobile robot ,etc) to reach and follow a geometric path. It only concerned with the geometrical error between the actual pose of the vehicle and the path to be followed, and the controller of the speed is usually provided by a additional velocity controller.
- **Trajectory tracking**: The trajectory tracking problem concerns on the minimization of a function error that includes the mismatch between the actual configuration of the object and the virtual configuration. A trajectory is normally defined as a time or velocity parameterized path.

Different kinds of approaches have been proposed for the motion control problem of mobile robot. The pure pursuit algorithm has been used in the robotics field as a tracking algorithm for many years. Amidi and Thorpe [20] proposed and tested this type of algorithm under various conditions and found it has the most outstanding results as a general-purpose tracking algorithm. Campbell [21] presents a novel approach to the implementation of the traditional "pure pursuit" style controller in which a dynamic vehicle model is used to map from the path curvature specified by the pure pursuit algorithm to the vehicle's actual steering angle. This pure pursuit controller is then contrasted with a simulation-based controller that uses a kinematic model to predict the vehicle's response to a series of different steering inputs.
Various advances made in guidance problem have been published in the mobile robot path following and trajectory tracking problem using linear control and nonlinear control. An path following of a dynamic wheeled robot in the presence of parameter uncertainty by Soetanto et al. [22], which a control law was developed to control the rate of progression of a virtual target to track the desired path and offered a formal demonstration of convergence of the robot to the path. In Kanjanawanishkul et al. [23] a path following for unicycle mobile robot with a optimal forward velocity is performed with model predictive control (MPC). Oelen and van Amerongen [24] proposed a linear tracking controller with two degrees of freedom, where the controller's accuracy only depends on the geometry of the reference trajectory. The fractional PI controller was applied for Four Wheel Drive Skid-Steered Mobile Robot in Orman et al. [25] in which the controller produces the torques of each motor of a mobile robot for trajectory tracking and stabilization in the desired position, and the comparison between the fractional PI controller and the conventional PI controller is evaluated.
Micaelli and Samson [26] present two types of controller for path following problem for the unicycle type and two steerings wheeled mobile robot based on the relative path to vehicle distance and orientation.

The first one is designed through feedback linearization, while the second one is based on the Lyapunov-oriented approach. It also studied a solution to overcome the mobile robot singularities. Both linear and nonlinear solutions achieved satisfactory results for path following problems. On many occasions, the linear control is preferable to nonlinear control because of its simplicity compared to the nonlinear controller; however, its robustness is limited. The study of De Luca and Oriolo [27] showed that the absence of smooth stabilizability for nonholonomic systems is limited to stabilizing an equilibrium point, which means the problem is genuinely nonlinear. Since the nonlinear feedback controllers can be complicated in real applications and implementation, a model-based control design for trajectory tracking of mobile robot based LQR is often used for tracking problem in recent years. It is one of the most widely used static state feedback methods.

The research Morales et al. [28] presents an LQR trajectory tracking control of an omnidirectional wheeled mobile robot, where an LQR controller is designed for speed control at the motor level, and a comparison between the LQR controller and PI controller is discussed. Lin et al. [29] showed a LQR controller for a car-like robot, based on a linearized kinematic model around the origin. The study shows the effectiveness of the optimal LQR controller for the straight and curve trajectories. A LQR controller for a particular two wheeled mobile robots - KHEPERA IV is presented in Abbasi and Moshayedi [30]. The studies discussed this type of robot's complete dynamic model and applied several challenge path to the system. The simulation results showed the control design could track applied reference paths with an acceptable tracking error.

Both linear and nonlinear solutions have their advantages and disadvantages. Therefore, various researchers try to combine both solutions for better performance. Demirbaş and Kalyoncu [31] proposed a PID controller with a kinematic-based backstepping controller for a differential mobile robot trajectory tracking problem. The backstepping controller is used to overcome the system's non-linearity, and the PID controller was used for the motor speed adjustments. As for the rhombic-like vehicle, four types of path-following control of rhombic like vehicles are presented in Silva et al. [32], all based on the vehicle's kinematic model. However, this study suggests room improvement by include vehicle and actuators dynamics. It also demonstrates that as the vehicle's mass in the analysis is significant, the controller's performance deteriorates as the mass increases. And for the best of us knowledge, there is no designed controller for the type of RLVs in the study: a vehicle with two modules, each module has two driven wheels. This work aims to develop a controller based on the vehicle's kinematic and the modules, including the actuators' dynamic, for RLVs to track the desired reference.

# Chapter 3

# Mathematical Modeling

A formulation of the system aims to model the motion of the rhombic vehicle system. It is an essential phase in the simulation and control problems. For this reason, this chapter presents the RLVs modeling. In order to provide a general idea of this kind of vehicle system, like available actuators, their configuration and limitations, the RLVs prototype is described in Section 3.1. Section 3.2 to Section 3.5 describes the complete RLVs model. The kinematic control oriented model is introduced in Section 3.6.

## 3.1 Rhombic Like Vehicle System

The Computer-aided design (CAD) representation of the system is presented in Figure 3.1. The RLVs

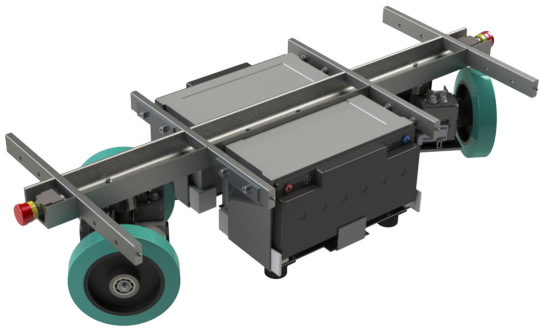


Figure 3.1: CAD representation of the RLVs without platform

is composed of a front module and a rear module. Each module has two wheels, and a motor is associated with each wheel. The system's planar representation is presented in Figure 3.2 to illustrate the main component of the system. To discuss the vehicle's motion, it is necessary to define the its positioning information and coordinate frame. According to the Figure 3.3, the vehicle motion is defined with the body-fixed coordinate frame $(x^B, y^B)$ and the fixed coordinate frame $(X^I, Y^I)$. $\psi$ is the projection of the

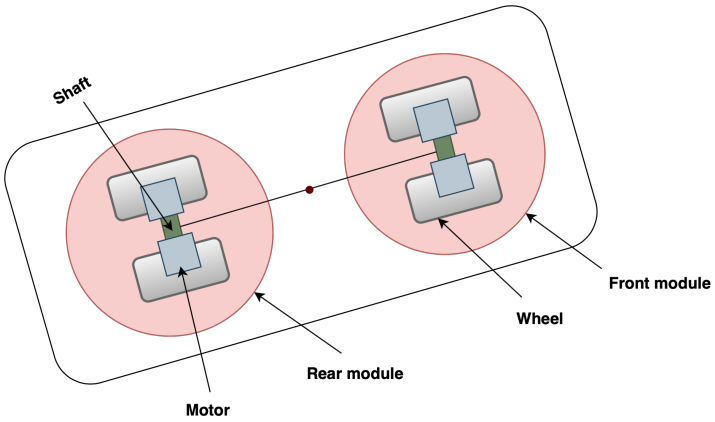



Figure 3.2: Planar representation of the vehicle

angle from the $X^B$ to the world's $X^I$ and $Y^I$ plane. It increases in the counter-clockwise direction, which implies that the vehicle's left turn signifies a positive radius. The vehicle pose is defined as $(x, y, \psi)$, and it moves with linear speed $v$ and angular speed $\omega$.

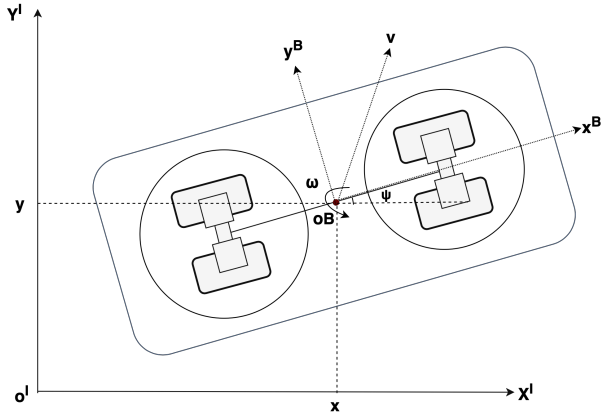



Figure 3.3: Vehicle coordinate system

The modeling of a complex system, when possible, the approach is to separate the system into subsystems and divide the task of modeling a complex system in modeling each of these simpler subsystems. In this case, the goal is to model the motion of the RLVs, thus the equations are divided into three parts as presented in Figure 3.4. Each system is described as follows:

- The drive system includes the four DC motors and the respect mechanical gear system. The input to the drive system is the voltage source $V_{ij}$, and the results of the drive system are the torque on the wheel $\tau_{ij}$ and the wheel angular speed $\omega_{ij}$, with the $i = f, r$ refer to the front or rear module and $j = r, l$ right or left side motor, respectively.
- The modules system includes the front and the rear module. Each module receives the actions from the motors, in this case $\tau_{ij}$ and $\omega_{ij}$ torque and angular speed of the wheel, and the result of the modules system are the forces and moments acting on the center gravity of the vehicle.
- The equations of motion, where the forces and moment are responsible for the vehicle motion according to the physics laws, dynamics, or kinematics equations. The state of the system is

noted as x. It vary depending on the modeling perspective and will be refereed at the proper time.

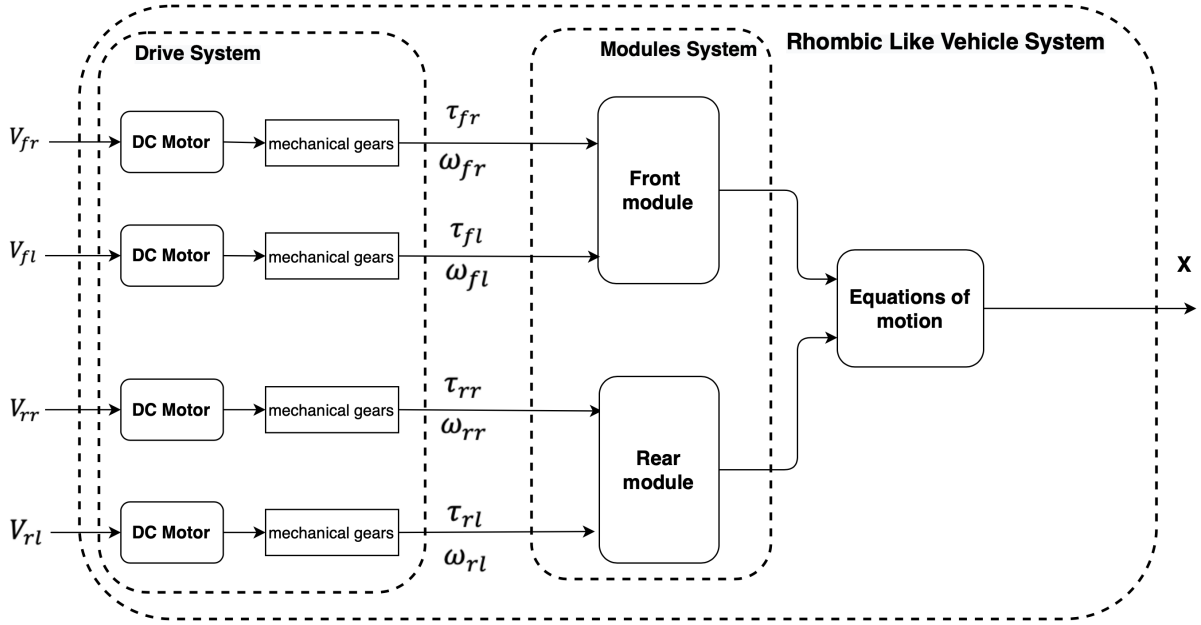



Figure 3.4: Decomposition of the system

In the following sections, each system will be described from left to right in detail.

## 3.2 Drive System Modeling

In this section, a modeling process of the drive system is developed. It is assumed that the vehicle is driven by four DC motors with mechanical gears. Therefore, the input of the drive system is four voltage source $V_{ij}$, while the output is the wheel's rotational speed $\omega_{ij}$ and torque $\tau_{ij}$, with the $i = f, r$ refer to the front or rear module and $j = r, l$ right or left side motor, respectively. The block diagram of the drive system is shown in Figure 3.5.


Figure 3.5: Drive system of the vehicle

Each motor will be modeled as a DC motor with mechanical gears (see Figure 3.6). A motor has current $I$, rotor angular speed $\omega_m$, inductance $L_e$, moment of inertia $J$, resistance $R_e$, the induced electromotive force $E_{emf}$,the voltage source $V$, $K_T$ is the constant that only depends on the force act on a conductor in the magnetic field, $K_b$ is the constant of friction.
The dynamic equation of the motor is presented as follows:

$$R_e I + L_e \dot{I} + E_{emf} - V = 0 \tag{3.1}$$

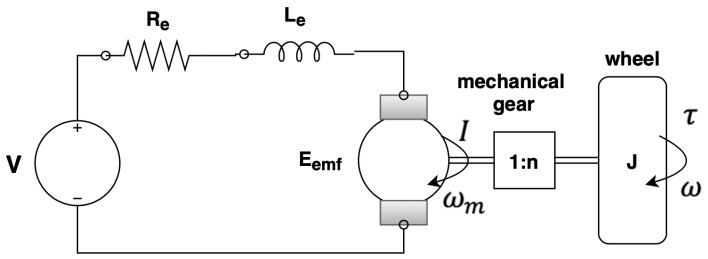



Figure 3.6: Schematic of DC motor with mechanical gear

$$J\dot{\omega}_m = \tau_m - c\omega_m \tag{3.2}$$

where the torque generated by DC motor $\tau_m$ is proportional to the current, and the electromotive force $E_{emf}$ proportional to the angular velocity as follows:

$$\begin{aligned} \tau_m &= K_t I \\ E_{emf} &= K_b \omega_m \end{aligned} \tag{3.3}$$

Using the equations (3.1) and (3.2), the motor system can be presented in the state space matrix form as follows:

$$state : x = \begin{bmatrix} I \\ \omega_m \end{bmatrix} \tag{3.4}$$

$$input : u = V \tag{3.5}$$

$$\begin{bmatrix} \dot{I} \\ \dot{\omega}_m \end{bmatrix} = \begin{bmatrix} -\frac{R_e}{L_e} & -\frac{K_b}{L_e} \\ \frac{K_t}{J} & -\frac{c}{J} \end{bmatrix} \begin{bmatrix} I \\ \omega_m \end{bmatrix} + \begin{bmatrix} \frac{1}{L_e} \\ 0 \end{bmatrix} V \tag{3.6}$$

The motor engine model used in the project FORMULA fusion is the MDXL61GN3IP, a single and integrated motor package (motor, drive and controller) sponsored by Applied Motion company (Reference: $J0200 - 408 - 4 - 000A$), as illustrated in Figure 3.7. According to the information provided by Applied

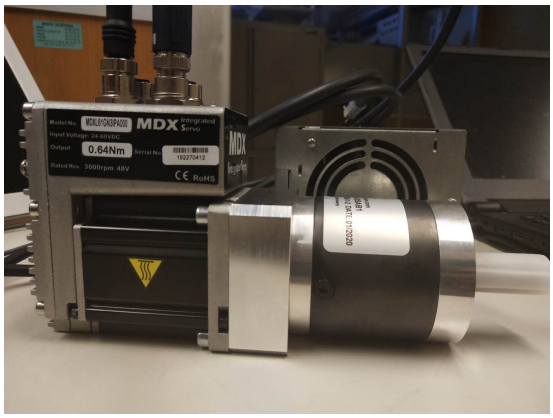



Figure 3.7: MDX Servo Motor

Motion company (see Annex B), the motor specifications and parameters are given in Table 3.1.

Table 3.1: Motor Model Parameters

| **Motor Parameters** | **Value** |
|---|---|
| Armature Resistance | $R_e = 0.192\,\Omega$ |
| Armature Inductance | $L_e = 0.56 \times 10^{-3}\,\mathrm{H}$ |
| Torque constant | $K_t = 0.192\,\mathrm{N \cdot m/A}$ |
| Voltage constant | $K_b = 0.392\,\mathrm{V/(rad/s)}$ |
| Friction coefficient | $c = 3.0 \times 10^{-3}\,\mathrm{N \cdot m/s}$ |
| Inertia | $J = 165 \times 10^{-7}\,\mathrm{kgm^2}$ |
| Supply Voltage | 24-60 $V$ |
| Maximum Speed | 6000 $rpm$ |
| Peak torque | $1.9\,\mathrm{N \cdot m}$ |

The block diagram of DC motor is presented in Figure 3.8, where the input is the voltage and the output is the current $I$ and rotor angular speed $\omega_m$. The drive system is equipped with a mechanical gear

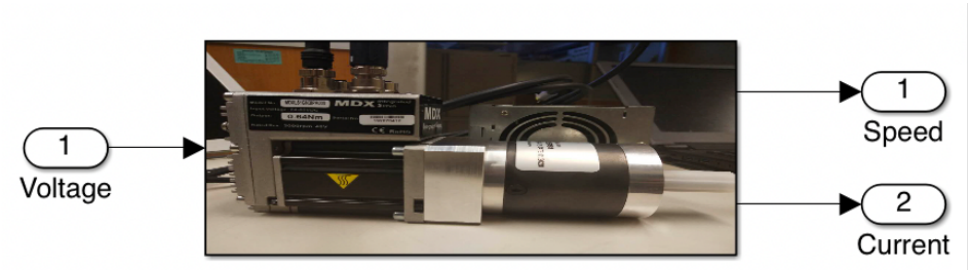



Figure 3.8: Block diagram of DC motor

system as presented in Figure 3.9 where the transmission is consists of two stages: a small gearbox and a set of gear wheels. The whole gear system provides the gear ratio $n$, where $n = 84$, therefore:

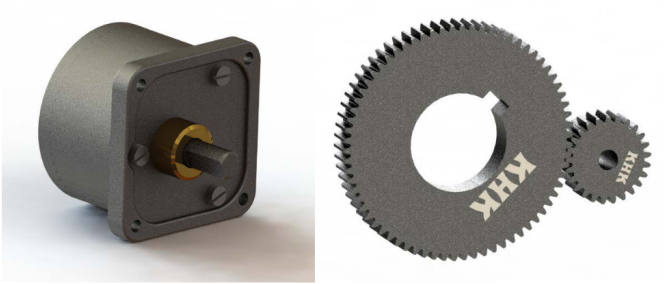


Figure 3.9: Mechanical gear system

$$\begin{aligned} \tau &= nK_tI \\ \omega &= \frac{\omega_m}{n} \end{aligned} \tag{3.7}$$

with the $\tau$ is the torque acting on the wheels, and $\omega$ is denoted as the angular speed of the wheel until a new definition is assigned. Additional dynamic effects related to the mechanical gear system are herein not considered.

## 3.3 Modules System Modeling

This section, a detailed description of the modules system is being studied. After analyzing the drive system, one obtains the torque $\tau_{ij}$ and wheel angular speed $\omega_{ij}$ as the drive system's output. To understand how the wheel torque effect the vehicle's motion, the relation between the module velocity and orientation with the wheel angular speed is presented in this section. The modules system is described in which one obtains the equations that have the inputs as $\tau_{ij}$ and $\omega_{ij}$. The system's outputs are forces $F_{px}$ and $F_{py}$, and moments $M_{pz}$ in the center of the vehicle, which is a function of the forces acting on the front module and rear module wheels, denoted as $Fp_{fj}$ and $Fp_{rj}$ respectively, with $j = r, l$ refer to the right and left wheels (see Figure 3.10).

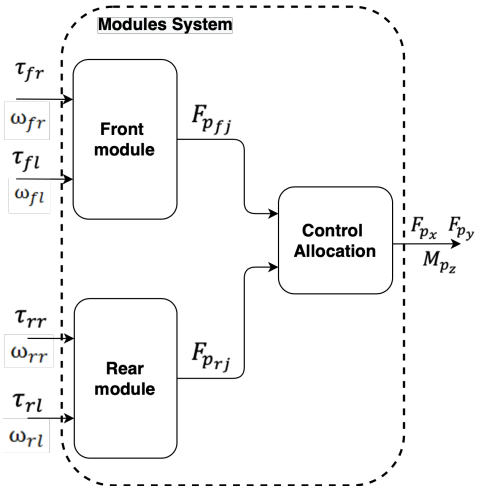



Figure 3.10: Representative diagram of modules system

The first module is implemented in the scope of the project. Figure 3.11 shows a module with two parallel driven wheels, each wheel is attached with a drive system and a power supply. For the mathematical analysis of the module model, based on the Figure 3.11, the model representative of the module is presented in Figure 3.12. The $(x^m, y^m)$ is the module fixed frame and $(X^I, Y^I)$ is the global fixed frame, and the $ith$ module moves with forward linear velocity $v_i$ and angular velocity $\omega_i$. A module in general moves in Two Dimensional (2D) world with forward speed and zero instantaneous lateral motion. As two modules have same configuration, the modeling of the module system is identical for the front and rear module. For notation simplicity, in this section, $v$ and $\omega$ is refer to the modules linear and angular speed until new definition is assigned.

Suppose that each wheel of the module has an angular velocity $\omega_r$ and $\omega_l$ with $(r, l)$ refers to the right and left wheel, and $R$ the wheel radius. The model representative of the modules in the module fixed frame $(x^m, y^m)$ is presented in Figure 3.13, and the module rotates around the Instantaneous Center of

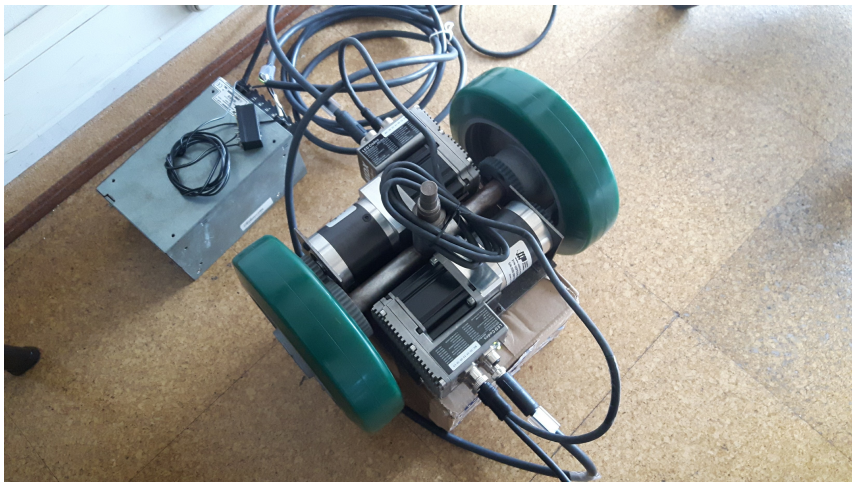


Figure 3.11: Module prototype attached with drive system

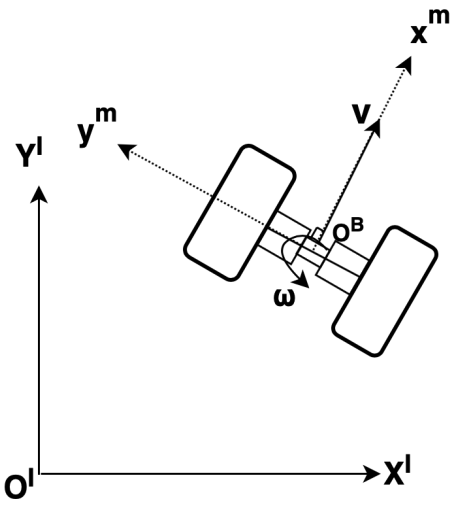



Figure 3.12: Model representative of the front module

Rotation (ICR). $D$ the distance between the wheels. According to the geometric relations presented in

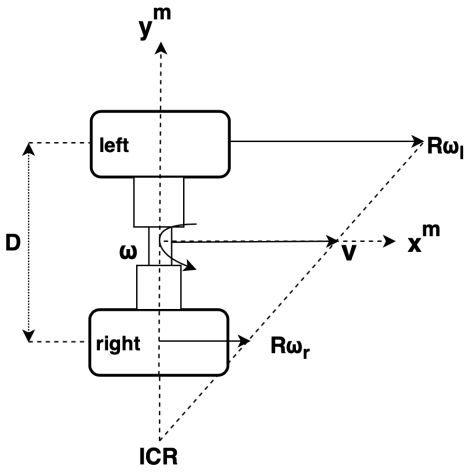



Figure 3.13: Geometric relation of the front wheels

Figure 3.13, the forward velocity $v$ and angular rate $\omega$ depends on both two wheels angular velocities as:

$$v_x = \frac{R\omega_r + R\omega_l}{2} \tag{3.8}$$

$$\omega = \frac{R\omega_r - R\omega_l}{D} \tag{3.9}$$

Therefore the geometric relation in the front module can be summarized as follows:

$$\begin{bmatrix} v_{fx} \\ v_{fy} \\ \dot{\theta}_f \end{bmatrix} = \begin{bmatrix} \frac{R(\omega_{fr}+\omega_{fl})}{2} \\ 0 \\ \frac{R(\omega_{fr}-\omega_{fl})}{D} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{D} & -\frac{R}{D} \end{bmatrix} \begin{bmatrix} \omega_{fr} \\ \omega_{fl} \end{bmatrix} \tag{3.10}$$

with the $\omega_{fr}$ the angular speed of the front module right wheel and $\omega_{fl}$ the angular speed of the front module left wheel. Analogously the same for the rear module:

$$\begin{bmatrix} v_{rx} \\ v_{ry} \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \frac{R(\omega_{rr}+\omega_{rl})}{2} \\ 0 \\ \frac{R(\omega_{rr}-\omega_{rl})}{D} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{D} & -\frac{R}{D} \end{bmatrix} \begin{bmatrix} \omega_{rr} \\ \omega_{rl} \end{bmatrix} \tag{3.11}$$

with the $\omega_{rr}$ the angular speed of the rear module right wheel and $\omega_{rl}$ the angular speed of the rear module left wheel. Combining the equation (3.10) and (3.11), the modules linear and angular speed is now expressed in function of $\omega_{fr}$, $\omega_{fl}$, $\omega_{rr}$ and $\omega_{rl}$ in the module frame. In the module frame, $v_{fy}$ and $v_{ry}$ are null, therefore the (3.10) and (3.11) can be simplified as:

$$\begin{bmatrix} v_f \\ \dot{\theta}_f \\ v_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} & 0 & 0 \\ \frac{R}{D} & -\frac{R}{D} & 0 & 0 \\ 0 & 0 & \frac{R}{2} & \frac{R}{2} \\ 0 & 0 & \frac{R}{D} & -\frac{R}{D} \end{bmatrix} \begin{bmatrix} \omega_{fr} \\ \omega_{fl} \\ \omega_{rr} \\ \omega_{rl} \end{bmatrix} \tag{3.12}$$

with $R$ the wheel radius and $D$ the distance between the wheels.

Now one gets the modules velocities and orientations, how the force and moment affects the vehicle motion is studied. The active force is generated by the motor, those forces are aligned to the wheels, how those forces acting on the center gravity of the vehicle depends on the action of the modules system. The force diagram with all active forces is presented in Figure 3.14: where $\theta_f$ and $\theta_r$ are the modules orientations, and $F_{ij}$ the active forces of the vehicle with $i = (f, r)$ refer to the front, and rear modules and $j = (r, l)$ to the left and right wheels, respectively. The $F_f$ and $F_r$ is the resulting force acting on the front and rear module, respectively. $F_{px}$ and $F_{py}$ are the resulting forces in the $x$ and $y$ direction, and $M_{pz}$ the moment, in the center of the vehicle.

The active force is generated by the motor, and can be calculated as:

$$F_{ij} = \frac{\tau_{ij}}{R} \tag{3.13}$$

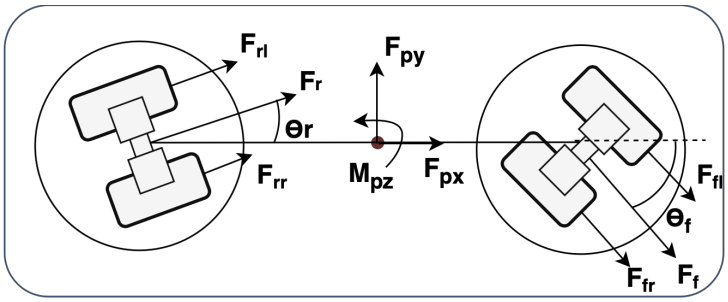



Figure 3.14: Force diagram for the vehicle

with $\tau_{ij}$ the torque action on the wheels, and $R$ the wheel radius.

The calculation of the resulting forces of the modules level in the vehicle frame is performed once each module's orientation is determined. For the analysis of the forces, the following relation is obtained according to Figure 3.14:

$$\begin{aligned} F_{ij_x} &= F_{ij}\cos\theta_i \\ F_{ij_y} &= F_{ij}\sin\theta_i \end{aligned} \tag{3.14}$$

with the index $i$ refers to the $ith$ module, $j$ refers to the $jth$ wheel.

The resulting forces in $x$ and $y$ direction, and moment from the action of the modules system are obtained in relation to the vehicle center of gravity's local coordinate system:

$$F_{px} = F_{f_x} + F_{r_x} = (F_{fr_x} + F_{fl_x}) + (F_{rr_x} + F_{rl_x}) \tag{3.15}$$

$$F_{py} = F_{f_y} + F_{r_y} = (F_{fr_y} + F_{fl_y}) + (F_{rr_y} + F_{rl_y}) \tag{3.16}$$

$$M_{pz} = \frac{D}{2}(F_{fr_x} + F_{rr_x}) - \frac{D}{2}(F_{fl_x} + F_{rl_x}) + L_f(F_{fr_y} + F_{fl_y}) - L_r(F_{rr_y} + F_{rl_y}) \tag{3.17}$$

where the $\theta_f$,$\theta_r$ the front and rear orientation, respectively. $D$ the distance between the left and right wheels, $L_f$,$L_r$ the distance of the front, rear module to the vehicle center of gravity, respectively.

## 3.4 Vehicle Motion: dynamics and kinematics

Two sets of equations is used to describe the vehicle motion: the equation of dynamics, which relates the forces acting in the body to it is acceleration or inertia, and the kinematics equations, which relate the speed of the body to its position. When the vehicle is in motion, there are several force acting on center of the vehicle (see Figure 3.15). As the type of the vehicle is expected to operate in the flat surface, the dynamic model is simplified. According to the Newton's second law of motion, the dynamic of the

vehicle in the vehicle body frame can be described as:

$$\begin{aligned} M\dot{v}_x + \omega M v_x &= F_x \\ M\dot{v}_y + \omega M v_y &= F_y \\ I_z\dot{\omega} &= M_z \end{aligned} \tag{3.18}$$

where the velocity vector expressed in the vehicle frame $(x^B, y^B)$ as:

$$v^B = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3.19}$$

and $M$ is the mass of the vehicle and $I_z$ the vehicle inertia, and from now on the $\omega$ is denoted as the angular speed of the vehicle.

After analyzing the drive system and modules system, one obtains the active forces $(F_{px}, F_{py})$ and moments $M_{pz}$ acting on the center of the vehicle. Additionally, the friction forces is considered. For

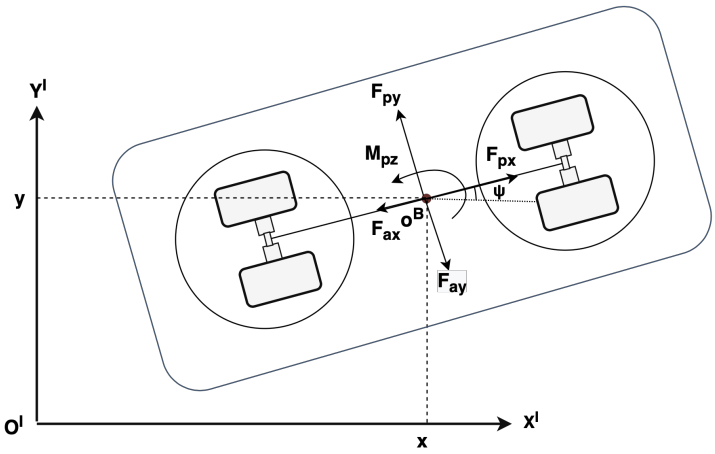



Figure 3.15: External forces acting on the vehicle

simplicity, all the resistant force presents in the vehicle is summarized as $F_a$, and defined as a function of the vehicle's speed :

$$\begin{aligned} F_{ax} &= \mu_x v_x \\ F_{ay} &= \mu_y v_y \end{aligned} \tag{3.20}$$

where $\mu_x$ and $\mu_y$ are the friction coefficients. The value of the friction coefficients are related to the actual pose of the RLVs and the surface condition of the work environment. It can be obtained experimentally. Due to lack of the information, the first estimation of the friction value is made based on the Fonte [5], and then adjusted to the RLVs model in studied.

Thus the velocities in the body frame is presented as follows:

$$\dot{v}_x = \frac{1}{M}(F_{px} - \mu_x v_x - \omega v_x) \tag{3.21}$$

$$\dot{v}_y = \frac{1}{M}(F_{py} - \mu_y v_y - \omega v_y) \tag{3.22}$$

$$\dot{\omega} = \frac{1}{I_z} M_{pz} \tag{3.23}$$

For control and navigation purposes, the velocity vector expressed in the vehicle frame $(x^B, y^B)$ as (3.19) must be transformed to the global frame $(X^I, Y^I)$:

$$v^I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} \tag{3.24}$$

and this leads to the kinematics relations.
Consider the coordinates $(x, y, \psi)$ give the vehicle position in the global frame; it may be regarded as an integration of the inertial velocities $v^I$ in the global frame with the following relations:

$$v^I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = R_B^I \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3.25}$$

with

$$R_B^I = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.26}$$

## 3.5 Rhombic Vehicle Simulator

Based on the modeling equations of the previous sections, the rhombic vehicle model can be expressed in state space form as follows:

$$\dot{x} = f(x, u) \tag{3.27}$$

where:

- the state $x = [x, y, \psi, v_x, v_y, \omega, \theta_f, \theta_r, I_{ij}, \omega_{ij}]^T$
  with $(x, y)$ define the position of the vehicle, $\psi$ define the orientation of the vehicle, $v_x, v_y, \omega$ define the velocities of the vehicle in the local frame, and $\theta_f$,$\theta_r$ the modules orientations. $I_{ij}$ and $\omega_{ij}$ are the current and angular velocity of four motors, respectively, with $i = (f, r)$ the front and rear module, $j = (l, r)$ the left and right wheels.
- the input vector $V_{ij} = [V_{fr}, V_{fl}, V_{rr}, V_{rl}]^T$ the voltage source of the four motors.

The RLVs model is based on (3.27) and block diagram presented in Figure 3.4. For simulation, the following model parameters (see Table 3.2) is used.

Table 3.2: Vehicle Parameters Simulation setup

| **Vehicle parameters** | **Value** |
|---|---|
| Distance of the front module to the Center of Gravity (CG) | $L_f = 0.5\,\text{m}$ |
| Distance of the rear module to the CG | $L_r = 0.5\,\text{m}$ |
| Vehicle width | $W = 0.7\,\text{m}$ |
| Vehicle length | $L = 1.1\,\text{m}$ |
| Distance between the wheels | $D = 0.25\,\text{m}$ |
| Wheel radius | $R = 0.1\,\text{m}$ |
| Vehicle Inertia | $I_z = 250\,\text{kgm}^2$ |
| Total Mass of the Vehicle | $M = 375\,\text{kg}$ |
| Maximum speed | $v = 0.5\text{m/s}$ |

A simulator was built in *Matlab/Simulink* for the validation of the control system (see Figure 3.16). In

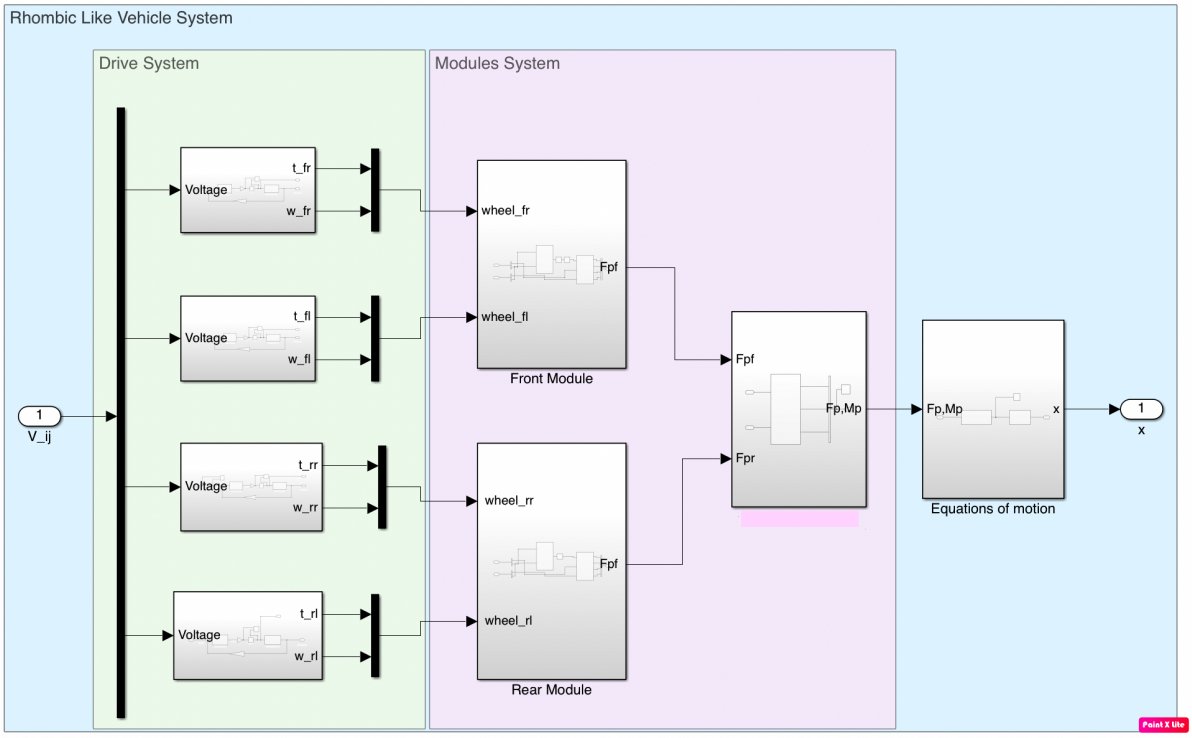
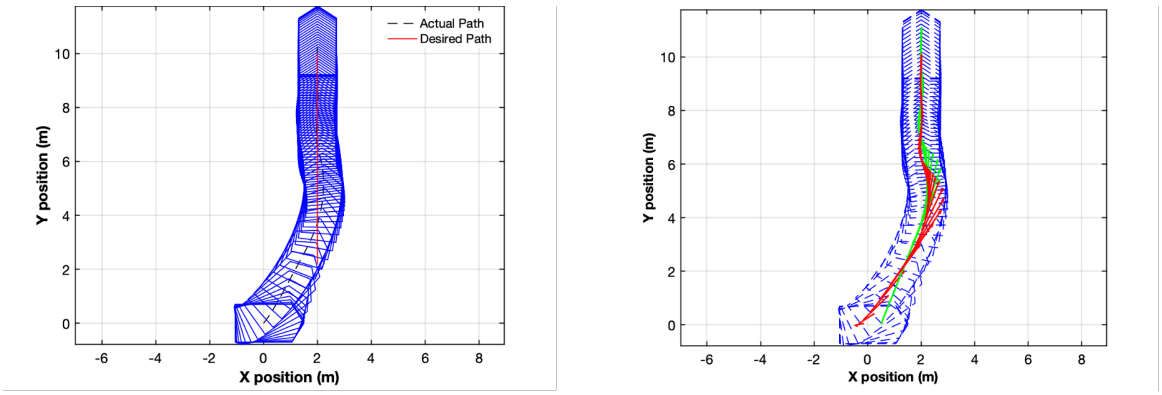



Figure 3.16: Simulator in *Simulink* environment

additional a *Matlab* function is created for better visualization of the vehicle motion and the modules orientations. The vehicle is simplified as a rectangle with nozzle on the head (see Figure 3.17(a)), the size of the vehicle is design proportional to the parameter indicated in Table 3.2. The modules velocity and orientation is expressed as vectors (see Figure 3.17(b)). The front and rear modules are presented as green and red arrow, respectively. The size of the arrow is proportional to the value of the modules speed. As for the better visualization, the ratio of velocities and size of arrow is changed case to case in

the next chapter, and the function is used mainly for illustration of the modules orientation.




(a) RLVs motion visualization

(b) Modules Orientation: The green arrow represents the front module and the red arrow represents the rear module

Figure 3.17: Motion visualization

## 3.6 Kinematic Model of the Rhombic Vehicle

The model presented in the previous sections provides a complete description of the RLVs behavior and will be used in the simulation of the designed controller in chapter 5 or whenever high accuracy simulations are needed. This model's limitation is the complexity; it does not provide an insight into the control design. Therefore a simple model based on the system kinematics is presented in this section that preserves the primary characteristic of the RLVs motion for control-oriented studies. The vehicle system and the respect coordinate system are presented in Figure 3.18. As mentioned in previous

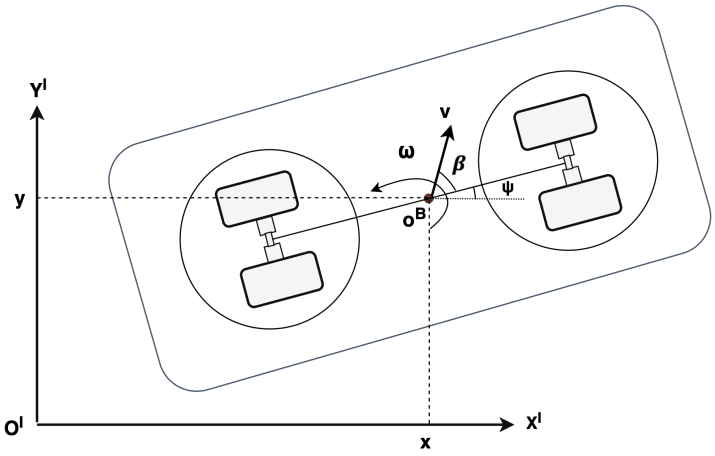



Figure 3.18: Model representative of the vehicle

chapter, in many studied, the kinematic model of the complex mobile robot are determined by combining

the kinematic model of a individual rolling system which is described as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{3.28}$$

with $(\dot{x}, \dot{y}, \dot{\psi})$ are the velocities in the global frame $(X^I, Y^I)$, and $v$ and $\omega$, the linear velocity and its angular velocity around the vertical axis, respectively,$\psi$ the robot's orientation with respect $x$-axis.
According to Ma [33] and Figure 3.18, a kinematic model is extended for rhombic like a vehicle as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v\cos(\beta+\psi) \\ v\sin(\beta+\psi) \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\beta+\psi) \\ \sin(\beta+\psi) \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{3.29}$$

with $\beta$ the side-slip angle of the vehicle.
Furthermore, for the studied of a rhombic like vehicle motion, it would be necessary to analyzes the relation of rhombic like vehicle motion concerning modules velocities $(v_f, v_r)$ and the steering angles $(\theta_f, \theta_r)$, therefore the bicycle based model is introduced.

### 3.6.1 Bicycle based model

Kinematics modeling based on the bicycle model for four-wheel steering vehicles is described in [34]. The kinematic bicycle model is one of the standard models used as a suitable control-oriented model, and it is adherent to the nonholonomic constraints. It allows the vehicle motion simulation directly through the modules velocities $v_f$ and $v_r$ instead of velocities of each wheel's linear speed. The bicycle model assumes the modules as a rigid body, as presented in Figure 3.19. According to the literature and

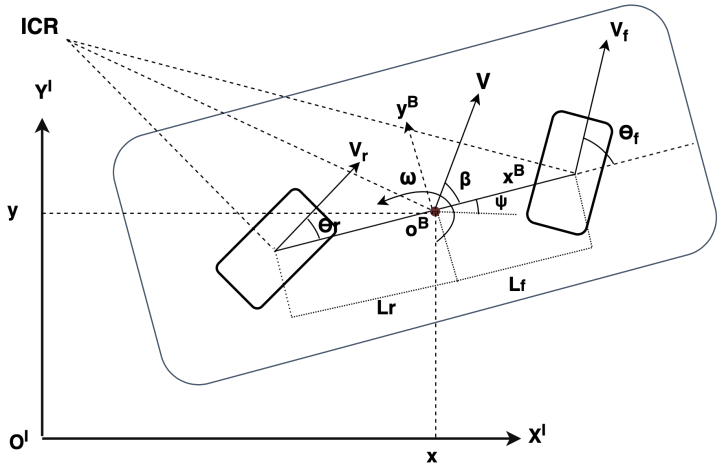



Figure 3.19: Bicycle model

geometric relation in Figure 3.19, the following equations are valid:

$$v^I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\beta + \psi) \\ \sin(\beta + \psi) \\ \frac{(\tan\theta_f - \tan\theta_r)\cos\beta}{L_r + L_f} \end{bmatrix} v \tag{3.30}$$

where $\beta$ is the slip angle of the vehicle.

$$\beta = \arctan \frac{L_r \tan\theta_f + L_f \tan\theta_r}{L_r + L_f} \tag{3.31}$$

$$v = \frac{v_f \cos\theta_f + v_r \cos\theta_r}{2\cos\beta} \tag{3.32}$$

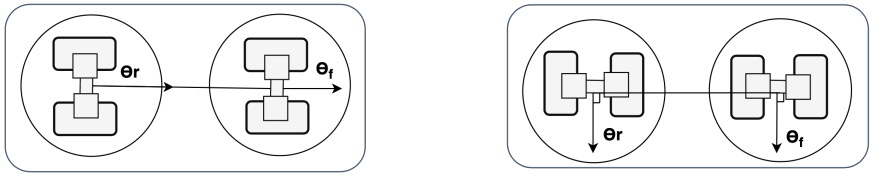
The RLVs is considered as rigid body itself therefore the geometric constraints must be obeyed. The geometric constraints is expressed as follows:

$$v_f \cos\theta_f = v_r \cos\theta_r \tag{3.33}$$

As mentioned above, Equation (3.30) assumes the modules as a rigid body, however the relationship between wheel velocities and modules velocities is already expressed in (3.12) in the previous section.

### 3.6.2 Motion capabilities and maneuvering ability

The mathematical modelling of the RLVs is presented above. It is essential now to analyze some considerations about the RLVs motion. Depending on the steering configuration of the modules system, the type of system enables high maneuverability. Figure 3.20 and Figure 3.21 shows a combination of the steering configuration that allows the rotation around the ICR. The RLVs presented a higher degree

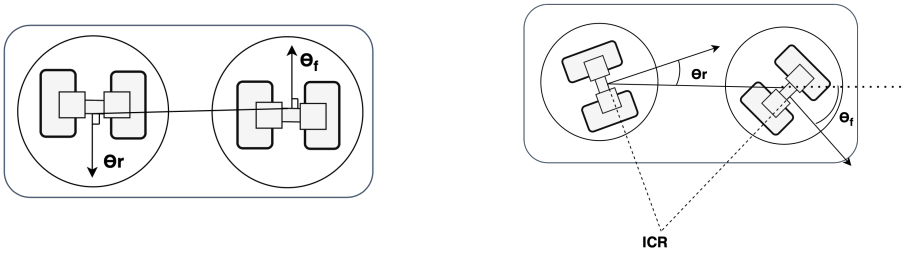



(a) wheels steered in equal angles, in the $x$-direction

(b) wheels steered in equal angles, in the $y$-direction

Figure 3.20: RLVs motion capabilities: I

of freedom compared to the typical car-like robot, especially for the cluttered environment in the ITER. In short, with this kind of vehicle configuration, which the motion of the vehicle is provided by modules movement, the vehicle tuned more easily.




(a) turning on spot, (b) rotation around the ICR

Figure 3.21: RLVs motion capabilities: II

**Singularities and sensible drive configurations**

The multitude of steering and driving configurations of each module allows the RLVs has higher manoeuvrability. However there are still some constrains of the RLVs motion in order to prevent vehicle physical damage. There is some combination of the steering and driving configurations of the RLVs that requires a strict coordination in order to achieve a sensible driving. Figure 3.22 showed possible motion singularities that might occur during the operations task. Most of this singularities cause the violation of the geometric constraints presented in Equation (3.33). In the motion control stage, the design of the control law is taking in account the geometric constraints in order to avoid those type of motion singularities.

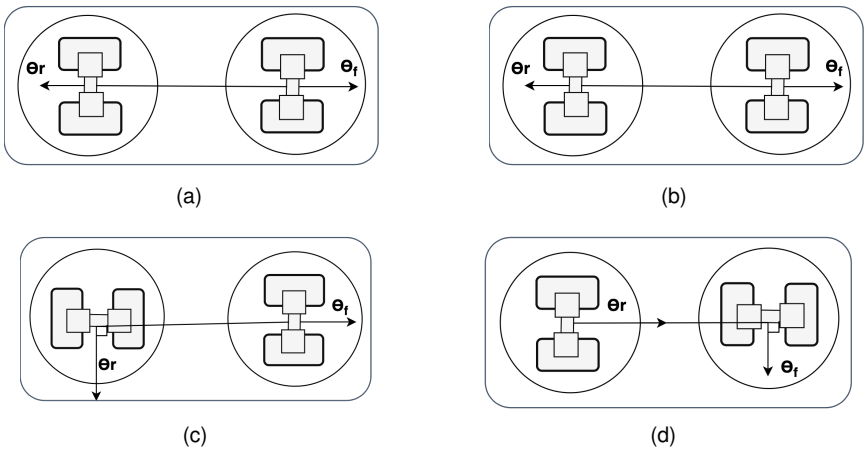



Figure 3.22: RLVs motion singularities

# Chapter 4

# Control Theory

## 4.1 Problem Formulation

As stated in the previous chapters, this work focus on the motion controller design. In chapter 3, the RLVs mathematical model is obtained. The design, implementation, and simulation of the control laws that solve RLVs' tracking problems will be the next task. The motion control is based on the schematic diagram of the control system, as shown in Figure 4.1, where the system's global input is the reference position $(x_{ref}, y_{ref}, \psi_{ref})$, and suppose the current position $(x, y, \psi)$ of the RLVs is measurable. The position control output is the four angular speed of the wheel. The low-level controller (PID controller in the motor level) is used to ensure the desired wheel velocities are reached. A designing process of a controller based on the hierarchical approach is presented: i) low-level law for the motor speed control and ii) high-level control law for the vehicle position control.

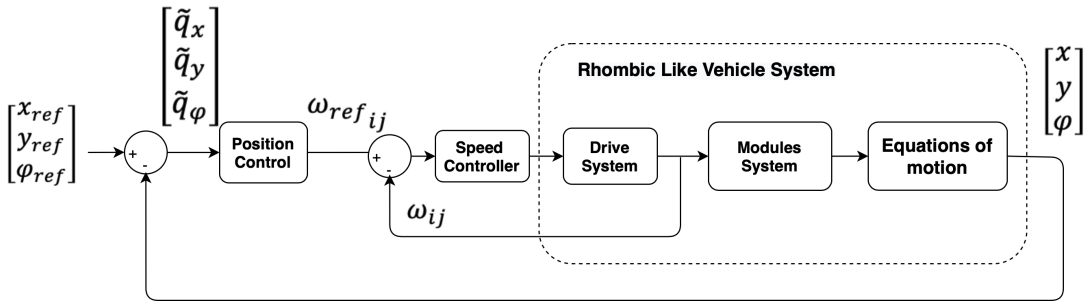



Figure 4.1: Schematic diagram of control system

Today's commonly used control theories are classical control theory, modern control theory, and robust control theory. In this chapter, a brief introduction of the analysis and design of the control system is presented. Section 4.2 has given the design of the PID controller based on the root locus approach. Section 4.3 offers the principle of the geometric based motion control approach for the motion control problem. Section 4.4 discusses state space design methods with the linear quadratic optimal regulator systems. The chapter closes by introducing some general concepts in the design and evaluating the control law in the next chapter.

## 4.2 PID Controller

The PID controller is widely used in the industrial controller, and can be used to improve both system transient response and steady state performance. Many different ways of tuning rules have been proposed in the literature. The PID transfer function is given by:

$$\frac{U(s)}{E(s)} = K_p + K_i\frac{1}{s} + K_d s \tag{4.1}$$

$K_p$ is the proportional gain, $K_i$ is the integral gain, $K_d$ is the derivative gain, $E(s)$ is a error tracking signal and $U(s)$ is the control actions. The proportional gain $K_p$ decrease the rise time but leads in more oscillation, the integral gain $K_i$ eliminate steady state errors, but increase the plant system overshoot; the gain derivatives $K_d$ will decrease the overshoot, but increase the setting time and may possibly cause instability. These suits as guideline for the parameter tuning.
If the mathematical modelling of the plant can be found, then the problem is addressed to apply the design techniques to determining the controllers parameters that fulfil the transient and steady-state specifications of the closed loop system. If the mathematical model of the plant is unknown, then a experimental approaches to the tuning of PID controllers can be used. For the control system design and the system analysis, the root locus method is often used. The closed loop system performance is closely related with the location of the closed loop poles, thus one can plot the root locus plot by hand or use software like *Matlab* to studied the basic characteristic of the system. For the control design, the root locus method is used to predict the effects on the location of the closed loop poles by adding poles and zeros to the system open loop transfer function, and try to force the root locus to pass the desired closed loop poles in the $s$ plan. The desired closed loop poles are often calculated according to the desired steady state and transient requirements.
The PID control design via root locus method presented in [35] can be outlined as:

- Evaluation of the uncompensated system to determine how much improvement in transient response is required
- Add zeros and/poles to the system to yield the transient and steady state requirement
- Simulation of the system to be sure all requirements have been met
- Redesign if simulation shows that requirements have not been met

## 4.3 Geometric Path Tracking Controller

Motion control problem aims to compute the control commands such that the vehicle can follow a reference path. The resulting control input should minimize the difference between the reference path and the actual path. There are various methods presented in the literature; two of the most popular types are a geometric and model-based method. In this section, a brief introduction of geometric methods is presented.

A geometric path tracking controller is any controller that tracks a reference path using only the geometry of the vehicle kinematics and the reference path. A very first approach is known as pure pursuit controller which initially was created to calculate the arc necessary to get the robot back to a path. Figure 4.2 intent to show the geometric relation between the desired path and the vehicle actual position. Consider $(x_{ref}, y_{ref}, \psi_{ref})$ are the reference pose of the vehicle, the $(x, y, \psi)$ are the actual pose in the global frame. The $(\delta_x, \delta_y, \delta_\psi)$ are the difference between the reference pose and the actual pose of the vehicle. From the geometric relations:

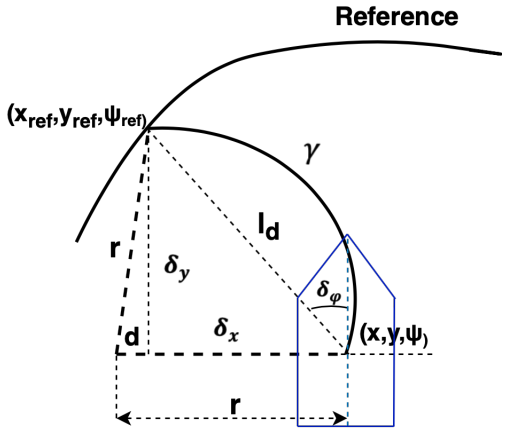



Figure 4.2: Geometry of the Pure Pursuit algorithm. The curvature of the arc indicates a circle of radius $r$. Parameter $l_d$ is the look ahead distance, with $x_{ref}$ and $y_{ref}$ defining the position of the look ahead point relative to the vehicle.

$$\begin{aligned} \delta_x^2 + \delta_y^2 &= l_d^2 \\ d^2 + \delta_y^2 &= r^2 \end{aligned} \tag{4.2}$$

where $d = r - \delta_x$.

According to Figure 4.2, the curvature of an arc is denoted as $\gamma$ and it can be calculated as:

$$\gamma = \frac{1}{r} \tag{4.3}$$

Combining the Equations (4.2) and (4.3), the curvature $\gamma$ is obtained:

$$\gamma = \frac{2\delta_x}{l_d^2} \tag{4.4}$$

In short, the pure pursuit algorithm is a proportional controller based on the error between the current heading angle and the desired heading angle $\delta_\psi$, where $\sin \delta_\psi = \frac{\delta_x}{l_d}$. For small heading angle error

$\delta_\psi = \frac{\delta_x}{l_d}$. Substitute this to the (4.4), one can get:

$$\gamma = \frac{2\delta_\psi}{l_d} \tag{4.5}$$

This algorithms (4.5) only has single parameter to tune: the looking ahead distance $l_d$. Smaller look ahead distance increases the maximum curvature of a path that can be tracked, and the system tracks the path more accurately. Bigger look ahead distance allows the system to begin turning before it reaches the curve, resulting in smoother trajectories.
The pure pursuit algorithm is effective because the control law is designed to calculate the arc based on the cross track error, if the error increase, the arc necessary to get the robot to the path is also bigger, therefore bring the robot to the path more aggressively. Knowing the necessary curvature to get the robot to the path, using the kinematic model, one can get the desired linear and angular speed of the robot, Coulter [36] describe in detail the implementation of the pure pursuit path tracking algorithm, and can be outlined as follows:

- Determine the current location of the robot.
- Find the path point closet to the robot.
- Define the desired point.
- Calculation of the linear and desired angular velocity.
- Update the robot position.

An improvement of the pure pursuit algorithm but also based on the geometry relation is stanley controller. It is the trajectory tracking approach used by Standford University's Darpa Grand Challenge team. Stanley algorithm not only take in account the heading angle error but also the cross tracking error, thus more effective and steady. In brief, it treats automobile trajectory tracking in a new manner, by considering the orientation of the front wheels, not the vehicle's body with respect to the desired trajectory, enabling collocated control of the system [37]. Based on the principle mentioned above, linear or nonlinear control law can be design for calculation of the desired velocity. The reason why this type of controller is often used in the autonomous navigation problem is because the implementation of this type of controller is usually easy, and tend to work better for lower speed driving. As the RLVs is suppose to work under a very low speed, the first approach of vehicle motion control is based on the geometric controller principle.

## 4.4 Quadratic Optimal Regulator

This section briefly describes the optimal control theory. A explanation of the theoretical tools is presented, along with an analysis of the effects of the controller parameters. The optimal control methodology aims to find an optimal control law for a given system with a respect cost function. Considering the

optimal regulator problem that, given the system equation:

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

determines the matrix $K$ of the optimal control vector

$$u = -Kx$$

so as to minimize a certain cost function. Consider the following case, where the standard cost function for a LQR problem is shown as:

$$\begin{aligned} J^* = min_u \int_{t_0}^{t_f} (x(t)^T Q x + u(t)^T R u(t))dt + x(t_f)^T (t_f) F(t_f) x(t_f) \\ \dot{x}(t) = Ax(t) + Bu(t) \\ x(0) = x_0 \end{aligned} \tag{4.6}$$

where $t_f$ is a fixed time.

This is called a finite horizon LQR where $Q$ is a positive-definite (or positive-semi-definite) Hermitian or real symmetric matrix and $R$ is a positive-definite Hermitian or real symmetric matrix. The matrices $Q$ and $R$ determine the relative importance of the error and the dissipation of the energy. Therefore, if the unknown elements of the matrix $K$ are determined so as to minimize the performance function $J$, then $u = -Kx$ is optimal for any initial state $x(0)$. The block diagram showing the optimal configuration is shown in Figure 4.3. The feedback control law $u_{opt}(t)$ that minimizes the value of the cost is given as:

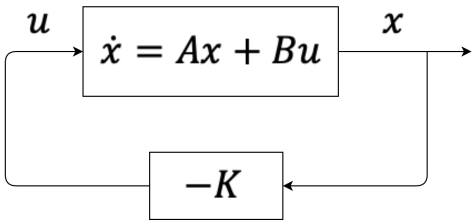



Figure 4.3: Optimal regulator system

$$u_{opt}(t) = -K(t)x(t) \tag{4.7}$$

where the $K(t)$ is the gain matrix determined by LQR controller optimally:

$$K(t) = -R^{-1}B^T P(t)$$

$P(t)$ is found by solving continuous time Riccati differential equation, numerically:

$$A^T P(t) + P(t)A - P(t)BR^{-1}B^T P(t) + Q = -\dot{P}(t) \tag{4.8}$$

with the boundary condition:

$$P(t_f) = F(t_f)$$

However a more simple case is when the matrices A,B,Q and R are constant together with an infinite horizon problem. The cost function and conditions now is defined as:

$$\begin{aligned} J^* = min_u \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) dt \\ \dot{x}(t) = Ax(t) + Bu(t) \\ x(0) = x_0 \end{aligned} \tag{4.9}$$

The system is consider in the steady state condition, therefore the solution can be considered as an Algebraic Riccati Equation (ARE) :

$$A^T P_{ss} + P_{ss} A - P_{ss} B R^{-1} B^T P_{ss} + Q = 0 \tag{4.10}$$

and

$$u_{opt}(t) = -Kx(t) \tag{4.11}$$

with

$$K(t) = -R^{-1} B^T P_{ss}$$

For the trajectory tracking problem, suppose given a system $\dot{x} = f(x, u)$ and a reference trajectory $(x_{ref}, y_{ref})$, the problem becomes to calculate a compensator of the form $u = K(x, x_{ref}, u_{ref})$ such that when $t$ goes to infinity, the position error is equals to zero.

## 4.5 Common Concepts

The design and evaluation of the control laws that solve the vehicle tracking problem requires the prior definition of some concepts. This section gathers all the common elements used hereafter.

### 4.5.1 Velocity transformation

The kinematic relations that allows the transformation of the control variables $(v, \beta, \omega)$ to modules velocities and orientation $(v_f, \theta_f, v_r, \theta_r)$ is proposed by Kelly and Seegmiller [9]. And in order to get output as $(\omega_{fr}, \omega_{fl}, \omega_{rr}, \omega_{rl})$, the kinematic relation of the module system is used. Once the relations between the modules velocities and vehicle velocities is obtained, the solution presented in this work can be used for RLVs with more than two modules.

According to Figure 4.4, to obtain the values of $(v_f, \theta_f, v_r, \theta_r)$, and meanwhile respecting the rigid body constraint

$$v_f \cos\theta_f = v_r \cos\theta_r$$

the following relations between the modules speed and vehicle speed must be verified:

$$\begin{bmatrix} v_{fx} \\ v_{fy} \\ v_{rx} \\ v_{ry} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & L_f \\ 1 & 0 & 0 \\ 0 & 1 & -L_r \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{4.12}$$

where

$$\begin{aligned} v_x &= v\cos\beta \\ v_y &= v\sin\beta \end{aligned} \tag{4.13}$$

$L_f$, $L_r$ is the distance of the front and rear module to the CG, respectively. The orientation can be obtained using the trigonometric relation:

$$\theta_i = \arctan 2\frac{v_{iy}}{v_{ix}} \tag{4.14}$$

and the linear velocity of the module:

$$v_i = \sqrt{{v_{ix}}^2 + {v_{iy}}^2} \tag{4.15}$$

The module is equipped with two wheels, and each wheel is attached with a motor. In body frame, the angular velocities of the wheels can be obtained from the modules speed as:

$$\begin{bmatrix} \omega_{fr} \\ \omega_{fl} \\ \omega_{rr} \\ \omega_{rl} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & \frac{D}{2} & 0 & 0 \\ 1 & -\frac{D}{2} & 0 & 0 \\ 0 & 0 & 1 & \frac{D}{2} \\ 0 & 0 & 1 & -\frac{D}{2} \end{bmatrix} \begin{bmatrix} v_f \\ \dot{\theta}_f \\ v_r \\ \dot{\theta}_r \end{bmatrix} \tag{4.16}$$

with $R$ the wheel radius, and $D$ the distance between two wheels. Hence, this transformation allows that

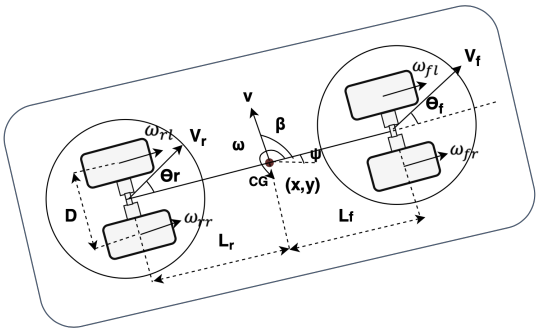



Figure 4.4: RLVs velocity transformation

motion control thought $(v, \beta, \omega)$, inside of $(\omega_{fr}, \omega_{fl}, \omega_{rr}, \omega_{rl})$.

### 4.5.2 Position errors

The position error concept is introduced to evaluate the performance of the path following and tracking controller. For the path following problem, the reference trajectory is defined as way-points, no time dependent. This means the errors $(\delta x, \delta y, \delta_\psi)$ are defined as the difference between the closet distances between the vehicle position $(x, y, \psi)$ and the reference trajectory $(x_{ref}, y_{ref}, \psi_{ref})$ as presented in Figure 4.5. Therefore, the position errors are defined as:

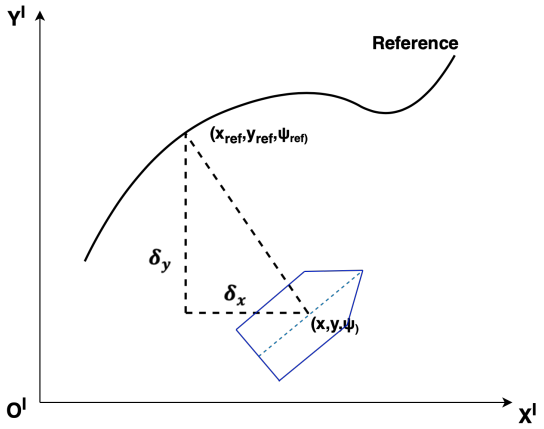



Figure 4.5: Errors definition

$$\begin{aligned} \delta_x &= x_{ref} - x \\ \delta_y &= y_{ref} - y \\ \delta_\psi &= \psi_{ref} - \psi \end{aligned} \tag{4.17}$$

The big difference between the path following and trajectory tracking problem, is in the path tracking problem, the reference trajectory in time-dependent variables. It can be given as function of the time or way-points with predefined velocities. Consider the vehicle is at position $(x(t), y(t), \psi(t))$, and the reference is given as $(x_{ref}(t), y_{ref}(t), \psi_{ref}(t))$. The position errors in the local frame is given as:

$$\begin{bmatrix} \delta_x(t) \\ \delta_y(t) \\ \delta_\psi(t) \end{bmatrix} = \begin{bmatrix} x_{ref}(t) - x(t) \\ y_{ref}(t) - y(t) \\ \psi_{ref}(t) - \psi(t) \end{bmatrix} \tag{4.18}$$

Both path following and trajectory tracking problem are studied in this work, and the conception of errors is used for the evaluation of the problem. The root mean square (RMS) of the position error is used for the comparison between the presented controllers.

### 4.5.3 Sensitivity and robustness to parameter uncertainty

In the described RLVs model, some of the parameter are likely to be uncertainty. And as the controller design is mainly based on the kinematic relations, the sensitivity and robustness of some of the param-

eter are performed to evaluated the controller performance.
The parameters for which some uncertainty is assumed are then:

- $M$- vehicle mass
- $L_f$ - distance of the front module to the vehicle CG
- $L_r$ - distance of the rear module to the vehicle CG
- $R$ - wheel radius

The $L_f$, $L_r$, and $R$ are geometric parameters used in the control design; therefore, it is essential to analyze how much the uncertainty can affect the controller performance. The $M$ is also an important parameter to study because the controller is a design based on the kinematic model, and how much the inertial influences the controller performance would be an interesting theme. The deterministic building model of the RLVs is used for the control law evaluation. The case study considers the controller's sensitivity and robustness to parameter uncertainty correspond to a circular trajectory with a radius of 4 m. For the baseline simulation, it is assumed no error in the model parameters. The simulation is repeated varying each of the above parameters for selected variables. The parameters vary one at a time to allow the evaluation of each one's influence with the comparison of the RMS of selected value in the simulations.

# Chapter 5

# Motion Control Implementation

## 5.1 Speed controller for the DC motor

For a motion control purpose, a speed controller in the motor level is needed to ensure the tracking features. The mathematical modelling of the DC motor is already presented in Section 3.2, the problem now is the determination of the controller parameters that satisfies the system requirements. Based on Equation (3.6), the transformation is done to take the ratio of $\frac{\omega_m(s)}{V(s)}$, with $V(s)$ the voltage source and $\omega_m(s)$ the motor angular speed, in the frequency domain. From Equation (3.6), the following equation is obtained:

$$R_e \frac{J\dot{\omega}_m(t) + c\omega_m(t)}{K_t} + L_e \frac{J\ddot{\omega}_m(t) + c\dot{\omega}_m(t)}{K_t} + K_b K_t \omega_m(t) = V(t) \tag{5.1}$$

where the time dependent variable now is no longer omitted.
A Laplace transform is defined as:

$$F(s) = \int_0^\infty f(t)e^{-st}dt \tag{5.2}$$

where $s$ is a complex number frequency parameter. Apply the (5.2) to (5.1), the transfer function of the DC motor is obtained:

$$\frac{\omega_m(s)}{V(s)} = \frac{K_t}{L_e J s^2 + (L_e c + R_e J)s + (R_e c + K_b K_t)} \tag{5.3}$$

Replacing the motor parameters in the Equation (5.3):

$$G(s) = \frac{\omega_m(s)}{V(s)} = \frac{1}{1.42 \times 10^{-7} s^2 + 7.46 \times 10^{-5} s + 9.20 \times 10^{-3}} \tag{5.4}$$

with the input to the DC motor system is voltage source $V$ and output of the angular speed $\omega_m$. Figure 5.1 shows the detailed implementation of the DC motor in *Simulink/Matlab* based on the transfer function presented above and later will be used for the speed controller design. The structure of the control system represented schematically in the vector form by the block diagram in Figure 5.2(a). The input to the system is the reference of the angular speed, with $V_{sat}$ the eventual voltage saturation of the DC motor. The speed controller implementation is based on the PID theory. In this case, the control law is

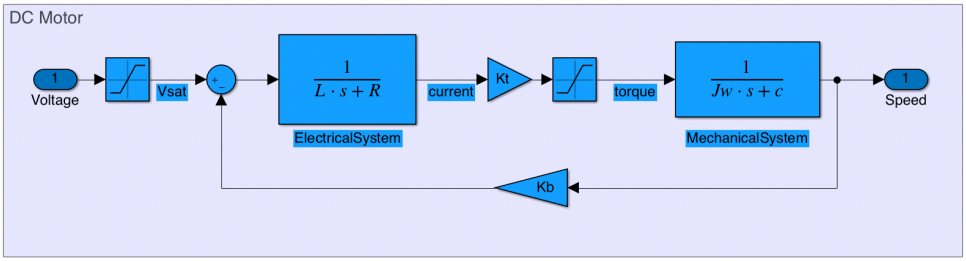



Figure 5.1: DC motor system implementation in *Simulink/Matlab*

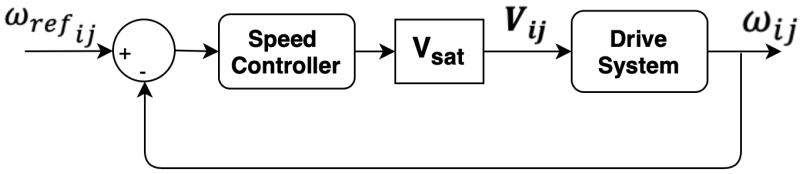
defined as:

$$\frac{V(s)}{E(s)} = Gc(s) = K_p + K_i\frac{1}{s} + K_d s \tag{5.5}$$

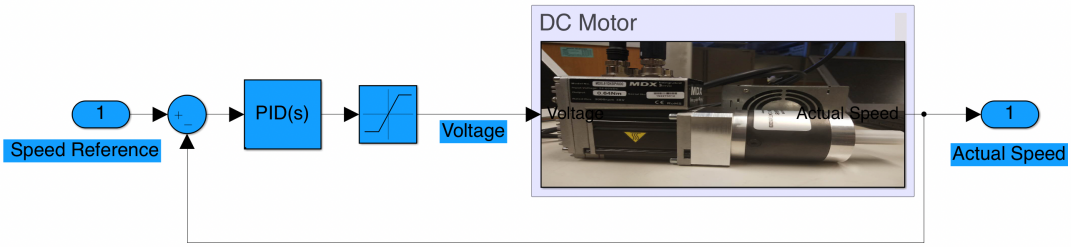
where the $E(s)$ is the speed tracking error:

$$E(s) = (\omega_{ref} - \omega)_{ij}$$

and $K_p$ is the proportional gain, $K_i$ is the integral gain, and $K_d$ is the derivative gain.
The illustrative implementation in the software *Simulink/Matlab* is showed in Figure 5.2(b). For the design




(a) Motor system with speed controller representation




(b) DC Motor with speed controller in*MATLAB/SIMULINK*

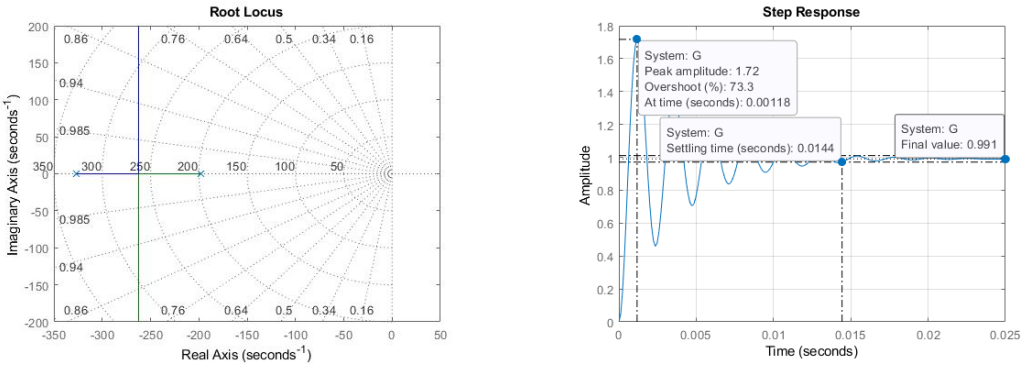
Figure 5.2: Drive system implementation

of the controller, the following design criteria are defined:

- Setting time less than 0.01 s
- Overshoot less than 5%
- Steady-state error less than 1%

The speed controller design via root locus method is performed, using the transfer function (5.4) derived in the previous section. First of all, an evaluation of the performance of system by drawing the root locus of the the closed-loop system with unit gain (see Figure 5.3(a)), and the closed-loop step response with

a unit gain is illustrated in Figure 5.3(b). Figure 5.3(a) shows the dominant poles are at $-326.43$ and




(a) Root locus of the the closed-loop system with unit gain

(b) Step response of the closed-loop system with unit gain

Figure 5.3: Speed controller design: system performance evaluation

$-198.25$. All poles reside in the left-half of the plane which means the system is stable. Moreover, the steady state error design requirements is satisfied with a unit gain as shown in Figure 5.3(b), however the overshoot is too large. Toward the system characteristics, in order to improve the system transient performance meanwhile not to compromise the steady state error, PID controller is desired. According to the design requirement, the Matlab Control System Design and Analysis APPs is used to tuning the controller parameter (see Figure 5.4). Final PID parameters are presented in Table 5.1. According to

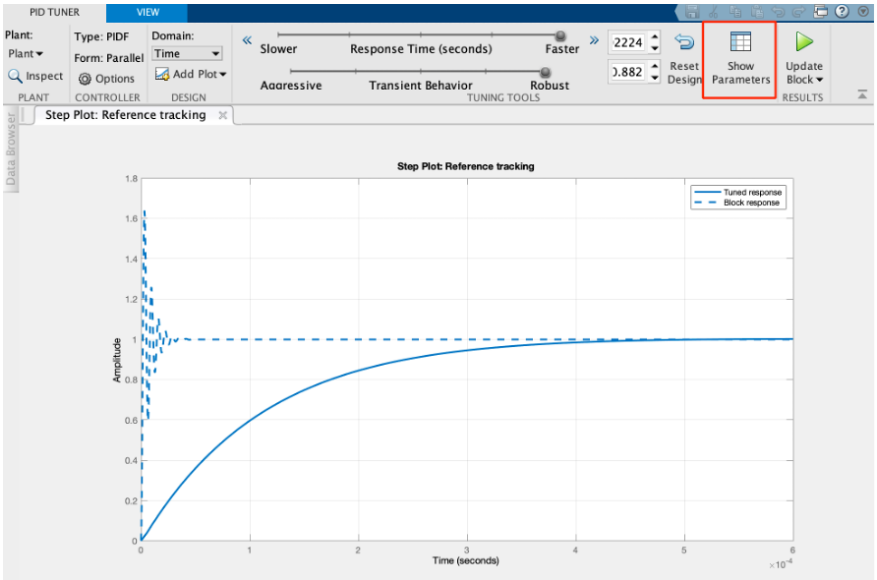



Figure 5.4: PID tuning in *Simulink/Matlab*

the Equation (5.5) and Table 5.1, the transfer function of the PID controller is obtained:

$$Gc(s) = K_p + K_i\frac{1}{s} + K_d s = 50 + 5\frac{1}{s} + 0.5s \tag{5.6}$$

Table 5.1: PID speed controller parameters

| Controller Parameters | Value |
|---|---|
| $K_p$ | 50 |
| $K_i$ | 5 |
| $K_d$ | 0.5 |

The system response with PID controller is presented in Figure 5.5 and all design criteria are satisfied. For the simulation, the reference speed for the motor with speed controller is $\omega_r = 500\,\mathrm{rad/s}$. The system

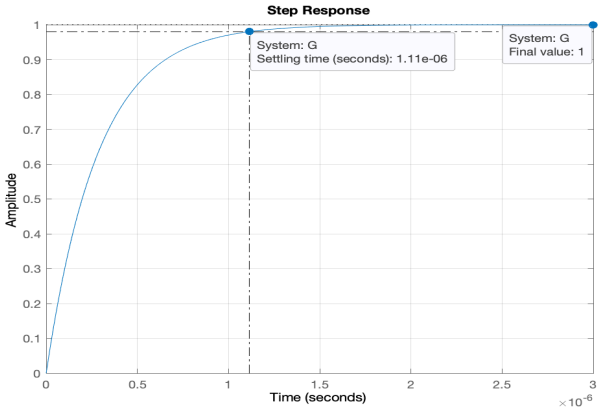
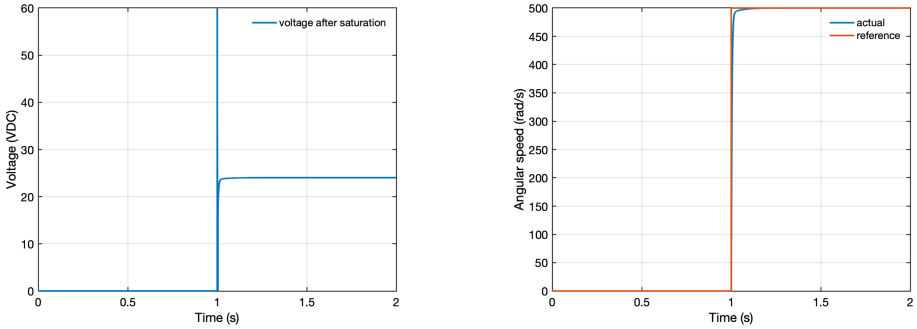



Figure 5.5: System step response with speed controller

response is presented in Figure 5.6(b) and the respect control actions after saturation is presented in Figure 5.6(a). The actuator saturation which results additional nonlinear element in the real application.




(a) Control Action (b) Speed response

Figure 5.6: PID speed controller simulation

The maximum speed allow in MDX serve motor is $\omega_r = 628.32\,\mathrm{rad/s}$, the respect control action and torques are limited according to system specifications.

### 5.1.1 Experimental Results

#### Simulation setup

The motor will be used is MDX Integrated Servo Motor, and it equips with a speed controller - PID with feedforward compensator. This section presents the simulation results of the speed controller in *Simulink*, and the experimental results in MDX Servo Suite. The comparison between them is analyzed. The MDX Servo Suite is an application provided by MDX Applied Motion company, and it is used to tuning the gained parameters before actual system operation in order to optimize the servo systems performance. The application standard configuration as well as the limitation of the motor is presented in Figure 5.7. The application provide the initial control parameters and a detailed explanation of the tuning

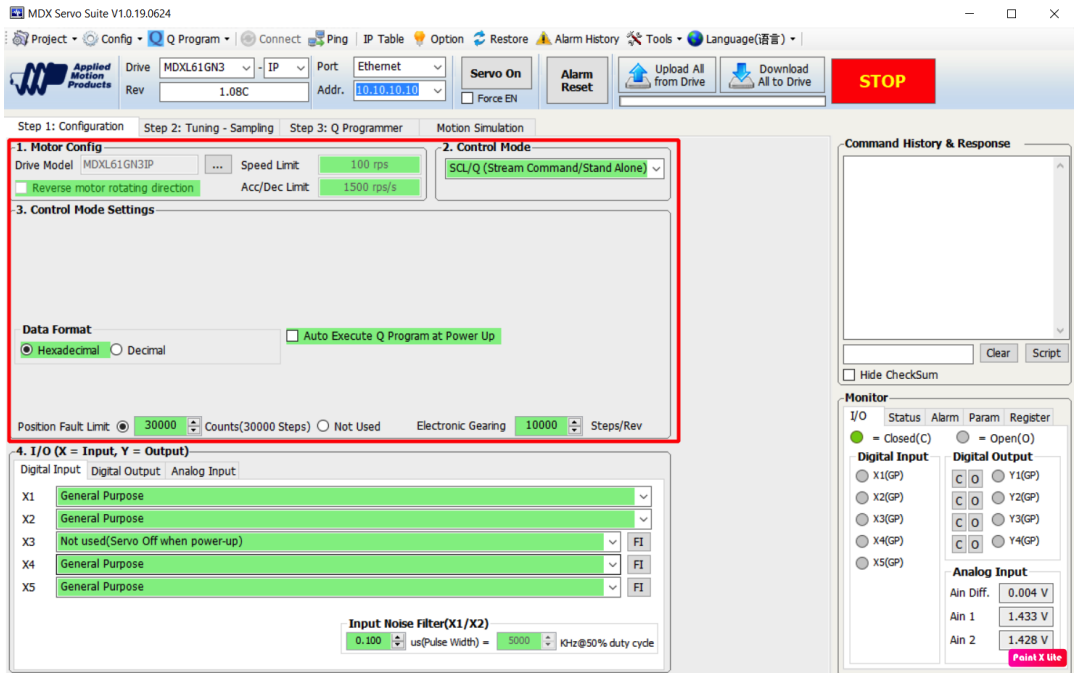



Figure 5.7: General configuration of MDX Servo Suite

process. For simplicity, each parameters is turned once at the time in order to get better performance. Figure 5.8 showed the final controller parameters in the left side, and in the right side is the speed plot where the red line is the reference speed and blue line is the actual speed. The speed signal of the motor presents significant noise, which may to related to the encoder. The simulation in *Simulink* using PID parameters in Table 5.1 and experimental results in MDX Servo Suite is presented in 5.10. The DC motor is model in noisy free condition, however during the tuning process in MDX Servo Suite, the presented noise can not be ignored. Regarding to get more realistic results as showed in Figure 5.10(b), additional noise was implemented to the motor model (see Figure 5.9). It is assumed the real angular speed $\omega_{real}$ have an uncorrelated zero mean noise although no bias is added, and can be presented as:

$$\omega_{real} = \omega + \sigma_{\omega}$$

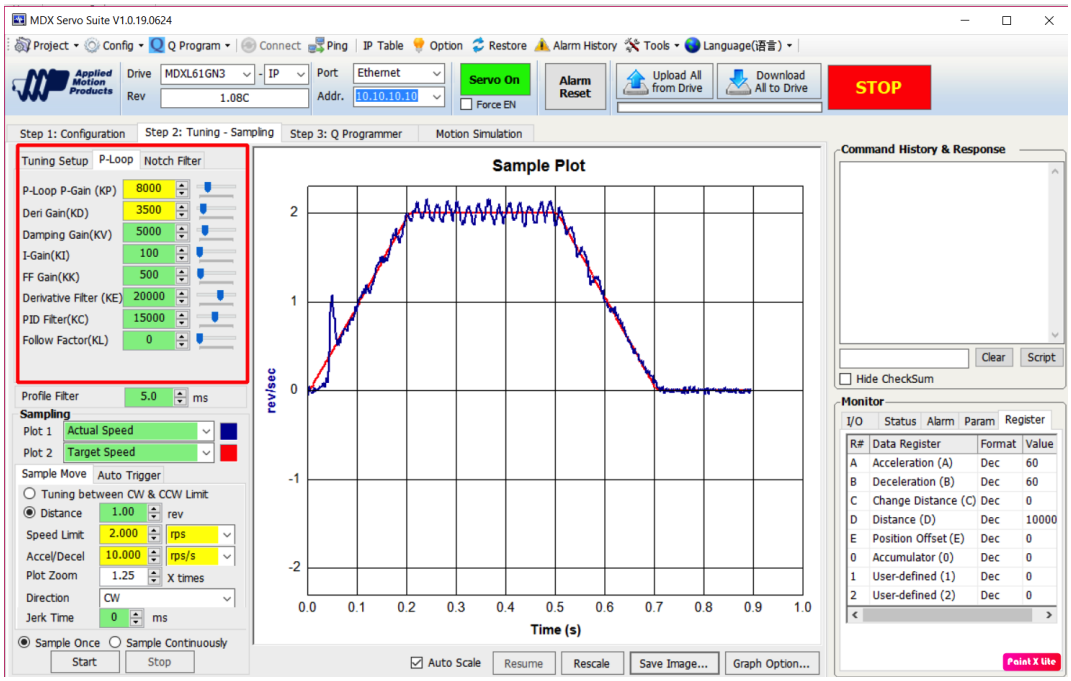



Figure 5.8: Tuned controller parameters in MDX Servo Suite

which $\omega$ is the actual speed and $\sigma_\omega$ is the random white noise [1]. The *Simulink* tool is used to model the noise, it requires the sample time of the sensor and the respective noise power, due to lack of the information, the estimation of those parameter is a iteration process based on the Figure 5.10(b). The

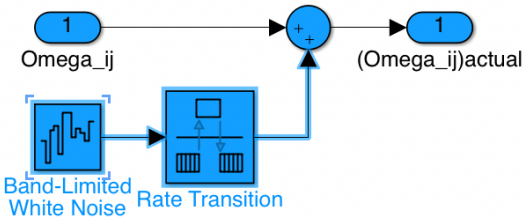
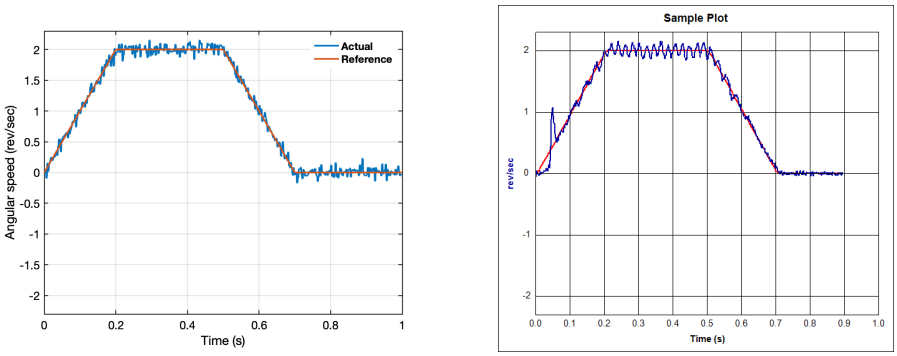



Figure 5.9: Noise implementation

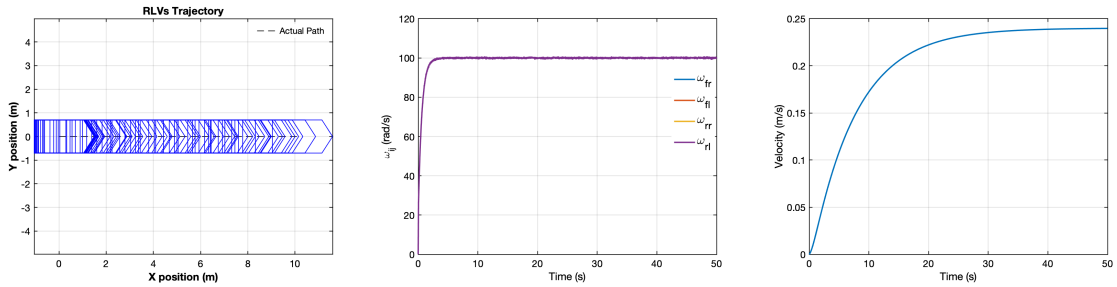
final result with implemented noise is presented in Figure 5.10. With the implemented speed controller for each motor, the system's input became four angular speed. For the simulation, the four angular speed is given as $\omega_{ref_{ij}} = 100\,\mathrm{rad/s}$. The resulting vehicle trajectory is presented in Figure 5.11(a). Four motors have the same angular velocity; therefore the vehicle intent to move in a straight line in the $x$-direction. The resulting motor speed is presented in Figure 5.11(b) with respective implemented noise. It can be seen in Figure 5.11(c) the resulting vehicle linear velocity starts to increase from zero and stabilized around $v = 0.24\,\mathrm{m/s}$. Hence the system is in equilibrium condition at which a dynamical system is steady

[1] White noise is a random signal having equal intensity at different frequencies, giving it a constant power spectral density




(a) Speed Control : Simulation Results (b) Speed Control: Experimental Results

Figure 5.10: Comparison between the simulation and the experimental results




(a) RLVs trajectory (b) Four motor angular speed (c) RLVs velocity

Figure 5.11: Simulation in RLVs simulator with speed controller

## 5.2 Alonzo Kelly Modified Controller

In this section, a geometric controller is presented with a good compromise between complexity and performance. Silva et al. [32] proposed the geometric controller based on the geometrical inspiration for a rhombic vehicle with two driven wheels. Based on this approach and principle presented in Section 4.3, a geometric controller is modified and extended to the RLVs in studied. It is named AKM controller because it is based on the Kelly and Seegmiller [9] study.

The main idea of the methodology is control the RLVs motion thought the control of $(v^c, \beta^c, \omega^c)$ (see Figure 5.12), which enable the front and rear modules follows the same path or different path. It requires only as input the desired path pose $(x_{ref}, y_{ref}, \psi_{ref})$ upon the control of $\beta^c$, $v^c$ and $\omega^c$. The output would be the angular velocities of the wheel $[\omega^c_{fr}, \omega^c_{fl}, \omega^c_{rr}, \omega^c_{rl}]$ that enable the RLVs follow the desired path.

The $v^c$ is assumed to be predefined and the speed information is contained in the reference trajectory. The control variable $\beta^c$ can be obtained using the geometric law as presented in Figure 5.12, and is used to direct the vehicle's to the reference pose. As for control variable $\omega^c$, it aims to orientating RLVs to the desired orientation. The control law for those two variables is presented as follows:

$$\beta^c = \arctan \frac{x_{ref} - x}{y_{ref} - y} - \psi \tag{5.7a}$$

$$\omega^c = \frac{k_\omega(\psi_{ref} - \psi)}{t_n} \tag{5.7b}$$

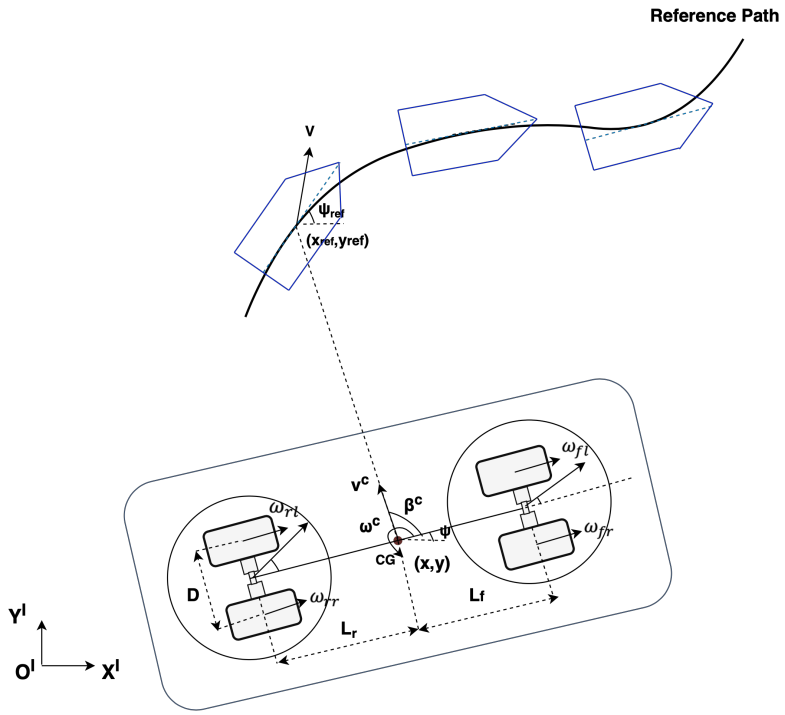



Figure 5.12: Alonzo Kelly Modified controller representation and variables definition

where the $(x_{ref}, y_{ref}, \psi_{ref})$ are the coordinate of reference path pose, and $(x, y, \psi)$ the actual coordinate of RLVs. $t_n$ the simulation step and $k_\omega$ a positive gain that tunes the RLVs to desired orientation. In brief, the AKM controller works as a proportional controller, where the control variables are proportional to the heading angle error and the cross track errors.

The implementation of AKM controller is straightforward. As mentioned in section 4.3, the use of look ahead distance can improve significantly the performance of the controller. The big challenge of implementation become how to deal with the path information and set a reasonable look ahead distance. For the path following problem, the path information can be represented as a set of discrete points and contained following information:

- $x$ location in global frame
- $y$ location in global frame
- $\psi$ heading angle in global frame

The $(x_{ref}, y_{ref}, \psi_{ref})$ appeared in the control law (5.7), is chosen as the closest point in the reference path. Therefore a treatment of the path information is required. The closest point in the path with in the vehicle actual pose is calculated, and the reference pose at that instant time is defined based on the closet point. One additional tuning parameters will be the choice of look ahead point. Longer look ahead distance cause the RLVs converge to the desired path more gradually with less oscillation. As for the trajectory tracking problem, the reference to the system is time dependent information or velocity parametric vector, hence the treatment of the path information is not necessary.

### 5.2.1 Simulation: case study

This section illustrates the simulation results of the AKM controller. The evaluation of the controller first tested using the kinematic model (see Figure 5.13.) with the kinematic model presented in (3.30), afterward a more precise simulation is done in the RLVs simulator which is presented in Section 3.5. For

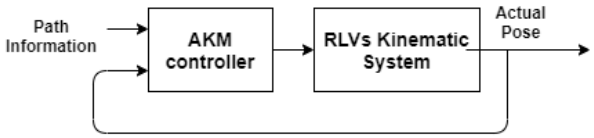



Figure 5.13: Alonzo Kelly Modified controller block diagram

the path following task simulation, as the RLVs will be operating the ITER with the cluttered environment, in order to simulate the work conditions, the reference path is designed (see Figure 5.14), where the RLVs starts in position $(0, 0)\mathrm{m}$ and stop at $(0, 5)\mathrm{m}$ and the desired speed of the RLVs is predefined $v^c = 0.1\,\mathrm{m/s}$.

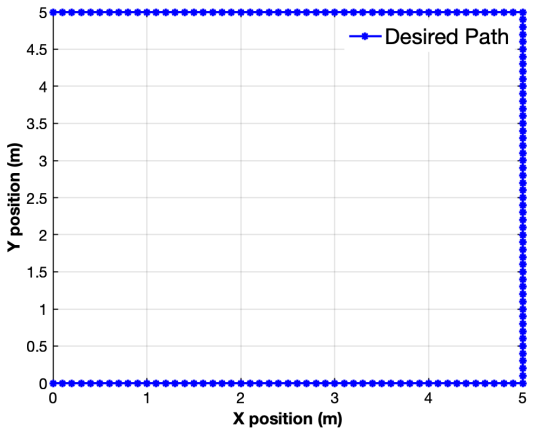
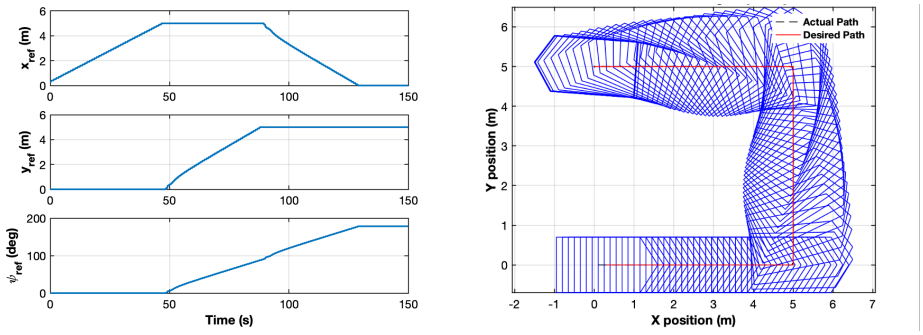
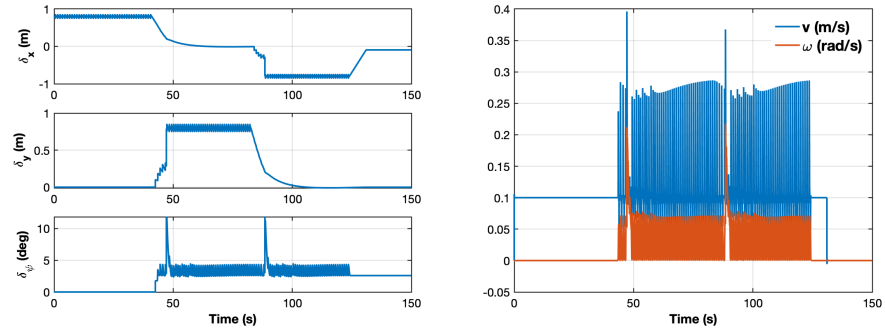



Figure 5.14: Reference path: directional path

The simulation is done in both the kinematic model and the simulator built in Section 3.5. Figure 5.15(a) shows the reference input for the AKM controller and one can notice there are nonuniform steps in the reference causes by the algorithms that calculate the closet point between the set of the reference point and the actual vehicle position. The resulting trajectory in Figure 5.15(b) showed with the AKM controller, RLVs track the path's geometric path with acceptable errors. And the stabilization of the system in the desired pose is easily reached. The oscillation of the position error in Figure 5.15(c) is mainly caused by introducing the look ahead distance. The 'error' here is the distance between the look ahead point and the actual pose. Figure 5.15(d) showed the RLVs oscillate around the desired speed $0.2\,\mathrm{m/s}$. The simulation with the kinematic model was able to capture the essential characteristics of the AKM controller. However, a more precise simulation is done with the simulator presented in chapter 3, which intends to evaluate how the vehicle inertial influences the controller's performance. The operating conditions are maintained for the simulation. Table 5.2 summarizes the controller's performance with the given reference in both two models. The position errors in $x$ and $y$ direction, denoted as $\delta_x$ and $\delta_y$,




(a) Reference input (b) Resulting trajectory




(c) Errors (d) Resulting vehicle speed

Figure 5.15: AKM controller: simulation in the kinematic model

respectively, and the orientation error $\delta_\psi$ are used for the evaluation of the controller performance. The simulation in the RLVs simulator is worst because the vehicle dynamic is not considered at the controller's design stage.

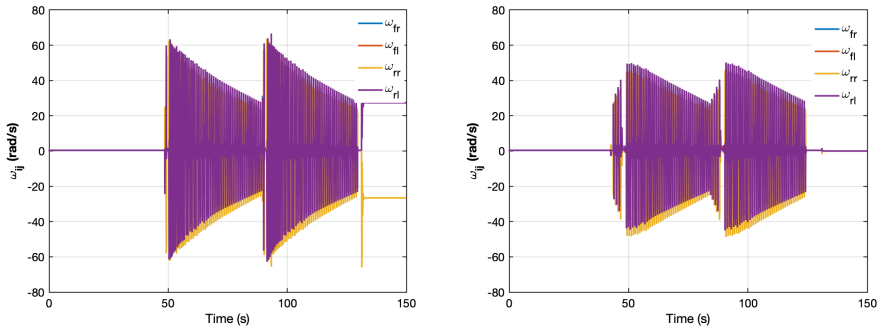
Table 5.2: AKM controller: simulation result (RMS values of selected variables)

| AKM controller | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi(deg)$ |
|---|---|---|---|
| Kinematic model | 0.59 | 0.40 | 2.79 |
| RLVs simulator | 0.87 | 0.68 | 6.63 |

The previous simulation's position error can be decreased if the look ahead distance is decreased. Table 5.3 demonstrates the position error decreased with the decrease of the look-ahead distance. However, a smaller look ahead distance can cause strong oscillations, and the system became challenging to stabilize. Figure 5.16 showed the RLVs's wheel speed for the look ahead distance of $0.2\,\mathrm{m}$ and look-ahead distance of $0.8\,\mathrm{m}$. With a smaller look ahead distance, the requested wheel speed has more oscillations. One principle drawback of this method is that without a high-quality speed controller, the constant fluctuation in the requested wheel speed (see Figure 5.15(d)) may cause damage to the system actuator.

Table 5.3: AKM controller: Influence of the look ahead distance

| Look ahead distance $(m)$ | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi(deg)$ |
|---|---|---|---|
| 0.2 | 0.15 | 0.10 | 2.58 |
| 0.5 | 0.38 | 0.26 | 2.67 |
| 0.8 | 0.87 | 0.68 | 6.63 |




(a) Look ahead distance = 0.2 m (b) Look-ahead distance = 0.8 m

Figure 5.16: Motor angular speed towards different look ahead distance

For the trajectory tracking problem, the simulation with the spiral path is detailed as follows, which intends to illustrate the essential point of the AKM controller facing a trajectory tracking problem. The spiral reference is presented in 5.17, where the trajectory equation is:

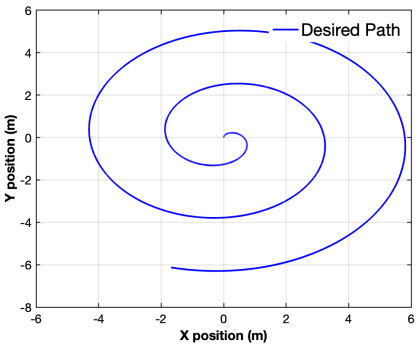
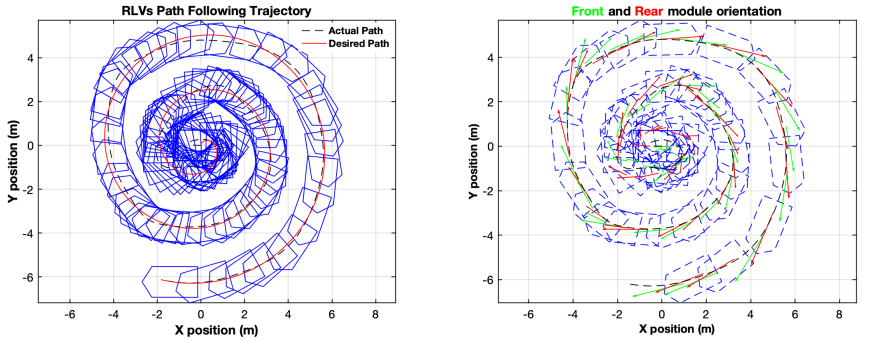



Figure 5.17: Reference trajectory: spiral

$$x_{ref} = 0.032t\sin(0.08t) \tag{5.8}$$
$$y_{ref} = 0.032t\cos(0.08t)$$

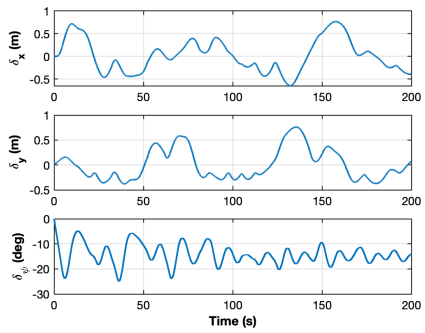
with sampling time of $0.01\,\mathrm{s}$. Figure 5.18(a) shows the RLVs were able to follow the desired path. The module orientation is presented in Figure 5.18(b), once the vehicle is in the right direction the variation in the orientation of the modules became smaller. The AKM controller was able to follow the desired



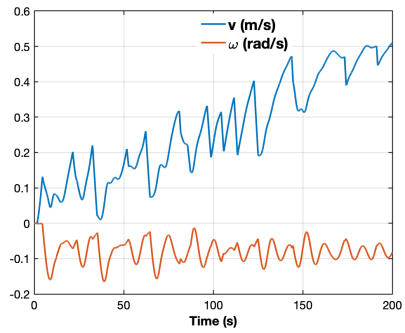

(a) Resulting trajectory




(b) Modules orientation




(c) Errors

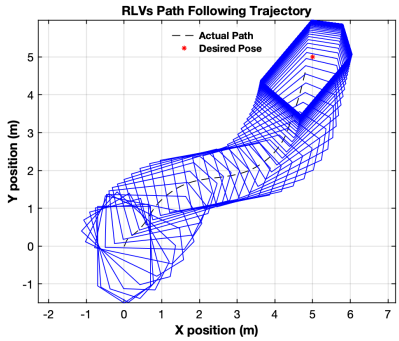



(d) Resulting vehicle speed

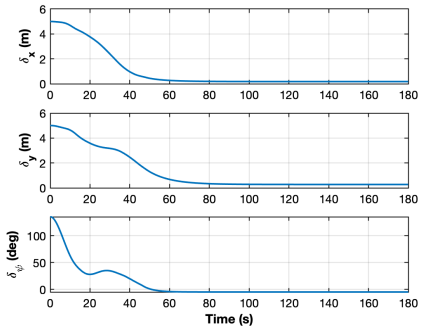
Figure 5.18: AKM controller: spiral

reference with $0.76\,\mathrm{m}$ safety margin in the $(x, y)$ position, and $25°$ in the heading angle (see Figure 5.18(c)). The resulting linear and angular velocity is presented in Figure 5.18(d).

For the pose tracking problem, the study is concentrated on stabilizing the RLVs system to a static point. The RLVs initial position is given as $(0, 0, -90°)$ and the reference pose is set to $(5, 5, 45°)$. The simulation results are presented in Figure 5.19(b). The system stabilized around $t = 80\,\mathrm{s}$.




(a) Resulting trajectory




(b) Errors

Figure 5.19: AKM controller: pose tracking

**Conclusion**

This chapter presents the AKM controller applied to the RLVs system. After the simulation with a different type of path, how the controller parameter influences the system is studied. The AKM controller is sufficient for the short-term goal such as pose tracking and path following problem, but there is no guarantee the solution is optimal, and the performance of the controller relied on the delicate tuning of control parameters. The simulation result in RLVs simulator is worse than the simulation in the kinematic model because the system with dynamics does not respond as quick as possible to a high speed requested by the controller. Although it has relatively good simulation results since the operating speed is low. A more agile methodology is proposed in the next section based on the optimal control theory, which calculates the desired linear and angular velocity optimally.

## 5.3 Optimal Control-Linear Quadratic Regulator

A LQR optimal control for trajectory tracking of RLVs is developed in this section. For the study case, the solution calculates a velocity variation that should be followed for a reference signal. As explained in Chapter 4, the motion control is based on the hierarchical approach. Therefore, the output of the solution should be the four angular speed reference. In the following, the procedure applied on the linearization of the system around the operating point is described.

### 5.3.1 Model Linearization

For the given system:

$$\dot{x} = f(x, u)$$

the linearization of the system corresponds to the first-order term of its Taylor expansion in the interest point $(x_e, u_e)$ is:

$$\dot{x} = f(x_e, u_e) + \frac{\partial f}{\partial x}(x_e, u_e)(x - x_e) + \frac{\partial f}{\partial u}(x_e, u_e)(u - u_e) \tag{5.9}$$

Substituting the jacobian matrices, one gets:

$$A = \frac{\partial f}{\partial x}(x_e, u_e) \tag{5.10}$$

$$B = \frac{\partial f}{\partial u}(x_e, u_e) \tag{5.11}$$

and the variation around the equilibrium point is given as:

$$\tilde{x} = x - x_e \tag{5.12}$$

$$\tilde{u} = u - u_e \tag{5.13}$$

One gets the linear model as:

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u} \tag{5.14}$$

For the trajectory tracking problem, suppose the system state is given as $q = (x, y, \psi)$, and the reference trajectory is $q_{ref} = (x_{ref}, y_{ref}, \psi_{ref})$. The goal is design a controller that $\lim_{t\to\infty}(q_{ref} - q) = 0$, to do so, the error system is construct based on the rhombic like vehicle kinematic model is presented above. Those errors in the body frame can be expressed as follows:

$$\begin{bmatrix} \tilde{q}_x \\ \tilde{q}_y \\ \tilde{q}_\psi \end{bmatrix}_B = \begin{bmatrix} x_{ref} - x \\ y_{ref} - y \\ \psi_{ref} - \psi \end{bmatrix} \tag{5.15}$$

A rotation matrix has been applied to obtain the error in the inertial frame $(\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi)_I$:

$$\begin{bmatrix} \tilde{q}_x \\ \tilde{q}_y \\ \tilde{q}_\psi \end{bmatrix}_I = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_\psi \end{bmatrix}_B \tag{5.16}$$

The linearization of the kinematic model (3.28) is done based on (5.9), however the system can be extended once variable $\beta$ is determinable.

Hence, the dynamic of state tracking error based on (5.15) and (3.28) is expressed as:

$$\begin{bmatrix} \dot{\tilde{q}}_x \\ \dot{\tilde{q}}_y \\ \dot{\tilde{q}}_\psi \end{bmatrix}_B = \begin{bmatrix} \dot{x}_{ref} - \dot{x} \\ \dot{y}_{ref} - \dot{y} \\ \dot{\psi}_{ref} - \dot{\psi} \end{bmatrix} = \begin{bmatrix} \tilde{v}\cos\tilde{\psi} \\ \tilde{v}\sin\tilde{\psi} \\ \tilde{\omega} \end{bmatrix} \tag{5.17}$$

with $\tilde{v} = v_{ref} - v$ and $\tilde{\omega} = \omega_{ref} - \omega$.

Linearization of the state space model form of the system around the point of interest, in this case is a set of reference that enable the $\tilde{q}_I = 0$ is performed which the equilibrium point are denoted by:

$$\begin{aligned} v &= v_{ref} \\ \omega &= \omega_{ref} \\ \psi &= \psi_{ref} \end{aligned} \tag{5.18}$$

Applying a first order Taylor approximation around the equilibrium point as follows:

$$\dot{\tilde{q}}_I = \frac{\partial f}{\partial q}(x^{ref}, u^{ref})\tilde{q}_I + \frac{\partial f}{\partial u}(x^{ref}, u^{ref})\tilde{u} \tag{5.19}$$

thus the tangent linearization of system about the reference trajectory in the global frame can be outlined as:

$$\dot{\tilde{q}}_I = \begin{bmatrix} 0 & \omega_{ref} & 0 \\ -\omega_{ref} & 0 & v_{ref} \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_I + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} = A\tilde{q}_I + B\tilde{u} \tag{5.20}$$

Detailed calculation is presented in Appendix A.

**Controllability study**

In the control problem, it is important to find out whether or not a system is controllable. A controllability about a trajectory of a derived linearized system is studied in this section. The linerized system (5.20) is time varying, a necessary and sufficient controllability is the controllability Gramian is non-singular [38], where the controllability matrix $C$ is defined as follows:

$$C = \begin{bmatrix} A & AB & A^2B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_{ref}^2 & v_{ref}\omega_{ref} \\ 0 & 0 & -\omega_{ref} & v_{ref} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.21}$$

The linear system is controllable if the matrix $C$ has same rank of matrix $A$, in this case $3$. This provided either $v_{ref}$ or $\omega_{ref}$ are nonzero.

For the tracking problem, with the reference trajectory defined as $(x_{ref}, y_{ref}, \psi_{ref})$, the $v_{ref}$ and $\omega_{ref}$ can be calculated from following equations according to Abbasi and Moshayedi [30]:

$$\psi_{ref} = \arctan \frac{\dot{y}_{ref}}{\dot{x}_{ref}} \tag{5.22}$$

having differentiation of (5.22) in order to get $\omega_{ref}$ as:

$$\omega_{ref} = \frac{\dot{x}_{ref}\ddot{y}_{ref} - \dot{y}_{ref}\ddot{x}_{ref}}{\dot{x}_{ref}^2 + \dot{y}_{ref}^2} \tag{5.23}$$

$$v_{ref} = \pm\sqrt{\dot{x}_{ref}^2 + \dot{y}_{ref}^2} \tag{5.24}$$

The sign for $v_{ref}$ will define forward or backward motion of vehicle. The desired cartesian motion reference $(x_{ref}, y_{ref})$ should be twice differential according to the Equation (5.23).

The system has three state variables $[\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi]$, representing the dynamics of tracking error in coordinate $(x, y, \psi)$, and two input $[\tilde{v}, \tilde{\omega}]$. In order to determine inputs of the closed-loop system the LQR optimal control is used. According to the LQR control theory, the solution can be obtained as

$$u_{opt}(t) = -K\tilde{x}(t)$$

where $K$ is the gain matrix determined by LQR controller optimally. The Bryson method is used for the first estimation of the weighting matrices of the states and inputs, respectively $Q$ and $R$, as diagonal

matrices where each term is the inverse square of the expected maximum for the variable during the motion:

$$\begin{aligned} Q &= diag(Q_i) \\ Q_i &= \frac{1}{q_{i,max}^2} \end{aligned} \tag{5.25}$$

with $q_{i,max}$ maximum acceptable value of $q_i$ in SI unit. According to the ITER project specification, the vehicle has to move in cluttered environments with safety margins of 30 cm, therefore the maximum accepted value for the $\tilde{q}_x$ and $\tilde{q}_y$ are approximately $0.3\,\mathrm{m}$, and $0.15\,\mathrm{radius}$ for the vehicle orientation. The final value is presented below based on the (5.25):

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 50 \end{bmatrix} \tag{5.26}$$

Same principle is used for the matrix $R$:

$$\begin{aligned} R &= diag(R_i) \\ R_i &= \frac{1}{u_{i,max}^2} \end{aligned} \tag{5.27}$$

with $u_{i,max}$ max acceptable value of $u_i$.

$$R = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.28}$$

By changing the elements of $Q$, the sensitivity of the system to the state variables can be adjusted. Therefore, to obtain the inputs of the system, the Equation (5.29) is available:

$$\begin{bmatrix} v^c \\ \omega^c \end{bmatrix} = -K \begin{bmatrix} \tilde{q}_x \\ \tilde{q}_y \\ \tilde{q}_\psi \end{bmatrix}_I \tag{5.29}$$

with $K$ the gain matrix determined by LQR controller optimally. The system is extended with determination of $\beta^c$ (5.7a) and one can get the four angular speed $\omega_{ij}$ using the velocity transformation presented in 4.5.1.

### 5.3.2 Simulation: case study

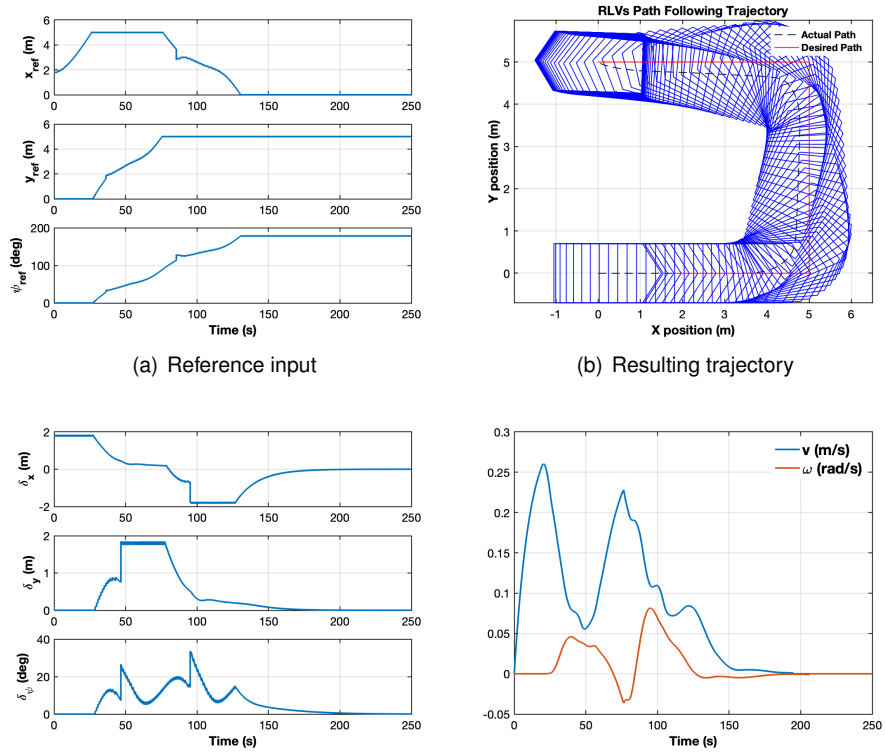
This section presents an illustrative simulation results of the RLVs linear controller based on the linear model. As mentioned above, RLVs are supposed to operate in a cluttered environment. Therefore the first simulation is done with the reference path (see Figure 5.14) used in the previous section, where the vehicle starts in $(0, 0)m$ and stops in $(0, 5)m$. The state $Q$ and input $R$ weighting matrices are set as (in

SI units):

$$Q(\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi) = diag(10, 10, 50) \tag{5.30}$$

$$R(\tilde{v}_{ref}, \tilde{q}_{ref}) = diag(5, 1)$$

A look ahead distance is also introduced as a tuning parameter, which cause steps in the reference input (see Figure 5.20(a)). The $v_{ref}$ is set as $0.2\,\mathrm{m/s}$, the variation of vehicle speed is mainly related with the error at the instant of simulation (see Figure 5.20(d)). The deviation of the vehicle trajectory from the reference trajectory in Figure 5.20(b) can be decreased by tuning the element of $Q$ matrix.



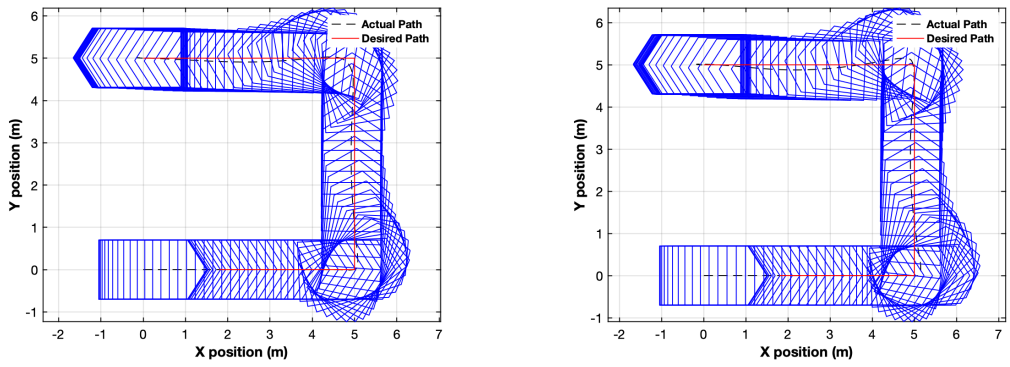

(a) Reference input

(b) Resulting trajectory

(c) Errors

(d) Resulting vehicle speed

Figure 5.20: LQR controller: Path following with look ahead distance




(a) $Q(\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi) = diag(50, 50, 50)$

(b) $Q(\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi) = diag(100, 100, 50)$

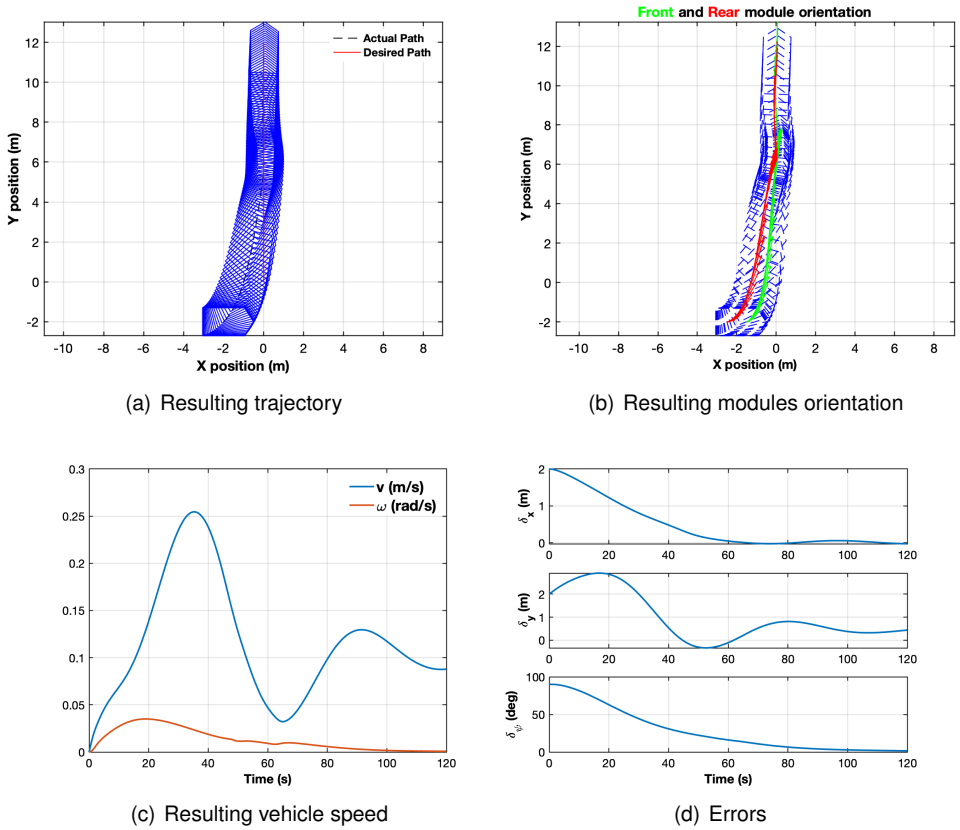
Figure 5.21: Effect of weighting matrices $Q$

Figure 5.21(a) and Figure 5.21(b) shows the resulting trajectory by increasing the values of the element in matrix $Q$. The static error decreases with the increase of matrix $Q$ elements (see Figure 5.21), and RLVs also needs less space to perform the assigned task in this circumstance. However if the value is too large, the system will begin to oscillate (see Figure 5.21(b)). Therefore, the value of matrix $Q$ should be chosen with a reasonable compromise between the stability and accuracy.

For the trajectory tracking problem, the first simulation intent to simulation the RLVs goes to the straight line in the $y$-direction, where the reference is defined as:

$$\begin{aligned} x_{ref} &= 0 \\ y_{ref} &= 0.1t \\ \psi_{ref} &= 90° \end{aligned} \tag{5.31}$$

with sample time $0.01\,\mathrm{s}$. The vehicle starts at $(-2, -2)\mathrm{m}$ and the reference trajectory starts in $(0, 0)\mathrm{m}$. The reference velocity $v_{ref} = 0.1\,\mathrm{m/s}$ is calculated based on the reference trajectory according to Equation (5.24). Figure 5.22(a) shows the resulting trajectory of the RLVs and how the vehicle enters the reference trajectory.




(a) Resulting trajectory

(b) Resulting modules orientation

(c) Resulting vehicle speed

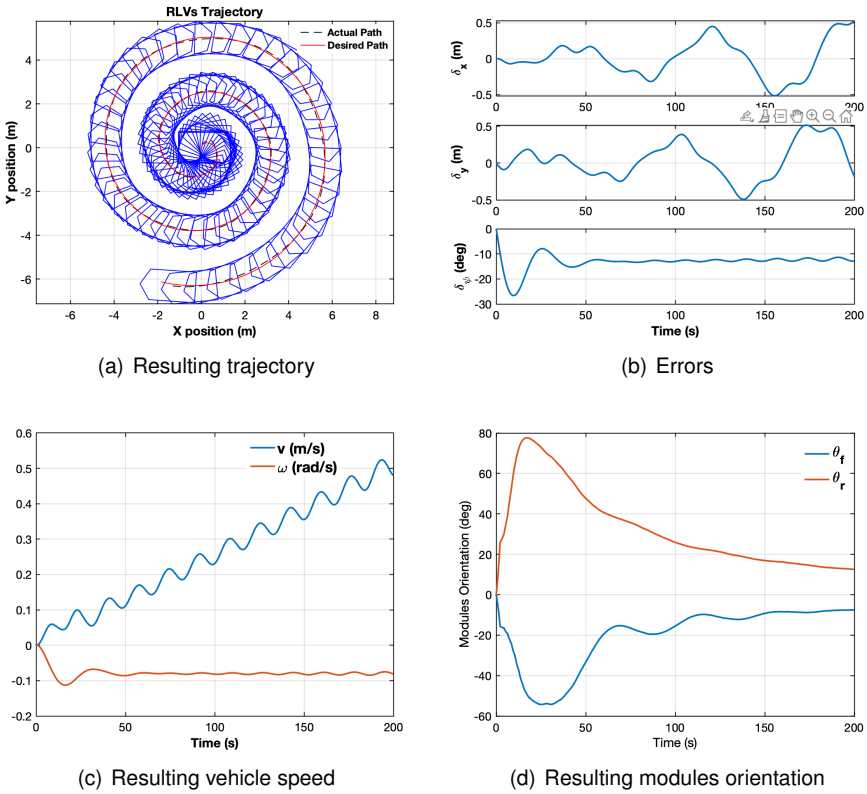
(d) Errors

Figure 5.22: LQR controller: trajectory $y$-direction

The vehicle enters the reference trajectory after it runs $6\,\mathrm{m}$ in the $y$-direction. The position in $y$ is well followed with an error inferior to $0.5\,\mathrm{m}$ after the reference trajectory entrance. The module orientation stabilized when the vehicle is in the right heading angle; the resulting module orientations are presented

in Figure 5.22(b). The reference speed is $v_{ref} = 0.1\,\mathrm{m/s}$; however, as the vehicle start at the position $(-2, -2)\mathrm{m}$, as the controller calculate the rate of the velocities based on the error signal, for more significant errors, the controller intent increase the speed up to $v_{ref} = 0.25\,\mathrm{m/s}$ (see Figure 5.22(d)), after the entrance to the reference trajectory, the velocity closes to the reference velocity as expected.

The simulation with the spiral path (5.8) is done to evaluate the controller performance toward different speed reference. Figure 5.23(a) shows the trajectory described by the RLVs system. The position error




(a) Resulting trajectory

(b) Errors

(c) Resulting vehicle speed

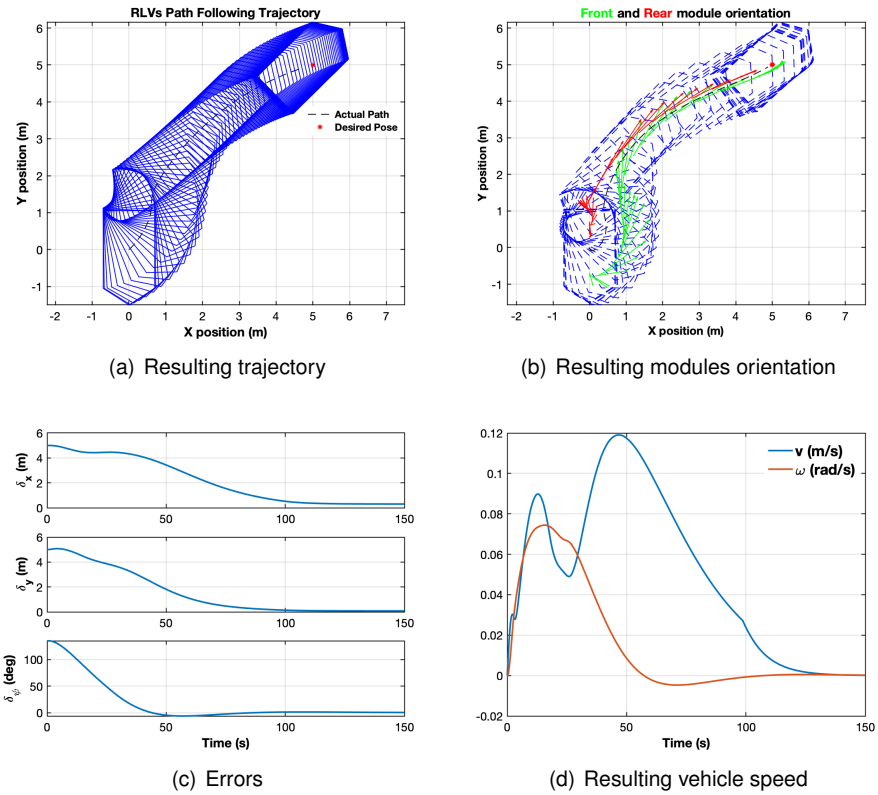
(d) Resulting modules orientation

Figure 5.23: LQR controller: spiral

in $x$ and $y$ increase when the vehicle speed increase (see Figure 5.23(b)). Compared with the simulation results of the AKM controller, the trajectory described by vehicle with LQR controller is smother and has smaller position error (around $0.5\,\mathrm{m}$).

Two pose tracking problems are now studied to investigate the stabilization feature and the relation between the module's orientation and the vehicle heading angle. The linear system is controllable if neither $v_{ref}$ nor $\omega_{ref}$ is non zero value. The system stabilizes when the error coordinate is close to zero. The first reference for the system are position $(5, 5)\mathrm{m}$ with $45°$ heading angle. The chosen initial conditions for the RLVs is $(0, 0)m$ with $-90°$ orientation, and the $v_{ref}$ is set to a small value $v_{ref} = 0.1\,\mathrm{m/s}$. The system stabilized around $125\,\mathrm{s}$ (see Figure 5.24(c)). For more significant speed, the stabilization takes more time. The module's orientation coincides with the vehicle heading angle (see Figure 5.24(b)). The oscillations at the begging of the vehicle linear speed in Figure 5.24(d) is because the controller intends to correct first the heading angle.

For the second case study, the desired pose is position $(5, 5)\mathrm{m}$ with zero heading angle. The chosen initial conditions for the RLVs is $(0, 0)\mathrm{m}$ with zero heading angle. This case study aims to illustrate how




(a) Resulting trajectory

(b) Resulting modules orientation

(c) Errors

(d) Resulting vehicle speed

Figure 5.24: LQR controller: initial conditions $(x, y, \psi) = (0, 0, -90°)$; Desired Pose: $(5, 5, 45°)$

the vehicle can slide to the desired position without changing the vehicle heading angle. The vehicle stabilized in the desired position around $100\,\mathrm{s}$ (see Figure 5.26(a)). The maneuver is possible with the combination of the two modules steering configuration presented in Figure 5.25.

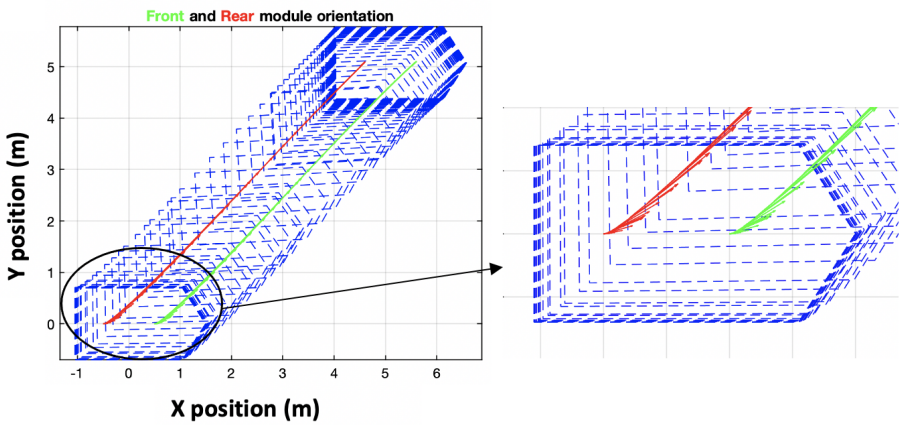
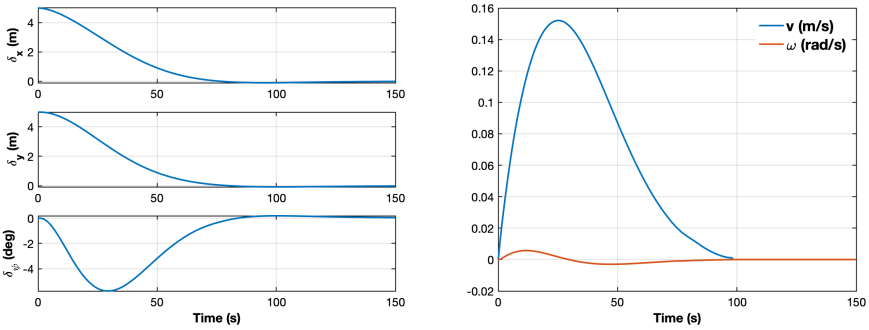



Figure 5.25: LQR controller: initial conditions $(x, y, \psi) = (0, 0, 0)$; desired pose: $(5, 5, 0)$




(a) Errors (b) Resulting vehicle speed

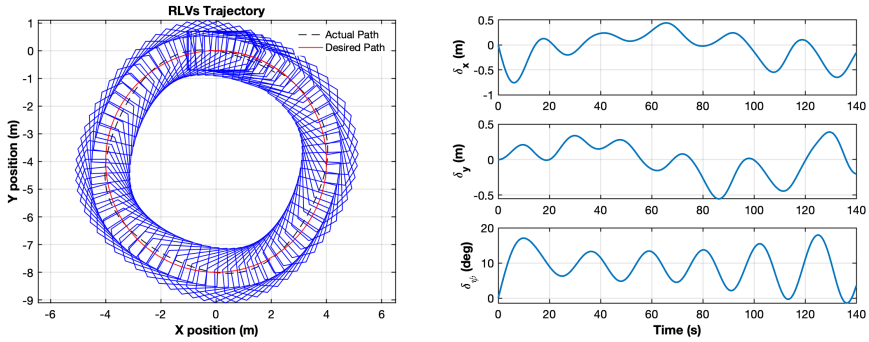
Figure 5.26: LQR controller: initial conditions $(x, y, \psi) = (0, 0, 0)$; desired pose: $(5, 5, 0)$

### 5.3.3 Study of the robustness of the LQR controller

In this section, a study of how the model parameters' variation on the controller performance is presented. For the baseline simulation, there is no variation of the model parameters. The simulation's case mission is considered a circle path, with the null initial conditions. The reference is defined as:

$$\begin{aligned} x_{ref} &= 4\sin(0.05t) \\ y_{ref} &= 4\cos(0.05t) - 4 \end{aligned} \tag{5.32}$$

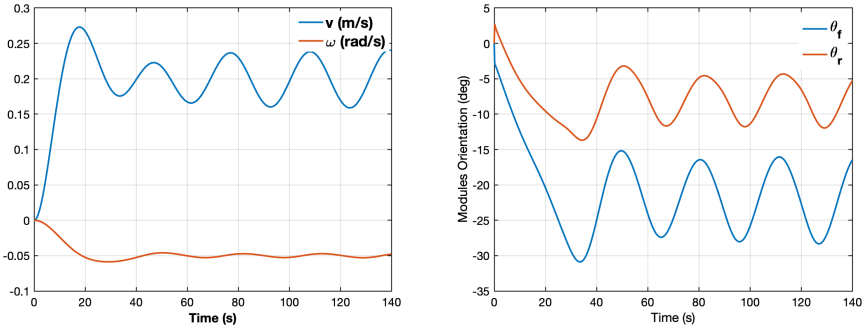
According to the Equation (5.24) and (5.23), the reference velocities are calculated, where the $v_{ref} = 0.2\,\mathrm{m/s}$ and $\omega_{ref} = 0.05\,\mathrm{rad/s}$ clockwise. Figure 5.27(a) is the resulting trajectory of the vehicle, and the vehicle follows the reference trajectory with the minimum position error (see Figure 5.27(b)). The




(a) Resulting trajectory (b) Errors

Figure 5.27: Baseline simulation for the study of the robustness of the LQR controller: resulting trajectory and tracking error

resulting vehicle velocities are presented in Figure 5.28(a), where the linear speed $v$ oscillated around the $v_{ref}$ and angular speed approximated to the $\omega_{ref}$. As the angular speed reaches the desired value, the module orientation variation becomes constant (see Figure 5.28(b)). Several simulations are done with the variation of the vehicle mass in the model. Figure 5.29 showed the position and orientation errors increase with the increase of the vehicle mass. This is due to the fact of the LQR controller is designed based on the kinematic model, and the controller performance deteriorated with the increase




(a) Resulting vehicle speed (b) Resulting modules orientation

Figure 5.28: Baseline simulation for the study of the robustness of the LQR controller: resulting speed and module orientations

of the mass. As the mass increases, the vehicle's inertial effect becomes significant, and the system can not respond to the request control action as quickly as it is supposed to be, which causes the controller performance to become worse. Table 5.4 summarized the RMS value of selected variables (position

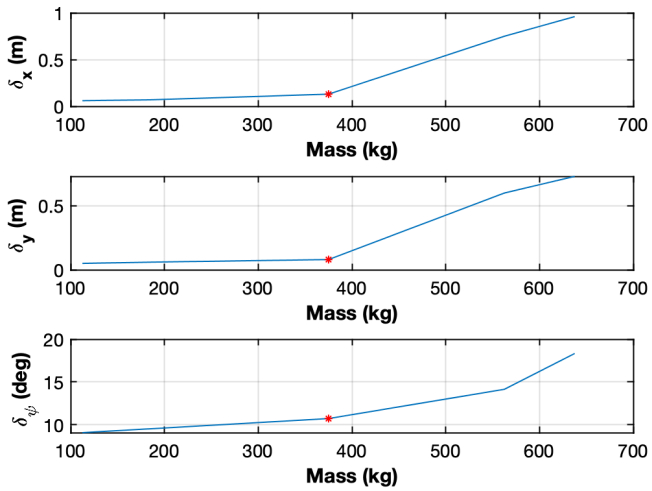



Figure 5.29: Influence of the vehicle mass: red mark represents the baseline simulation

errors and orientation errors) with different vehicle mass. The RMS values of the vehicle with mass of $375\,\mathrm{kg}$ is the baseline case and serves as a reference. The deviation of $+70\%$ of the vehicle mass, the position error is up to $0.96\,\mathrm{m}$ which is out of the acceptable margin of the work condition, thus the LQR is consider to be robust to $50\%$ variation of the mass parameter.

Table 5.4: Robustness tests on Mass of the RLVs (RMS values of selected variables)

| **Mass** | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi$(deg) |
|---|---|---|---|
| $M = 375\,\mathrm{kg} * 30\%$ | 0.06 | 0.05 | 9.05 |
| $M = 375\,\mathrm{kg} * 50\%$ | 0.07 | 0.06 | 9.52 |
| $M = 375\,\mathrm{kg}$ Baseline | 0.13 | 0.08 | 10.70 |

Table 5.4: Robustness tests on Mass of the RLVs (RMS values of selected variables)

| **Mass** | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi$(deg) |
|---|---|---|---|
| $M = 375\,\text{kg} * 150\%$ | 0.75 | 0.60 | 14.13 |
| $M = 375\,\text{kg} * 170\%$ | **0.96** | **0.73** | **18.33** |

How the variation of the wheel radius influence the controller performance is also studied. According to the vehicle specification in Table 3.2, the wheel radius is $R = 0.1\,\text{m}$. Several simulations are done with the variation of the wheel radius in the controller. LQR controller demonstrates to be robust to a $\pm 0.02m \approx 20mm$ in the wheel radius parameter as presented in Table 5.5. The difference more significant than $20mm$ in the controller's design loses control of the RLVs system ( RMS value of $\delta_\psi$ is up to $40°$) which means the mismatch between the wheel's diameter of the model and the control project can not vary more the $40mm$. Figure 5.30 showed the variation of the wheel radius effects more in the position $x$ and $y$ than the vehicle orientation. This kind of mismatch can be caused by various reasons, such as tire wear or measurement errors. In reality, the tire's wear can be a severe problem, and depends on the type of tire will be used, the scale of variation can be considerable. However, as the wheel radius parameter is a measurable variable, and generally speaking, for a large and heavy-duty vehicle, the maximum tire wear permitted is around $60mm$[2], including the eventual measurement error, the LQR is considered to be robust to $50\%$ variation of the wheel radius parameter.

Table 5.5: Robustness tests on wheel radius of the RLVs (RMS values of selected variables)

| **Wheel radius** | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi$(deg) |
|---|---|---|---|
| $R = 0.06\,\text{m}$ | 0.36 | 0.55 | 27.96 |
| $R = 0.08\,\text{m}$ | 0.24 | 0.17 | 9.13 |
| $R = 0.1\,\text{m}$ Baseline | 0.13 | 0.08 | 10.70 |
| $R = 0.12\,\text{m}$ | 0.32 | 0.47 | 14.65 |
| $R = 0.14\,\text{m}$ | **0.47** | **0.35** | **39.73** |

The $L_f$ and $L_r$ are the distance of the front and rear module to the vehicle CG, respectively. They are parameters used in the velocity transformation during the control design. The methodology used in the wheel radius applies to the $L_f$ and $L_r$ robustness study. Simulation result shows the influence of $L_f$ and $L_r$ are similar, therefore only the $L_r$ result is presented in Table 5.6. The variation of the $L_r$ influence the performance of the controller-less than $0.1\text{m}$ in the $\delta_x$ and $\delta_y$, and less than $3°$ in the $\delta_\psi$. Compare to the

[2]Information provided by Technical Engineer in CRRC Tangshan Co., Ltd., a manufacturer of rolling stock located in Tangshan, Hebei province,China

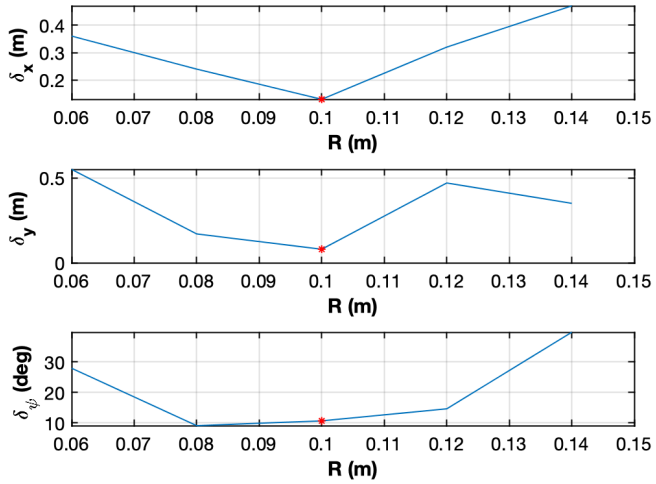



Figure 5.30: Influence of the wheel radius: red mark represents the baseline simulation

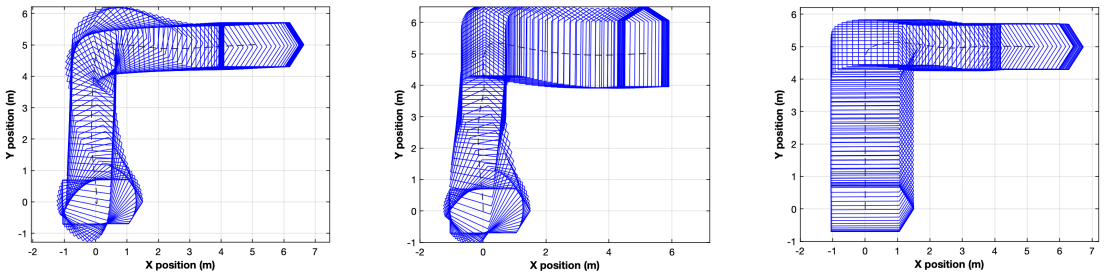
dimension of RLVs, this kind of variation can be negligible. Hence the LQR is considered to be robust to the parameters $L_f$ and $L_r$.

Table 5.6: Robustness tests on $L_r$ of the RLVs (RMS values of selected variables)

| $L_r$ | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi$(deg) |
|---|---|---|---|
| $L_r = 0.40$ m | 0.22 | 0.18 | 10.11 |
| $L_r = 0.45$ m | 0.22 | 0.17 | 8.04 |
| $L_r = 0.5$ m Baseline | 0.13 | 0.08 | 10.70 |
| $L_r = 0.55$ m | 0.23 | 0.18 | 8.70 |
| $L_r = 0.6$ m | 0.22 | 0.20 | 10.35 |

### 5.3.4 Influence of system constraints

As mentioned in Section 3.6.2, the RLVs presented high maneuverability because the variety on the steering configuration of the modules system. The RLVs maneuverability advantage for a case study presented in Figure 1.5 is now simulated in the RLVs simulator with a LQR controller. The directional path is designed, several simulation is performed to illustrate the maneuverability advantage of the RLVs. The simulation result is presented in Figure 5.31, the same reference path is reached with different maneuver.




(a) Scenario I (b) Scenario II (c) Scenario III

Figure 5.31: Maneuverability advantage of the RLVs: three scenarios simulation

The controller was able to perform different maneuvers of the system. The modules orientation of the second scenario is illustrated in Figure 5.32 to highlight some essencial point of the case study. Figure

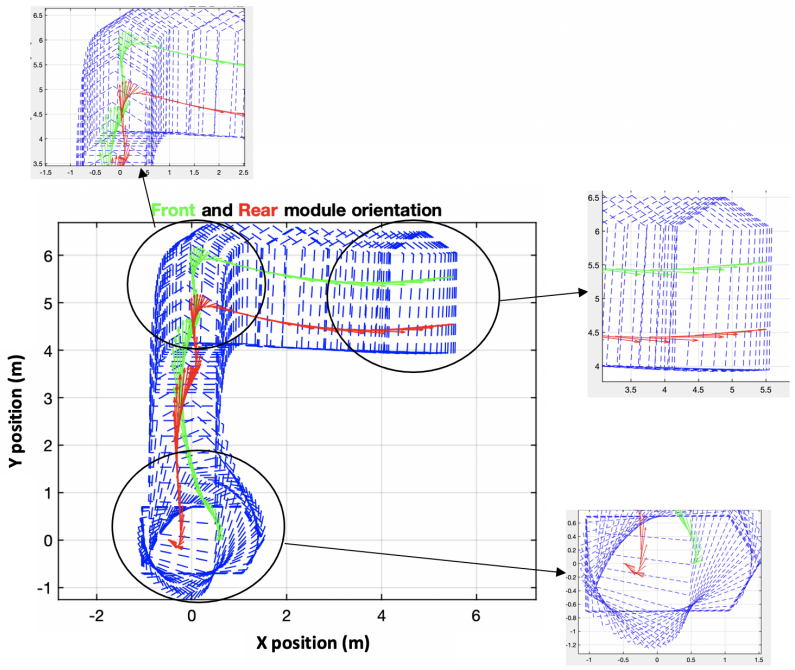



Figure 5.32: LQR controller: Scenario II- modules orientation

5.32 showed the modules system rotate intensively at the beginning to get the vehicle into a path with the right orientation, and stabilized during the movement in the $y$-direction. When the vehicle reaches the corner, the modules rotate again to ensure the vehicle move in $x$-direction without change of the vehicle orientation.

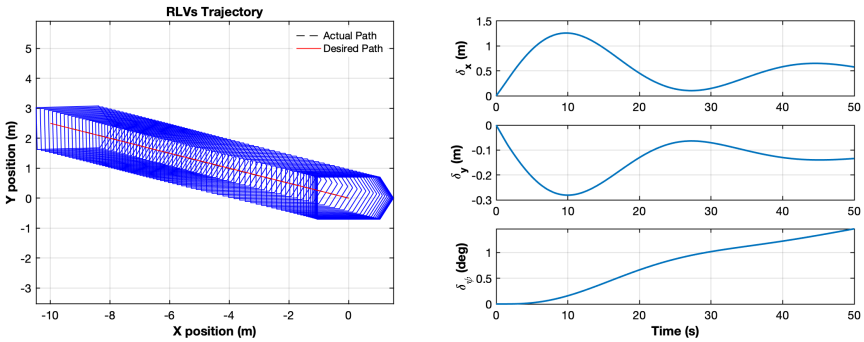
However, all simulation until now assumes the modules system works as a differential robot without any constraints in the physical configurations. This section presents an additional study for a particular situation in which the module orientation is constrained. The reference input for the simulation is defined

as:

$$\begin{aligned} x_{ref} &= -0.2t \\ y_{ref} &= 0.05t \\ \psi_{ref} &= 0 \end{aligned} \tag{5.33}$$

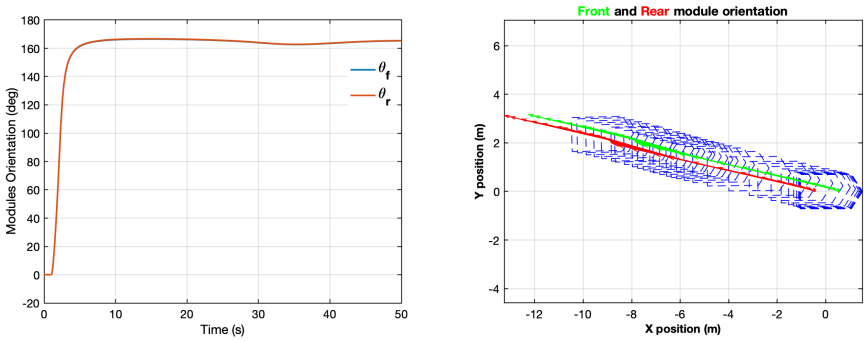
with sampling time of $0.01$ second.

For the case mission study, the first simulation is made without any additional system constrain. The vehicle was able to follow the reference trajectory with minimum errors (see Figure 5.33(a). The heading angle error is increasing during the simulation time, however still small the $1°$ as showed in Figure 5.33(b). As the RLVs' desired orientation is $0°$, the front and rear modules have to rotate more than $135$ degree (see Figure 5.34). This same simulation is done where now the modules can only turn




(a) RLVs trajectory (b) Errors

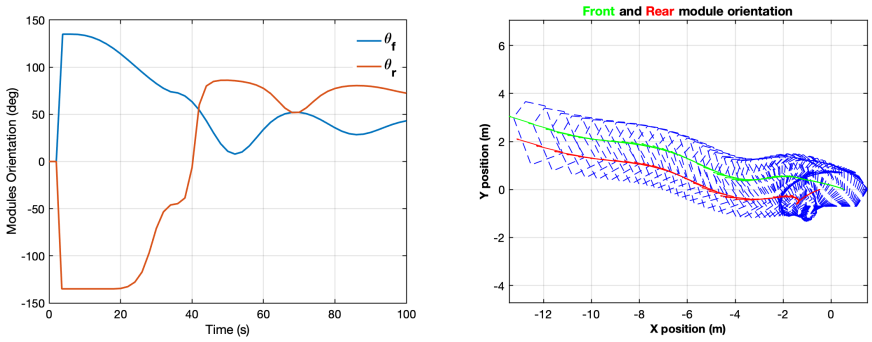
Figure 5.33: LQR controller: without constraints




(a) (b)

Figure 5.34: Modules orientations without constraints

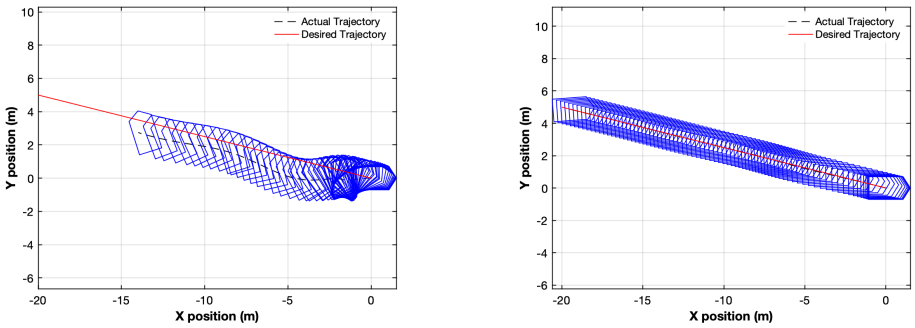
$[-135° + 135°]$. Figure 5.35(a) showed the module orientations are constrained in the defined intervals. As in the design stage, no considerations of the orientation constraints are considered; the vehicle loses the tracking initially; With the module orientation constraints, the vehicle starts to rotate with a small movement at the beginning. After the $50$ s, when the modules are in admissible orientations, the vehicle starts to track the reference trajectory. The simulation time for the case study mission is extended to $100$ s, and the resulting trajectory of the vehicle with and without modules orientation constrains are




(a) (b)

Figure 5.35: Modules orientations with constraints

presented in below: The additional module orientation constraints influence most the correction of the




(a) Resulting trajectory with modules orientation constrains

(b) Resulting trajectory without modules orientation constrains

Figure 5.36: Comparison of the resulting trajectory: with and without modules orientation constrains

vehicle orientations. Although in some cases, the correction of the orientation error is not as much as necessary as the position errors, and some fiscal constraints in the module system are unavoidable. The adjustment in the control algorithm should be made considering the real situation. One possible solution to face this type of system constraints is in the motion planning stage where the algorithm considers the module orientation constraints and gives the feasible reference input; another solution is the designed controller for each module and introduces the module system constraints in the design phase. However, the additional fiscal constraints between the different modules have to be taken into account for this approach.

### Conclusion

Three types of tasks are studied to understand the feature of the controller in the study. All simulation are done in the RLVs simulator. Several challenges trajectory is implemented to examine the system capabilities facing the path tracking problem. The simulation with path in $y$-direction and the spiral trajectory is used to illustrate essential points of the LQR controller, and the circle trajectory to study the LQR controller's robustness. The simulation result is summarized in Table 5.7, and LQR controller was

able to executed the mission with an acceptable error margin.

Table 5.7: LQR controller: simulation results (RMS value of selected variables)

| Type of path | reference | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi(deg)$ |
|---|---|---|---|---|
| $y$ direction | $x_{ref} = 0$<br>$y_{ref} = 0.1t$ | 0.15 | 0.42 | 43.86 |
| Spiral | $x_{ref} = 0.032t \sin(0.08t)$<br>$y_{ref} = 0.032t * 4\cos(0.08t)$ | 0.25 | 0.24 | 13.53 |
| Circle | $x_{ref} = 4\sin(0.05t)$<br>$y_{ref} = 4\cos(0.05t) - 4$ | 0.13 | 0.08 | 10.7 |
| Direction Path see Figure 5.14 | constant speed | 0.32 | 0.25 | 19.80 |

# Chapter 6

# Results and Discussion

Two types of control solutions described in the previous chapters present their advantages and disadvantages, many of which are discussed in the respective chapters. An overall comparison between them is crucial to provide a better overview of the different control options. In this chapter, this assessment is made considering parameters such as path-tracking performance for the several challenge paths (see Section 6.1), a case study mission which intent to evaluate the control effort of two controllers in Section 6.2.

## 6.1 Performance for case-study mission

This section summarises the simulation result of two controllers facing several challenging reference trajectory. For the tracking error assessment, the cross-track error of the center of the RLVs, denoted by $e_c$, is used, and it is calculate as:

$$e_c = \sqrt{\delta_x^2 + \delta_y^2} \tag{6.1}$$

where the $\delta_x$ and $\delta_y$ are the position error of the center of vehicle in $x$ and $y$ direction, respectively.
Table 6.1 listed all the simulation results of two controllers. Both controllers can accomplish the mission in the study; however, the different performances are noticeable. The results showed for all the path-tracking case in study, the LQR was able to follow the reference with less errors. However, the performance of two controllers relied a lot on the tuning parameters, therefore a more precise study is presented in the next section to evaluate the difference between the two controllers.

Table 6.1: RMS of $e_c$ for each of the controllers

| Type of path | Reference | Type of controller | $e_c(m)$ |
|---|---|---|---|
| Spiral | $x_{ref} = 0.032t\sin(0.08t)$ | AKM | $0.45$ |
| | $y_{ref} = 0.032t\cos(0.08t)$ | LQR | $0.36$ |
| Circle | $x_{ref} = 4\sin(0.05t)$ | AKM | $0.43$ |
| | $y_{ref} = 4\cos(0.05t) - 4$ | LQR | $0.15$ |

Table 6.1: RMS of $e_c$ for each of the controllers

| Type of path | Reference | Type of controller | $e_c(m)$ |
|---|---|---|---|
| Direction Path | constant speed | AKM | $0.92$ |
| | | LQR | $0.40$ |

## 6.2 Comparison of controller performance: case study

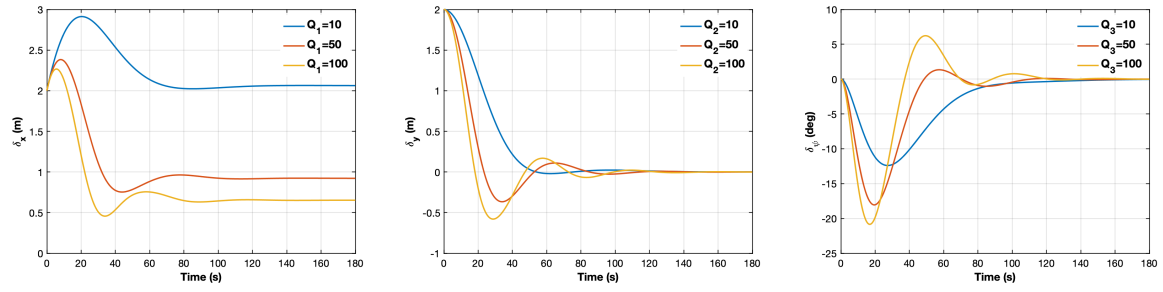
The mission case study used to evaluate the control performance of two controller is defined as:

$$\begin{aligned} x_{ref} &= 0.1t \\ y_{ref} &= 0 \\ \psi_{ref} &= 0 \end{aligned} \tag{6.2}$$

with sampling time of $0.01\,\mathrm{s}$, the vehicle initial conditions are $(-2, -2)\mathrm{m}$ with zero heading angle.
For evaluating the LQR controller, the $v_{ref}$ and $\omega_{ref}$ are calculated according to the Equations (5.24) and (5.23). The baseline value for matrix $Q$ and $R$ are:

$$\begin{aligned} Q(\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi) &= diag(10, 10, 50) \\ R(\tilde{v}_{ref}, \tilde{q}_{ref}) &= diag(5, 1) \end{aligned} \tag{6.3}$$

The effect of weighting matrices $Q$ and $R$ on the control system performance for the case mission is presented. Several simulation is done with the element of the matrix $Q$ and $R$ vary one at a time, allowing the evaluation of each element's influences. Figure 6.1 shows the variation of matrix $Q$ in time domain

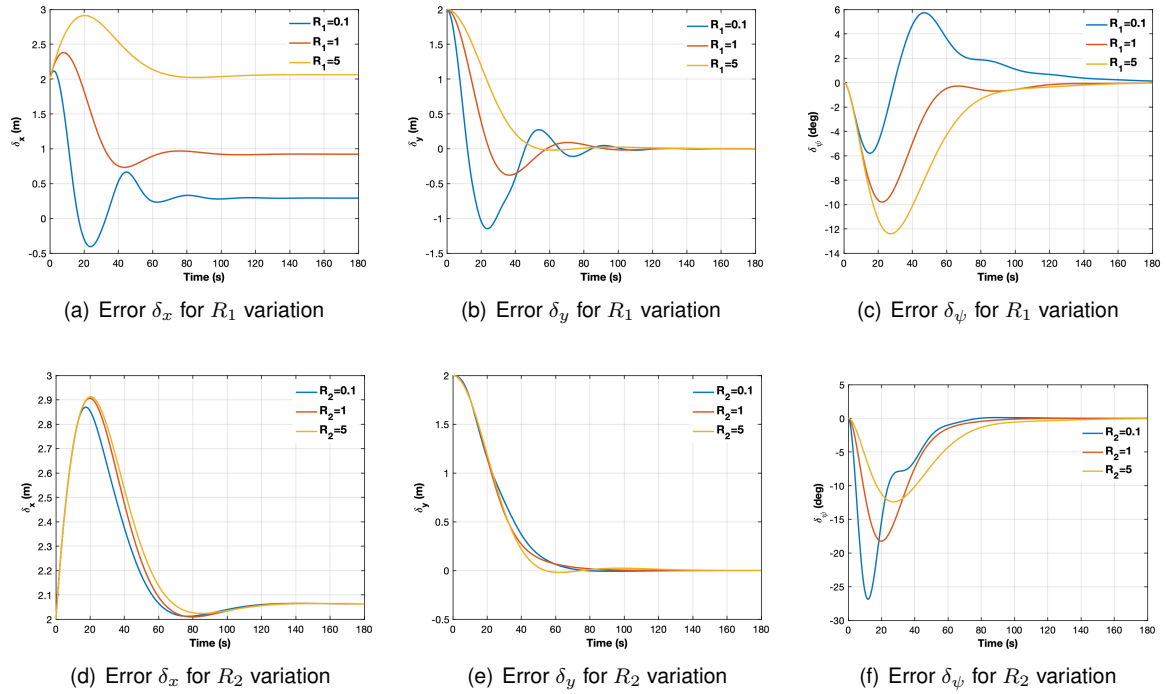



(a) Error $\delta_x$ for $Q_1$ variation (b) Error $\delta_y$ for $Q_2$ variation (c) Error $\delta_\psi$ for $Q_3$ variation

Figure 6.1: Effect of weighting matrices $Q$ on the time response: $R(\tilde{v}_{ref}, \tilde{q}_{ref}) = diag(5, 1)$

performances. With the variation in matrix $Q$ element, the overshoot slightly increases with a gradual fall in rising time, and the system error decreases with the increase of the $Q$. Similar observations could have been found for the matrix $R$. For high value of $R$ the time response becomes quicker with increased

overshoot. Based on the previous analysis of matrix parameters and the RLVs work conditions, the final




(a) Error $\delta_x$ for $R_1$ variation (b) Error $\delta_y$ for $R_1$ variation (c) Error $\delta_\psi$ for $R_1$ variation

(d) Error $\delta_x$ for $R_2$ variation (e) Error $\delta_y$ for $R_2$ variation (f) Error $\delta_\psi$ for $R_2$ variation

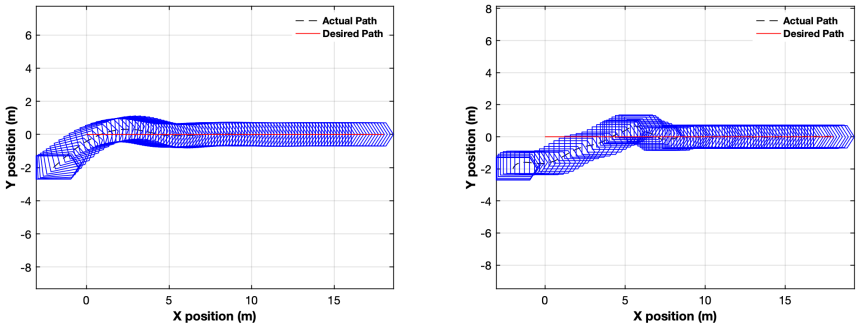
Figure 6.2: Effect of weighting matrices $R$ on the time response: $Q(\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi) = diag(10, 10, 50)$

value is set as:

$$Q(\tilde{q}_x, \tilde{q}_y, \tilde{q}_\psi) = diag(50, 50, 50) \tag{6.4}$$

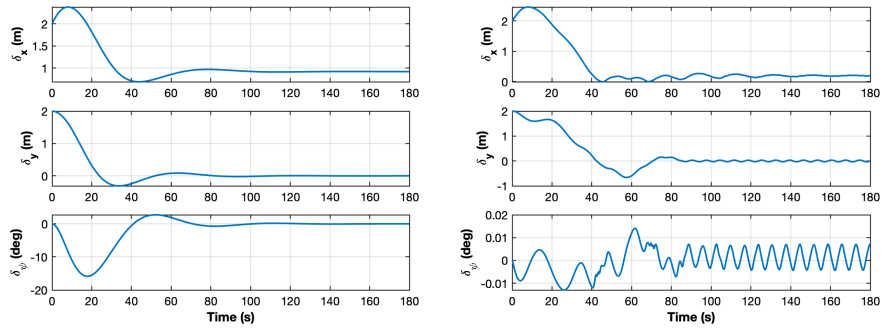
$$R(\tilde{v}_{ref}, \tilde{q}_{ref}) = diag(1, 1)$$

As for the AKM controller, the same reference is used. Different from the LQR controller, the LQR controller is a heuristic controller; the performance of the controller depends much more on the tuning process than the LQR controller. In this case study, without an introduced offset distanced, the only tuning parameter of the AKM controller is the value of $k_\omega$. A bigger value of $k_\omega$ decreases the static error with an increase of the overshoot. The tuning process of $k_\omega$ should consider the compromise between the position error and the transient system performance. Figure 6.3 showed the final path-tracking results. The LQR has a smooth transition in the entrance of the reference trajectory compared with AKM controller. After the entrance to the reference trajectory, both controllers are able to follow the reference trajectory. The vehicle with LQR controller was able to enter the reference trajectory quicker than vehicle with AKM controller. However, the deviation of the vehicle in the heading angle in LQR controller up to $15°$ while in AKM controller is less than $1°$. Figure 6.4 shows the comparison of errors of two controllers. It is observable, after the entrance to the reference trajectory, the AKM controller presents less static errors than the LQR controller. For better validation of two controller, the RMS of position and orientations errors are calculated and presented in Table 6.2. The AKM has smaller value in $\delta_\psi$ as expected, because as showed in Figure 6.3(b), the AKM controller intent to correct first the vehicle orientation, only after minimize the position errors. This type of performance has to do a lot with the tuning process of the AKM controller. In the error point of the view, for the selected reference, the




(a) LQR controller (b) AKM controller

Figure 6.3: Comparison of two controllers: resulting trajectories




(a) LQR controller (b) AKM controller

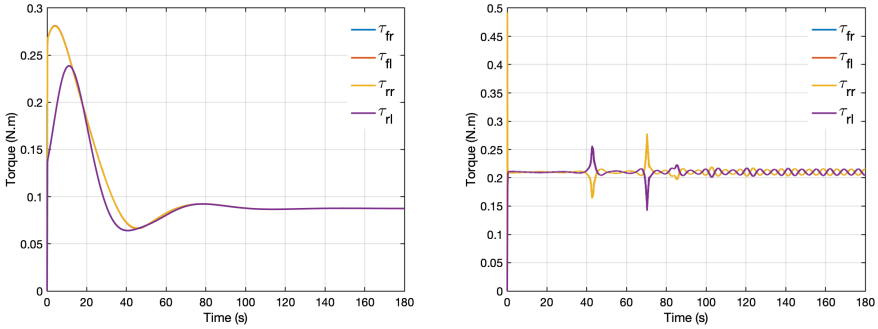
Figure 6.4: Comparison of two controllers: errors

LQR controller is preferably than AKM controller.

Table 6.2: Comparison between two controllers (RMS values of selected variables)

| **Type of controller** | $\delta_x(m)$ | $\delta_y(m)$ | $\delta_\psi$(deg) |
|---|---|---|---|
| LQR | 1.16 | 0.49 | 5.95 |
| AKM | 0.87 | 0.69 | 0.01 |

Figure 6.5 showed the requested torque of four motors for both two controllers. The value of torque is much smaller in the LQR controller, as the principle of the LQR is calculate the optimal control input that minimized the defined cost function. Moreover, the excitation of the requested torque of AKM controller in Figure 6.5(b) indicate that the AKM controller for each iteration try to get as quickly as possible, but the RLVs may not reaches the reference speed as desired.




(a) LQR controller (b) AKM controller

Figure 6.5: Comparison of actuators request

### Conclusions

In this chapter we provides an overview of the advantages and disadvantages of the two control solutions considered in this work, a geometric controller - AKM controller and a optimal controller - LQR controller when applied to the path-tracking problem. In the previous sections we compared the results obtained for each controller regarding a case study mission, tracking errors, a control effort and parameter tuning process. In order to better visualize the relative results, Table 6.3 presents a qualitative overall comparison between controllers.

Table 6.3: Overall Comparison between two controllers: (++ good, + average, – poor)

| **Type of controller** | AKM | LQR |
|---|---|---|
| Tracking errors | ++ | + |
| Requested control effort | – | ++ |
| Code simplicity | ++ | + |
| Design parameters tuning | – | + |

The table overall comparison between two controller. The evaluation of the design parameters tuning provides a comparative idea of the necessary effort of the designer to correctly tune the controllers parameters. The AKM controller as a heuristic controller, the performance relies a lot in the selection of the parameter while the LQR controller only requires the adjustment of the state and input control matrices parameters. This evaluation is merely based on the knowledge acquired throughout this work, and serves only as an indication of the future work.

# Chapter 7

# Conclusions and Future Work

The WMR have a broad spectrum of transportation applications where there is a need for material delivery, such as as transportation in factories, warehouses, and places where there is a need for material delivery. The rhombic like vehicle system is a type of a wheeled mobile robot with high maneuverability. Considering its particular configuration, this type of especially suitable for the work environment like ITER. RLVs' role as autonomous transportation equipment depends on their autonomous navigation capacity. The development of control solutions for autonomous vehicle navigation is essential to execute different missions in the critical working environment.

The work presented in this thesis played an important contribution for the project FORMULAfusion, with the RLVs modeling followed by the development of two types of motion control solutions for a scale model of RLVs, valid for several challenge paths.

First, the development of an overall control solution depends on a good knowledge of the system behavior. The modeling of vehicle system based on the kinematic and dynamic equations is performed which the RLVs is divided into three system: drive system, modules system and vehicle system. The motor engine used in the project Forumulafusion is modeled with the respective actuation constraints. The control-oriented kinematic model is also studied and linearized. A detailed analysis of the kinematic models provided the necessary insight into the vehicle behavior characteristics, allowing the control solution's design.

The motion control solution is based on a hierarchical approach: i) low-level law for the motor speed control and ii) high-level control law for the vehicle position control. A speed controller at the motor level is developed based on the PID control theory. Two control solutions for the position control are proposed : i) geometrical controller and ii) optimal linear control. The PID speed controller is implemented using *Simulink/Matlab*, and the comparison between the simulation data and the real data is performance to optimized the controller's performance.

As for the position control, the geometrical controller, namely, the AKM controller, is a model-free controller. This control algorithm based on the geometric relation between the reference path and the vehicle's actual pose. The simulation for the path following and set point case are done to illustrate the controller performance facing a different control parameters. The simulation results also showed the

AKM controller's performance depends a lot on the tuning process of the controller parameter. This method's main advantages are the design simplicity and the main disadvantage is the time-consuming tuning of the control design parameters.
The LQR controller is designed based on the kinematic model. The main advantages of this method, compared with AKM controller, are the simplicity of tuning process. It still takes some time-consuming tuning of the control design parameters, namely the state and input control matrices. However, this time compared with the AKM controller is insignificant.
Both two control solutions, AKM and LQR controller, proved to be capable of executing different missions: trajectory tracking and set-point cases mission. An assessment of each controller's advantages and disadvantages and a comparison between them was also made, providing a comprehensive insight into the RLVs autonomous navigation control problem and the solutions proposed.
During the development of this work, several issues remain unresolved, and some other methods can be developed. However, given the time limitations for this thesis development, they have not been addressed:

- Development of the mathematical model of the RLVs including tire model and design of the controller with the dynamic vehicle effects.
- Performed the control design in discrete time, it would be of interest to analyse the differences in the continuous case.

# Bibliography


[1] Reuben Hoggett. A history of cybernetic animals and early robots, 2020. URL `https://en.wikipedia.org/wiki/File:Atomic_force_microscope_block_diagram.svg`. [Online; accessed September 27, 2020].

[2] C. Gonzalez Gutierrez, C. Damiani, M. Irving, J.-P. Friconneau, A. Tesini, I. Ribeiro, and A. Vale. ITER Transfer Cask System: Status of design, issues and future developments. *Fusion Engineering and Design - FUSION ENG DES*, 85:2295–2299, 2010. doi: 10.1016/j.fusengdes.2010.09.010.

[3] A. Vale and I. Ribeiro. Mobile robot navigation for remote handling operations in iter. In *III Workshop de Robotica: Robotica Experimental*. Escuela Superior de Ingenieros de la Universidad de Sevilla, España, 2011.

[4] F. Rubio, F. Valero, and C. Llopis-Albert. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2):1–22, 2019. ISSN 17298814. doi: 10.1177/1729881419839596.

[5] D. Fonte. *Motion Planning for a Rhombic-Like Vehicle Operating in ITER Scenarios*. PhD thesis, Instituto Superior Técnico, University of Lisbon, 2011.

[6] X. Yun and N. Sarkar. Unified formulation of robotic systems with holonomic and nonholonomic constraints. *IEEE Transactions on Robotics and Automation*, 14(4):640–650, 1998.

[7] Holonomic and Nonholonomic Constraints. In *The Calculus of Variations*, pages 119–133. Springer New York, New York, NY, 2004. ISBN 978-0-387-21697-3. doi: 10.1007/0-387-21697-9{_}6. URL `https://doi.org/10.1007/0-387-21697-9_6`.

[8] P. F. Muir and C. P. Neuman. Kinematic modeling of wheeled mobile robots. *Journal of Robotic Systems*, 4(2):281–340, 1987. ISSN 10974563. doi: 10.1002/rob.4620040209.

[9] A. Kelly and N. Seegmiller. A vector algebra formulation of mobile robot velocity kinematics. In *Field and Service Robotics*, pages 613–627. Springer, 2014.

[10] A. M. Bloch, P. S. Krishnaprasad, J. E. Marsden, and R. M. Murray. Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis*, 136(1):21–99, 1996. ISSN 00039527. doi: 10.1007/BF02199365.

[11] M. Trojnacki. Dynamics model of a four-wheeled mobile robot for control applications–a three-case study. In *Intelligent Systems' 2014*, pages 99–116. Springer, 2015.

[12] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme. Robust localization using relative and absolute position estimates. *IEEE International Conference on Intelligent Robots and Systems*, 2:1134–1140, 1999. doi: 10.1109/iros.1999.812832.

[13] G. Li, D. Qin, and H. Ju. Localization of Wheeled Mobile Robot Based on Extended Kalman Filtering. *MATEC Web of Conferences*, 22:1–5, 2015. ISSN 2261236X. doi: 10.1051/matecconf/20152201061.

[14] M. Doumbia and X. Cheng. State estimation and localization based on sensor fusion for autonomous robots in indoor environment. *Computers*, 9(4):1–15, 2020. ISSN 2073431X. doi: 10.3390/computers9040084.

[15] D. Glavaški, M. Volf, and M. Bonkovic. Robot motion planning using exact cell decomposition and potential field methods. In *Proceedings of the 9th WSEAS international conference on Simulation, modelling and optimization*, pages 126–131. World Scientific and Engineering Academy and Society (WSEAS), 2009.

[16] G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, 2 edition, 2010. doi: 10.1017/CBO9780511780929.

[17] M. Defoort, J. Palos, A. Kokosy, T. Floquet, and W. Perruquetti. Performance-based reactive navigation for non-holonomic mobile robots. *Robotica*, 27(2):281–290, 2009. doi: 10.1017/S0263574708004700.

[18] J. P. Laumond, P. E. Jacobs, M. Taïx, and R. M. Murray. A Motion Planner for Nonholonomic Mobile Robots. *IEEE Transactions on Robotics and Automation*, 10(5):577–593, 1994. ISSN 1042296X. doi: 10.1109/70.326564.

[19] B. Triggs. Motion Planning for Nonholonomic Vehicles: An Introduction, 1993. URL `https://hal.inria.fr/inria-00548415`. Survey paper presented at Seminar on Computer Vision and Robotics, Newton Institute of Mathematical Sciences, Cambridge, England.

[20] O. Amidi and C. Thorpe. Integrated mobile robot control. *Fibers' 91, Boston, MA*, 1388(May): 504–523, 1991.

[21] S. F. Campbell. *Steering control of an autonomous ground vehicle with application to the DARPA urban challenge*. PhD thesis, Massachusetts Institute of Technology, 2007.

[22] D. Soetanto, L. Lapierre, and A. Pascoal. Adaptive, Non-Singular Path-Following Control of Dynamic Wheeled Robots. *Proceedings of the IEEE Conference on Decision and Control*, 2(January): 1765–1770, 2003. ISSN 01912216. doi: 10.1109/CDC.2003.1272868.

[23] K. Kanjanawanishkul, M. Hofmeister, and A. Zell. Path following with an optimal forward velocity for a mobile robot. *IFAC Proceedings Volumes*, 43(16):19–24, 2010.

[24] W. Oelen and J. van Amerongen. Robust tracking control of two-degrees-of-freedom mobile robots. *Control Engineering Practice*, 2(2):333–340, 1994. ISSN 09670661. doi: 10.1016/0967-0661(94)90215-1.

[25] K. Orman, A. Basci, and A. Derdiyok. Speed and direction angle control of four wheel drive skid-steered mobile robot by using fractional Order PI controller. *Elektronika ir Elektrotechnika*, 22(5):14–19, 2016. ISSN 13921215. doi: 10.5755/j01.eie.22.5.16337.

[26] A. Micaelli and C. Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots To cite this version :. *INRIA technical report*, 2097(RR-2097), 2006. ISSN 0249-6399. URL http://cat.inist.fr/?aModele=afficheN&cpsidt=49682%5Cnhttp://hal.inria.fr/inria-00074575.

[27] A. De Luca and G. Oriolo. Modelling and control of nonholonomic mechanical systems. *Kinematics and Dynamics of Multi-Body Systems*, 01 1995. doi: 10.1007/978-3-7091-4362-9_7.

[28] S. Morales, J. Magallanes, C. Delgado, and R. Canahuire. LQR Trajectory Tracking Control of an Omnidirectional Wheeled Mobile Robot. *2018 IEEE 2nd Colombian Conference on Robotics and Automation, CCRA 2018*, 2018. doi: 10.1109/CCRA.2018.8588146.

[29] F. Lin, Z. Lin, and X. Qiu. LQR controller for car-like robot. *Chinese Control Conference, CCC*, 2016-Augus:2515–2518, 2016. ISSN 21612927. doi: 10.1109/ChiCC.2016.7553742.

[30] A. Abbasi and A. J. Moshayedi. Trajectory tracking of two-wheeled mobile robots, using lqr optimal control method, based on computational model of khepera iv. *Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering*, 10(3):41–50, 2018.

[31] F. Demirbaş and M. Kalyoncu. Differential drive mobile robot trajectory tracking with using pid and kinematic based backstepping controller. *Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi*, 5(1):1–15, 2017.

[32] N. Silva, L. Baglivo, A. Vale, and M. D. Cecco. Four Path Following Controllers for Rhombic Like Vehicles. *2013 IEEE International Conference on Robotics and Automation*, pages 3204–3211, 2013. doi: 10.1109/ICRA.2013.6631023.

[33] Y. Ma. *Path planning and control of non-holonomic mobile robots*. PhD thesis, Doctorat délivré par l'École Centrale de Lille, 2013.

[34] R. Solea, A. Filipescu, S. Filipescu, and B. Dumitrascu. Sliding-mode controller for four-wheel-steering vehicle: Trajectory-tracking problem. In *2010 8th World Congress on Intelligent Control and Automation*, pages 1185–1190. IEEE, 2010.

[35] N. Nise, M. Perez, A. Perez, E. Perez, N. Nise, S. Simrock, N. Siddique, and A. Carrillo. *Control Systems Engineering*, volume 517. 2011. ISBN 9783319021348. URL http://150.185.9.18/fondo_editorial/images/PDF/CUPUL/SISTEMADECONTROL1.pdf.

[36] R. C. Coulter. Implementation of the Pure Pursuit Path Tracking Algorithm. Technical Report January, The Robotics Institute Camegie Mellon University, Pittsburgh, Pennsylvania, 1992.

[37] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. *Proceedings of the American Control Conference*, pages 2296–2301, 2007. ISSN 07431619. doi: 10.1109/ACC.2007.4282788.

[38] A. Bicchi and V. Kumar. *Control of Wheeled Mobile Robots: An Experimental Overview*, volume 270. 2001. ISBN 978-3-540-42090-3. doi: 10.1007/3-540-45000-9. URL `http://www.springerlink.com/index/10.1007/3-540-45000-9`.

# Appendix A

# Linearization Calculus

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{A.1}$$

where the $(\dot{x}, \dot{y}, \dot{\psi})$ are the velocities of the vehicle in the global frame, $v$ and $\omega$, the linear velocity of the vehicle in the local frame and its angular velocity around the vertical axis respectively, $\psi$ is the orientation of the vehicle with respect $x$ axis.

The reference state is defined as $(x_{ref}, y_{ref}, \psi_{ref})$ and the dynamic of the error related to each coordinate in the body frame is expressed as follows:

$$\begin{bmatrix} \dot{\tilde{q}}_x \\ \dot{\tilde{q}}_y \\ \dot{\tilde{q}}_\psi \end{bmatrix}_B = \begin{bmatrix} \dot{x}_{ref} - \dot{x} \\ \dot{y}_{ref} - \dot{y} \\ \dot{\psi}_{ref} - \dot{\psi} \end{bmatrix} = \begin{bmatrix} \tilde{v}\cos\tilde{\psi} \\ \tilde{v}\sin\tilde{\psi} \\ \tilde{\omega} \end{bmatrix} \tag{A.2}$$

with $\tilde{v} = v_{ref} - v$, $\tilde{\omega} = \omega_{ref} - \omega$ and $\tilde{\psi} = \psi_{ref} - \psi$.

The linear quadratic solution is used, therefore the non-linear model from (A.1) needs to be linearized. Linearization of the state space model form the system around the point of interest, in this case, a set of reference $v_{ref}$, $\omega_{ref}$ and $\psi_{ref}$ that enable the $\tilde{q}_B = 0$. Therefore, the equilibrium point are denoted by:

$$v = v_{ref} \tag{A.3}$$

$$\omega = \omega_{ref} \tag{A.4}$$

$$\psi = \psi_{ref} \tag{A.5}$$

Applying a first order Taylor approximation around the equilibrium point to the (A.2):

$$\dot{\tilde{q}}_B = \frac{\partial f}{\partial q_B}(q_B^{ref}, u^{ref})\tilde{q}_B + \frac{\partial f}{\partial u}(q_B^{ref}, u^{ref})\tilde{u} \tag{A.6}$$

$$\dot{\tilde{q}}_B = \begin{bmatrix} 0 & 0 & -v_{ref}\sin\psi_{ref} \\ 0 & 0 & v_{ref}\cos\psi_{ref} \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_B + \begin{bmatrix} \cos\psi_{ref} & 0 \\ \sin\psi_{ref} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \tag{A.7}$$

Those errors are expressed in base frame coordinate system, therefore a rotation matrix has been applied to get the error in the inertial frame:

$$\begin{bmatrix} \tilde{q}_x \\ \tilde{q}_y \\ \tilde{q}_\psi \end{bmatrix}_I = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{q}_x \\ \tilde{q}_y \\ \tilde{q}_\psi \end{bmatrix}_B = \tilde{q}_I = R\tilde{q}_B \tag{A.8}$$

According to the chain rules:

$$\dot{\tilde{q}}_I = \dot{R}\tilde{q}_B + R\dot{\tilde{q}}_B \tag{A.9}$$

where:

$$\dot{R} = \begin{bmatrix} -\sin\psi_{ref}\omega_{ref} & \cos\psi_{ref}\omega_{ref} & 0 \\ -\cos\psi_{ref}\omega_{ref} & -\sin\psi_{ref}\omega_{ref} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.10}$$

Using (A.8) and (A.10) the following eqations are obtained:

$$\dot{\tilde{q}}_I = \dot{R}\tilde{q}_B + R\dot{\tilde{q}}_B = \dot{R}R^{-1}\tilde{q}_I + R\dot{\tilde{q}}_B \tag{A.11}$$

with

$$\dot{R}R^{-1}\tilde{q}_I = \begin{bmatrix} -\sin\psi_{ref}\omega_{ref} & \cos\psi_{ref}\omega_{ref} & 0 \\ -\cos\psi_{ref}\omega_{ref} & -\sin\psi_{ref}\omega_{ref} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\psi_{ref} & -\sin\psi_{ref} & 0 \\ \sin\psi_{ref} & \cos\psi_{ref} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q}_I \tag{A.12}$$

Thus

$$\dot{R}R^{-1}\tilde{q}_I = \begin{bmatrix} 0 & \omega_{ref} & 0 \\ -\omega_{ref} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_I \tag{A.13}$$

And

$$R\dot{\tilde{q}}_B = \begin{bmatrix} \cos\psi_{ref} & \sin\psi_{ref} & 0 \\ -\sin\psi_{ref} & \cos\psi_{ref} & 0 \\ 0 & 0 & 0 \end{bmatrix} \left( \begin{bmatrix} 0 & 0 & -v_{ref}\sin\psi_{ref} \\ 0 & 0 & v_{ref}\cos\psi_{ref} \\ 0 & 0 & 0 \end{bmatrix} R^{-1}\tilde{q}_I + \begin{bmatrix} \cos\psi_{ref} & 0 \\ \sin\psi_{ref} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \right) \tag{A.14}$$

$$\Longleftrightarrow$$

$$R\dot{\tilde{q}}_B = \begin{bmatrix} \cos\psi_{ref} & \sin\psi_{ref} & 0 \\ -\sin\psi_{ref} & \cos\psi_{ref} & 0 \\ 0 & 0 & 0 \end{bmatrix} \left( \begin{bmatrix} 0 & 0 & -v_{ref}\sin\psi_{ref} \\ 0 & 0 & v_{ref}\cos\psi_{ref} \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_I + \begin{bmatrix} \cos\psi_{ref} & 0 \\ \sin\psi_{ref} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \right) \tag{A.15}$$

$$\Longleftrightarrow$$

$$R\dot{\tilde{q}}_B = \begin{bmatrix} 0 & 0 & v_{ref}(-\cos\psi_{ref}\sin\psi_{ref} + \sin\psi_{ref}\cos\psi_{ref}) \\ 0 & 0 & v_{ref}(\sin^2\psi_{ref} + \cos^2\psi_{ref}) \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_I + \begin{bmatrix} \sin^2\psi_{ref} + \cos^2\psi_{ref} & 0 \\ -\cos\psi_{ref}\sin\psi_{ref} + \sin\psi_{ref}\cos\psi_{ref} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \tag{A.16}$$

Thus

$$R\dot{\tilde{q}}_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_{ref} \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_I + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \tag{A.17}$$

Combining the (A.13) and (A.17) the tangent linearization of system about the reference trajectory in the global frame is obtained:

$$\dot{\tilde{q}}_I = \begin{bmatrix} 0 & \omega_{ref} & 0 \\ -\omega_{ref} & 0 & v_{ref} \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_G + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} = A\tilde{q}_I + B\tilde{u} \tag{A.18}$$

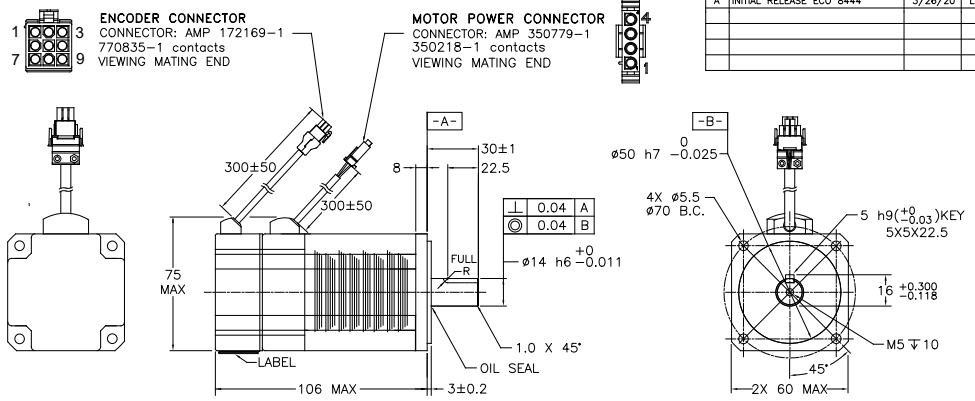
with $\tilde{q}_I$ as trajectory error in the inertial frame.

# Appendix B

# Technical Datasheets

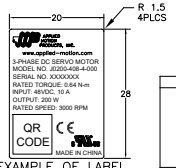
## B.1 DC SERVO MOTOR Datasheet

| REVISIONS | | | |
|---|---|---|---|
| REV | DESCRIPTION | DATE | APPROVED |
| A | INITIAL RELEASE ECO 8444 | 3/26/20 | LEO. L |




| ENCODER CONNECTION TABLE | | |
|---|---|---|
| PIN | LEAD COLOR | SIGNAL |
| 1 | BLUE | U+/A+ |
| 2 | GREEN | V+/B+ |
| 3 | YELLOW | W+/Z+ |
| 4 | BLUE/BLACK | U-/A- |
| 5 | GREEN/BLACK | V-/B- |
| 6 | YELLOW/BLACK | W-/Z- |
| 7 | RED | +5V |
| 8 | BLACK | GND |
| 9 | SHIELD | SHIELD |

| MOTOR CONNECTION TABLE | | |
|---|---|---|
| PIN | LEAD COLOR | SIGNAL |
| 1 | RED | PHASE U |
| 2 | YELLOW | PHASE V |
| 3 | BLUE | PHASE W |
| 4 | YELLOW/GREEN | GROUND |



EXAMPLE OF LABEL

NOTES: UNLESS OTHERWISE SPECIFIED
1. A SHAFT SEAL IS SHIPPED WITH MOTOR, BUT NOT INSTALLED.
2. TIMING LOGIC IS CCW AS VIEWED FROM FRONT SHAFT.
3. MOTOR IS IP65, EXCEPT SHAFT OPENING AND CABLE END CONNECTOR.
4. INSTALL INDOORS, AWAY FROM DIRECT SUNLIGHT AND FLAMMABLE GASES.
5. 1000M MAX ALTITUDE.

| TOLERANCES | THIRD ANGLE PROJECTION |
|---|---|
| DECIMALS: MM<br>X.XXX= ±0.013<br>X.XX = ±0.25<br>X.X = ±2.5<br>ANGLES:<br>MACH. = ±.5°<br>CHAM. = ±5° | |
| COMPUTER DATA BASE DRAWING | |

| APPROVALS | DATE |
|---|---|
| DRAWN ALAN.N | 3/26/20 |
| CHECKED | |
| APPROVED | |

APPLIED MOTION PRODUCTS, INC.

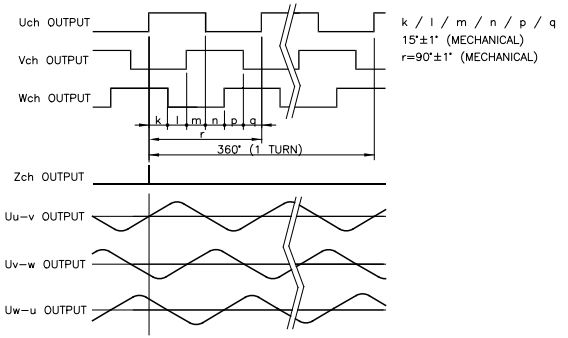
DC SERVO MOTOR

B | DWG NO. J0200-408-4-000 | REV A

SCALE: NONE | SHEET 1 OF 2

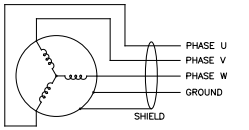
| DC SERVOMOTOR SPECIFICATION – PERMANENT MAGNET 8 POLES | | |
|---|---|---|
| INPUT POWER SUPPLY | VDC | 48 |
| RATED OUTPUT | W | 200 |
| VOLTAGE CONSTANT | V(rms)/K(rpm)±10% | 4.1 |
| RATED WINDING CURRENT | A (rms) | 10 |
| PEAK WINDING CURRENT | A (rms) | 30 |
| WINDING RESISTANCE | Ω ±10%@20°C | 0.192 LINE-TO-LINE |
| WINDING INDUCTANCE | mH ±20% | 0.56 LINE-TO-LINE |
| RATED TORQUE | N*m (ozf*in) | 0.64 (90.63) |
| PEAK TORQUE | N*m (ozf*in) | 1.9 (269.06) |
| TORQUE CONSTANT | N*m/A(rms)±10% | 0.065 |
| RATED SPEED | rpm | 3000 |
| MAXIMUM SPEED | rpm | 6000 |
| WEIGHT-WITH ENCODER | kg (lbs) | 1.1 (2.3 ) |
| INSULATION CLASS | | B 130°C |
| ENCODER RESOLUTION | | 2500 LINES/REV. |
| SHAFT LOAD – AXIAL | N (lbf) MAX | 70 (15 ) |
| SHAFT LOAD – RADIAL | N (lbf) MAX | 200 (45 ) |
| MOTOR RATING | REF NOTE 3 | IP65 |
| AMBIENT TEMPERATURE | OPERATING | 0 TO 40C° |
| AMBIENT TEMPERATURE | STORAGE | -20 TO 80C° |
| AMBIENT HUMIDITY MAX | | 85% |
| INERTIA-WITH ENCODER | g*cm² (ozf*in²) | 165 (0.0902) |

## TIMING LOGIC / ENCODER SIGNALS (CCW)
(NOT TO SAME SCALE AS OTHER DETAIL)

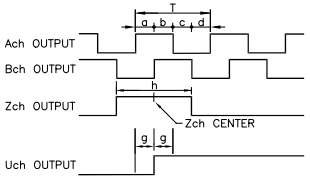



k / l / m / n / p / q
15°±1° (MECHANICAL)
r=90°±1° (MECHANICAL)

## MOTOR PHASES




## TIMING LOGIC / ENCODER SIGNALS
ALL THE TIMING LOGIC IS OBTAINED IN CCW ROTATION AS VIEWED FROM THE FRONT SHAFT.




T= 360 DEGREES / 2500C/T
a, b, c, d=T/4±T/8
h= T±T/2
g=±1° (BETWEEN Zch CENTER AND Uch RISE)

| CONTRACT NO. | |
|---|---|
| APPROVALS | DATE |
| DRAWN ALAN.N | 3/26/20 |
| CHECKED | |
| APPROVED | |
| APPROVED | |

APPLIED MOTION PRODUCTS, INC.

SERVO MOTOR SPECIFICATIONS

B | FSCM NO. | DWG NO. J0200-408-4-000 | REV A

SCALE: NONE | SHEET 2 OF 2