POLITECNICO DI TORINO

Master Degree in Mechatronic Engineering

Master Degree Thesis

Modelling, Control and Simulation of an Unmanned Ground Vehicle for Agriculture 4.0

Supervisors

Dr. Fabrizio DABBENE

Prof. Davide RICAUDA AIMONINO

Dr. Martina MAMMARELLA

Candidate

Simone SCIVOLI

JULY 2020

Abstract

This thesis focuses on the design of an innovative algorithm of automatic control for a four wheel steering (4WS) unmanned ground vehicle (UGV) within the framework of agriculture 4.0. The 4WS is obtained by implementing two Ackermann steering mechanisms (ASM) on both front and rear axes: such mechanism generates wheels' slippage whenever both axes are steered.
The main purpose of the thesis is to design a controller able to pursue the automatic tracking of a given path and in the meantime optimize the velocity of each wheel in order to minimize the slippage produced by the ASM.
Moreover, a simulation environment has been realized to assess the behaviour of the vehicle on a given scenario. Such environment is divided into three different parts: i) the UGV model that includes both kinematic (how it moves in a two dimensional space) and dynamics (how its motors behave when receiving some inputs), ii) the navigation system that stores informations about the path to be followed, and sends the informations the controller requires to work and iii) a controller, based on the information received from the navigation system, that computes the required steering angles of the axes to comply automatic guidance and the optimized velocity of each wheel to minimize their slippage. A substantial simulation campaign has been performed in which the UGV behaviour has been simulated within field-like scenarios while performing different tasks. The goal was to find the best controller configuration. The results of the simulation campaign show that the vehicle is able to track the path correctly and to assign the best velocity to each wheel to minimize the error generated by the ASM.

Dedications

I would like to dedicate my work made in these months to my parents and my brother, my uncle Giuseppe and aunt Giusy that have been like parents in these years in Turin, my friends that shared everything with me in the last 6 years, and to my whole family (relatives and childhood friends) who have been physically far, but have always supported me and my work.

Contents

Introduction **4**

1 Ackermann steering mechanism **5**
1.1 Ackermann steering equation 6
1.2 Case study: four steering wheels 8
1.2.1 Computation of the slipping angles 9

2 Modelling the Unmanned Ground Vehicle **13**
2.1 Kinematic equations of motion 13
2.2 Dynamics of the motors 17
2.2.1 Steering motors equations of motion . . . 17
2.2.2 Wheels motors equations of motion 18
2.3 Description of the UGV 20

3 The navigation system **23**
3.1 Assignment of the Frenet frame to C(k) 23
3.1.1 The starting point 25
3.2 Computation of the angular deviation 26
3.3 The choice of v_d 28

4 The controller **30**
4.1 Steering controller 30
4.2 Velocity optimizer 33

5 Testing and simulation results **36**
5.1 Simulation results on path 1 38
5.2 Simulation results on path 2 43

Conclusion and future work **48**

Introduction

Worldwide growth in population, together with the necessity of preserving the planet in terms of both pollution and resources, as described in [1], requires an evolution of agricultural work, permitting both an increment of the level of production of primary goods and a reduction of the impact of human activity on the environment. To bring solutions to such a contradictory situation, robotics appears to be a promising alternative. The benefits of automatic devices in terms of accuracy, repeatability, and durability, indeed, constitute interesting potentials in taking up the agricultural challenge, as reported in [3].

The most important feature when working inside an agricultural field is to move from a point to another pursuing a certain task. Automation of motion seems to be a valid solution when an higher efficiency in production is required. But such automation requires a quite high precision in motion, due to the low grip conditions,as stated in [19], and the confined spaces these vehicles are intended for.

To contribute to the technological progress in field application, the DIpartimento di Scienze Agrarie, Forestali, Alimentari (DISAFA) department of Università di Torino, in partnership with Politecnico di Torino and CNR-IEIIT Institute of Electronics, Computer and Telecommunication Engineering of National Research Council of Italy (CNR-IEIIT) is developing a model of a four wheel steering (4WS) unmanned ground vehicle (UGV) able to achieve some goals in terms of automation and mobility in fields, with the purpose of providing a concrete help to farmers when they have to harvest the production from their plants. This vehicle has been designed to mount an Ackermann steering mechanism (ASM) on both front and rear axes, while it is set to have four independent motor on each wheel.

In the last decades, four wheel steering vehicles have been designed for a variety of applications. Indeed, a vehicle that has the possibility of steering

with all wheels has two great advantages with respect to a front wheel steering one: when it is going at low speed, the front and rear wheels turn to the contrary, which can reduce the turning radius of the vehicle and improve the flexibility of steering; while increasing velocity, i.e. moving at high speed, the front and rear wheels turn equally, which can reduce the side-slip angle of the vehicle center, the yaw rate and the heeling angle, and improve the handling stability of the vehicle, as widely described in [2].

When moving into a field, the vehicle is constrained to move into narrow spaces; this means that its trajectory cannot be casual, but it has to move following a given path that considers all the obstacles along it in order to avoid the collision with any of them. It is possible to find in the literature many control systems applied to UGVs able to achieve the automatic tracking of a given path and these systems are designed starting from the modelling of the vehicle. There are two main categories of vehicle models: i) dynamic model, ii) kinematic model. These two types are used for different purposes: the dynamics is kept into account when the vehicle is able to reach quite high velocities, thus its stability is an important feature; the kinematic models are used when a vehicle is used to fulfill tasks that require low speeds, thus the inertias are neglected with respect to kinematic features. Regardless of the model used, path tracking is the main target, but it is achieved in different ways depending on which kind of model is considered. A non-linear feedback control is used in [20] to make a four wheel independent steering vehicle achieve path tracking; according to [22], the 4WS vehicle is controlled with an adaptive pure pursuit controller; the strategy of [4] is using an Adaptive Fuzzy PID algorithm to control the vehicle. A simpler controller, a feedback path tracking one, is used in [17] to pursue automatic tracking of a given path.

All of the aforementioned control algorithms model the four wheel independent steering describing its dynamic features; some examples of control systems based on kinematic models are given by the following papers: according to [9], the UGV could be guided by a feedback control law aimed to correct the lateral and angular deviations of the vehicle with respect to the path; in [12] are proposed two different approaches, the first controller based on a feedback linearisation approach, a second one based on a Lyapunov oriented control design.

Agricultural four wheel steering (4WS) vehicles have been studied mainly with the aim of increasing the driveability of vehicles operating in confined space, e.g., handlers, loaders, or farm vehicles in headlands, as stated in [13];

according to [3], other application of 4WS have been thought for small, specialized autonomous mobile robots; in the latter paper it's illustrated a feedback control algorithm that keeps into account also the sliding effect due to the low grip conditions that affect the terrains where the vehicle has to move. The model presented in [3] was extremely useful to generate the model and the control system that are described in this thesis, but it is again tailored for a four wheel independent steering.

This thesis is intended to envision an UGV that mounts an ASM on the axes; this design generates slippage on wheels whenever the vehicle is steering with both axes. To overcome this problem, a new non-linear proportional controller that keeps into account the relationships among wheels due to the ASM is proposed.

However, steering controller alone is not able to minimize such slippage; on the other side, the four wheels are driven individually. This gives the opportunity to introduce a second controller that optimizes the velocities of each wheel in order to minimize the errors introduced by the ASM.

A simulation environment is required to study the behaviour of the UGV when driven by the aforementioned controllers. Such simulator,developed as part of this thesis on MATLAB/Simulink, is divided into three main blocks:

- the kinematic and dynamic models of the UGV;
- the navigation system, storing information on the path to be followed, that analyses the position of the UGV and returns the reference data required by the controller;
- the controller, which is divided into two parts: the steering controller responsible for the automatic tracking of the given path; the velocity optimizer, responsible of the minimization of wheels slippage.

Path tracking, as already stated, is the main purpose of the UGV but to achieve this target the path has to be defined by an external algorithm, able to study the environment and to generate a series of points that give the possibility to the vehicle to move among obstacles without colliding. Such algorithm has been developed by my colleague Mohammadreza Beygfard: this guidance scheme analyses the environment where the robot has to move and returns a series of points describing desired position and orientation it has to maintain in order to start from the first point and approach the goal complying with the safety constraints.

This paper is organized as follows: the first chapter focuses on the Ackermann Steering Mechanism, what it is used for and which kind of vehicle it is intended to install it, focusing particularly on the 4WS UGV designed by the DISAFA research group. Chapter 2 is about the design of the kinematic and dynamic models of the UGV and explains how the simulation environment exploits them; Chapter 3 explains the operation of the navigation system: how it uses information on the path to be followed and how it manipulates these data to return them as required by the controller; Chapter 4 focuses on the two controllers and their main features to control the UGV. Last, Chapter 5, explains the main features of the tests applied on the simulation environment used to configure the controller and shows the results of tracking two different field-like paths, given by the algorithm of my colleague Mohamadreza Beygfard, assigning to the UGV different set-ups to illustrate the various behaviours assumed.

Chapter 1

Ackermann steering mechanism

The Ackermann steering mechanism is a geometric arrangement of linkages in the steering of a vehicle designed to turn the inner and outer wheels of the steering axis at appropriate angles. It has been named after the inventor Rudolph Ackermann, who introduced this system in 1817 for horse-drawn vehicles. Such geometrical arrangement implies that the rotation axis of all wheels are arranged as radii of circles with a common centre point, as reported in [15], called Instantaneous Center of Rotation (ICR).

The primary goals of this geometry are the following, as stated in [23]:

- minimum wheel-slip and symmetric steering control for left and right turns;
- ensuring minimum cross-coupling between steering and axis oscillation;
- maintaining favourable pressure angles in the joints;
- avoiding interference between the moving parts of the mechanism and between them and the body of the vehicle.

The mechanism that complies better such purpose consists of two inclined levers, associated to each steering wheel, coupled together by means of a rod, as widely described in [7]. This configuration forms a trapezoid, if seen from above. The trapezoid steering four-bar mechanism can be modelled in different ways: the first one analysing the Ackermann configuration was Charles Jeantaud in 1878. He stated that the best configuration in order to follow the trajectory from a straight line to the maximum curvature was the case when

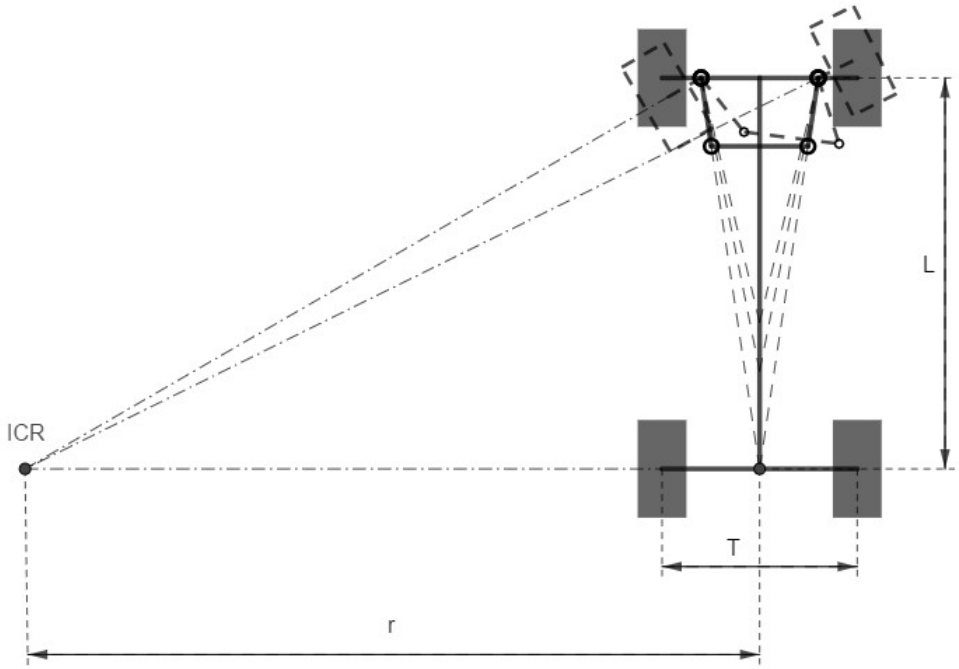


Figure 1.1: *Trapezoid steering four-bar mechanism. In figure are represented different convergence points of the inclined rod projections.*

the projection of the inclined levers converge in correspondence to the center of the rear axis,as reported in [16]. According to [6], others, such Dixon, studied the mechanism and drew different conclusions about the point along the wheelbase where the lines should intersect and the length of the rods attached to the wheels. An example of the Jeantaud diagram is shown in Figure 1.1. Although trapezoid steering four-bar mechanism is very good, it remains an approximation, in fact it is not able to always follow the Ackermann criteria, specially when approaching the physical limits of the steering angle.[14]

1.1 Ackermann steering equation

For a generic turn, there are two wheels that are closer to the ICR (the inner ones) and two wheels that are further to such point (the outer ones). To obtain a single ICR, the two wheels belonging to the steering axis have different steering angles. The equation describing the angle of the inner wheel δ_i and the angle of the outer one δ_o can be derived by making some trigonometry

computations.

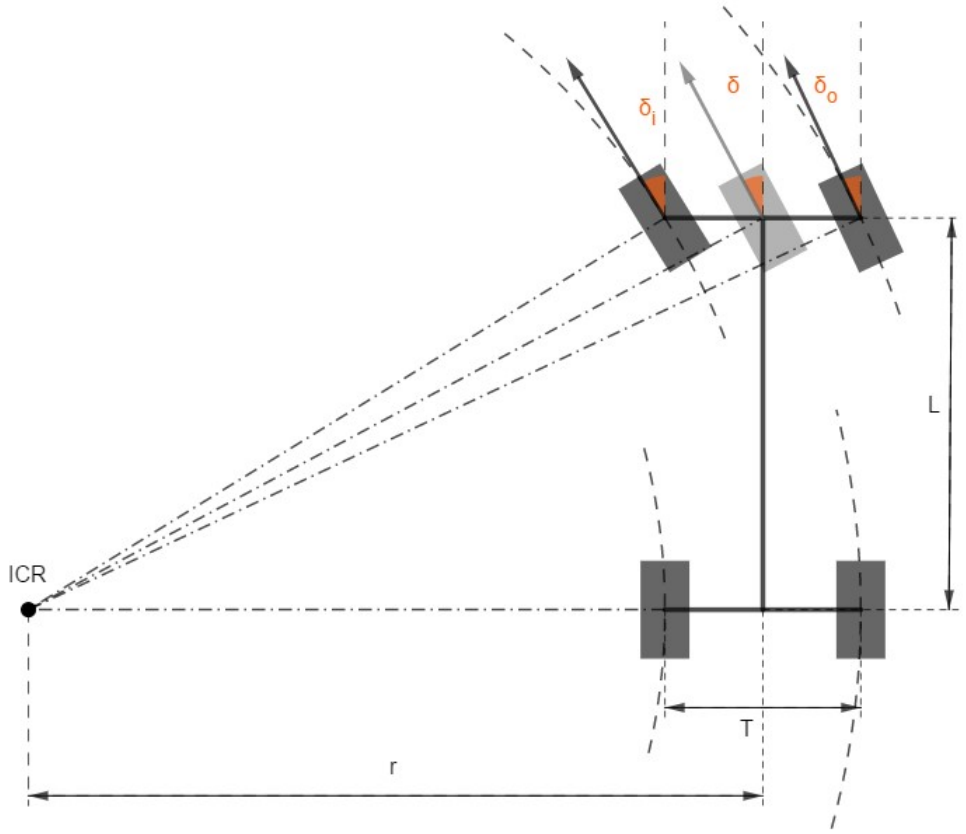


Figure 1.2: *Ideal Ackermann turning geometry. the wheel at the center of front axis is virtual and is used to define its steering angle δ; it's useful to mark the geometric relationships between inner and outer wheels.*

Let's apply a virtual wheel on the center of the steering axis. Let's then call the steering angle of such wheel δ. This angle defines how much the axis is steering.
We can consider three right triangles composed by the wheelbase L as height and the radius of rotation r (see Figure 1.2), by adding or subtracting the length $\frac{T}{2}$,as base of the triangle, obtaining, thus, three equations, as stated in [6]:

$$\tan(\delta_i) = \frac{L}{r - \frac{T}{2}} \tag{1.1}$$

$$\tan(\delta) = \frac{L}{r} \tag{1.2}$$

$$\tan(\delta_o) = \frac{L}{r + \frac{T}{2}} \tag{1.3}$$

By subtracting the reciprocal of the equations (1.3) and (1.1), we arrive at the Ackermann steering equation:

$$\frac{1}{\tan(\delta_o)} - \frac{1}{\tan(\delta_i)} = \frac{r + \frac{T}{2}}{L} - \frac{r - \frac{T}{2}}{L} \Rightarrow \cot(\delta_o) - \cot(\delta_i) = \frac{T}{L} \tag{1.4}$$

Following the same procedure, angles δ_i and δ_o can be expressed as function of the axis angle δ:

$$\frac{1}{\tan(\delta_o)} - \frac{1}{\tan(\delta)} = \frac{r + \frac{T}{2}}{L} - \frac{r}{L} \Rightarrow \cot(\delta_o) = \cot(\delta) + \frac{T}{2L}$$

$$\frac{1}{\tan(\delta_i)} - \frac{1}{\tan(\delta)} = \frac{r - \frac{T}{2}}{L} - \frac{r}{L} \Rightarrow \cot(\delta_o) = \cot(\delta) - \frac{T}{2L}$$

Let's substitute $\cot(\alpha) = \frac{1}{\tan(\alpha)} = \frac{\cos(\alpha)}{\sin(\alpha)}$ in order to avoid an undefined function when $\alpha = 0$. We are now able to write the angles δ_i and δ_o as a function of δ as

$$\delta_i = \tan^{-1}(\frac{2L\sin(\delta)}{2L\cos(\delta) - T\sin(\delta)} \tag{1.5}$$

$$\delta_o = \tan^{-1}(\frac{2L\sin(\delta)}{2L\cos(\delta) + T\sin(\delta)} \tag{1.6}$$

1.2 Case study: four steering wheels

The ASM, as stated above, is thought for only two steering wheels' vehicles, since its main purpose is to obtain a single ICR in correspondence to the non-steering wheels' rotation axis.
Our UGV mounts two different ASM on front and rear axes. This design

generates two different ICR whenever both axis are steering together: indeed, the front steering axis generates an ICR in correspondence to the projection of the rear one and the rear steering axis does the same along the projection of the front one.

Analysing this specific situation in geometric terms, the four wheels' rotation axes form a quadrilateral, as we can see in Figure 1.4; the actual ICR can be wherever inside this shape. The unknown actual position of the ICR within the quadrilateral influences the motion of the UGV: if the ICR is situated in correspondence of one of the four vertices, then only two wheels are adhering with the ground and the remaining two are slipping; in the case the ICR is situated along a side, the adhering wheels number decrease to only one, while the other three wheels are affected by side slip; eventually, any other position of the ICR involves the slippage of all the four wheels.
Last but not least, if the UGV is moving with a crab-wise motion (CWM) (see Figure 1.3), wheels situated on the same axis are not parallel due to the ASM and, even if the wheels on the diagonal are parallel one to each other, such configuration bring to the slippage of two wheels: the front left and the rear right one or the front right and the rear left one, depending on which pair of wheels is the dominating one.

1.2.1 Computation of the slipping angles

Let's now analyse which kinds of steering errors are generated in this system, making reference to Figure 1.4. Let's consider angles δ_F and δ_R which control the front and rear axes; then let's assign an angle to the wheels belonging to the same axis according to Eq. (1.5) and (1.6). The reference value of these angles is set to zero when it is going straight with the vehicle body; the angle is positive when the wheel is rotating counter-clockwise (if seen from above), while in case of a clockwise rotation, the angle is considered negative.

We have to distinguish 2 different cases:

- δ_F and δ_R are equal both in magnitude and in sign; this means that vehicle is moving with a CWM.

- δ_F and δ_R are not equal, meaning that an ICR is generated.

In the first case, as already stated, wheels on the diagonals of the vehicle are parallel one with each other, but the ones on the same axis are not.

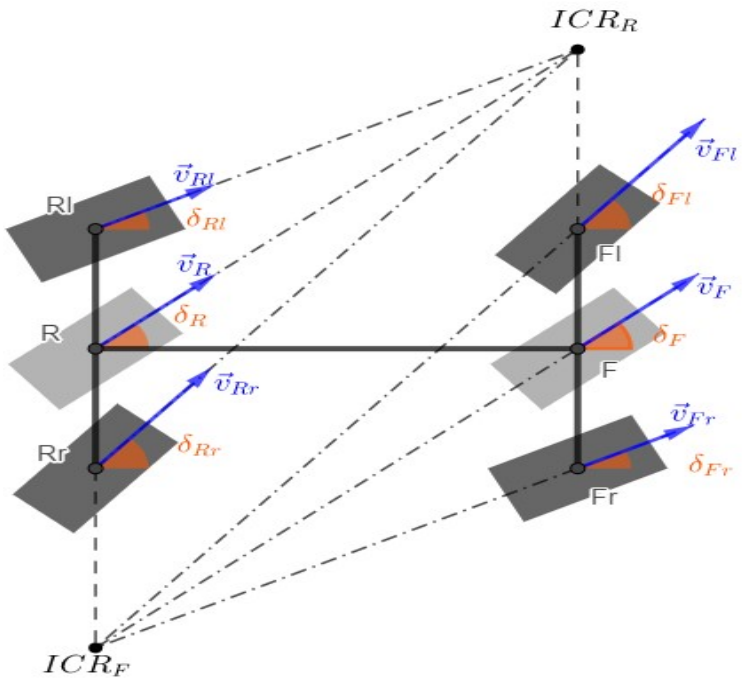


Figure 1.3: *Crab-wise motion. It's possible to see how the wheels situated diagonally are parallel, but they aren't with respect to the same axis.*

This produces slippage on the two wheels that are not following the vehicle's direction, defined by the other two. Such wheels are subjected to the error:

$$\beta = \delta_i - \delta_o \tag{1.7}$$

that can be either positive or negative, depending on which are the wheels that are slipping.

The second case is a bit more tricky but we can compute the sliding error using some trigonometry concepts, supposing that the actual ICR corresponds to the virtual one.
Let's consider the reference frame (R, x, y) in Figure 1.4. Its origin is situated in correspondence with the rear virtual wheel R and the x axis positioned along the vehicle body; consequently, y axis points toward the left wheel. The aim is finding the coordinates of I_{CR} with respect to this frame.
Now, let's consider the triangle $\overset{\triangle}{FI_{CR}R}$; the angles $\hat{F}$ and $\hat{R}$ are known;

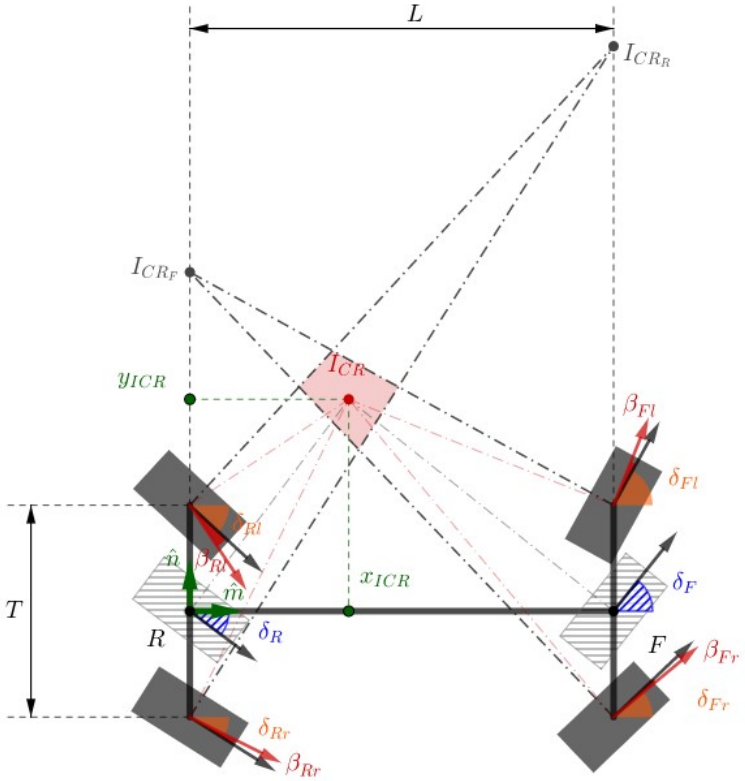


Figure 1.4: *Representation of the UGV steering principle. The Ackermann mechanism applied on both front and rear axes generate an error angle β on the wheels, causing slippage on al least two of them.*

moreover, the sum of the three internal angles must be equal to π:

$$\begin{cases} \hat{F} + \hat{R} + \hat{I_{CR}} = \pi \\ \hat{F} = \dfrac{\pi}{2} - \delta_F \\ \hat{R} = \dfrac{\pi}{2} + \delta_R \end{cases}$$

thus the angle $\hat{I_{CR}}$ is defined as:

$$\hat{I_{CR}} = \delta_F - \delta_R \tag{1.8}$$

that is positive if the vehicle is turning left, while it's negative if vehicle is turning right.

We are now able to apply the sine rule to find segment $\overline{FI_{CR}}$ (or $\overline{RI_{CR}}$, it's the same procedure with slightly different considerations).

$$\frac{L}{\sin(\hat{I_{CR}})} = \frac{\overline{FI_{CR}}}{\sin(\hat{F})} = \frac{\overline{RI_{CR}}}{\sin(\hat{R})}$$

This means that

$$\overline{FI_{CR}} = \frac{L}{\sin(\delta_F - \delta_R)} \sin(\frac{\pi}{2} - \delta_R) = \frac{L}{\sin(\delta_F - \delta_R)} \cos(\delta_R)$$

It's now possible to find (x,y) coordinates of the virtual ICR as

$$\begin{cases} x_{ICR}\big|_R = \overline{FI_{CR}} \sin(\hat{F}) = \dfrac{L\cos(\delta_R)}{\sin(\delta_F - \delta_R)} \cos(\delta_F) \\ y_{ICR}\big|_R = L - \overline{FI_{CR}} \cos(\hat{F}) = L * \left(1 - \dfrac{\cos(\delta_R)}{\sin(\delta_F - \delta_R)} \sin(\delta_F)\right) \end{cases} \tag{1.9}$$

Once the position of ICR is known, we can compute the theoretical angles γ_{ij}, with $i = F, R$ and $j = l, r$, and thus the slipping angle relative to each wheel:

$$\beta_{ij} = \gamma_{ij} - \delta_{ij} \tag{1.10}$$

where:

- δ_{ij} is given by Eq. (1.5) and (1.6),
- $\gamma_{ij} = \arctan\left(\dfrac{A - x_{ICR}}{y_{ICR} \mp \frac{T}{2}}\right)$ *where* $\begin{cases} A = 0 \ if \ i = R \\ A = L \ if \ i = F \end{cases}$

Although this computations are quite easy and useful, we don't know the exact ICR; this means that we have to follow another approach in order to set these errors to zero, or at least minimize them.

Chapter 2

Modelling the Unmanned Ground Vehicle

This chapter is focused on modelling the UGV, trying to obtain a model that is as precise as possible. The UGV has to work in farm fields where the coefficient of friction, fundamental element to define the dynamic of the vehicle, varies a lot, making it difficult to be computed in real time; moreover the vehicle will maintain a low speed: if a planar motion is assumed, when the speed is low the effects of the dynamics of the system, which appear in the form of fast modes, can be neglected compared to the effect of the kinematics which appear in the form of slow modes, as introduced in [9] and [21]. The whole model considers the motion of the robot and simulates the behaviour of the electric motors applied on wheels and the ones applied on the steering axes to make the ASM work. The desired velocities v_{ij} and steering angles δ_i, where $i = [F, R]$ and $j = [l, r]$ are known (they are received as inputs from the controller). The main purpose is to compute the position (X_M, Y_M) and the heading θ_M of the vehicle.

2.1 Kinematic equations of motion

To describe vehicle motion, a global reference frame $(O, \hat{i}, \hat{j})$ is considered (it is marked with $[G]$). Then a new body reference frame is positioned on the UGV: the Center of Gravity (CoG) of the vehicle will be the origin of the frame $(M, \hat{m}, \hat{n})$ (defined by $[V]$), with the x_m-axis going along its length and the y_m-axis pointed to its left side (see Figure 2.1). The CoG is positioned at distance a from the front axis, and at distance b from the rear one; thus the

length is computed by:

$$L = a + b$$

T is the distance, along the y_m-axis, from the center of the right to the center of the left wheel.

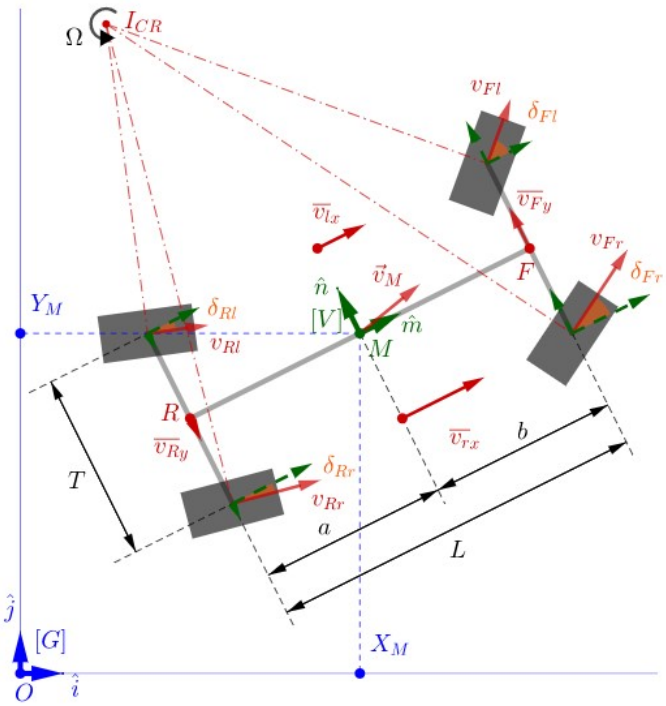


Figure 2.1: *Representation of the 4WS kinematic model that introduces the error generated by the ASM: velocities are discomposed into components, then a mean of longitudinal and lateral velocities is computed and applied to the CoG M*

To design a 4WS kinematic model able to simulate the behaviour of the UGV, some non-holonomic constraints (NHC) must be defined to avoid wheels' slippage, taken from [5] and [18] and reinterpreted in this paper as:

$$\begin{cases} v_{Fl}\cos(\delta_{Fl}) - v_{Rl}\cos(\delta_{Rl}) = 0 \\ v_{Fr}\cos(\delta_{Fr}) - v_{Rr}\cos(\delta_{Rr}) = 0 \\ v_{Fl}\sin(\delta_{Fl}) - v_{Fr}\sin(\delta_{Fr}) = 0 \\ v_{Rl}\sin(\delta_{Rl}) - v_{Rr}\sin(\delta_{Rr}) = 0 \end{cases} \tag{2.1}$$

that can be rewritten as:

$$\begin{cases} v_{Fl_x} = v_{Rl_x} \\ v_{Fr_x} = v_{Rr_x} \\ v_{Fl_y} = v_{Fr_y} \\ v_{Rl_y} = v_{Rr_y} \end{cases}$$

To explain such constraints let's consider the dimensions of the UGV: L is the longitudinal distance from the front to the rear wheel and it's the same on both left and right side; while T is the lateral distance from left to right wheels, and it's equal both on front and rear axes. The purpose of the NHC is to make sure that such distances remain constant in time: let's consider for example the first equation of (2.1), which is applied on the left side of the vehicle; if $v_{Fl_x} > v_{Rl_x}$ then the front extreme of length L is "going faster" than the rear one; this would implies that such length is not maintained in time, or better, considering the vehicle as a rigid body, one of the two wheels would slip in order to maintain the distance L between the wheels. The second equation applies the same reasoning but it's referred to the right side, while the third and the fourth ones consider the lateral velocities applied on front and rear wheels respectively; in this way distance T between the wheels of the same axis is maintained constant, or again, none of the two wheels of such axis will be subjected to lateral slippage.

Another interpretation can be promoted by considering the angular velocity Ω of the vehicle around the ICR: to avoid slippage, every wheel must maintain the same angular velocity around ICR

$$\frac{v_{ij}}{d_{ij}} = \Omega$$

where d_{ij} is the distance from the ICR to the ij-th wheel. The parameter is later used within the velocity optimizer (see section 4.2 to compute wheels' velocities whenever an ICR is generated.
However this is only a particular case of the NHC: indeed they include also the case the vehicle is moving with a CWM; during such motion no ICR is generated.

As already stated in chapter 1, the UGV is designed with two ASM that produce a steering error. Consequently, the NHC cannot be easily respected and this means that a little error on all the four constraints is produced.
To consider these errors into the kinematic model, we have to discompose

velocities into their components with respect to $[V]$ as

$$\begin{cases} v_{ij_x}\big|_{[V]} = v_{ij} \cdot \cos(\delta_{ij}) \\ v_{ij_y}\big|_{[V]} = v_{ij} \cdot \sin(\delta_{ij}) \end{cases} \qquad with\ i = [F, R], j = [l, r]$$

and to compute their mean for left/right side (along x_m) and for front/rear axis (along y_m) we obtain

$$\begin{cases} \overline{v_l}x\big|_{[V]} = \dfrac{v_{Fl_x} + v_{Rl_x}}{2} \\ \overline{v_r}x\big|_{[V]} = \dfrac{v_{Fr_x} + v_{Rr_x}}{2} \\ \overline{v_F}y\big|_{[V]} = \dfrac{v_{Fl_y} + v_{Fr_y}}{2} \\ \overline{v_R}y\big|_{[V]} = \dfrac{v_{Rl_x} + v_{Rr_x}}{2} \end{cases}$$

Once obtained the mean velocities, which also consider the errors introduced by Ackermann steering mechanisms, we can study how the CoG M of the vehicle moves:

$$\begin{cases} \dot{x}_M\big|_{[V]} = \dfrac{\overline{v_l}x + \overline{v_r}x}{2} \\ \dot{y}_M\big|_{[V]} = \dfrac{\overline{v_F}y + \overline{v_R}y}{2} \\ \dot{\theta}_M\big|_{[V]} = \dfrac{\overline{v_F}y - \overline{v_R}y}{a + b} \end{cases} \tag{2.2}$$

The latter introduces the kinematic equations of motions in the frame $[V]$. To make things work we have to generalize these equations, thus they have to be moved into the global frame $(O, \hat{i}, \hat{j})$. To do this, we have to integrate the angular velocity $\dot{\theta}_M\big|_{[V]}$, which is the same both in frame $[V]$ and in frame $[G]$, thus obtaining the orientation of robot at time t

$$\theta_M(t) = \int_0^t \dot{\theta}_M dt \tag{2.3}$$

the latter is used to compute the rotation matrix $R^{[G]}_{[V]}$ as

$$R^{[G]}_{[V]} = \begin{bmatrix} \cos(\theta_M) & \sin(\theta_M) \\ -\sin(\theta_M) & \cos(\theta_M) \end{bmatrix}$$

Hence we can rotate the velocity vector [1] $\vec{v}_M = \begin{bmatrix} \dot{x}_M \\ \dot{y}_M \end{bmatrix}$ from the inertial frame $[V]$ to the global one $[G]$ as

$$\vec{v}_M\big|_{[G]} = R^{[G]}_{[V]} \cdot \vec{v}_M\big|_{[V]}$$

In this way we have obtained all three kinematic equations of motion with respect to the global frame $[G]$.

Finally, it's easy to compute the position of the UGV's CoG at time t as

$$X_M = \int_0^t \dot{x}_M\big|_{[G]} dt, \qquad Y_M = \int_0^t \dot{y}_M\big|_{[G]} dt \tag{2.4}$$

Naturally, to integrate the velocities obtaining the position/heading we need the initial conditions (X_{M_0}, Y_{M_0}) and θ_{M_0}: they represent the starting position/heading of the UGV and must be chosen correctly according to the path's starting point (see Section (3.1.1) to understand how the robot behaves changing its initial pose).

2.2 Dynamics of the motors

The UGV moves thanks to some electric motors mounted on the wheels and on the steering axes. To make these motors spin, they have to be controlled so that they can start still and stop still, while they reach maximum allowed speed in between these extremes. This behaviour is represented in Figure 2.2 where it's possible to see that the velocity profile has a trapezoidal shape; an initial positive acceleration is considered; when the maximum speed is reached, the acceleration is set to zero, and when the desired position is almost approached, the acceleration is set to its negative maximum.

2.2.1 Steering motors equations of motion

The behaviour of the electric motors applied on the steering axes follows the one described above: a desired angular position δ_{des} is set. The motor moves from its current position δ_{act} at time t_0 to the desired one at time t_f. To do this it firstly accelerates until its maximum speed:

$$\delta_{act}(t) = \delta_{act}(t_0) + \dot{\delta} \cdot (t - t_0) + \frac{1}{2}\ddot{\delta} \cdot (t - t_0)^2$$

[1] $\vec{v}_M$ is a physical vector. To pass from a coordinate system to another, it only has to be rotated and not translated.

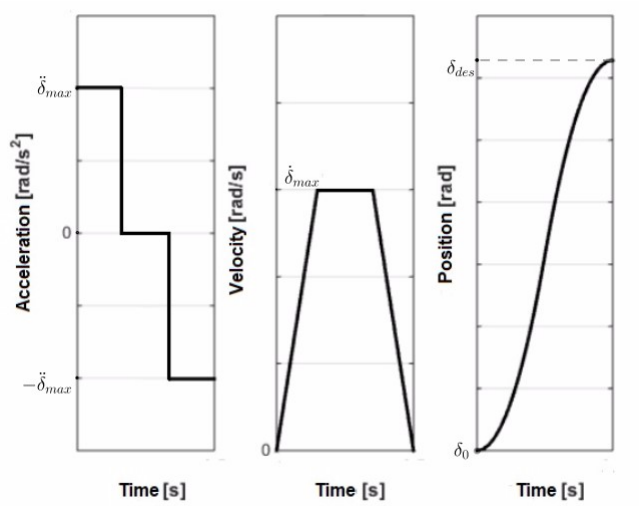


Figure 2.2: *Representation of the behaviour of an electric motor. Here is represented the case where the motor starts and stops still and reaches the maximum spin speed in between. Image taken from [11] and modified*

whereas when the maximum speed is reached $\dot{\delta} = \dot{\delta}_{max}$ at time $t = t_1$ we have

$$\delta_{act}(t) = \delta_{act}(t_1) + \dot{\delta_{max}} \cdot (t - t_1)$$

if $\delta_{des} - \delta_{act} < \dfrac{\dot{\delta}^2}{2\ddot{\delta}}$ (at time $t = t_2$) then the angular acceleration becomes negative until the angular velocity $\dot{\delta}$ returns to zero when $\delta_{act} = \delta_{des}$ at time $t = t_f$:

$$\delta_{act}(t) = \delta_{act}(t_2) + \dot{\delta} \cdot (t - t_2) - \frac{1}{2}\ddot{\delta} \cdot (t - t_2)^2$$

This cycle is repeated every time $\delta_{act} \neq \delta_{des}$.

2.2.2 Wheels motors equations of motion

To simulate the behaviour of the electric motors mounted on the wheels we have to consider that the trapezoidal velocity profile of the UGV is a global consideration: once the vehicle starts, it moves at constant speed until it is close enough to the final point of the path (see section 3.3). This means that each electric motor have to run at constant speed $\omega_{ij_{act}}$ unless the controller requires a different velocity $\omega_{ij_{des}}$ [2] to the single wheel (see section 4.2). During

[2] $\omega_{ij_{act}} = \omega_{act}$ and $\omega_{ij_{des}} = \omega_{des}$ in 2.5

a single cycle from t_0 to t, the behaviour of each motor ($\omega \equiv \omega_{ij}$) is described as follows:

$$\omega_{act}(t) = \begin{cases} \omega_{act}(t_0) + \dot{\omega} \cdot (t - t_0), & if\ \omega_{act}(t_0) < \omega_{des}(t) \\ \omega_{act}(t_0), & if\ \omega_{act}(t_0) = \omega_{des}(t) \\ \omega_{des}(t_0) - \dot{\omega} \cdot (t - t_0), & if\ \omega_{act}(t_0) > \omega_{des}(t) \end{cases} \tag{2.5}$$

The actual angular velocity $\omega_{ij_{act}}$ is then converted into a linear one:

$$v_{ij} = v_{ij_{act}} = \omega_{ij_{act}} \cdot R$$

and used into the equations of motion of the kinematic model.

2.3 Description of the UGV

In this section all the features of the UGV designed by the DISAFA (DIpartimento di Scienze Agrarie, Forestali, Alimentari) of the Università di Torino are reported.

Figure 2.3: *Pictures of the UGV: a) and b) are photos taken from different angles; c) is a detail of the front ASM and the electric motors applied on the steering axis and on the wheels.*

The dimensions of the vehicle are:

- $L = 1.5\,m \Rightarrow$ Distance between the center of front and rear wheels

(Length of the UGV);

- $T = 1\,m \Rightarrow$ Distance between the center line of each of the two wheels on the same axis (Width of the UGV);

- $a = 0.75\,m \Rightarrow$ Distance between the center of front wheel and the CoG of the UGV;

- $b = L - a = 0.75m \Rightarrow$ Distance between the CoG of the UGV and the center of rear wheel.

Wheels implemented on the UGV have the following dimensions:

- $R = 254\,mm \Rightarrow$ Wheel radius;

- $W = 203\,mm \Rightarrow$ Wheel width.

The electric motors installed on each wheel have the following characteristics:

- $\omega_{max} = 8.38\,rad/s \Rightarrow$ maximum angular velocity of the wheels' motors ;

- $\dot{\omega}_{max} = 1.40\,rad/s^2 \Rightarrow$ maximum angular acceleration of the wheels' motors;

- $v_{max} = \omega_{max} \cdot R = 2.13\,m/s = 7.67\,km/h \Rightarrow$ maximum speed of the CoG of the vehicle.

The steering axes mount electric motors with the following features

- $\dot{\delta}_{max} = 1.87\,rad/s \Rightarrow$ maximum angular velocity of the motors applied on the steering axes;

- $\ddot{\delta}_{max} = 1.56\, rad/s^2 \Rightarrow$ maximum angular acceleration of the motors applied on the steering axes;

- $\delta_{max} = \pm 23^\circ = \pm 0.401\, rads \Rightarrow$ steering range.

These data are used in the modelling of both kinematic features of the vehicle and dynamic characteristics of the motors.

Chapter 3

The navigation system

The aim of the navigation system is to give informations about the path the vehicle has to follow and about the velocity it has to maintain.
The path to be tracked by our UGV is described by a series of N points. Each point $C(k)$, with $k \in [1, N]$, is defined by (X_C, Y_C) coordinates and is associated with an orientation θ_C with respect to the Global frame $[G]$. With these informations we are able to associate a mobile reference frame $(C, \hat{p}, \hat{q})$ for each point belonging to the path; this means that the frame changes depending on which point is chosen: such a frame is called Frenet Frame $[F]$, as widely explained in [10].

3.1 Assignment of the Frenet frame to C(k)

For each cycle the informations about the position (X_M, Y_M) of the vehicle are received from the kinematic model, computed with Eq. (2.3) and (2.4). The main aim of the navigation system is to analyse the position of the robot and, depending on these data, to choose which is the best point belonging to the path where the Frenet frame has to be assigned. This is done by computing the x- and y- coordinates of the CoG of the UGV with respect to frame $[F]$ assigned to a bunch of points (for example, the analysed points are from $C(k-1)$ to the chosen one); let's call these coordinates $(s, d)\big|_{[F]}$. If the vehicle's x-coordinate (with respect to $[F]$) is less than $0.5m$ far from a certain point ($s(C(k-1)) \leq -0.5m$, then the next one is chosen as reference (in the previous example, $C(k)$ will be our reference) , and the following ones are no more analysed. The space considered $s = -0.5m$ is chosen so that the vehicle has enough room for manoeuvre to understand how it has to behave and to

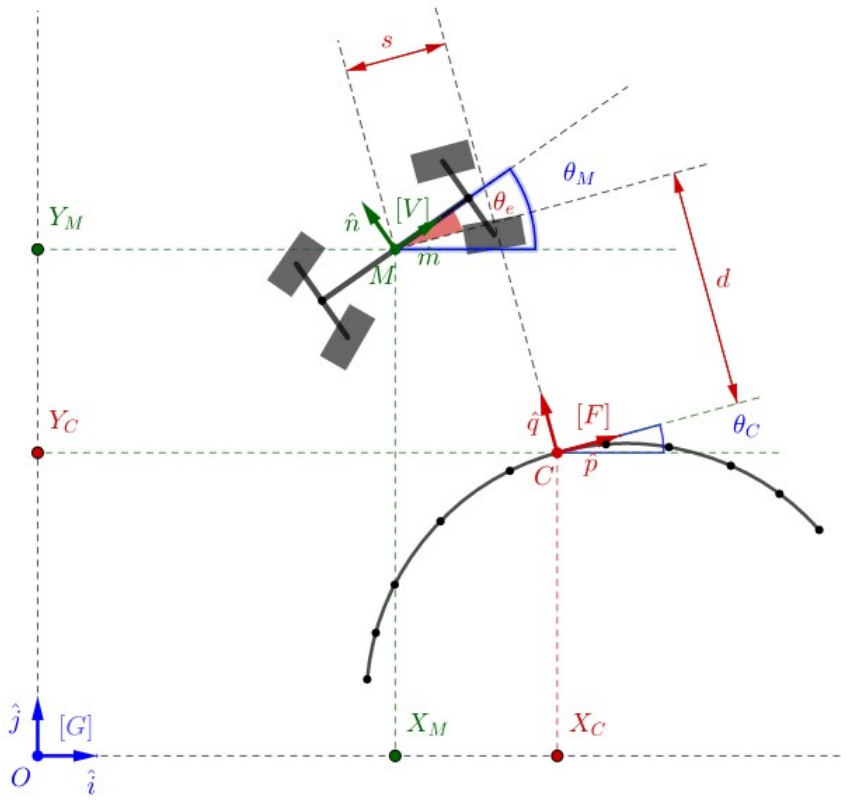


Figure 3.1: *How the vehicle approaches the path: the navigation system uses the longitudinal distance s as reference to change the frame position, while d and θ_e are returned as outputs*

make motors work in advance before it approaches the reference point.

Let's analyse what happens if point $C(k)$ (called C for simplicity) is chosen as reference (see Figure 3).
The assigned frame $[F]$ will have origin in C: $\overline{OC} = (X_C, Y_C)$ and orientation θ_C with respect to global frame $[G]$; to compute the position (s, d) of the vehicle we have to change the reference of the position vector $\overline{OM} = (X_M, Y_M)$ from the global frame to the Frenet one: $\overline{CM} = (s, d)\big|_{[F]}$:

$$\overline{CM} = \overline{OM} - \overline{OC} \tag{3.1}$$

where:

$$\overline{CM} = \begin{bmatrix} s \\ d \end{bmatrix}\Bigg|_{[F]} \quad \overline{OM} = \begin{bmatrix} X_M \\ Y_M \end{bmatrix} \quad \overline{OC} = \begin{bmatrix} X_C \\ Y_C \end{bmatrix}$$

In mathematical terms, a simple roto-translation of the position vector $\overrightarrow{p_M}$ is applied:

$$\overrightarrow{p_M}\big|_{[F]} = R^{[F]}_{[G]} \cdot (\overrightarrow{p_M}\big|_{[G]} - \overrightarrow{p_C}\big|_{[G]}) \tag{3.2}$$

where position vectors are

$$\overrightarrow{p_M}\big|_{[F]} \equiv \overline{CM}; \quad \overrightarrow{p_M}\big|_{[G]} \equiv \overline{OM}; \quad \overrightarrow{p_C}\big|_{[G]} \equiv \overline{OC};$$

and rotation matrix is

$$R^{[F]}_{[G]} = \begin{bmatrix} \cos(\theta_C) & -\sin(\theta_C) \\ \sin(\theta_C) & \cos(\theta_C) \end{bmatrix} \tag{3.3}$$

Once the roto-translation is fulfilled, the x-coordinate of $\overrightarrow{p_M}\big|_{[F]}$ is analysed. The output of this script will be the y-coordinate of $\overrightarrow{p_M}\big|_{[F]}$ (i.e. longitudinal distance d) and the orientation θ_C with respect to the global frame $[G]$.
Once the vehicle has performed a sufficient displacement so that $s(C(k)) > -0.5m$, then $C(k+1)$ will be chosen as reference for the next cycle and this procedure is repeated until it has not reached the last point, or goal, of the path.

3.1.1 The starting point

A different approach is followed if the simulation is at its first cycle. The vehicle, indeed, assumes different behaviours based on which is its initial position. The first thing to do is to compute the absolute distance from the UGV to each point of the path as

$$d_{abs} = \sqrt{(X_M - X_C(k))^2 + (Y_M - Y_C(k))^2} \tag{3.4}$$

There are two possible cases:

- the vehicle is close enough to one of the points belonging to the path $C(k)$;
- the vehicle is too far from the path.

In the first case, the orientation is immediately considered as the required one:$\theta_C = \theta_C(k)$; no matter the heading of the vehicle, the latter starts immediately the path tracking.
In the second one, the UGV has to get closer to the path before starting to track it; the starting point $C(1)$, in this specific case, is the target of the Navigation system. The vehicle approaches the starting point of the path with a straight line. This is made possible by computing the orientation (in the Global frame) of such straight line as

$$\theta_C = \arctan\left(\frac{Y_M - Y_C(1)}{X_M - X_C(1)}\right) \tag{3.5}$$

Such orientation is used to compute rotation matrix (3.3) then used in the roto-translation fulfilment. Once the vehicle is at distance $dabs \leq d_{min}$ then the navigation system starts following the main algorithm.

When approaching this case, the navigation system does not consider any obstacle; this means that, to avoid any accident, the robot has to be correctly positioned before starting the autonomous guidance.

3.2 Computation of the angular deviation

To make the robot track the path, the controller has to receive two data: how much the robot is far from path and how much its heading is shifted with respect to the one required from the reference point $C(k)$. These data are provided by the navigation system: the position error, or lateral deviation, d is directly provided once the "best" point $C(k)$ is chosen, while the orientation error, or angular deviation, θ_e, that has to be computed.

To compute the angular deviation we need to take care of two different considerations:

- is there a specific angle the vehicle has to maintain with respect to the path?
- what if the orientation error is so big ($|\theta_e| \geq 2\pi$) that the UGV starts to turn into a round?

The first problem is easy to solve: the angular deviation is a feedback that considers the heading of the vehicle with respect to the orientation required to

the chosen point in that loop; to make the UGV follow a different angle it's sufficient to add the crab angle to the reference one as

$$\theta_e = (\theta_C + \theta_{crab}) - \theta_M \tag{3.6}$$

then the crab angle must be sent to the controller to keep it in consideration (Eq. (4.2)).

The second issue can be solved by adding some other considerations: the aim is limiting heading error in between two values that can describe all the possible orientations the vehicle can assume; such values therefore must be able to include each orientation error in between the interval $[0, 2\pi]$.
To prevent the UGV from always steering to the same side, it's better to shift the aforementioned interval into positive and negative values (see Figure 3.2):

$$\theta_e \in [-\pi, \pi] \tag{3.7}$$

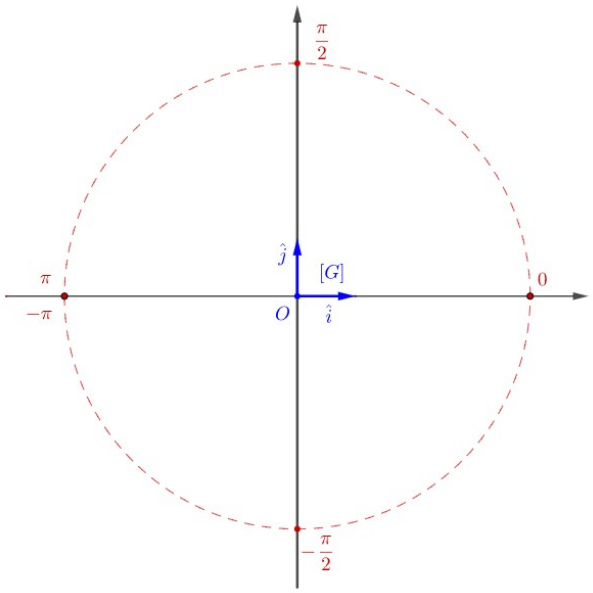


Figure 3.2: *Global Reference frame showing how the orientation error θ_e is manipulated so that the controller won't make the vehicle turn in a round.*

The way to achieve such a purpose is to verify which value assume the orientation error θ_e and, if it exceeds the aforementioned values, to translate such value by adding or subtracting 2π until it's not included in it:

$$\theta_e = \begin{cases} \theta_e + 2k\pi \ \ if \ \ \theta_e < -\pi \\ \theta_e - 2k\pi \ \ if \ \ \theta_e > \pi \end{cases} \qquad k \in \mathbb{N} \tag{3.8}$$

3.3 The choice of v_d

The vehicle, tracking the path, is supposed to start still and to approach the end of the path with a null velocity. In the middle, it has to maintain the velocity v_{manual} that is decided during the setup. At the begin, the navigation system sets the desired velocity $v_d = v_{manual}$, thus, the UGV accelerates until such velocity is reached, and it is maintained during the whole path. When approaching the end of the path, the vehicle must brake. This means that the desired velocity must be set to zero: $v_d = 0$.

Braking the vehicle requires a certain distance d_{brake} defined as

$$d_{brake} = -\frac{v_M^2}{2a_M} \tag{3.9}$$

where v_M is computed as magnitude from the components of the vehicle's CoG taken from Eq. (2.2) [1]:

$$v_M = \sqrt{\dot{x}_M^2 + \dot{y}_M^2}$$

and a_M is the acceleration of the UGV and it is supposed to be the same as the wheels' ones (the acceleration will be negative because the vehicle is braking) where

$$a_M = \dot{\omega} \cdot R$$

To stop in correspondence to the goal, the vehicle has to know the distance d_{end} from its position to the destination, i.e. the last point $C(N)$ belonging to the path. considering the point C(k) as the one chosen as reference by the navigation system, the length of the path that is not been tracked yet is given by

$$d_{end} = \sum_{i=k}^{N-1} \left(\sqrt{(x_C(i) - x_C(i+1))^2 + (y_C(i) - y_C(i+1))^2} \right) \tag{3.10}$$

This means that the vehicle should be able to choose between $v_d = v_{manual}$ manually set and a null velocity $v_d = 0$ and this change of happens when the distances described right above are the same:

$$v_d = \begin{cases} v_{manual} \;\; if \;\; d_{end} > d_{brake} \\ 0 \;\; if \;\; d_{end} \leq d_{brake} \end{cases} \tag{3.11}$$

[1] remember that $\vec{v_M}$ is a physical vector, thus its magnitude does not change from a a reference frame to another

A trapezoidal velocity profile of the UGV's CoG during the whole path tracking is obtained(see Figure 3.3).

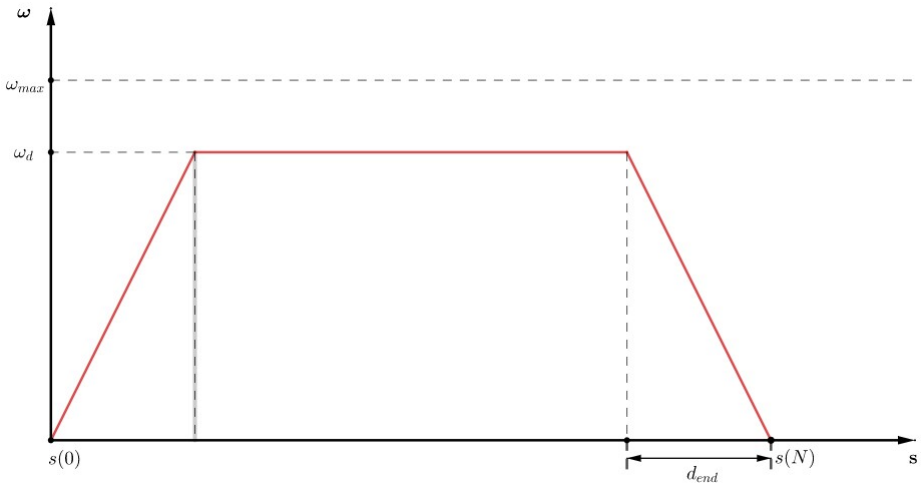


Figure 3.3: *Trapzoidal velocity profile of the CoG of the robot. s is the curvilinear abscissa that determines the distance travelled along the path. When the length of the path to be tracked is equal to the distance the vehicle requires to brake, the navigation system switches the desired velocity to zero.*

Chapter 4

The controller

The UGV's kinematic model receives two different inputs: the desired steering angle $\delta_{ij_{des}}$ of the wheels and the linear velocity $v_{ij_{des}}$ each wheel should maintain. These inputs are decided by the controller that works with two different modules:

- a steering controller that receives the position d and steering θ_e errors from the navigation system (see Section 3.3) and computes the angle $\delta_{F_{des}}$ and $\delta_{R_{des}}$ to be sent to the steering motors applied on each axis to bring these errors to zero;
- a velocity optimizer that measures the current axes' steering angles $\delta_{F_{act}}$ and $\delta_{R_{act}}$ and computes the best velocity to be assigned to each wheel in order to minimize the NHC errors depending on the desired velocity v_d the vehicle should maintain.

4.1 Steering controller

Path following is the main purpose of the steering controller; it can be achieved by controlling the two steering axes, assigning them the angles $\delta_{F_{des}}$ and $\delta_{R_{des}}$ [1].

The controller receives two inputs from the navigation system:

- The lateral deviation d of the vehicle's CoG from the path,
- The angular deviation θ_e between the orientation required by the path and the vehicle's heading.

[1] to make things easier, these angles are defined as δ_F and δ_R

Path tracking is achieved once these two inputs are reduced to zero.

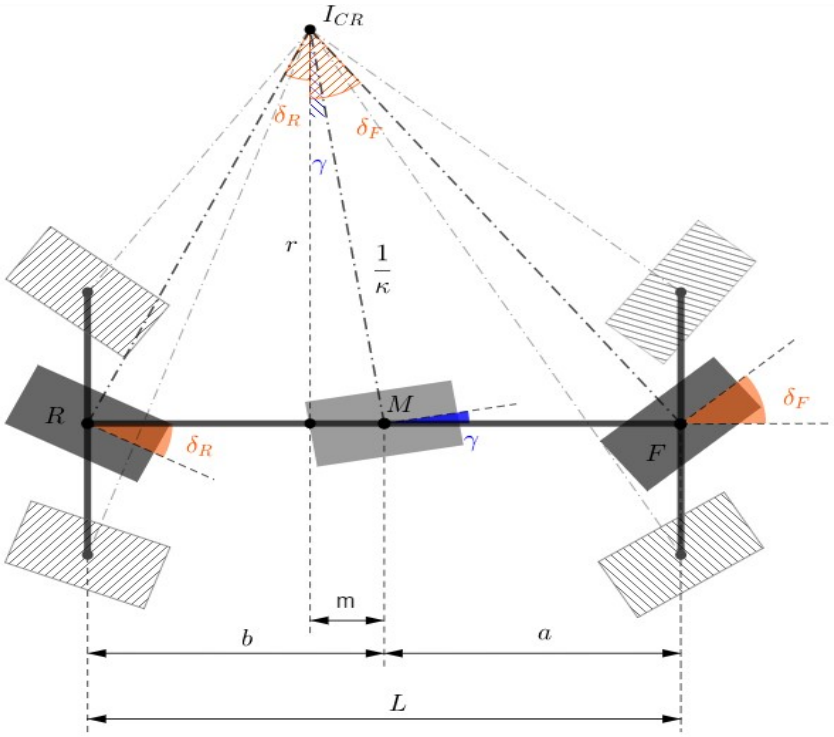


Figure 4.1: *A virtual wheel is applied on the CoG of the vehicle; by acting on it, the controller is able to assign the correct angles to both front and rear axes to correct its trajectory and make it follow the path correctly*

Referring to Figure 4.1, a virtual steering wheel is positioned into the CoG M of the UGV. This steering angle is called γ. This is used to correct the position error d by steering both front and rear axes by the same angle, i.e.

$$\delta_{F_{des}} = \delta_{R_{des}} = \gamma$$

This means that the vehicle is moving with a CWM so that the lateral velocity component is opposite with respect to distance d (it can be positive or negative, see Section 3.1).
The idea is to convert the input d into a steering command γ_c, with an appropriate proportional gain G_d; the bigger the gain the more severe is the lateral motion, as defined hereafter

$$\gamma_c = -G_\gamma \cdot d \tag{4.1}$$

The more the distance vehicle-path, the bigger will be the steering angle γ_c.

If a crab angle θ_{crab} is required during the tracking of the path, this is kept into account in the navigation system when computing the orientation error θ_e (see Section 3.2). To make the vehicle fulfill a CWM, the angle θ_{crab} is subtracted to the steering command γ_c as

$$\gamma = \gamma_c - \theta_{crab} \tag{4.2}$$

in this way the vehicle maintains a certain heading with respect to the orientation required by the path.

On the other hand, to correct the angular deviation θ_e: it is not possible to act directly on the virtual wheel, but we have to act on the front and rear steering angle individually. To do this it's convenient to exxploit the curvature parameter κ. It is defined as the reciprocal of the distance of the reference point M from the ICR of the vehicle, i.e.

$$\overline{MI}_{CR} = \frac{1}{\kappa}$$

Again, a proportional gain G_κ is used to transform the deviation $theta_e$ into curvature κ:

$$\kappa = -G_\kappa \cdot \theta_e \tag{4.3}$$

The bigger the orientation error, the greater will be the curvature κ.

To compute the steering angles δ_F and δ_R we have to consider that the rotation axis of each wheel, front, rear and virtual ones, passes through the ICR. By looking at Figure 4.1), it's possible to notice that the rotation axes of the wheels and the body of the vehicle form several right triangles, thus it's possible to apply some trigonometric relationships as in the follows

$$\begin{cases} b - m = r\tan(\delta_R) \\ a + m = r\tan(\delta_F) \\ m = r\tan(\gamma) \\ r = \dfrac{1}{\kappa}\cos(\gamma) \end{cases}$$

Rearranging the previous system we are able to compute the steering axes angles as function of the curvature κ and the steering angle γ applied on the

virtual wheel[2]:

$$\begin{cases} \delta_{F_{des}} = \arctan(\dfrac{a\kappa + \sin(\gamma)}{\cos(\gamma)}) \\ \delta_{R_{des}} = \arctan(\dfrac{b\kappa - \sin(\gamma)}{\cos(\gamma)}) \end{cases} \tag{4.4}$$

In conclusion the steering controller is a non linear proportional one.

After several tests, the correct values for the gains have been obtained in order to fulfil a correct path tracking (the path must be designed to respect the physical constraints of the robot).

4.2 Velocity optimizer

The NHC defined in section 2.1 cannot be respected only by considering the ICR and computing velocities as a function to the desired velocity manually imposed. Another method to compute velocities is required.

Let's first define the same reference frame $R, \hat{m}, \hat{n}$ we already saw in Figure 1.4: its origin is placed on R and its x-axis is along the length of the vehicle. In the case there exists an ICR ($\delta_{F_{act}}$ and $\delta_{F_{act}}$ are not equal), we can compute the theoretical one, given by the intersection among the actual steering angles. Its coordinates defined will be (x_{ICR}, y_{ICR}) with respect to the reference frame just defined.
It's now possible to compute the distance of each wheel from the ICR (the position of each wheel is known):

$$r_{ij} = \sqrt{(x_{ICR} - x_{ij})^2 + (y_{ICR} - y_{ij})^2}$$

The NHC in the case just analysed will be respected if every wheel maintains an angular velocity around the ICR $\Omega_{ij} = \dfrac{v_{ij}}{r_{ij}}$ that is the same of the angular velocity of the vehicle's body $\Omega_{des} = \dfrac{v_d}{y_{ICR}}$, but we also know that the double Ackermann steering mechanism generates an orientation error such that the NHC cannot be reduced to zero.

On the other hand, if the steering angles of the axes are equal, the easiest method to avoid any slippage on the wheels is to assign to each wheel the desired velocity the vehicle should maintain: $v_{ij} = v_d$; but again the NHC are

[2] γ and κ are calibrated so that the steering range of the motors applied on the ASM are respected.

not respected due to the ASM (see Figure 1.3): wheels placed in the diagonal of the vehicle are parallel, while they are not if considering the same axis.

The most efficient way to find the best velocity to be assigned to each wheel is to use quadratic programming, taking into account both the conditions above and the NHC. The quadratic programming is the process of solving a linearly constrained quadratic optimization problem, as described in [8].

The first thing to do is to set up the problem in order to use correctly this powerful tool. The quadratic optimization problem can be written as:

$$\left|\left|\alpha \cdot v - \beta\right|\right|^2 \tag{4.5}$$

where v is the variable to be optimized defined as

$$v = \begin{bmatrix} v_{Fl} \\ v_{Fr} \\ v_{Rl} \\ v_{Rr} \end{bmatrix}$$

while α and β are defined in different ways depending whether $\delta_{F_{act}}$ and $\delta_{F_{act}}$ are parallel or not. In the first case:

$$\alpha = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}, \quad \beta = 4 \cdot v_d$$

while in the second one:

$$\alpha = \begin{bmatrix} \frac{1}{r_{Fl}} & \frac{1}{r_{Fr}} & \frac{1}{r_{Rl}} & \frac{1}{r_{Rr}} \end{bmatrix}, \quad \beta = 4 \cdot \Omega_{des}$$

Vector v is constrained by:

$$|A \cdot v| \leq b \tag{4.6}$$

These constraints are the NHC: A is taken from Eq. (2.1), writing the trigonometric data in matrix form while b is a vector defining the boundary limits that substitute the zeros:

$$A = \begin{bmatrix} \cos(\delta_{Fl}) & 0 & -\cos(\delta_{Rl}) & 0 \\ 0 & \cos(\delta_{Fr}) & 0 & -\cos(\delta_{Rr}) \\ \sin(\delta_{Fl}) & -\sin(\delta_{Fr}) & 0 & 0 \\ 0 & 0 & \sin(\delta_{Rl}) & -\sin(\delta_{Rr}) \end{bmatrix}$$

and:

$$b = \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix}$$

ε is a variable small enough to make wheel not to slip, but big enough to make possible to find solutions in each and every situation. Its order of magnitude is:

$$\varepsilon = 10^{-3} \div 10^{-2}$$

Chapter 5

Testing and simulation results

A lot of tests have been done to in order to obtain the best tracking on the given paths. Such tests consist of:

1. assign to the navigation system different kinds of paths, some with smooth curves, other with more severe ones;
2. set the initial position of the vehicle in various points along the path and see if the algorithms of the starting point designed in the navigation system work properly;
3. give any kind of initial orientation to verify if the robot is able to return to the desired position given by the path;
4. combine the last two point and verify if there are singularities in the algorithms;
5. set up different desired velocities to verify if the robot is able to follow the path if it requires motor speeds that are close to the maximum ones;
6. assign the field-like paths to see if the UGV is able of moving among narrow spaces without colliding with any obstacle.

The first 4 points have been used to find the best values of the gains G_d and G_θ of the steering controller and of the value ε that constitute the upper and lower bound for the NHC used in the velocity optimizer. Among these ones, the fourth point is the most important one: once the values of the gains and the bounds are found, it was important to understand which are the initial conditions where the steering range δ_{max} of the ASM do not cause any

problem. Indeed, this was useful to set a bound of d_{min} in the algorithm defined in Section 3.1.1 after which the navigation system starts to follow the main algorithm (the one defined in Section 3.1). is able to re-orientate itself in order to come back onto the start of the path, no matter its initial position or orientation.

The fifth point was exploited to understand the limits of the motor dynamics: as you increase the desired velocity, the steering motors require an higher angular acceleration $\ddot{delta}$ to achieve path tracking. The higher the v_d the more the errors during the tracking of the path, specially when severe manoeuvres are required.

The best values for the gains and the bounds are found after many tests and they are inserted as in the follow

- $G_\theta = 8$ steering controller gain that controls the heading of the UGV;
- $G_d = 0.7$ steering controller gain that controls the lateral deviation of the UGV with respect to the path (see Section 4.1);
- $\varepsilon = 0.005$ upper and lower bound for the NHC defined in the velocity optimizer (see Section 4.2), this value allows to always find a solution to the optimization problem;
- $d_{min} = 1.5\,m$ distance bound where the navigation system switches from the "starting point" algorithm to the main one (see Section 3.1);
- $v_{d_{max}} = 1.5\,m/s$ maximum desired velocity after which the vehicle does not manage to follow the path correctly.

In conclusion, the UGV was tested on paths designed to fit into field-like environments: such paths have been generated by the algorithm developed by Ing. Mohammadreza Beygfard. This algorithm keeps into account all the mechanical features of the vehicle and it is designed to generate paths on a given map (representing the environment where the vehicle has to move in), given the starting point and the goal; it considers any kind of obstacle within the map and produces a series of points that constitute the path so that the vehicle is able to start from the starting point and reach the goal without colliding any obstacle.

In the next sections are represented the results of the simulation environment tested on two different paths generated by the aforementioned algorithm. The monitored data are :

- the behaviour of the vehicle following the path, i.e. position and orientation of vehicle and wheels.
- the speed of the vehicle CoG during the whole tracking, useful to see whether the UGV is able to maintain the desired velocity set;
- the linear velocity of each wheel used to understand how well the velocities required by the velocity optimized are differentiated;
- the steering angles of both axes, to see whether the limits applied for the curvature κ and the virtual steering angle γ comply with the steering range of the steering motors or not.

5.1 Simulation results on path 1

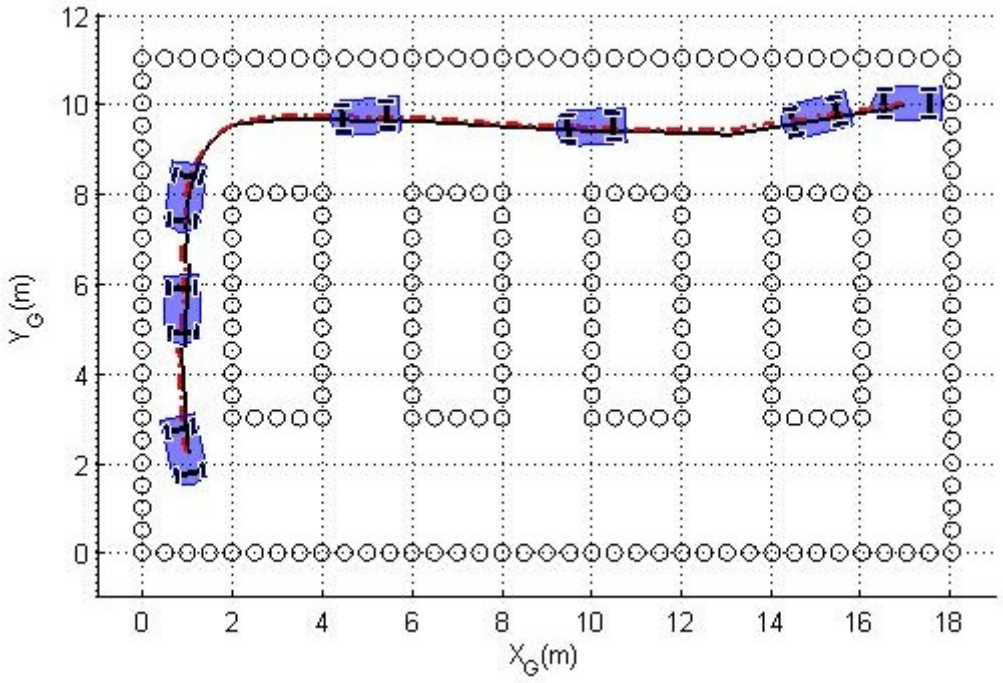
In the first on-field test the robot is set to maintain a desired velocity $v_d = 0.5\,m/s$ and a heading $\theta_{crab} = 5^\circ = \pi/36\,rad$ during the whole tracking of the path. The UGV is positioned right at the start of the path to be tracked

$$\begin{bmatrix} X_{M_0} \\ Y_{M_0} \end{bmatrix} = \begin{bmatrix} X_C(1) \\ Y_C(1) \end{bmatrix} = \begin{bmatrix} 17 \\ 10 \end{bmatrix}$$

and its heading is the same of the one required on $C(1)$:

$$\theta_{M_0} = \theta_C(1) = \pi$$

In Figure 5.1 we can observe how the UGV manages to follow the path without any particular change of trajectory, maintaining a CWM during the whole tracking:



[h]

Figure 5.1: *UGV tracking the first path: the vehicle's initial position and heading are the same of the path's start. A small CWM is required and the vehicle achieves such purpose easily.*

It's possible to see in Figure 5.2 that the vehicle starts still and maintains the desired velocity during the whole path, even iv a CWM is required; at the end it starts to brake in order to approach the goal with null velocity.

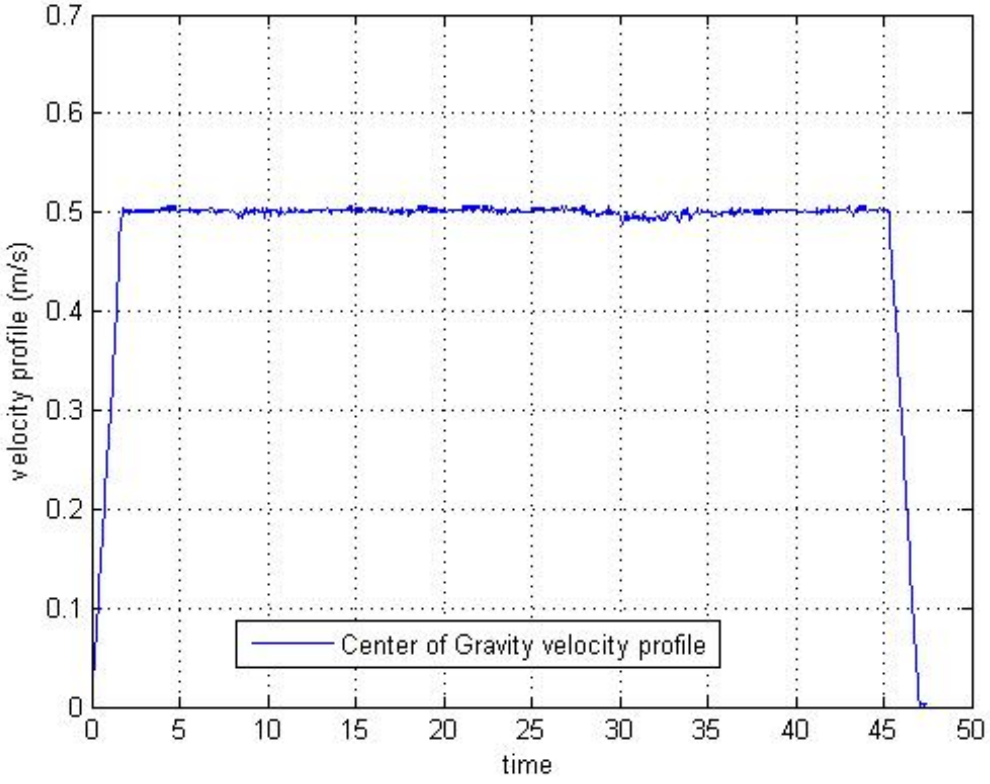


Figure 5.2: *Vehicle's CoG velocity profile of the first path; the speed is maintained more or less constant during the whole tracking of the path. Moreover the vehicle starts with a null velocity and stops in correspondence with the goal.*

Figure 5.3 shows how the velocity of each wheel is adapted both in linear strokes (there is a slight difference among the wheels velocities) and during the curves (the difference among wheels velocities increases). The wheels start still and stop still, as the vehicle's CoG

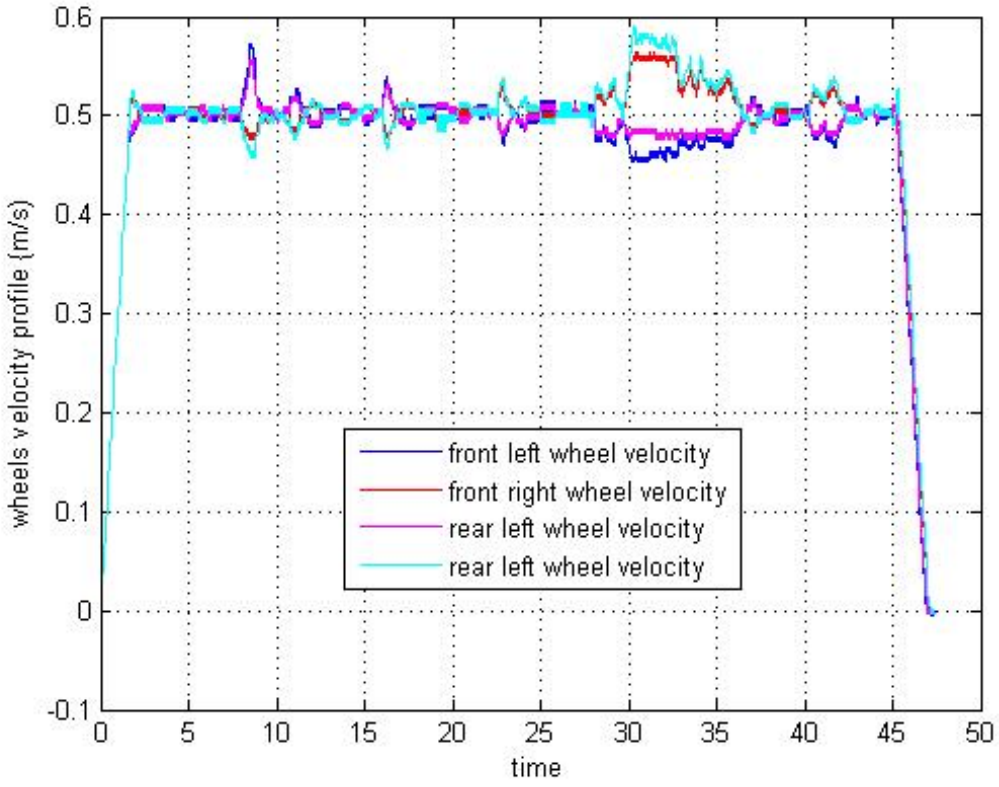


Figure 5.3: *wheels' velocity profiles of the first path: it's possible to see how each wheel's velocity is chosen so that the velocity of the CoG is maintained constant. The NHC are kept in between the bounds ε along the whole path*

Last but not least, we can see in Figure 5.4 the behaviour of the steering motors: they hold a certain heading with respect to the body frame of the UGV during the linear strokes of the path ($\delta_F \simeq \delta_R \simeq -0.1\,rad$) in order to fulfill the CWM; moreover they keep their value in between the allowed steering range $\delta_{max} = \pm 23^\circ = \pm 0.401\,rad$ when approaching the curves.

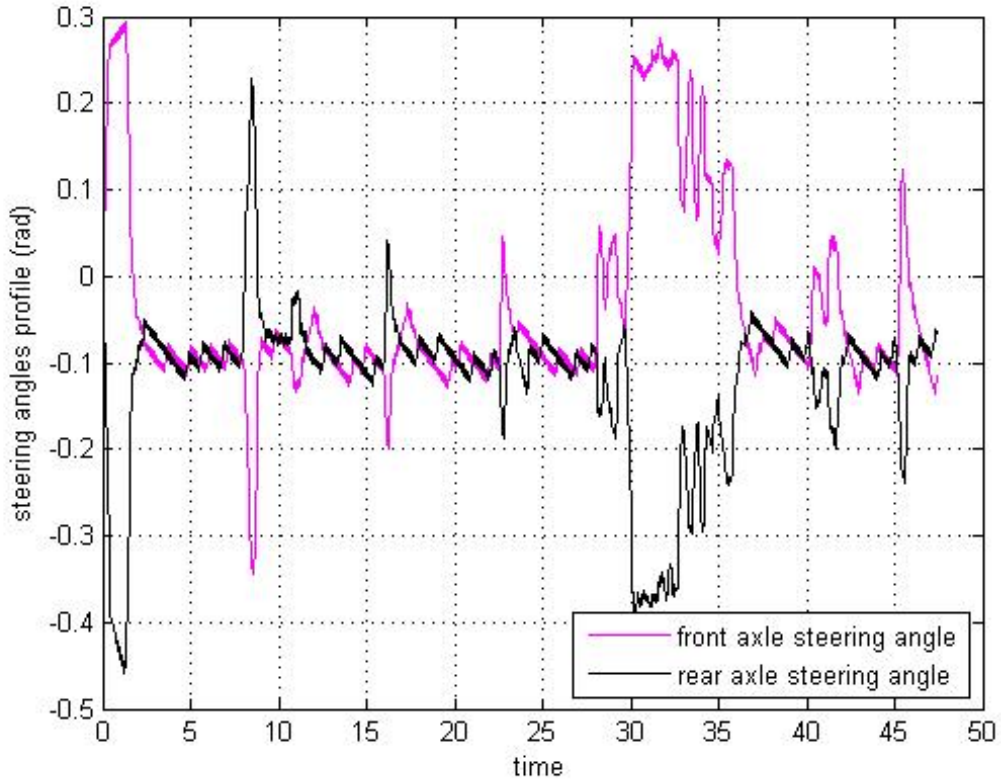


Figure 5.4: *Steering angle profiles: both angles are kept in between the steering range of the electric motors during the tracking. A Crab angle is maintained in the straight parts of the path to allow the CWM.*

5.2 Simulation results on path 2

The second on-field path is a bit more tricky: distances among obstacles are less than the previous case, thus the manoeuvres to be effectuated are more severe.

This time, the UGV starts with a different position

$$\begin{bmatrix} X_{M_0} \\ Y_{M_0} \end{bmatrix} = \begin{bmatrix} 17 \\ 9 \end{bmatrix}$$

and heading

$$\theta_{M_0} = \pi$$

with respect to the path's start

$$\begin{bmatrix} X_C(1) \\ Y_C(1) \\ \theta_C(1) \end{bmatrix} = \begin{bmatrix} 18 \\ 8 \\ -\pi/2 \end{bmatrix}$$

and has to maintain a null crab angle $\theta crab = 0$. Therefore, the desired velocity is greater than the one maintained along the first path $v_d = 1\, m/s$.

In Figure 5.5 it's possible to see that the first objective of the navigation system is the first point of the path: the UGV reaches the path's start with the the desired heading; once the path is approached, the vehicle tracks it correctly: small deviations occur along the most severe curves(like the one in the first turn), but the trajectory is rapidly corrected by the controller. After all, the vehicle is able to track this second path smoothly:

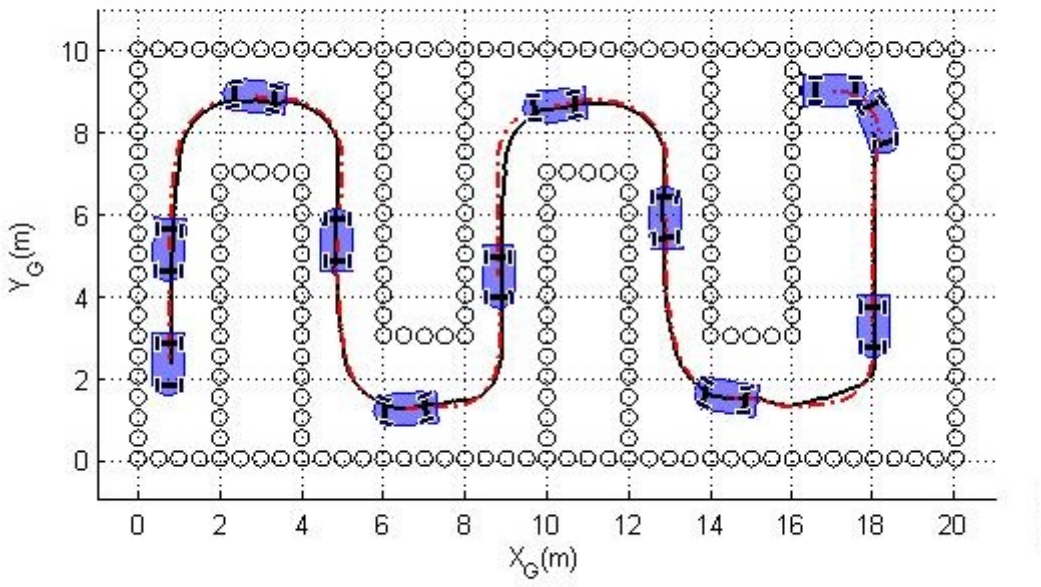


Figure 5.5: *UGV tracking the second path: the vehicle's starting point is close to the path, with a certain lateral and angular deviation. The navigation system gives the correct data to the controller which makes possible to approach the path's start avoiding any kind of abrupt manoeuvre; once the path is reached, the tracking is smooth until the end*

As in the previous case, the vehicle is able to maintain a more or less constant velocity during the entire tracking of the path, starting still and approaching the goal still:

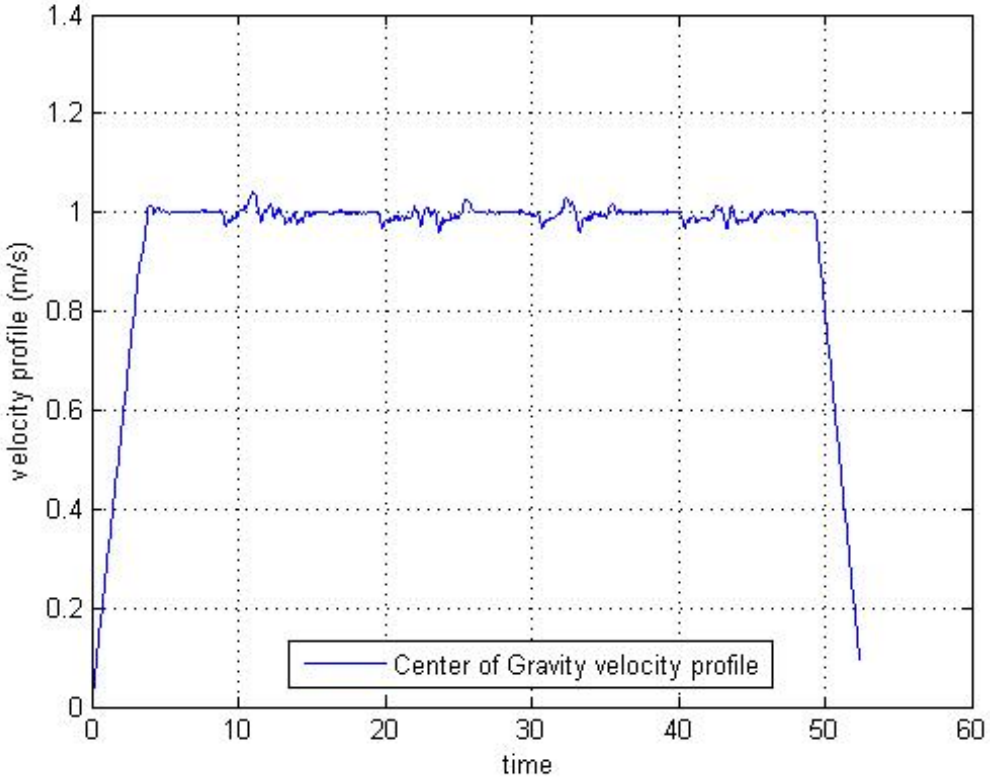


Figure 5.6: *Vehicle's CoG velocity profile; it complies the trapezoidal profile: starts still and approaches the goal still.*

Being the second path a series of "S" turns, the velocity optimizer assigns to each wheel values that change along every curve: the inner wheels are slower than the outer ones, no matter if the curve is to the right or to the left.

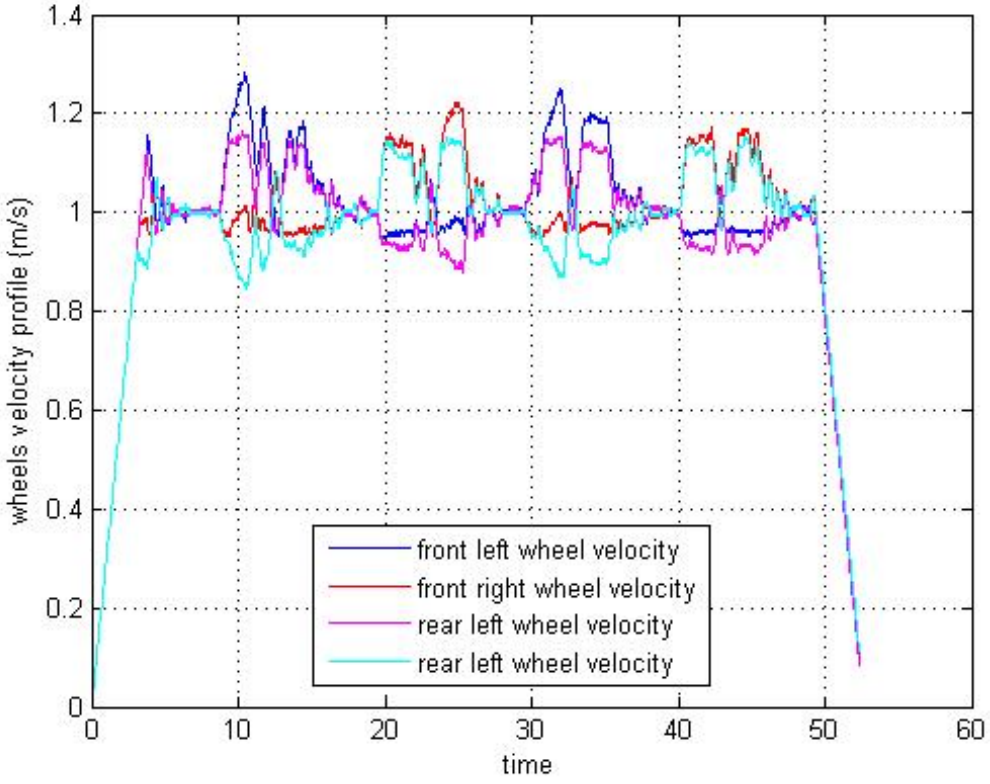


Figure 5.7: *wheels' velocity profiles: the Velocity Optimizer chooses the best velocity to be assigned to each wheel in order to comply with the NHC and to maintain the CoG velocity constant along every turn*

In conclusion, the steering profile of the second path shows how the vehicle is able to maintain its trajectory by keeping its steering angles almost always between the range allowed by the motors.

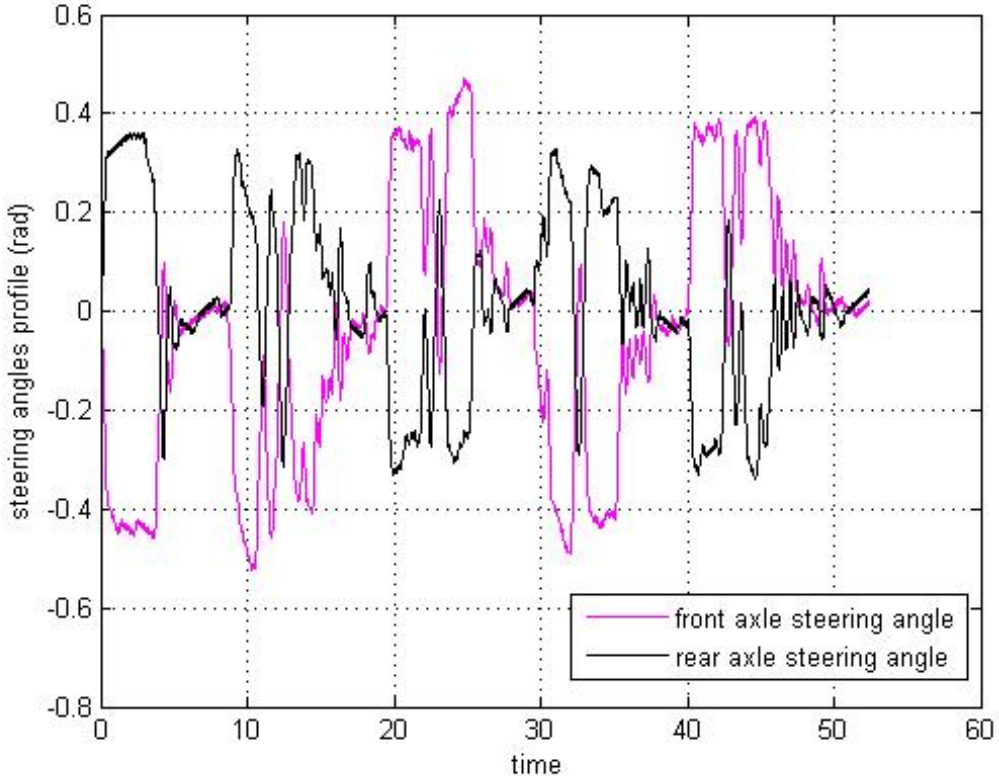


Figure 5.8: *Steering angle profiles: both angles are almost always bounded between the physical constraints; the only peaks that exceed the allowed range are in correspondance to the most severe curves.*

Conclusion and future work

This thesis focused on the design of a controller able to make the vehicle tracking a given path and to optimize the velocities to be assigned to each wheel in order to minimize the errors due to the ASM. During the simulation campaign, the 4WS model of the UGV has proven to be capable of simulating its motion and suitable for path tracking, thanks to the navigation system able to provide the required data to the controller. In particular, the controller, thanks to a long process of testing along various simple paths, has been configured in the best way possible and it has shown its capability of pursuing the target. Last but not least, the optimal paths for driving the UGV in vineyard-like scenarios, that have been generated by the algorithm of Mohammadreza Beygfard and used as ultimate test field, have proven to be reliable: the vehicle manages to track them smoothly, without colliding any kind of obstacle.

In conclusion, the work done until now has been really important to achieve path following thanks to the steering controller and, on the other hand, to succeed in solving the issues of slippage on wheels thanks to the velocity optimizer. The next step would involve hardware-in-the-loop testing: implementing the algorithms described on this paper on a board and try to optimize this platform for such algorithms. Subsequently, a vehicle-in-the-loop testing could be done to see if the vehicle is truly capable of tracking a given path with the aforementioned algorithms.

Bibliography

[1] J. Bruinsma. *World agriculture: towards 2015/2030: an FAO perspective*. Earthscan, 2003.

[2] Y. Cao and M. Qiao. Application of fuzzy control in four wheel steering control system. In *2017 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pages 62–66. IEEE, 2017.

[3] C.Cariou, R. Lenain, B. Thuilot, and M. Berducat. Automatic guidance of a four-wheel-steering mobile robot for accurate field operations. *Journal of Field Robotics*, 26(6-7):504–518, 2009.

[4] X. Chen, Q. Bao, and B. Zhang. Research on 4wis electric vehicle path tracking control based on adaptive fuzzy pid algorithm. In *2019 Chinese Control Conference (CCC)*, pages 6753–6760, 2019.

[5] A. De Luca and G. Oriolo. Modelling and control of nonholonomic mechanical systems. In *Kinematics and dynamics of multi-body systems*, pages 277–342. Springer, 1995.

[6] J. C. Dixon. *Suspension geometry and computation*. John Wiley & Sons, 2009.

[7] B. T. Fijalkowski. *Automotive Mechatronics: Operational and Practical Issues: Volume I*, volume 47. Springer Science & Business Media, 2010.

[8] C. A. Floudas and V. Visweswaran. Quadratic optimization. In *Handbook of global optimization*, pages 217–269. Springer, 1995.

[9] A. Hemami. A control scheme for low speed automated vehicles with double steering. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 3, pages 2452–2454. IEEE, 1994.

[10] S. Hu, M. Lundgren, and A. J. Niemi. Discrete frenet frame, inflection point solitons, and curve visualization with applications to folded proteins. *Physical Review E*, 83(6):1–20, 2011.

[11] P. P. Keumejio. Ottimizzazione delle prestazioni dinamiche di meccanismi articolati mediante elementi elastici. 2016.

[12] A. Micaelli and C. Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. *Proceedings of ICARV*, 1993.

[13] M. A. Miller, B. L. Steward, and M. L. Westphalen. Effects of multi-mode four-wheel steering on sprayer machine performance. *Transactions of the ASAE*, 47(2):385, 2004.

[14] W. C. Mitchell, A. Staniforth, and I. Scott. Analysis of ackermann steering geometry. Technical report, SAE Technical Paper, 2006.

[15] W. Norris. *Modern steam road wagons*. Longmans, Green, and co., 1906.

[16] M. J. Nunney. *Light and heavy vehicle technology*. Routledge, 2007.

[17] R. Potluri and A. K. Singh. Path-tracking control of an autonomous 4ws4wd electric vehicle using its natural feedback loops. *IEEE Transactions on Control Systems Technology*, 23(5):2053–2062, 2015.

[18] B. Triggs. Motion planning for nonholonomic vehicles: An introduction. 1993.

[19] D. Wang and C. B. Low. Modeling skidding and slipping in wheeled mobile robots: control design perspective. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1867–1872. IEEE, 2006.

[20] R. Wang, C. Hu, F. Yan, and M. Chadli. Composite nonlinear feedback control for path following of four-wheel independently actuated autonomous ground vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):2063–2074, 2016.

[21] A. H. Wickens. Dynamics of actively guided vehicles. *Vehicle System Dynamics*, 20(3-4):219–242, 1991.

[22] M. R. Woods and J. Katupitiya. Modelling of a 4ws4wd vehicle and its control for path tracking. In *2013 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pages 155–162, 2013.

[23] J. S. Zhao, X. Liu, Z. J. Feng, and J. S. Dai. Design of an ackermann-type steering mechanism. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 227(11):2549–2562, 2013.