Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA BIOMEDICA, ELETTRICA E DEI SISTEMI

Ciclo 33

Settore Concorsuale: 09/G1 - AUTOMATICA

Settore Scientifico Disciplinare: ING-INF/04 - AUTOMATICA

DESIGN, MODELING AND CONTROL OF A TRACKED UGV FOR PRECISION AGRICULTURE

Presentata da: Roberto Tazzari

Coordinatore Dottorato

Michele Monaci

Supervisore

Lorenzo Marconi

Esame finale anno 2021

ABSTRACT

The population growth, and the consequent raise in food demand, are worldwide problems which are strictly related to the capacity of agricultural systems to withstand these increasing requests, despite climate changes and the lack of manpower. In the last decades, nations, companies and research groups are facing this problem, with the aim of introducing new techniques to optimize and preserve the available resources, as well as developing new tools aimed at increasing productivity and efficiency of human labor. Indeed, *Precision Agriculture* merges agronomy and farming techniques to data science and robotics in order to support and improve farmers' decision making and the whole production process.
Mobile robotics can be the turnkey to bridge automation and agricultural worlds. In fact, farmers still carry out a large part of the operations manually, in particular within orchards and greenhouses. For orchard agricultural robots to become established, it is necessary to be able to navigate autonomously through harsh environments, perceiving the surrounding space and adapt to it, while carrying the required tools for performing tasks such as spraying, or thinning.

Motivated by the previous framework, this thesis focuses on the design and assembling of an Unmanned Ground Vehicle for precision agriculture specifically tailored for orchards, greenhouses and horticulture. In particular, the contributions of this thesis are multiple, from on hand mechatronic design of the vehicle is presented. Moreover, we developed an optimized model-based design tool for tracked vehicles, which leverages their dynamic models in order to drive the sizing procedure. In the end, we describe how autonomous navigation is achieved, proposing novel algorithms for GPS-free in-row localization and navigation, exploiting the *semi-structured* nature of the environment. We also report how we improved path-following performance of tracked vehicles, designing an adaptive observer for slip estimation and compensation.
In the last part of this dissertation, we show the results of the experimental on-field tests carried out using the assembled tracked robot, with the aim of validating the proposed design and navigation approaches.

SOMMARIO

L'incremento della popolazione e il conseguente aumento della domanda di cibo sono problemi globali strettamente legati alla capacità del sistema agricolo mondiale di soddisfare questa richiesta crescente, nonostante la carenza di manodopera e le variazioni climatiche. Dagli inizi del ventunesimo secolo, enti nazionale, aziende e gruppi di ricerca hanno affrontato questa problematica, introducendo nuove tecniche per ottimizzare e preservare le risorse disponibili. Allo stesso modo, sono stati sviluppati nuovi strumenti capaci di aumentare la produttività e l'efficienza degli operatori. Tra questi si annovera l'*Agricoltura di Precisione*, la quale unisce tecniche e conoscenze agronomiche all'analisi dei dati e alla robotica, al fine di supportare le decisioni degli agricoltori durante tutto il processo produttivo.
La robotica mobile può rappresentare la chiave di volta per unire il mondo dell'automazione a quello agricolo. Attualmente, gli agricoltori svolgono ancora la maggior parte delle lavorazioni manualmente, in particolare nelle attività relative alla frutticoltura, l'orticoltura e l'agricoltura da serra. Affinché i robot progettati per la frutticoltura prendano piede nel settore, è necessario che acquistino la capacità di navigare autonomamente in un ambiente accidentato come quello agricolo, analizzando e adattandosi allo spazio circostante. Allo stesso tempo, devono anche essere in grado di trasportare l'equipaggiamento necessario per svolgere le lavorazioni richieste, come ad esempio l'irrorazione o il diradamento.

Questa tesi descrive gli aspetti più critici del processo di progettazione ed assemblaggio di un veicolo cingolato autonomo per la frutticoltura. Infatti, si descrive lo sviluppo di uno strumento di progettazione ottimale per veicoli cingolati, basato sulla modellazione dinamica di questi. Viene anche illustrato come siano stati risolti i problemi relativi alla localizzazione e alla navigazione, sia all'interno del campo che all'esterno. Infine, si descrivono le tecniche utilizzate al fine di risolvere il problema della navigazione autonoma, proponendo un nuovo algoritmo che sfrutti le conoscenze parziali che si hanno dell'ambiente. Viene anche mostrato un algoritmo per migliorare le prestazioni di inseguimento di traiettorie dei veicoli cingolati grazie ad un osservatore adattativo.
Nella parte finale della discussione, vengono mostrati i risultati sperimentali raccolti durante le prove in campo effettuate con il prototipo sviluppato al fine di validare gli algoritmi proposti.

To Sara
and to my family.

Acknowledgement

I would like to thank my supervisor Prof. Lorenzo Marconi for allowing me to *enter the game*, giving me the opportunity to challenge myself, pursuing the goal of merging my two life-long passions: technology and agriculture.
Many thanks to Alessà, Samo, Flavio and Nicola for welcoming me in the Lab, sharing with me ups and downs, as well as constantly supporting and pushing me. Without you all, this journey would have been a completely different thing.
My gratitude also goes to all the friends from CASY and lab mates: Ilario, Dario, Lorenzo G., Andrea E., Simone, Mario, Jonathan, Michelangelo, Roberto, Andrea T, Andrea C., Ivano, Guido and Lorenzo S. for being way more than just colleagues. You guys rock!
Thanks to the Aslatech crew: Andrea, Michele, Bruno, Radu, Marian and Ermes. I want to thank you for the fruitful discussions and the always-on-focus suggestions. Thank you all for helping us in making it real!
Thanks to Prof. Davide Scaramuzza for giving me the opportunity of being part of the Robotics and Perception Group allowing me to prove myself in an exciting new environment. New lab, same story: once there I met plenty of wonderful people who welcomed me since the very first moment. Thank you Davide F., Giovanni, Javier, Thomas, Yunlong, Tammy, Sihao, Dimche, Daniel, Mathias, Henri, Zichao, Dario, Antonio, Elia, Manuel, Manasi and Christian.

Thanks to my family, for always supporting me and leaving me the opportunity to make my own mistakes and decisions. Thanks for your trust and wisdom, as well as for doing also part of my share of duties at home. A special thank goes to my grandparents, who left an unforgettable mark. Thank you all for shaping me in the man I am today.
Thanks to all my friends for being still here after the uncountable times I skipped a night out because I had to study.
Last but not least, thanks to my girlfriend Sara, for her unconditional love and patience. Thank you for being always there for me and for all those times we missed something just because of me. Thank you for your precious advice. I love you Sara.

Grazie alla mia famiglia, per avermi sempre supportato, dandomi la possibilità di prendere le mie decisioni, così come di compiere da solo i miei errori. Grazie per aver sempre compensato anche per la mia parte di doveri. Grazie per la vostra fiducia e saggezza. Un grazie speciale va ai miei nonni, che hanno lasciato un segno indelebile. Grazie a tutti voi per avermi reso l'uomo che sono ora.
Grazie a tutti i miei amici, per esserci sempre nonostante le innumerevoli uscite rifiutate siccome dovevo studiare.
Ultima ma non per importanza, ringrazio Sara, per il tuo amore e la tua pazienza incondizionati. Grazie per esserci sempre per me nonostante tutto. Sei la migliore compagna di viaggio che potessi sperare trovare. Ti amo Sara.

Contents

Notation

$\mathbb{R}$	set of real numbers
$\mathbb{R}_{\geq 0}$	set of non-negative real numbers
$\mathbb{Z}$	set of integer numbers
$\in$	belongs to
$:=$	defined as
$\forall$	for all
$E^{n\times m}$	set of matrices with n rows and m columns and coefficients in E
$\|\cdot\|$	Euclidean norm of vectors and induced norm of matrices
$\|\cdot\|_p$	$\mathcal{L}_p$ norm of time signals, with $p \in [1,\infty]$
$M^\top$	transpose matrix
M^{-1}	inverse matrix
$M \geq 0$	positive semidefinite matrix
$M > 0$	positive definite matrix
det$M$	determinant of M
rank$M$	rank of M

$0_{n\times m}$	matrix of dimension $n \times m$ whose entries are all zeros. When $n = m$ we write 0_n and when the dimension is clear from the context the subscript is omitted and we simply write 0.
I_n	n-dimensional identity matrix. When the dimension is clear from the context the subscript is omitted and we write only I.
$f : A \to B$	a function from A to B
$\gamma \in \mathcal{K}$	γ is a class-$\mathcal{K}$ function, i.e. $\gamma : [0, a) \to \mathbb{R}_+$ $(a \in \mathbb{R}_+^*)$ is continuous, strictly increasing and $\gamma(0) = 0$.
$\gamma \in \mathcal{K}_\infty$	γ is a class-$\mathcal{K}_\infty$ function, i.e. $\gamma \in \mathcal{K}$ and $\gamma(x) \to_{x\to\infty} \infty$
$f \in \mathcal{L}$	f is a class-$\mathcal{L}$ function, i.e. $f : \mathbb{R}_+ \to \mathbb{R}_+$ is continuous, strictly decreasing and $f(x) \to_{x\to\infty} = 0$
$\beta \in \mathcal{KL}$	β is a class-$\mathcal{KL}$ function, i.e. $\beta(\cdot, t) \in \mathcal{K}$ for each t and $\beta(s, \cdot) \in \mathcal{L}$ for each s
C^n	the set of n-time continuously differentiable functions (C^0 is the set of continuous functions)

I

Introduction

In this chapter the key concepts and needs motivating the combination of mobile robotics and agriculture are introduced. We also provide the rationale which motivated the development of this thesis, as well as an overview of the main contributions brought in by this thesis.

The use of mobile robots in outdoor applications is already widespread in different fields such as Search&Rescue [2, 3], Security&Surveillance [4] or Space Exploration [5]. Alongside, agriculture is one of the latest fields in which the use of mobile robotics is rapidly increasing [6, 7], as the global population growth requires new agricultural methods and tools to optimize production and preserve the environment, [8, 9]. Current agricultural robotics ranges from huge tractors self-navigating within the field, to farming tools aimed at autonomously monitoring health status of weeds and plants [10]. This branch of agronomy, named Precision Agriculture (PA), deals with farming techniques merging data science and robotics in order to optimize farming processes [11]. The main goal of PA is to gather as much data as possible about terrain, plants and farm as a whole, to support farmer decisions, such as spraying pesticides only where needed. In this context, GPS/GNSS-driven systems play an important role, since they allow tractors to self-steer in order to till or seed the farm in the most accurate way. They allow one to create maps relating plants and terrain properties to a specific and accurate global position, endowing seeding and spraying tools with the

capability of working with punctual precision.
Currently, PA autonomous platforms are mostly deployed in *open-field* farming, while for Orchard, Horticulture and Greenhouses they are still at research level. The main difference between the first and the second ones lies in the navigation paradigms. In open field, in fact, GPS-GNSS data are enough to navigate reliably through the field, while, in the other farming environments, trees canopy and orchard/greenhouse structure can obstruct GPS signals, thus requiring GPS-free navigation strategies, as well as obstacle perception and avoidance capabilities.

This dissertation frames within the latter scenario, describing how we proposed, designed, assembled and tested a tracked Unmanned Ground Vehicle (UGV) for precision agriculture, specifically tailored for orchard, greenhouse and horticulture activities. This thesis focuses on three main topics: (i) **mechatronic design** of the vehicle, from locomotion to I/O peripherals and actuation management, as well as selection of the proper sensor suite and design of the customized farming tools; (ii) kinematic and dynamic modeling of skid-steering phenomena, leveraging this knowledge to create a **model-based design tool for tracked vehicles**; (iii) robot localization and **autonomous navigation**, proposing novel algorithms for in-row relative localization and **slip estimation**.

The development from scratch of a medium-size robotic tractor required quite a lot of work, under many different aspects, by several members of CASY laboratory at University of Bologna, among whom a former PhD student, Dr. Flavio Callegati, whose PhD dissertation, [12], includes related topics partially discussed also in this thesis.

This thesis is organized in four main parts. In the first part, the rationale behind the mechatronic design of the robot, motivating the choices taken during the prototype development process, as well as describing the main hardware and software components are presented. The second part reports the kinematic and dynamic models of skid-steering tracked vehicles and a model-based design tool is presented. Finally, the third part addresses the problem of localization and autonomous navigation taking advantage of the available partial knowledge of the (semi-structured) environment. Experimental results conclude the forth and last part.

Before entering into technical details, the following section provides more information about the context and the state of the art of this research field.

I.1 Motivations and state of the art

In the last two decades, the idea of *agricultural robots* has changed, initially the concept of robot was closer to computer aided fruit selection tools carried by standard tractors, e.g. an automatic picking system for melon harvesting, [13], or a robotic arm for radicchio selection and collection, [14]. Nowadays, with the term *robot*, the community tends to mean a *robotic platform, equipped with a suite of sensors by which it can navigate autonomously in different kinds of crops, collect data from the field, monitor the health of plants and interact with them or the user.* Either they are autonomous ground vehicles (UGVs) which allows the users to remotely assign missions and monitor their progress, [8, 15, 16], or they are aerial vehicle (UAVs) carrying visual sensors for plant/fruit monitoring [17, 18], and even a combination of the two, where UGVs and UAVs cooperate together in order for each one to overcome the limitations related to their motion domains, [19, 20], as also shown in Figure I.1.

Figure I.1: Cooperation between aerial and ground robots, [21].

The ones reported above are just few representative examples of the several interesting applications available all over the world. Nevertheless, they show the possibility of building agricultural robotic platforms capable of navigate in the farm environment, while autonomously performing different operations, as well as interacting and communicating with human operators. Ideally, this opens the doors towards a future in which agriculture will be completely automated, and the farmers relies on

a fleet of robots which carry out assigned works and allowing the farmers to monitor their progresses from a remote position. In fact, it is the pursue of the final goal of bringing this future closer that pushed us toward the development of an agricultural robot. The robot assembled was nicknamed "Surus Prime" and it is shown, equipped with a mower, in Figure I.2. We selected the name *Surus* because it was the name of the last and favorite elephant of Hannibal Barca, [22, 23], the Carthaginian general who led the African troops during the Alps crossing in the Second Punic War. In order to highlight the strength of the robot, despite its smaller size if compared to standard tractors. On the other hand, the term *Prime*, comes from the famous multimedia franchise *Transformers*, [24], in which *Optimus Prime*, [25], is the leader of a group of living autonomous robots, who fights for tight and beneficial collaborations between robots and human beings. We picked part of this name since the rationale behind the construction of this agricultural robot is not to remove human labor from farm activities, rather to co-operate, taking care of all the repetitive and simple tasks, leaving to human unmatchable expertise and dexterity the most critical and relevant activities.

Figure I.2: Surus Prime equipped with mower.

In line with this, it is natural to require Surus Prime to be able to carry out on-field farming tasks, in different ground and weather conditions, either replacing human

labor or supporting it. In doing so, the robot needs to be able to simultaneously perform: (i) autonomous **navigation**, both outside and inside the field; (ii) management of the **farming tools**. The latter is related to the specific mission and can range from activation of a camera, to the ignition of a smart nozzle for punctual pesticides distribution. Autonomous navigation on the other hand, is a common aspect of any mission, therefore, a *fundamental* requirement for any practical application of automation in the agricultural field.

Navigation within agricultural environment, both inside and outside the orchard/-greenhouse is really challenging, due to the variable nature of this particular outdoor environment: orchard maintenance operations (e.g. pruning, thinning), missing trees, protruding branches, light variations[1] or weather/season changes affect the environment and its perception, making the detection of recurrent features or schemes much more involved. Consequently, the position estimation based on visual feedback tends to be inaccurate. On the other hand, occlusions in GPS signals are recurrent due to cloudy weather or dense vegetation. Thus, GPS-free robust navigation algorithms are strictly required.
In addition to this, it is important to highlight that open-field farming environment can be considered *structured*, such as the field shape, dimensions and the position of static obstacles (e.g. electric grid poles, trenches, etc.) are known a priori, whereas outdoor navigation in *nonstructured environments* cannot rely only on global positioning systems such as GPS/GNSS receivers, due to the presence of moving obstacles and the absence of complete maps. Therefore, the robot envisioned has to sense/perceive the surrounding environment in order to safely traverse it. Perception systems can either be visual cameras, laser scanners, ultrasonic sensors or the like. We will discuss about the onboard sensor selection in Chapter 2, while the localization and navigation algorithms implemented are described in Chapter 4.
Finally, since the designed robot is a ground vehicle, **interaction with the terrain** plays an important role, in particular, given the choice of tracks as locomotion mean. It follows that, slip has to be analyzed and considered during motion, more details concerning dynamic and kinematic modeling of these phenomena are provided in Chapter 3, while in Chapter 4, it is shown how we developed an adaptive observer for **slip estimation and compensation**.

[1] Light variations can be caused by shadow zones projected by the trees canopy, as well as different lighting conditions due to different daytime.

1

Design Concepts

In this chapter, the main concepts behind the design of our agricultural robot are presented. Particular focus is given to the idea of "all-weather" vehicle, which led to a small electric-powered tracked robot.

In this chapter, we describe the main design concepts which have driven the development of the agricultural robot assembled at CASY laboratory (University of Bologna)[1]. We point out the advantages of a **small electric-powered tracked** vehicle both in terms of enhanced traction and reliable use either within orchard lanes or inside greenhouses. In the end, we show how the use of small multi-agents systems guarantees an higher fault-tolerance and faster response when compared to the use of a single bigger unit. In fact, we present our solution to the **all-weather** concept described by Blackmore [26], that basically highlights how effective in-field operations have to rely on a platform able to timely enter the field even if terrain conditions are not optimal (for instance after a storm), and without damaging the terrain itself both in terms of soil compaction and displacement. Furthermore, in terms of power and mechanical design, the platform is compliant with existing orchards (or greenhouses) dimensions and the requirements imposed by the activities to be performed inside them. The prototype is designed to target small-to-medium farms, which is not a real limitation as they represent more than 80% of all the farms within European Union,

[1] https://centri.unibo.it/casy/en/research

[27].
The resulting small and lightweight robot fulfills the specification of a platform easy-to-maintain, affordable, efficient, and "low-weight" to allow field operations also in presence of adverse terrain conditions. The UGV design augments the concepts discussed in [26] with an *expandability* concept based on replication of a basic "small" platform if higher performances are needed, rather than making a single unit bigger. In the following, we will describe in details each of the aspects and design objectives considered during the robot development process.

1.1 Adaptability and flexibility

The current prototype relies on a core locomotion system given by two rubber tracks (caterpillars) that provides to the vehicle high static stability and traction. Several farming tools (implements) can be attached according to the target operation and working environment. We also want to point out that, the inter-axis distance between the caterpillars can be potentially adapted to different row dimensions, thanks to the specific mechanical design. The adaptability and flexibility features that characterize the platform are used to accomplish operations in orchard management with "plug & play" implements. Indeed, the rover adopts the concept of "Task Oriented Automatic Subsystems" [26]. The platform is conceived to be a "socket in the field", a source of power easily accessible, able to dispense energy to all the interchangeable implements.

1.1.1 Implement design

In our design, the idea of a big tractor pulling farming tools and supplying power to them is replaced by the concept of "motorized implement". The final goal is to optimize the whole platform in terms of weight, power efficiency, and maneuverability through an integrated design between the different components. Currently, two customized implements, specifically tailored for our robot prototype, have been developed. Both the implements share a common power-train that provides adequate torque. The common mechanical interface can be adapted to both tools in order to ensure quick plug & play change operations. Namely, the implements that have been customized for our robotic platform are:

- **Branch shredder and lawn mower** - Since the robot features a standard Power

Take Off (PTO) plug [28], it can be easily equipped with traditional farming tools. In fact, a small commercial shredder/mower was adapted for our application, but we kept the same power transmission plug for sake of compatibility. This implement allows the robot to shred pruned branches or scrubs and mowing the orchard lane, even when the terrain conditions are wet or muddy. Figure 1.1a shows the robot equipped with the shredder.

- **Orchard sprayer** - A commercial multi-nozzle sprayer, [29], with an embedded turbine has been customized to be mounted directly on the implement support interface of the robot, without using a separate tow trolley. The PTO is used again for power transmission, nonetheless, we added a belt with pulleys in order to provide the suitable reduction factor to achieve nominal speed of the turbine. Figure 1.1b shows the robot equipped with shredder. Each nozzle is equipped with solenoid valves for selective spraying according to the field treatment prescription map.

One of the advantages deriving from the use of caterpillars rather than steering wheels is that the robot can be straightforwardly driven backward or forward, simply by changing the sign of the motor commands. This feature is quite important as different implements have different preferred working direction, e.g. the shredder requires to be pushed, therefore mounted in the front of the robot, while the sprayer is best placed in the rear part of the robot, otherwise the sprayed water droplets could reduce the visibility of the vision sensors.
Compliant with the idea of *motorized implement*, the water tank has been used as a frame element to increase the robot rigidity and, in addition, to enhance the static stability moving the center of gravity of the body almost in the central position with respect to the tracks length. More details about the mechatronic design and components features are provided in Chapter 2.

1.2 Versatility

We interpreted the concept of versatility in different ways: on one hand Surus Prime *multi-purpose* nature, [7], makes the robot versatile for as many autonomous applications as the number of available implements, but it can also be used as a co-worker, supporting human labor. On the other hand, thanks to its medium sizes this UGV is suitable for working within orchards as well as greenhouses and even in certain open-

(a) Robot in shredder configuration.

(b) Robot in sprayer configuration.

Figure 1.1: Robot equipped with different farming tools.

field scenarios (e.g. weed pests spraying). But what makes it most interesting for greenhouses applications is the fact that the locomotion system is electric-powered, meaning lower level of gas-emissions. In this section we will take a closer look to the proposed versatility shades.

1.2.1 Facilitator

A further feature of this rover is its flexibility in being used in totally autonomous scenarios as well as "facilitator" in contexts where human workers are directly involved. For instance, in the shredding or spraying operation, the platform operates autonomously, properly equipped with the dedicated implement and navigation algorithms. On the other hand, the same platform, carrying the appropriate implements could help human labor in more complex tasks. For instance, as shown in Figure 1.2, during harvesting autonomous navigation capabilities can help workers to focus only on their duties, leaving logistics or easier jobs to the machine.

1.2.2 Electric actuation

In an interview, [30], Blackmore stated that current tractors are oversized for their applications and these words are supported by recent studies, [31, 32]. In particular, they show via simulations, that two small electric tractors can theoretically replace

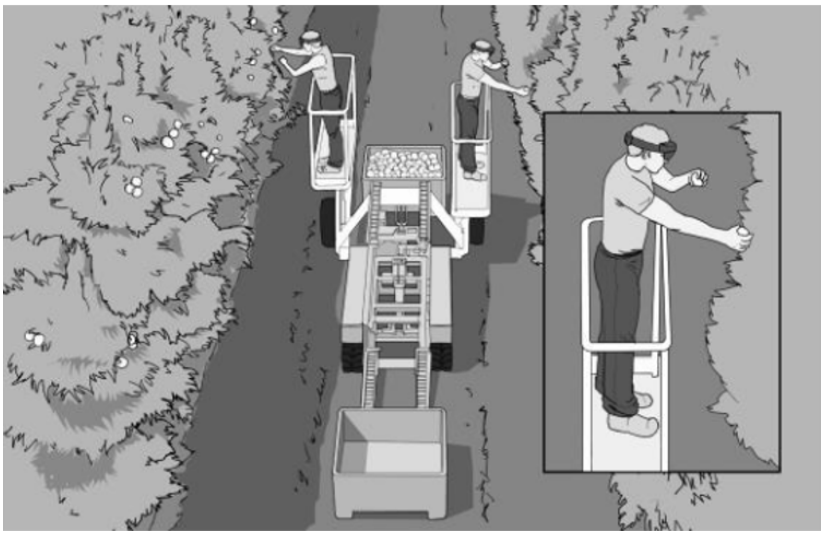


Figure 1.2: Use of the autonomous tractor as “facilitator” for human workers.

one standard tractor (160kW) in a 200 ha dairy farm, reducing energy consumption with respect to diesel powered tractors, by a factor of 58% if compared to a simulated diesel tractor and by 17-46% than real diesel systems. In addition to these results, they also claimed that with electric-powered tractors, the estimated reduction of greenhouses gas emission is about 92% if compared to diesel, even when energy consumption and gas emissions generated during battery manufacturing are included.
These recent results support our vision of multiple electric-powered small-sized tractors, capable to work effectively both inside orchard rows and within greenhouses, without any risk for gas emission or combustion.

Electric actuation allows us to use two smaller motors, one for each track, rather than one single bigger central motor, with consequent gearboxes and transmission (leading to efficiency losses similar to those inherently present in standard tractors). Having two independent motors is important since it makes the robot a differential-driven robot, granting a finer level of traction control than vehicles featuring a differential gear train.

1.3 Traction and Soil compaction

The problem of soil compaction is a critical aspect for agriculture, in particular during cold seasons or in general after heavy rain. Orchard agriculture is particularly

(a) Ruts caused by a tractor on muddy terrain.

(b) Effects on soil after tractor passage, [33].

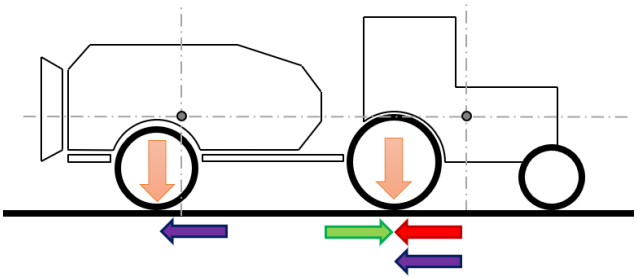
Figure 1.3: Ruts caused by the passage of a tractor on muddy terrain.

sensitive to this phenomenon: rainfalls increase the humidity, causing an optimal environment for fungal infections, to prevent this, trees and plants have to be treated as soon as possible. Therefore, it is often required to spray pesticides immediately after rainfalls, to effectively stop the proliferation of fungi [34]. Thereby, the sooner a tractor can enter the orchard, the better it is, nevertheless, heavy rains cause the terrain to become muddy, which can be easily compacted and rutted by tractors, negatively affecting orchard traversability, as shown in Figure 1.3 (for an extreme case we also refer the reader to this video[1]). In line with this concept, we started analyzing the role of weight in the soil compaction phenomenon, and, as expected the role of tractor weight is twofold, on one side the heavier the vehicle it is, the bigger is the normal pressure (measured in Pa or kg/m^2) applied to the terrain underneath the tractor, causing larger soil displacement and compaction. On the other hand, weight (or normal pressure on the ground) is important to determine the level of traction of the tractor, therefore wheeled farming machines requires large weight in order to achieve good traction levels on the ground.
Figure 1.4 shows two important cases in which, extra weight is added to standard tractors in order to increase their traction capabilities, as well as ensuring an higher level of static stability (avoiding overturning). As already mentioned, spraying is an important task among orchard farming activities, and in general the sprayer used can either be pulled, e.g. the one reported in Figure 1.4a, or carried on the rear linkage, as shown in Figure 1.4d.
Figure 1.4b shows traction and resistive forces acting on the tractor/terrain contact

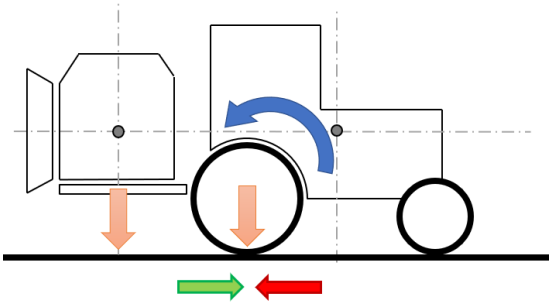
[1] https://youtu.be/lboML2FxSP4?t=60

(a) Example of a pulled sprayer.

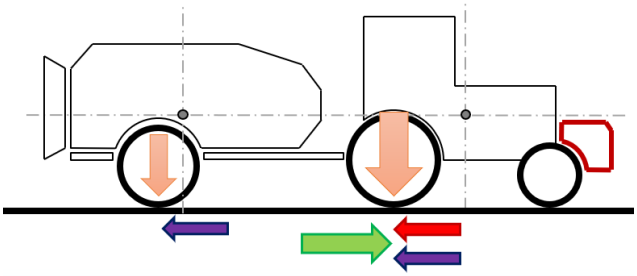
(d) Example of a carried sprayer.



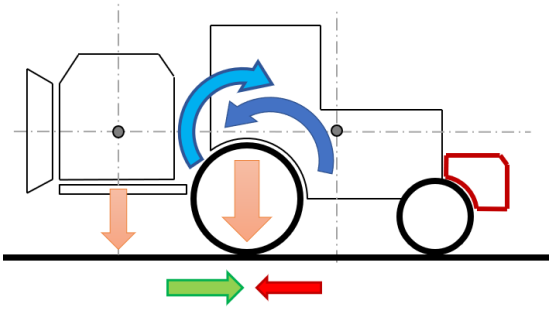
(b) Traction and resistive forces.



(e) Traction and resistive forces.



(c) Traction and resistive forces with additional frontal weight.



(f) Traction and resistive forces with additional frontal weight.

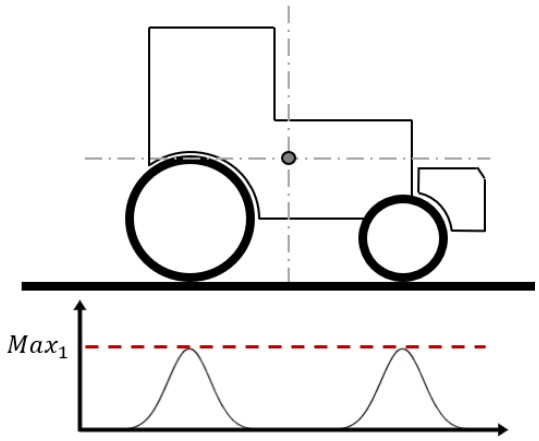
Figure 1.4: Effects on additional weight on traction.

in case of 2WD tractors, in particular, what we want to highlight is that the tractor has to move its own weight (red arrow), and in addition it also has to pull the sprayer (purple arrow). Therefore, the driving force generated by the tractor (green arrow) needs to be bigger than the sum of the previous two, if it is not the case[1] the tractor wheels could start to slip on the ground without moving the vehicle forward. Figure 1.4c shows that, by increasing the overall weight of the tractor, the traction perfor-

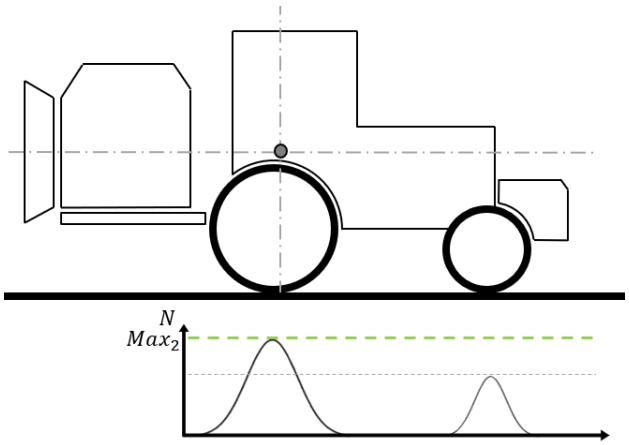
[1] In the following considerations we will only consider the case in which the slip is generated by poor traction on the terrain.

mance increases and then it is possible to achieve effective motion capabilities.
On the other hand, Figure 1.4e shows the carried-sprayer case, in which the sprayer becomes integral part of the tractor body, causing a displacement of the vehicle center of mass, or in other words a resistive torque about the rear wheel axles, as represented by the dark blue arrow. This torque changes the static stability of the vehicle, eventually leading to overturning. In order to counter balance the "overturning" torque, it is common practice to put additional weight in the front of the tractor, as shown in Figure 1.4f, affecting the overall weight, and in turn the compaction of the ground. Notice that, in addition to the cases described, ballasts can be added to the rear wheels of the tractor, we decided to omit this scenario as it is not common in orchard farming, rather more used in tractors working commodity crops.
For each sprayer design, it can be observed that the position of the water tank and of the turbine is quite relevant from a mechanical and structural point of view. In fact, in the design of Surus Prime, we opted for embedding the tank directly into the vehicle body, within the track contact length, in order to avoid overturning and to exploit the water mass to increase traction with the terrain.
Figure 1.5 and Figure 1.6 introduce the main reason why we decided to use tracks, rather than wheels, as locomotion mean (this discussion will be further developed, with analytical details in Chapter 3). As already discussed, achieving good traction on the ground is a crucial requirement for farming tractors, and this is typically improved by bulky tractor frames or by adding ballasts. Unfortunately, additional weight results in higher levels of normal force (weight) on the ground. We recall that, in our case, normal pressure on the ground is given by the ratio of normal force and the contact surface. Wheeled tractors feature smaller contact areas than tracked vehicles, thereby resulting in higher normal pressure peaks if compared with tracked tractors. Once again, we want to point out that in this section we only want to provide the intuition behind the design choice of building a small tracked vehicle, therefore the following quantities have to be considered as "rule-of-thumb" values.
Figure 1.5a shows the pressure peaks underneath a generic wheeled tractor, notice that we considered the center of gravity (gray dot in the image) to be at the same distance from the rear and front axle, in this way the pressure peaks in correspondence with the wheels are ideally equal. Figure 1.5c, instead, shows the case of a wheeled tractor[1] carrying a sprayer on the rear linkage, in this case the overall weight

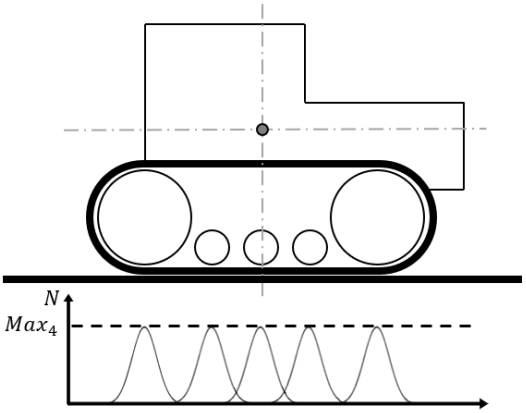
[1]We assume the weight and sizes of the wheeled tractor to be constant among the different cases considered.



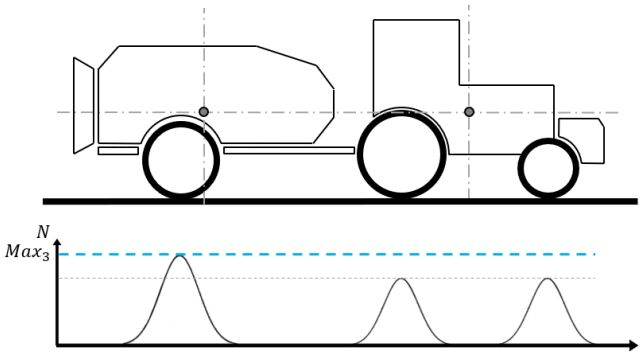
(a) Standard tractor.



(c) Tractor carrying a sprayer.



(b) Tractor with caterpillars.



(d) Tractor pulling a sprayer.

Figure 1.5: Normal pressure peaks generated by different tractor design.

is increased and the center of gravity is moved backward, closer to the rear wheels. From this, it follows that, being the contact surface the same, the pressure peaks increase in the rear part and decrease in the front part, as shown in Figure 1.5c. Let's now consider Figure 1.5d, where we have to consider both the weight of the trailed sprayer[1] and the tractor's[2]. In particular, as trailed sprayers have only two wheels and their weight is comparable to the one of the pulling tractors, the pressure peaks underneath the sprayer's wheels is potentially higher than tractor's, as shown in Figure 1.5d. In the end, tracked vehicles feature a higher number of contact points than wheeled tractors, and in turn a wider contact surface with the ground. This leads to lower pressure peaks if compared to standard wheeled tractors, even if the overall

[1]Trailed sprayers typical weight ranges from 1500 to 3500 kg when full of water, [35, 36, 37].
[2]Orchard tractors typically weigh about 2500-3500 kg, [38, 39].

weight is the same, as shown in Figure 1.5b[1].

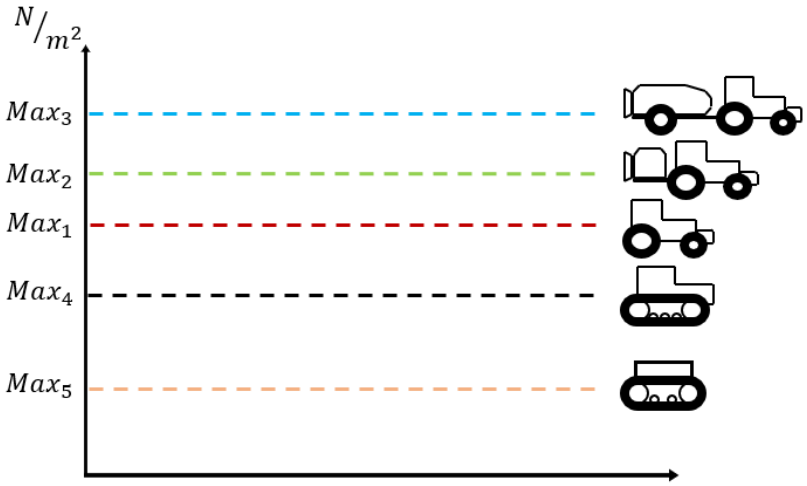


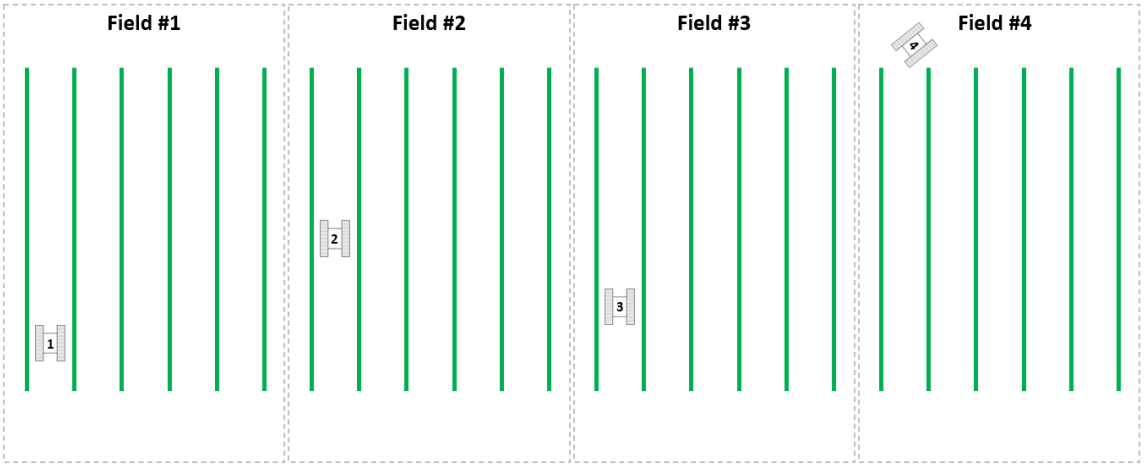
Figure 1.6: Comparison of the normal pressure peaks corresponding to the different cases discussed.

Figure 1.6 recaps the considerations done above, qualitatively comparing the pressure peaks relative to each tractor design. Looking at this picture it is straightforward to conclude that, increasing the tractor-ground contact surface and/or decreasing the overall vehicle's weight we obtain smaller pressure peaks, and in turn smaller soil compaction. Therefore the dashed orange line in Figure 1.6 graphically represents the advantages of designing a farming robot featuring tracks and smaller overall weight.

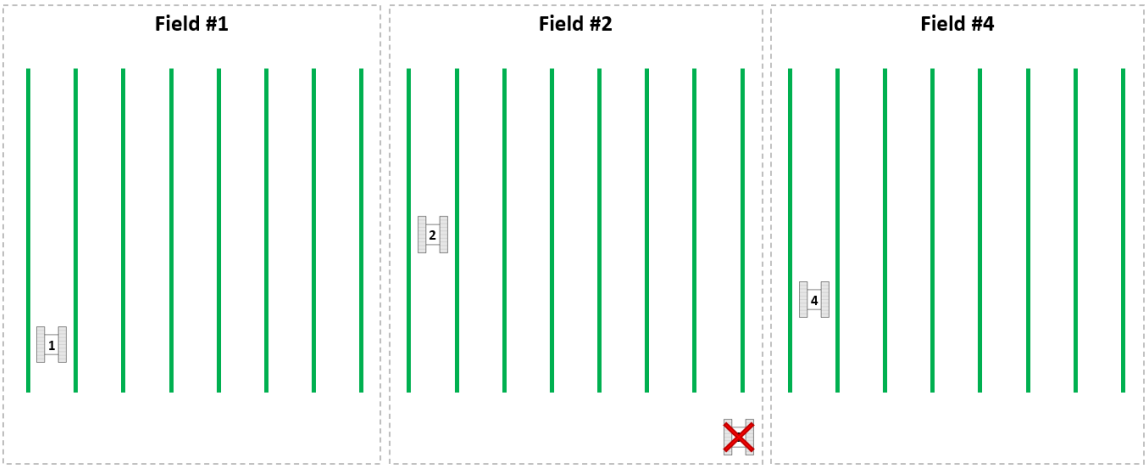
1.4 Scalability and Expandability

Although the mechatronic design is scalable to be suited to larger/smaller scenarios, it naturally fits the concept of expandability by replication of identical "small" modules: if higher work rates are needed, the vision is to add further "small" modules rather than designing a larger one. This concept allows one to keep the basic platform sustainable (in terms of initial investment, maintenance, maneuverability, robustness, incremental costs while expanding, etc.), but to increase work rates keeping the same interesting performance in terms of soil compaction and displacement.

[1]We refer the reader to the following video: https://youtu.be/yTAWAsmyG_k, where commercial types of tires and tracks are compared, reporting the same qualitative results discussed above.



(a) Example of multi-agents workload distribution.



(b) Workload distribution with one robot out of order.

Figure 1.7: Fault tolerant task reassignment paradigm allowed by the use of multiple small agents.

Redundancy of platforms brings a twofold advantage, on one hand, as just mentioned, it allows one to reduce the execution time of critical tasks which forces narrow time-windows to be executed. Figure 1.7a shows a possible workload distribution among several agents in order to timely complete the urgent task. On the other hand, redundancy introduces the concept of fault tolerance of the whole *multi-agent* system, in case of a fault in one of the agents, the remaining robots can re-assign their workloads in order to compensate for the missing agent, as shown in Figure 1.7b. Notice that fault tolerance is an extremely relevant added value if compared to actual operations with one single bigger tractor.

1.5 Conclusions

In this chapter, we described the main design concepts which drove the development of the agricultural robot assembled in our laboratory. In particular, we highlighted the idea of "all-weather" vehicle, which allows timely interventions independently on weather conditions. Moreover, we pointed out the advantages brought by a small electric-powered tracked vehicle both in terms of better traction and reliable use within orchards. In the end, we showed how the use of multi-agents systems guarantees an higher level of fault-tolerance and faster actions than one single bigger unit.

2

Mechatronic Design

In this chapter a presentation of the robot's hardware elements and their integration is given. Then, an overview of the software logical structure is provided, as well as a description of the main functionalities of the software components.

2.1 Introduction

The design of complex systems such as a robotic tractor requires the interaction and integration of mechanical, electrical, electronic and computer science elements. This chapter has indeed the aim of listing the different components required to assemble an agricultural robot, as well as describing how we integrated them into a single mechatronic system. We organized the hardware and software components in a vertical logical structure, from electric motors moving the tracks, up to the management software controlling I/O devices accordingly to the instructions received by the topmost logical level, which is in charge of providing high-level commands allowing the robot to operate in complete autonomy.
We want to give credit to Drover Srl[1] for the mechanical design and assembly of the vehicle, supporting us throughout the development of this project.

[1] http://www.drover.it/drover/

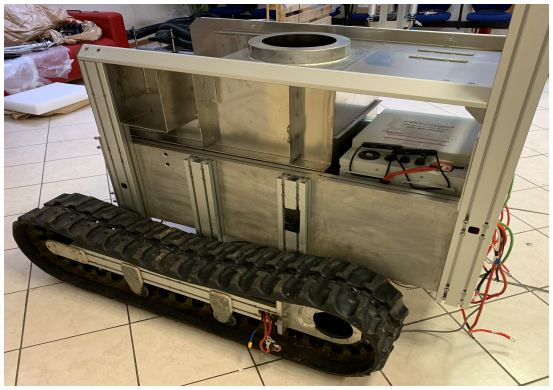
2.2 Mechanical Frame

The mechanical frame of the robot represents the backbone of the entire vehicle, as it has to host all the required equipment, withstand mechanical stresses resulting from the specific applications targeted by this project, and allow for agile navigation in irregular terrains.
To comply with these requirements, Bosch aluminum bars (Figure 2.1a) have been used. They merge lightweight properties of the aluminum, with great rigidity given by their particular shapes. In addition, they allow for on-the-go changes without the need of soldering, while, on the other hand, their specific shapes also allowed us to exploit the holes in the inner section for wiring purposes, as shown in Figure 2.1c.

(a) Bosch aluminum profiles.

(c) Use of bars cavities for wiring.

(b) Particular view of the bare frame.

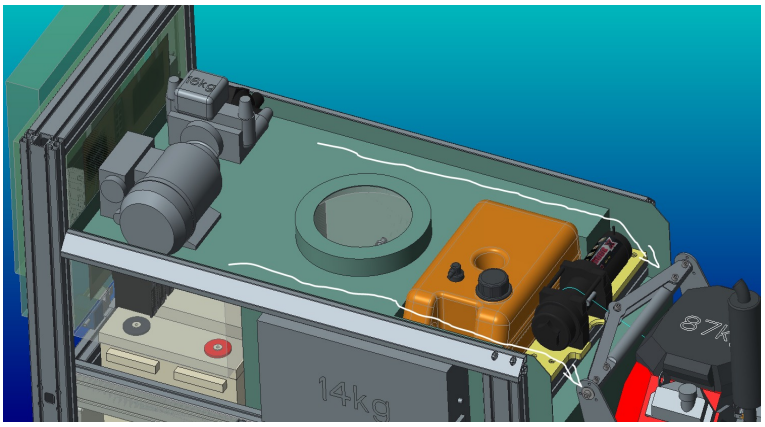


(d) Embodied battery and water tank.

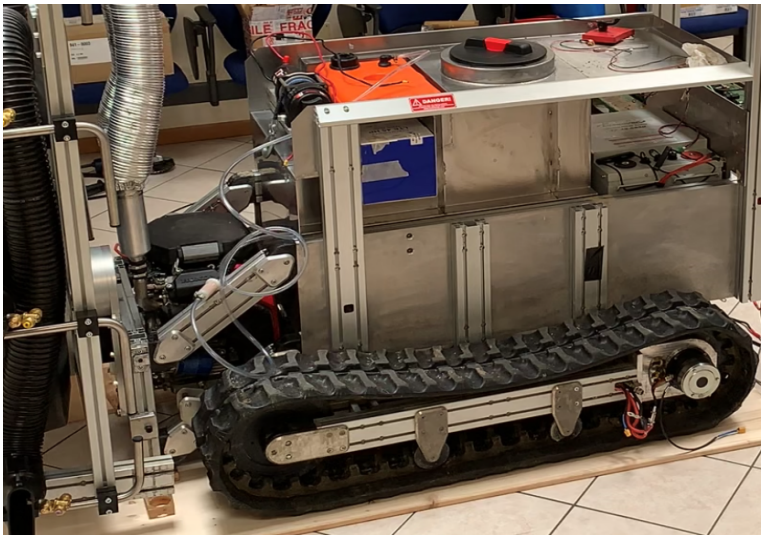
Figure 2.1: Mechanical frame of the prototype during assembly procedures.

Accordingly to what has been said in Chapter 1, the mechanical frame is conceived to embody the farming tools and components directly in the vehicle hull. Indeed, Figure 2.1d shows how we shaped the water tank so that its function is twofold, on one side, it allows the vehicle to carry the pesticides required for spraying purposes, and on the other hand, it gives more rigidity to the vehicle and it can host the components

required for powering and managing the attached farming tools, Table 2.1 lists all the components embodied in the water tank, along with their most relevant features. This specific tank layout and functionalities gave rise to the patent application [40]. Figure 2.2a and Figure 2.2b show the final assembly of the robot and how the water tank can host and support other components. In particular, Figure 2.2a shows how the upper plate of the tank is used to keep all the components managing liquids apart from the battery and computational units. Figure 2.2b reports a detailed view of the terminal connection with the implements, it is designed in a modular way in order to guarantee a quick and reliable change of the tools.



(a) CAD representation of the upper part of the water tank. The white arrows show the drain path imposed by the specific design.



(b) Corresponding view of the final prototype.

Figure 2.2: Mechanical frame of the prototype during assembly procedures.

Component	Parameter	Value
Additional Battery	Battery Type Nominal Voltage Capacity Max Const. Current	LiFePO4 12 VDC 40 Ah 400 A
Winch	Nominal Voltage Peak Current Max Const. Current	12 VDC 135 A 45 A
Inverter	DC Input Voltage AC Output Voltage Max Output Current	48 V 220 V 15 A
Spraying Pump	AC Voltage Max Current Power	220 V 6.5 A 1.4 KW
Mixer Pump	Voltage Max Current Power	12 VDC 10 A 120 W
Electro-valves	Voltage Max Current	12 VDC 0.7 A

Table 2.1: Farming tools actuation components.

2.2.1 Implements

As said above, the robot has to be capable of actuating different farming tools (implements), for some of them (e.g. mowers) it is required to adjust the relative distance/-position with respect to the ground. For this reason, the implements are attached to a four bar linkage mechanism, which originated the patent application [41], this allows one to regulate the height of the implement. The latter mechanism is then particularly shaped in such a way to host the fuel engine actuating the tool, as shown in Figure 2.2b. This expedient, reduces the distance between the motor power outlet and the corresponding inlet of the implement, making the robot structure shorter.
In terms of implement actuation, power consumption is a key factor to consider. Currently the robot is equipped with a Honda GX690 (22.1 HP) petrol engine[1], [42], directly connected to the tool via the standard PTO used for tractors, [28], in order to make the robot capable of adopting standard agricultural tools, with no need of adap-

[1]The petrol can is the orange container in Figure 2.2a and Figure 2.2b.

tations or additional components.

2.3 Locomotion systems

Locomotion of the robot relies on tracks rather than wheels. While more details about the rationale behind this choice are provided in Chapter 3, in this section we give a description about the components required for actuating them. In particular, the main elements are:

1. Battery
2. Electric Motors
3. Parking Brakes
4. Electric Drives

2.3.1 Battery

Locomotion of the robot is completely electric and, as well as all the electronic components, it is powered by the main LiPo battery (Figure 2.3) placed in the central part of the vehicle body, as shown in Figure 2.1d and Figure 2.2. This is a **LiFePO4 battery** designed for heavy automotive applications, whose features are summarized in Table 2.2. Notice that, positioning the battery in the rear bottom part of the vehicle allows for an enhanced weight balance, which has been presented as a patent application [43] currently under examination.
Three parallel connections are plugged into the battery terminals, one for direct power supply to the two motors, the second one is used to feed all the low-power electric and electronic devices, and the third one ends in a power plug suited for battery recharge. For safety reasons, all the three branches are electrically separated one another by contactors and fuses (Figure 2.15b shows, for instance, the contactor installed inside the I/O box), in particular, when the recharge plug is connected, a specifically-designed circuit cuts-off the other two, in order to prevent power surges or uncontrolled movements of the robot.
Electric and electronic components present on the robot require different voltage levels, therefore, DC-DC converters are installed onboard, transforming the 48 VDC output of the battery to 24 VDC, 12 VDC and 5 VDC, detailed overview of these components are provided in Section 2.7.

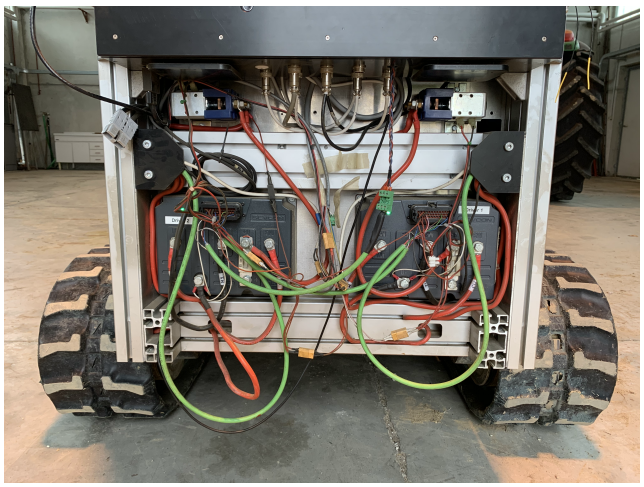
Figure 2.3: 48 VDC LiPo battery

Parameter	Value
Nominal Voltage	51.2 VDC
Capacity	130 Ah
Energy	6.65 KWh

Table 2.2: Battery specifications

2.3.2 Motor controllers, Motors and Brakes

(a) Zoomed view of the embedded casing of motors within the track body.



(b) Drives with wiring.

Figure 2.4: Detailed views of the robot locomotion systems.

As said above, the motion of the robot relies on two parallel tracks and each one of them is propelled by a dedicated brushless gearmotor[1], Figure 2.4a shows a zoomed view of the latter. This specific location of the motors, as well as the embedded gearbox designed to be integral with track frame are described in our patent application

[1]The motor outlet is connected to a planetary (epycicloidal) gearbox in order to reduce the rotational speed and increase the output torque.

[43]. In this configuration the vehicle can be steered as a *differential driven robot*, actuating each side at a different speed or direction, using the two separated motor controllers shown in Figure 2.4. These devices are *Sevcon Gen4*™ 48VDC drives, [44], they receive low-power commands through CANBus by the Computational Units and convert them into power signals used to feed the motors. As the motors are three-phase brushless motors, the electric drives have as well the duty of transforming the DC current provided by the battery into 48V three-phase AC.
For safety reasons, the robot is also equipped with parking brakes, the silver-colored components on top of motors (black) in Figure 2.4a. The benefit of these brakes is twofold, as the name suggests they are used when the robot is not working or moving, as well as for emergency stops: if required, these brakes can be activated and the vehicle will suddenly stop, preventing any collision or dangerous situation.
As it is possible to see in Figure 2.4a, parking brakes are mounted on top of the motors and the whole assembly is inserted directly into the track frame, reducing the cooling efficiency of the gearmotor. In addition to this, each parking brake is composed by an electromagnet which is activated when powered on: the braking effect is achieved when the power supply is cut-off[1], this means that during motion the coils are continuously fed, generating heat which increases further the motor temperature. To mitigate this effect, we designed a two-voltage-level power supply circuit, exploiting the nonlinear behavior of this family of brakes. The nonlinearity is given by the difference in the activation ($V_{T,on}$) and deactivation ($V_{T,off}$) voltage thresholds, namely in our case, $V_{T,on} = 20V$ and $V_{T,off} = 3.5V$. Therefore during nominal conditions, the brake is triggered at 24 VDC and then the supply voltage is lowered to 5 VDC, in this way the current passing through the coil is smaller, leading to a reduced power dissipation and heat generation.

2.4 Sensor Suite

In this section we will discuss the selection of sensors equipped on the robot, pointing out the importance of picking the sensors most suitable for the specific application and environment. The relevance of this choice is clear ever since the '60s when, at the Artificial Intelligence Center of Stanford Research Institute, *Shakey the Robot* was developed [45, 46], introducing for the first time the *sense-plan-act* paradigm, according to which perception of the robot itself or of the surroundings is processed

[1] This allows the robot tracks to be always locked when powered off.

in order to define the proper action to be undertaken. Even if this paradigm has been evolved in many different ways since that moment, [47], the selection of the proper set of sensors is still a crucial aspect, in the design of an autonomous mobile vehicle. In particular, for outdoor applications the sensor suite must provide information about robot motion, as well as global position, and proper perception of the surrounding static and dynamic environment.
As we already mentioned in the introduction, it is important to enrich the sensor suite with exteroceptive sensors (e.g. cameras, laser scanners, ultrasonic sensors, etc.), in order to extract meaningful information of the surrounding environment. Then, the main problem to face is that agricultural applications are particularly affected by large variations in lighting conditions (*luminance dynamic range*). The sun light generates areas in which the robot is directly exposed to light rays as well as large zones shadowed by trees canopy. Therefore, perception capabilities of an agricultural robot have to be robust to sudden variation in lighting conditions, and at the same time it has to allow the autonomous machine to perform tasks anytime during the day or night[1]. Last but not least, it has also be robust to visibility variations due to weather conditions (e.g. clouds, fog) or to the different seasons.

Given the above considerations, as also described in our work [48], Surus Prime is equipped with the following sensors:

- Motor encoders and telemetry
- Inertial Measurement Unit
- GNSS Receiver
- 3D Laser Scanner

whose measurements are then properly fused together in order to obtain a more accurate and reliable pose estimate.
It is important to mention that, before coming up with this list of sensors, other technologies have been studied in order to evaluate the benefits brought in by their possible introduction in the robot sensor suite. In particular, among those sensors featuring the Dynamic Range required for our application, we considered also HDR cameras, ultrasonic range sensors[2] and dynamic vision sensors (also know as event cameras). We selected the LiDAR technology as it provides a 360° field of view, it is

[1] Some of the farming tasks is most effective if carried out during night time.
[2] More experiments on the use of ultrasonic sensors are still ongoing.

not affected by motion blur and it has a really high Dynamic Range. More information about the reasons why event cameras were not picked can be found in Section C.3 of Appendix C.

2.4.1 Motor encoders and telemetry

As mentioned above, electric drives are not only meant to send commands to the motors, but also to receive and read feedback from them and then forward the latter to the computational units in order to be processed. In particular, readings from the two *sin-cos* motor encoders provide measurements about motor angular position and velocity, as well as motors torque and their internal temperature. On the other hand, the drives can also provide indirect measurements about the battery status and instantaneous current absorption.

2.4.2 Inertial Measurement Unit

The HMC5883L Inertial Measurement Unit (IMU) is composed of a three-axis gyroscope, a three-axis accelerometer and a three-axis magnetometer. This provides data about 3D rotations and twist of the body, as well as 3D linear accelerations, particularly relevant and useful within a sensor-fusion context.
The sensor used for this application is a low-cost Arduino-compatible IMU, placed as close as physically possible to the Center of Gravity of the robot itself.

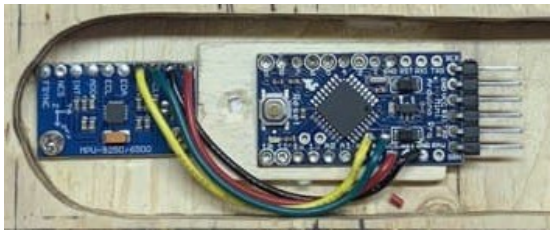


Figure 2.5: Zoomed view of the IMU-Arduino connection.

The communication protocol is I^2C, which is very sensitive to Electro-Magnetic (EM) disturbances. To solve this problem, we placed an Arduino Nano Every, [49], as close as possible to the sensor, as shown in Figure 2.5, connecting the arduino-board to the computational unit, via serial connection which is more robust to EM disturbances.

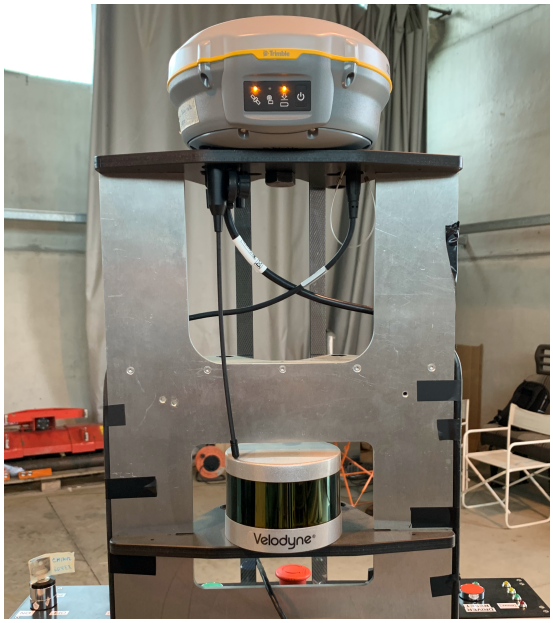


Figure 2.6: Zoomed view of GNSS and LiDAR on-board.

2.4.3 GNSS Receiver

We used the *Trimble R8s* GNSS system, [50], shown in the top part of Figure 2.6. A RTK GPS/GLONASS receiver that provides global positioning with sub-meter accuracy. The RTK correction acquired from existing topographic stations is applied to enhance precision up to centimeter accuracy. The standard NMEA interface is used to allow transparent substitution of sensor type.
The whole system has been configured also to work with non-RTK GNSS sensors, therefore producing a reduction of position accuracy, that can be improved by using an appropriate sensor fusion algorithm.

2.4.4 3D Laser scanner

The 3D *Velodyne LiDAR PUCK VLP-16* laser scanner, [51], is used to provide environment perception, thus making possible to detect rows of trees, obstacles, branches or other orchard features, as well as ditches, tranches and possible non-transitable areas. A zoomed view of the LiDAR and how it has been attached to the robot is presented in the bottom part of Figure 2.6.
This sensor is of paramount importance to provide reliable navigation inside orchards rows, as within orchard rows, GPS/GNSS signal is not always reliable, being obstructed

by tree canopy and foliage, more about how we used LiDAR for localization in Chapter 4.

2.5 Computational Units

The logical architecture managing both hardware and software components of the robot is an evolution of our former work [52], and in line with it, the architecture is organized in two main logical levels: a High-Level (HL) and a Low-Level (LL) as represented in Figure 2.7.

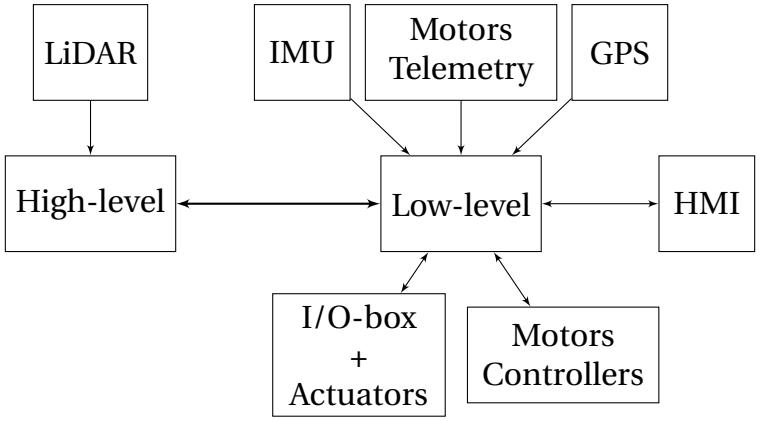


Figure 2.7: HW/SW architecture scheme.

In particular, the core of each of one the two levels is represented by the two computers upon which the whole architecture is based. The LL features an industrial standard (in terms of Electro-Magnetic Compatibility (EMC)) rugged PC, whereas the HL leverage the computational power of a desktop mini PC. The choice of using two different machines was driven by the fact that localization, navigation and control algorithms are most efficiently implemented using the Robotic Operating System (ROS), [53, 54], commonly used and developed on Linux OS, whereas, the most reliable and stable drivers released for industrial-standard components are available for Windows OS.[1]

[1]The choice to not rely on virtual machines was taken in order to maximize the efficiency on the two sides, being both low-level and high-level duties equally critical for the accomplishment of the robot tasks.

2.5.1 Low level components

As mentioned above, LL hardware and software components have the main function of collecting measurements from the different sensors and of communicating with external devices such as (i) the Human-Machine Interface or (ii) the remote controller as well as to (iii) Input-Output devices and the motors. In fact, the main functionalities of the LL are:

- *Sensor readings* - Read measurements from all the sensors available except for the LiDAR (which is read by the HL) and pre-process them in order to make them ready to be processed both by the mission planner in the HMI and by the HL navigation tasks. Thus, the LL has to feature the capability of communicating with different protocols, guaranteeing the possibility of gathering info from heterogeneous sensors.
- *HMI and remote control* - The LL implements the interaction between human operator and the machine. In fact, the Human-Machine Interface is the main tool for the user in order to plan the mission tasks for the robot, as well as to monitor the mission and robot status in real time. In addition, from the HMI, the user can also be enabled to drive manually the vehicle, therefore the LL has also to take care of receiving commands from the radio remote controller.
- *Communication with HL* - A reliable bilateral communication channel has to be established between the two levels allowing for a tight information exchange. This was implemented, exploiting the reliability of Ethernet communication, in particular leveraging the flexibility and good documentation available of the MAVLink protocol, [55].
- *I/O devices and actuators* - Along with sensors for navigation, the LL has also to read inputs coming from logic and analogical devices, as well as to trigger relays, logical outputs and of course actuate track motors. All these actions are performed over Ethernet using standard communication protocols such as MAVLink, CANBus and Modbus.

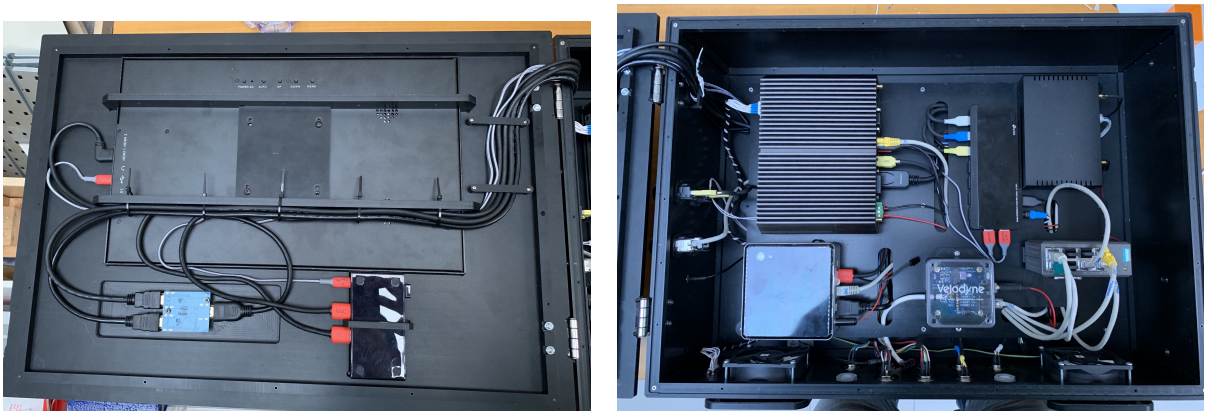
In order to be compliant with all the requirements listed above, a Cincoze DI-1000-i7, [56], has been selected. This choice is optimal as this device features several native USB, serial and Ethernet ports, which is important to avoid hubs, increasing the vulnerability to electro-magnetic disturbances.

2.5.2 High level components

HL hardware and software components have the main function of data processing, namely, endow the vehicle with autonomous navigation capabilities, which can be identified as:

- *LiDAR reading and processing* - Data coming from a 3D laser scanner (3D point-clouds) are heavy to be transferred, so to not saturate the transmission channel between the two logical layers, we decided to read this sensor directly in the HL. Another reason that pushed us to read it in the HL is that these information are used for localization and navigation purposes only by the HL.
- *Sensor fusion and Localization*[1] - In order to robustify the raw measurements of the sensors collected, *Sensor Fusion* techniques are implemented in order to reliably localize the robot.
- *Navigation and Control*[1] - To carry out the assigned farming task, the robot has to properly plan its trajectory, detecting and avoiding obstacles and occlusions. Once the path is defined, then the closed loop controller ensures accurate tracking.

To endow the robot with the computational power required to carry out all the functionalities listed above, an Intel NUC mini-PC, [57], was selected. This device is suitable for heavy computational payloads and it features a compact form factor, making it the ideal choice for the HL core unit. Then, we installed Linux and ROS on it.



(a) PC box lid side.

(b) PC box core part.

Figure 2.8: Robot PC electronic box detailed view.

[1] More about this topic will be presented in Chapter 4.

2.6 HMI and Remote Controller

The interaction with users/operators is a critical aspect, since it allows them to enter next mission specifications, as well as to monitor the actual state of both the robot and the current or scheduled missions.

2.6.1 Human-Machine Interface

The HMI code, beside allowing the operator to check the system status, has several other functionalities from managing the electronics to detecting the presence of faults on the CAN bus which controls the motors, up to mission assignment and monitoring. In fact, the features of the HMI can be summarized in:

- Sensor reading
- Management of all the I/O devices
- Communication with HL units, as well as ensuring reliable communications with all the components onboard
- Missions assignment

The robot HMI is implemented as a Visual Studio *form*, developed in C# code and run under Windows 10 OS. Since each feature is linked to visual elements of the form, the interface will be presented by reporting its different screens along with a description of their main items.

Home screen

As the HMI is started, the screen displayed in Figure 2.9 appears to the user.
In this screen the more relevant information about robot state are provided to the user. The *satelline map* shows the robot position (orange dot) at any point on earth's surface. It is updated thanks to the position feedback coming from the GPS receiver. In addition to this, for each motors the complete *telemetry* is shown, along with colored flags, which are triggered in case of warning feedback received by the motor controller (e.g. motor temperature too high). In line with this, state feedback about the most critical components of the robot are shown as well.
Below these boxes, several tabs can be accessed in order to enable/control/monitor different functionalities of the robot. Here below a list of the main tabs.

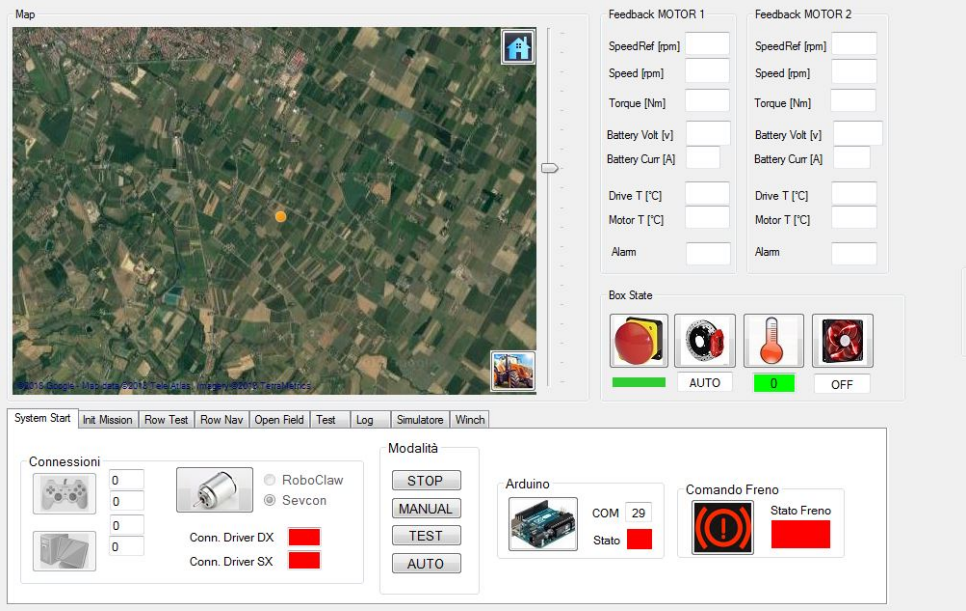


Figure 2.9: HMI home screen.

System Start

As shown in the bottom part of Figure 2.9 the first tab "System Start" allows to start the system. From left to right, the following items are displaced in this tab:

- Joy connection. The first thing to do is connecting to the RC remote controller, so that the user is always able to manually control the robot or trigger the brakes, in case of emergency, once the system is started.
- Simulator start. This feature start a simulator software which emulates feedbacks received by the real robot and it is mainly meant for debugging purposes as it allows to test new software offline.
- Motors connection. Starts Ethernet communication with I/O devices, in particular it enables the CAN communication with the motor controllers, from this moment on, the robot is active and it can be manually moved.
- Mode selection. Allows to chose the operating mode, among
  - *Stop*, safety mode in which motors are stopped and brakes are always active. It is useful during system boot and initialization procedure, or to safely carry out debugging activities.
  - *Manual*, in this mode the robot receives the user's manual commands. The motors reference speeds are directly computed by the HMI on the basis of the remote controller sticks position, rather than reading reference values

from the HL computer.
  - *Test*, useful for testing the tracks motion, through the imposition of different speed levels or some predefined trajectories, for instance straight, circular or "8"-shaped trajectories.
  - *Auto*, automatic mode in which the High Level defines the tracks reference speeds, accordingly to control loops developed and explained in details in Chapter 4.
- I/O Electronic Box. It is some sort of heritage from previous designs, but it still has the function of displaying that the connection with the electronic box in charge of managing all the I/O devices has been established properly.
- Brake management. Allows to activate or deactivate the parking brakes, directly from HMI software buttons. This function mirrors the one present on the Radio controller. The state flag shows not only if the brakes are active or not, but also if they are manually bypassed or working according to the automatic procedure described previously.

Init Mission

This tab is designed in order to manage the communication with the Robot Control (HL), check for errors in sensors reading and start missions (see figure 2.10).

Figure 2.10: HMI panel used for establishing the communication with the HL and setting the fundamental mission parameters.

From left to right, the panel is composed by the following boxes:

Switch and Laser

These two buttons allow to power on the laser scanner and the Ethernet switch, indeed enabling the communication between the HMI and the HL. This feature is useful just for a matter of energy savings, since the HL devices are basically employed

only in the automatic mode.

Rover Com
Manages the communication between the HMI and the HL programs, through the following items:

- URL. Allows to set or modify the High Level IP address and port.
- Start communication. Starts UDP communication between the LL and the HL. The label below reports the communication status.
- Reset. In case of necessity or problems, the communication can be reset, in case of communication failure, an automatic routine is anyway started in order to try to reactivate stable connection between the two level, in the meanwhile the automatic mission is suspended.

The first two functionalities are enabled once the "R2D2"-button[1] is pressed.

Pose e Campo
It allows to monitor the satellite coverage, manage the initial GPS settings and choose the orchard in which the mission has to be carried out. From top to bottom, the box is composed of:

- Sat. This label reports the number of GPS satellites currently visible from the receiver. If the GPS receiver is capable of receiving signals from different satellite constellation (GNSS), this label always shows the sum of them all.
- Home. If the number of visible satellites is high enough, the underneath button is enabled, allowing the user to set the *home* position from actual GPS latitude and longitude coordinates. When this button is pressed the "home position", considered as the origin of the ENU inertial reference system is defined. If the home position is not defined before starting the mission, the robot uses the predefined one[2].
- No GPS. Here the user can decide whether to use the GPS or not during the current mission. This can be useful for indoor test experiments, or in case the GPS receiver is not present onboard or out of order. Moreover, this button can also be toogled during operation, in order to simulate a fault in the GPS receiver.
- Campo. Allows the user to decide the working field where the robot has to carry

[1] R2-D2 [58].
[2] This option has to be used with caution: if the farm has more than one base station, relying on a predefined home position can negatively affect the accuracy of the pose estimation.

out its mission. Once the orchard is defined, the system automatically loads all the parameters describing the selected field, such as GPS positions of orchard rows, number and length of rows and so on.

Rover State

This tab contains labels which are meant to give feedback about sensors and task status, and have to be checked before starting each mission. Red color means that a fault occurred or the item is not yet initialized, green that the component is setup properly, while grey means that the device is missing or not plugged properly, like a sensor not correctly installed on the robot. In particular, these labels report if (from top to bottom)

- Onboard sensors (Encoder, GPS, laser scanner and IMU) are read correctly
- GPS is used or not in the current mission
- Home position has been defined (of course, only if GPS is used)
- Mission parameters have been set
- Robot parameters have been defined

Missione

The buttons in the top part of this box allow to *Stop*, *Start* or *Suspend* the mission. At the bottom, two labels show the motor reference speeds calculated by the HL program.

The following two tabs let the user define mission or test parameters, as well as monitoring robot working status during the operations, defined and started through this panel.

Row Test

This panel allows the user to test navigation within orchard rows as well as to monitor the work progression. The screen as shown in Figure 2.11 is composed by the following boxes.

Test Parameters

This box allows to define test parameters and send them to the High Level. In Figure 2.11, from left to right, it possible to find the following items:

- Test type. Using the buttons the test type can be selected, choosing between navigation along the row, row change or some combinations of the previous two.

Figure 2.11: HMI panel for test definition, assignment and monitoring.

- Speed. It allows to set the longitudinal speed that the robot has to keep during the test.
- RowDistPerc. This label defines the lateral distance (%) that the robot has to keep during navigation within rows, positive values result in later shift on the right hand side, on the left for negative values.
- NHCurveDir. In case the test includes row changes, this label defines the curve direction, towards left or right.
- GO!. When the button at the right most side of this box is pressed, the robot checks if all the sensors are read correctly and that the initialization process was completed properly. Then parameters related to the test are sent to the HL, the latter, in turn, starts the autonomous navigation, generating accordingly motor commands.

Test Status

This debugging box is designed to enable the user to monitor the different phases of autonomous navigation. From top to bottom:

- State machine operation. This label reports the operating condition (state) of the HL, namely the *navigation state machine*, this allows an easier debug of the state flow.
- Dentro/Fuori dal filare. Reports whether the robot is inside or outside the rows, based on the "End-of-row" detection algorithm.
- Hough/N-Hough attivo. This labels give a feedback about in-row localization algorithm status, telling if they are active or not.
- Filari percorsi. Indicates the number of rows completed by the robot during the current test.

Row Nav

By means of this panel the user can set and send missions to the robot, considering the robot to be at the entry point of the field, as well as monitor its progresses once the work has started. The screen, reported in Figure 2.12, is composed by the following boxes:

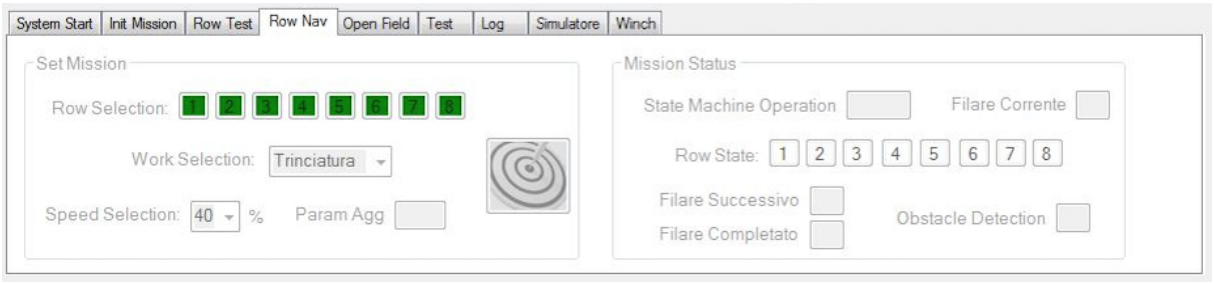


Figure 2.12: HMI tab for setting missions, send them to the robot and monitor their progresses.

Set Mission

It allows to set the mission parameters by means of the following items (from top to bottom):

- Row Selection. The buttons allow to select in which rows the robot has to work and which ones it has to skip.
- Work Selection. Through this item the user can define the task that the robot has to carry out, like fertilizer distribution, trees visual inspection, and so on. Based on this choice, a series of parameters is automatically loaded and passed to the HL (e.g. dimensions and weight of the farming tool, or speed which has to be maintained in order to successfully carry out the task).
- Speed Selection. Allows to select between different percentage values of the speed at which the mission will be carried out.
- Send. Pressing this button (right hand side of this box), the mission is sent to the robot HL.

Mission Status

This box, placed at the right side of the tab, contains some labels showing the mission status. From top to bottom:

- State Machine Operation. Reports the operating condition of the HL, indicating if the platform is waiting for starting the mission, rather than busy or stuck.

- Filare corrente. Shows which row the robot is currently working in.
- Row State. This labels provide an overview of the mission progresses, indicating which rows the robot has already completed, the one currently under work and those still to be done.
- Filare Successivo. Reports which row will be the next one, based on the trajectory pre-calculated by the Mission Planner.
- Filare completato. Indicates if the robot has finished to work in the current row.
- Obstacle Detection. Communicates whether the robot has encountered or detected obstacles on its path. In this case, if the robot managed to avoid it, a flag (mark) is left in the data logs, otherwise the robot travels back since the beginning of the current row and switches to the following one, triggering a warning reported in the log files.

Open Field

This mode has been designed in order to guide the robot in open field environments, i.e. to work on wheat fields or automatically move the rover from its base to the orchard where it has to carry out its mission. As better explained in Section 4.2.1, open field navigation receives a sequence of GPS waypoints, describing the global path that the robot has to follow in order to reach its destination. From this tab, it is possible to set the waypoints of interest, and then start the *mission planner*. The latter generate the proper sequence of GPS points and then send them to the HL controller which interpolates them, starting the corresponding mission.

Test

This panel activates when the user selects the *Test mode* from the first tab and it allows one to manually define the parameters of the trajectories to be negotiated by the robot.

2.6.2 Remote Radio-controller

The remote controller shown in Figure 2.14 allows the user to remotely control the robot, in particular, it allows to operate it manually, both in terms of navigation and in terms of actuating the attached farming tool. From the remote it is also possible to enable the *Autonomous* mode, keeping anyway functionalities of emergency stops.
As depicted in Figure 2.14, the remote controller is a radio device usually used for

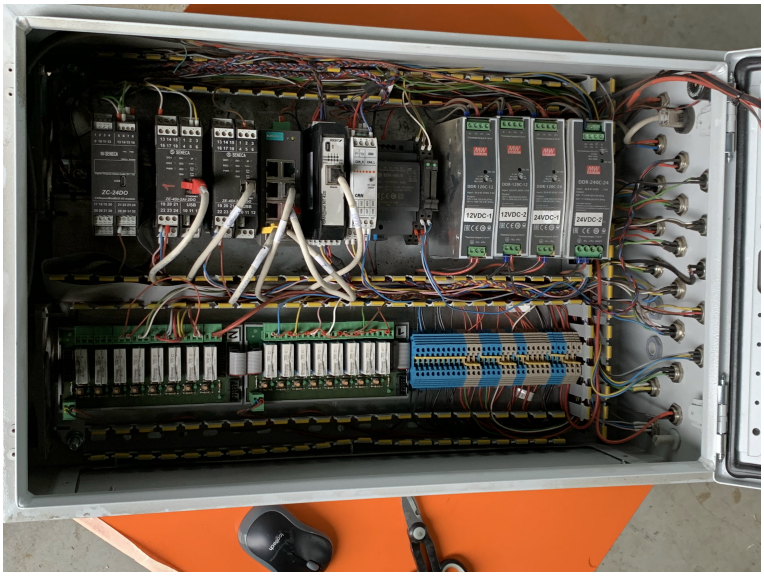
Figure 2.13: HMI panel designed to set and start testing experiments for robot tracks movements

Figure 2.14: Radio remote controller.

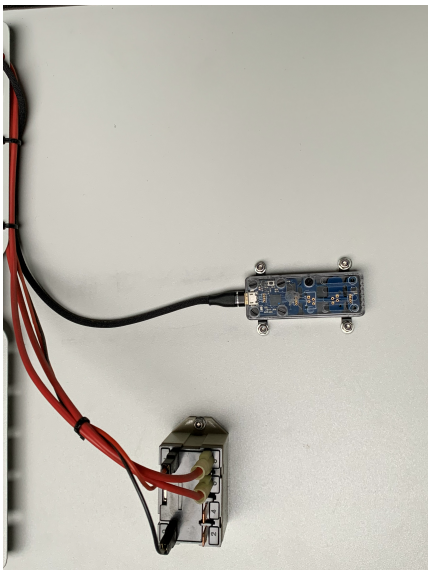
flying drones. It has been adapted since it ensures a reliable communication within a medium range distances.

2.7 I/O Box

Another key subsystem composing the mechatronic core of Surus Prime is the electric box managing the I/O devices. Figure 2.15a shows the inside of this electric box, in 2.15 it is possible to see all the I/O devices reading both digital and analogue signals from all different components of the robot. This box also hosts the DC-DC converts required to supply the right voltage levels to all the electronics devices the robot is equipped with.
It is also worthy to notice that the electric box is designed using only components compliant with the EMC industrial standards, and according to this CANOpen and



(a) I/O box core part.



(b) I/O box contactor.

Figure 2.15: Zoomed view of the I/O management box.

Modbus compatible devices have been selected[1].
Figure 2.15b shows two additional components added to the I/O box for safety reasons. The one on the bottom, as said already above, is a contactor which provides power supply directly from the battery to the components into this electric box, to the other electronics, and low-power electrical components. The presence of this device allows us to interrupt this power branch in case of battery recharge or of emergency. The black components is a temperature and moisture sensor, which is read by the HMI and allows us to monitor the state of this box during operations, either enabling the cooling system in case of too high temperature or turning off the whole system if the moisture level is not safe for the electric components.

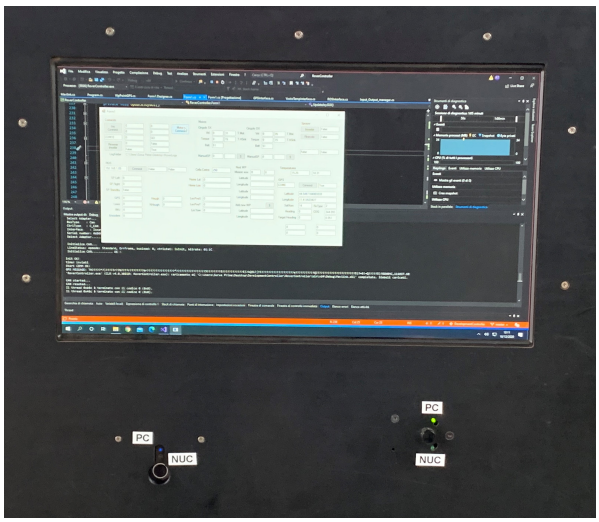
2.8 Manual User interaction

In order to allow the human operator to manually control part of the robot equipment, or to stop everything in case of emergency, Surus Prime is equipped with a button panel and a touch screen monitor.

[1]We decided to use multiple communication protocols in order to learn how to use the most common ones, in particular, this will allow us to switch to the use of a single protocol if required at higher TRL levels, [59].

(a) Rover backside view.

(b) Button panel.



(c) Touch screen and PC switches.

Figure 2.16: Surus Prime backside detailed view.

2.8.1 Button panel

The button panel is composed both by buttons and by LED indication lights, in particular, as shown by 2.16b, the following LEDs are placed on the panel:

- Brake indicator - It shows whether the parking brakes are enabled or not.
- Motor Controller State - Underneath the “Driver Reset” button two LEDs are placed, these have the function of mimicking the status LED present on each one of the two motor controllers. In fact, during normal working condition the motor controllers are not visible, and it would not be possible to quickly debug unusual behaviors.
- Emergency Stop - It show whether the emergency button has been pushed.
- Battery State - Below the “48 VDC” button, a LED indicator is placed, this shows whether the main battery is active or not, and in turn, if the battery is supplying power.

On the other hand, the button or manual controllers available on the panel are:

- Driver Reset button - It resets both the motor controllers.
- Brake switch - It allows to manually enable the parking brakes or to switch to

automatic mode, in which they are managed autonomously by the robot.

- Winch switch - It allows to manually command the winch connected to the equipped farming tool in order to adjust its height.
- Emergency button - This button is to be used in case of emergency, and it cuts out the power to the locomotion systems and to all the devices but the computers.
- NUC and PC buttons - They switch on the NUC and the rugged mini-PC, respectively.
- 48 VDC button - It enables the Battery Management System (BMS) of the main battery, closing the internal contactor and in turn enabling the power output port of the battery.
- Vehicle key - It triggers the contactor inside the I/O box, therefore supplying power to all the system.

2.8.2 Monitor touchscreen

Figure 2.16c shows the main interface with the user, in particular, the touchscreen monitor. During the design, special attention was paid to this aspect, since Surus Prime features two different computers but a single monitor, two switches have been connected to each PC, the first one is an HDMI switch and the second one is an USB switch. The use of these two devices allows one to visualize the output of both of them (not simultaneously) on the same screen and at the same time to have some USB peripherals moved accordingly from one to the other. In our case the USB devices connected to the switch are the touchscreen of the monitor, the mouse and the keyboard.
The most interesting use of the touchscreen is anyway with the Windows-running PC, as it runs the C# interface, allowing the user to interact directly with the visual buttons without the aid of mouse or other pointing device, which is a “nice-to-have” feature in outside applications.

2.9 Conclusions

In this chapter we discussed the different components necessary for assembling a tracked agricultural robot, for sake of completeness Figure 2.17 provides an overview of all the interconnections between each of the described components. Along with

this, we provided detailed views of the main design choices, and a description of the logical organization of hardware and software. In the end, we described how the user can interact/communicate with the robot sending manual commands or schedule autonomous missions.

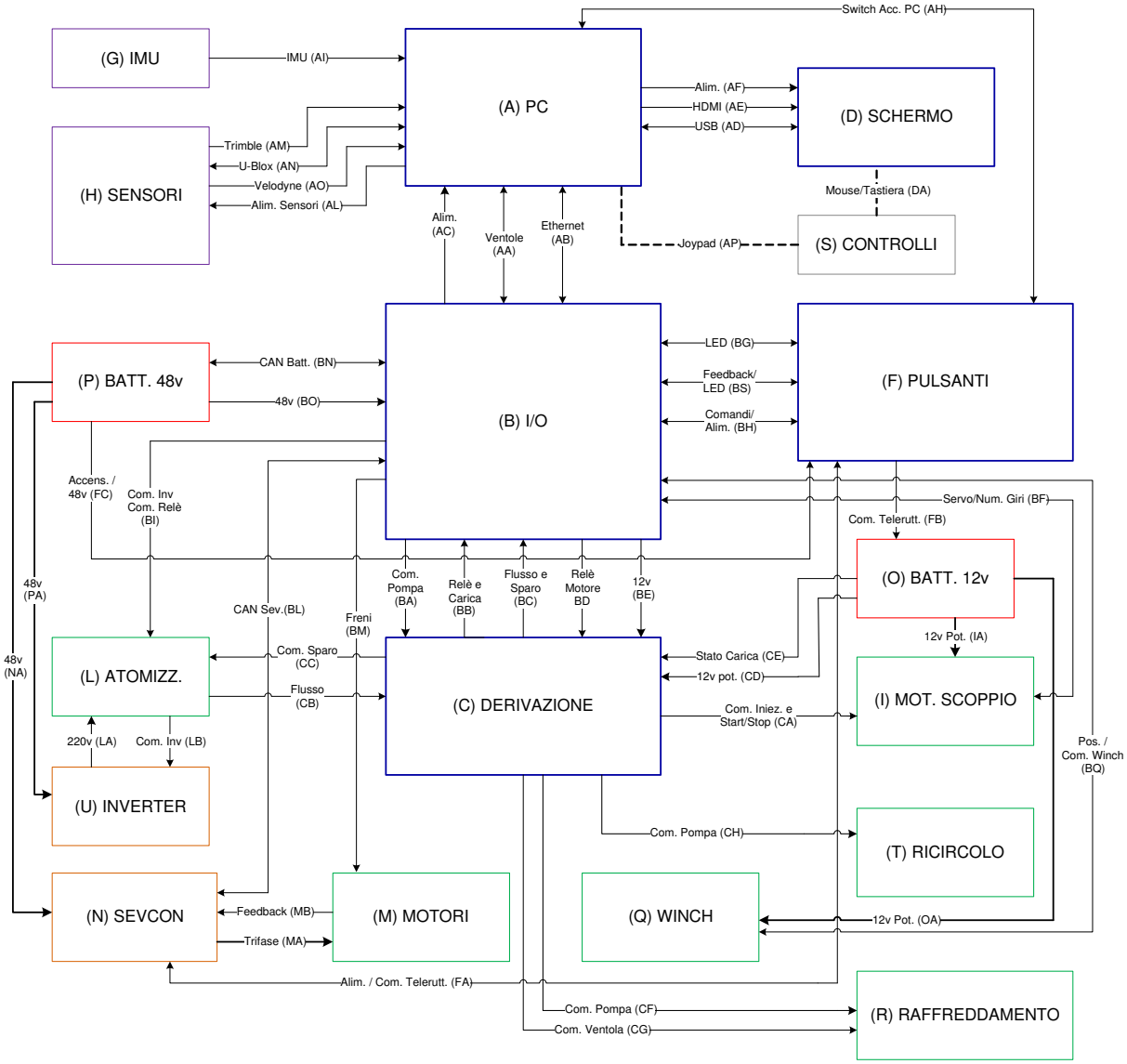


Figure 2.17: Logic scheme of all the elements composing the UGV with their relative interconnections.

3

Vehicle Model

In this chapter the kinematic and dynamic models of a tracked skid-steering vehicle are derived. A novel function describing the pressure distribution on the track is proposed. Then, a model-based design tool for tracked vehicles is presented.

3.1 Introduction

The work presented in this chapter is an extension of our paper [60], where the model of a tracked Skid-Steering Vehicle (SSV) is developed, both from a kinematic and a dynamic point of view. The inherent skidding effects generated by this kind of vehicles are accurately studied and then exploited to achieve a model-based design tool for tracked vehicle.
In addition, a novel approach for modeling the normal pressure distribution under the track is proposed. Experimental results validating the model are presented.

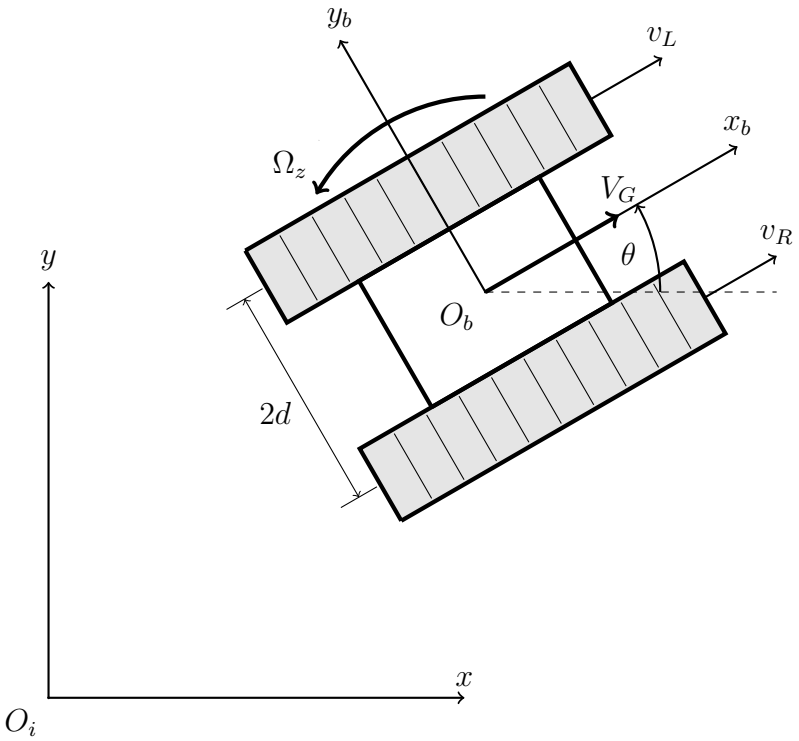


Figure 3.1: Planar motion of a tracked vehicle.

3.2 Kinematic Modeling

Under the main assumption of planar motion, we consider the following kinematic model described with respect to an inertial reference frame $\mathcal{F}_i(O_i, x, y, z)$ as

$$\begin{aligned} \dot{x} &= V_G \cos\theta \\ \dot{y} &= V_G \sin\theta \\ \dot{\theta} &= \Omega_z \end{aligned} \tag{3.1}$$

where V_G is the linear velocity of the Center of Gravity (CoG) in body coordinates, $\mathcal{F}_b(O_b, x_b, y_b, z_b)$, and Ω_z represents the angular velocity of the vehicle around the axis normal to the motion plane. The main quantities of interest are also presented in Figure 3.1.

Due to their nonholonomic nature, both SSVs and Differential Wheeled Robots (DWRs) kinematics can be described using (3.1). The main difference between them

lies in the expressions of V_G and Ω_z. In fact, for DWRs they can be computed as

$$\begin{aligned} V_G &= \frac{v_R + v_L}{2} = \frac{r}{2}(\omega_R + \omega_L) \\ \Omega_z &= \frac{v_R - v_L}{2d} = \frac{r}{2d}(\omega_R - \omega_L) \end{aligned} \tag{3.2}$$

while for SSVs the same quantities are expressed as

$$\begin{aligned} V_G &= \frac{v_R(1 - i_R(t)) + v_L(1 - i_L(t))}{2} = \frac{r}{2}(\omega_R(1 - i_R(t)) + \omega_L(1 - i_L(t))) \\ \Omega_z &= \frac{v_R(1 - i_R(t)) - v_L(1 - i_L(t))}{2d} = \frac{r}{2d}(\omega_R(1 - i_R(t)) - \omega_L(1 - i_L(t))). \end{aligned} \tag{3.3}$$

The quantities ω_R, ω_L are the right and left motor speed, respectively, and similarly, v_R, v_L represent the linear velocities of the right and left wheel in case of DWRs, or track for SSVs. The quantity r is the wheel radius for the first scenario, while in the second one it represents the radius of the driving sprocket. Then, d is half the distance between the center lines passing through the two wheels or tracks, also shown in Figure 3.1. The difference between (3.2) and (3.3) is given by the presence of $i_R(t), i_L(t) \in (-1, 1)$ that are the time-varying slip coefficients associated with the right and left track, respectively.

3.2.1 Slip coefficients model

While it is a well known fact ([61, 1, 62, 63]) that the coefficients i_R, i_L depend on the soil type as well as on the track velocities and, to the best of our knowledge, a mathematical model capturing the underlying complex relation is not available in literature.
In [62] and [63] it is observed that there is a nonlinear relation between the ratio $i_R(t)/i_L(t)$ and v_R, v_L but this result is limited to small turning radii, and a complete relation also including soil dependency is still missing.
In general, from a kinematic point of view, track slip coefficients are expressed as

$$i = 1 - \frac{V_t}{v_i} = 1 - \frac{V_t}{\omega r} \tag{3.4}$$

where i is the slip coefficient of a track, v_i is the ideal velocity of the track given by the product of the driving sprocket rotational speed ω and its radius r, and V_t is the actual

speed of the contact point of the track with the ground (which is hardly measurable). Equation (3.4) shows the relation between the value of the slip coefficient and the track velocity, but it does not show any dependency on soil-related parameters.
On the other hand, [61] and [1] report a relation between slip coefficients and the traction force of the track on the ground, which depends on the physical properties of the terrain according to

$$F = F_{max}\left[1 - \frac{K}{il}\left(1 - e^{-il/K}\right)\right] \tag{3.5}$$

where l is the length of the contact surface between the track and the ground, K represents the soil shear deformation modulus and F_{max} is the maximum traction force developed by the track. This force can be expressed as

$$F_{max} = S_c\tau_{max} = S_c\left(c + p\tan\varphi\right) = S_c c + \frac{W}{2}\tan\varphi \tag{3.6}$$

in which S_c is the track-ground contact surface, τ_{max} the maximum shear strength of the terrain, c the apparent cohesion of the terrain and φ the angle of internal shearing resistance of the soil, p is the normal pressure acting on the track and W the total weight of the vehicle. Inverting (3.5) it is then possible to retrieve the slip coefficient of the track, given soil parameters and the traction forces applied by each of them, as done in [64]. Again, the relation found is incomplete, as it does not relate both track velocities and soil features in a single equation.
In conclusion, both (3.4) and (3.5) are incomplete and therefore there is no advantage in considering the dynamic model over the kinematic one for problems of slip estimation. Our approach to solve the problem of model-free slip estimation, also presented in [65], will be explained in details in Chapter 4.
On the other hand, the dynamic model is fundamental for making estimates of power consumption as well as for vehicle sizing. Indeed the next section describes how to use this kind of model for these particular purposes.

3.3 Dynamic Modeling

The main goal of this section is to develop the dynamic analysis of skid-steering tracked vehicles, following the rationale outlined by Steeds in [66], Bekker in [67], [61] Wong in [68], [1] then extended to soft terrains by Al-Milli in [69]. Then, we enriched

Figure 3.2: Ground shearing caused by turning maneuver of a skid-steering tracked vehicle.

it with non-uniform Normal Pressure Distribution Functions inspired by the work of Wills [70], Keller [71] and by Liu's [72]. The robotic prototype built was then used to perform experimental tests aiming at the validation of the proposed model. Static stability is a key aspect while designing an autonomous vehicle which has to travel through an irregular terrain such as the one typically encountered in agriculture. This is the reason why caterpillars, rather than wheels, have been chosen: crawler mechanisms feature an higher level of stability because of the wider contact surface that increases the capability to adapt to the irregularity of the ground.
As already highlighted in the previous section, the main drawback of this enhanced stability lies into the turning mechanism: in SSVs, in fact, turning maneuvers are accomplished by actuating each side at a different rate or in a different direction, causing the tracks to slip, or skid, on the ground, causing the robot to shear the terrain (see Figure 3.2). Modeling these phenomena is important because it allows us to define a relation between geometric design parameters of the vehicle, trajectory coefficients and ground features.
The first part of the following dynamic analysis is strongly built on the work done in the field of SSVs by [61, 68, 1] and more details about the whole derivation of the model can be found in Appendix A.

3.3.1 Dynamic Model

The model of the forces acting onto a tracked vehicle during planar steering motion (as shown in 3.3) is given by

$$m\ddot{s} = m\dot{V_G} = F_o + F_i - R_{tot} \tag{3.7}$$

$$J_z\ddot{\vartheta} = J_z\dot{\Omega}_z = (F_o - F_i)d - M_r \tag{3.8}$$

where s and V_G are the linear displacement and velocity of the COG, ϑ and Ω_z denote the angular displacement and velocity about the turning center O, m and J_z are respectively the linear and angular inertia, F_i and F_o the motor thrusts generated by the inner (i) and outer (o) track, d is half of the distance between the longitudinal center lines of the tracks and R_{tot},M_r are respectively the resistive force and torque.
Considering (3.7), (3.8) in the case of steady-state turning maneuvers and following the derivation given in [1], further refined in [68], it is possible to express the forces required by the two tracks as follows

$$F_o = \frac{R_{tot}}{2} + \beta\omega_o + \frac{M_r}{2d} = \frac{f_r mg}{2} + \beta\omega_o + \frac{M_r}{2d} \tag{3.9}$$

$$F_i = \frac{R_{tot}}{2} + \beta\omega_i - \frac{M_r}{2d} = \frac{f_r mg}{2} + \beta\omega_i - \frac{M_r}{2d} \tag{3.10}$$

in which f_r represents the rolling friction coefficient along the longitudinal direction of motion, β represents the viscous friction coefficient that is due to the damping effects of the mechanisms connected to the motors, ω_o,ω_i are the rotational speed of the sprocket driving the outer and inner track, respectively, and g represents the gravitational acceleration. The turning moment M_r is dependent on the footprint of the track on the ground and, clearly, on terrain parameters. Considering uniform normal pressure distribution along the track, and letting the lateral resistance per unit length of the track eb $R_L = \frac{\mu_t mg}{2L}$, the resisting turning moment can be expressed as

$$M_r = 4\mu_t \frac{mg}{2L}\int_0^{L/2} x dx = \mu_t \frac{mgL}{4}, \tag{3.11}$$

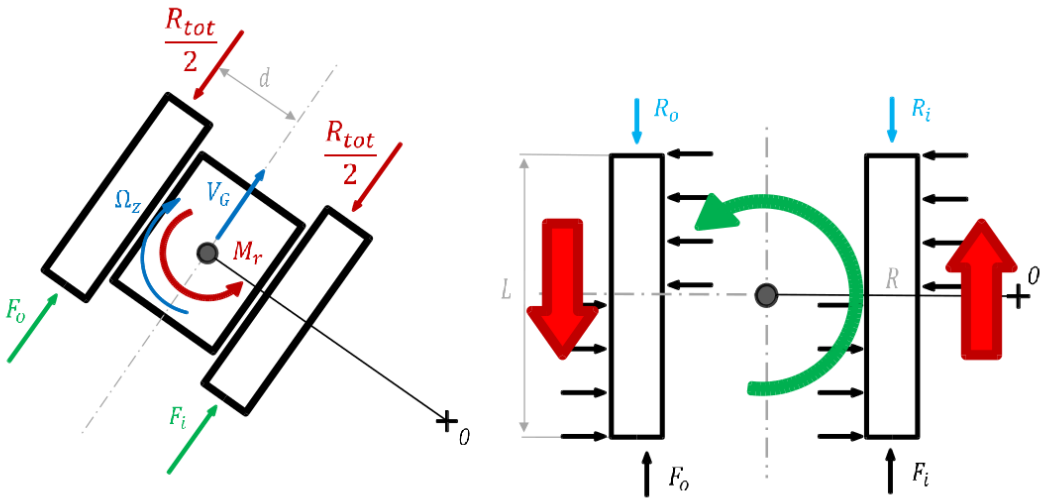


Figure 3.3: Forces acting on a tracked vehicle.

where μ_t is the lateral friction coefficient and L the length of the track contact surface.[1] Thereby, (3.9) and (3.10) become

$$F_o = \frac{f_r mg}{2} + \beta\omega_o + \mu_t \frac{mgL}{4 \cdot 2d} \approx \frac{mg}{2}\left(f_r + \mu_t \frac{L}{4d}\right) \tag{3.12}$$

$$F_i = \frac{f_r mg}{2} + \beta\omega_i - \mu_t \frac{mgL}{4 \cdot 2d} \approx \frac{mg}{2}\left(f_r - \mu_t \frac{L}{4d}\right). \tag{3.13}$$

In the final expression of the (3.12) and (3.13) we neglected the $\beta\omega_o$,$\beta\omega_i$ terms, as these effects are not relevant for the purpose of this section. Taking a closer look at (3.12) and (3.13), it is possible to notice that they do not express any dependency of F_o and F_i on either the specific trajectory (in terms of V_G, Ω_z, turning radius R') or design parameters (such as COG position, number of rollers).
Nevertheless, empirical tests show that the motor thrusts highly depend on these parameters, e.g. decreasing the turning radius R' results in an increase of M_r.
Therefore, embedding in the expression of F_o, F_i the dependence on the above listed parameters is of crucial importance.
The main cause of this problem is given by the assumption implicitly made at the beginning of the analysis, [66], where the shear stress developed along the track-ground

[1] Notice that, for sake of simplicity, we are neglecting centrifugal acceleration, since the magnitude of the velocity (V_G, Ω_z) is quite limited in our study case. Nevertheless, in A the effects related to this phenomenon are studied in details.

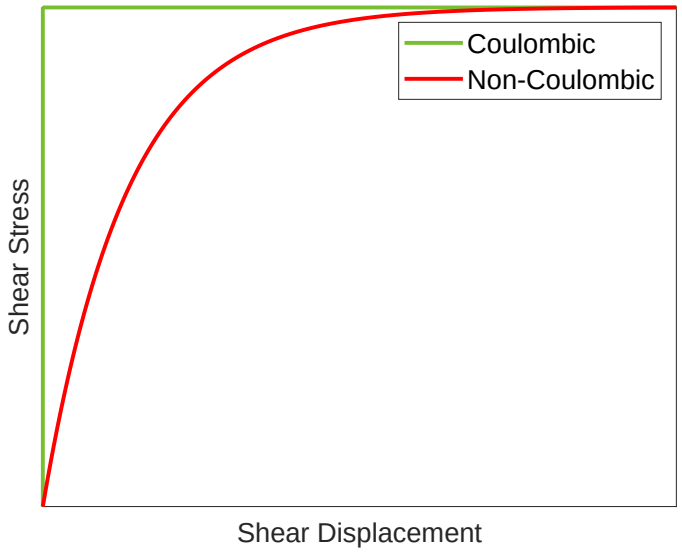


Figure 3.4: Comparison between Coulombic friction model and *"Extended Coulombic friction law"*.

contact surface is assumed to obey the Coulomb law of friction. Thus, the resultant shear stress on each track element is a discontinuous function of the shear displacement, and it acts in the opposite direction of the relative motion between the track element and the ground. Experimental evidences, [1], show that the shear stress developed on the track-ground interface is indeed a continuous function of the shear displacement, as reported by the red line in Figure 3.4, and it is more accurately modeled as an *Extended Coulombic friction law* of the form

$$\tau = \left(c + \sigma\mu\right)\left(1 - e^{-j/K}\right) \tag{3.14}$$

where τ is the shear stress, σ represents the normal pressure, μ the coefficient of friction between the track and the ground, j the shear displacement, K is the shear deformation modulus (with $K = 0$ capturing the Coulombic friction law) and c denotes the cohesion of the ground. The cohesion term was introduced in [69], allowing us to take into account soft grounds as well as firm ones.
Furthermore, notice that in [1], σ is assumed to be constant along the whole track contact surface. This assumption can be relaxed to take into consider also more realistic normal pressure distribution, as also described in Section 3.4.
The (3.14) can be now used to derive a more realistic expression of M_r. In particular,

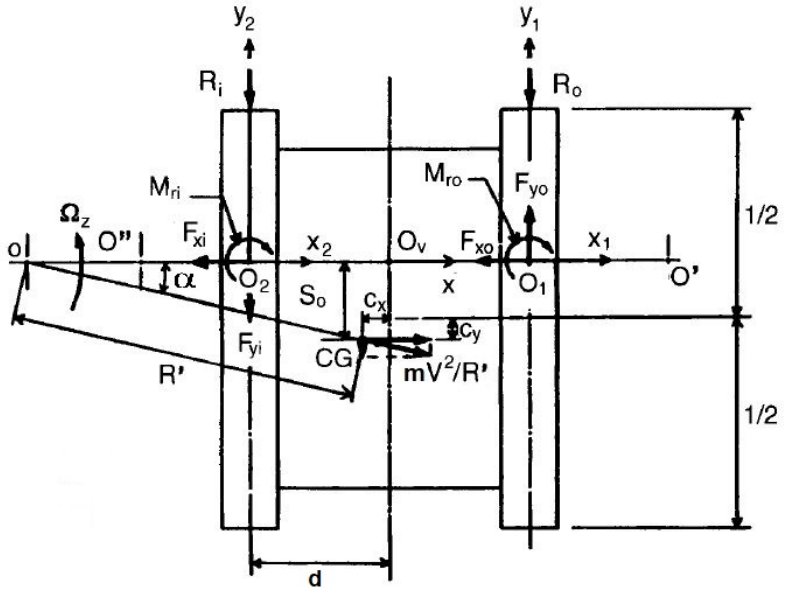


Figure 3.5: Forces acting on a tracked vehicle during a steady-state turn, [1]

the resistive lateral moments, generated by the turning motion of the outer (M_{r_o}) and inner track (M_{r_i}), caused by the lateral shear forces with respect to O_1 and O_2, shown in Figure 3.5, are given by

$$M_{r_{o,i}} = -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0}\int_{-W/2}^{W/2} y_{1,2}\left(c+\sigma_{o,i}\mu\right)\left(1-e^{-j_{o,i}/K}\right)\cos\delta_{1,2}\,dx_{1,2}dy_{1,2} \tag{3.15}$$

$$\cos\delta_{1,2} = \frac{-y_{1,2}\Omega_z}{\sqrt{\left[\left(R''+d+c_x+x_{1,2}\right)\Omega_z - r\omega_{o,i}\right]^2 + \left(y_{1,2}\Omega_z\right)^2}}, \tag{3.16}$$

where L, W are the length and width of the tracks contact surface (assuming the two tracks to be equal), c_x, c_y are the lateral and longitudinal distances between the COG and the longitudinal and lateral center lines of the vehicle hull respectively, s_O is the shift in the center of turn due to the presence of slippage, r is the sprocket radius, $(x_1, y_1), (x_2, y_2)$ are respectively the coordinates of a generic point on the outer or inner track, σ_o, σ_i the normal pressures considered for the outer/inner track, j_o, j_i shear displacement of the outer/inner track, δ_1, δ_2 are the angles between the resultant sliding velocities and the lateral directions of the points on the outer/inner track respectively and $R'' = \sqrt{R'^2 - s_O^2}$ is the lateral distance between the center of rotation O and the COG.

In the end, considering that $M_r = M_{r_o} + M_{r_i}$, the coefficient of lateral resistance μ_t can be obtained from the (3.11) as

$$\mu_t = \frac{4(M_{r_o} + M_{r_i})}{mgL} \tag{3.17}$$

and turns out to be function of $W, L, c_x, c_y, V_G\Omega_z, R'', \mu, c$. According to (3.17) the parameter μ_t is expressed as function of the design parameters (W, L, c_x, c_y), trajectory parameters (V_G, Ω_z, R'') and terrain features (μ, c).

3.4 Non-Uniform Normal pressure distribution

Until this point, it has been assumed that the weight was uniformly distributed via the tracks to the ground in contact with them.
This assumption is quite strong, since most of the tracks have grousers which are quite stiff but not infinitely rigid, and this affects the distribution of the normal pressure (or weight) under the track, especially if the length of the track is big and if they are made of plastic.
Throughout this section it will be discuss how different normal pressure distributions can be modeled and how these distributions affect the resulting resistive torques M_r.
In Figure 3.6 different typologies of Normal Pressure Distribution (NPD) are shown, in particular the *(b)* and *(c)* are compared with the uniform one (*a*). The main common features between these two distributions is that they both present peaks in correspondence of roadwheels position, those peaks represent that right under the central position of the wheels, the pressure reaches its maximum and then it decreases to zero as long as the considered point get further from the wheel center, and pressure increases again getting closer to the following wheel mid-point.
The NPD taken into consideration model in a more accurate way the presence of roadwheels along the track length and the grouser elasticity, therefore, an important parameter that the pressure distribution function has to include is the number of roadwheels. We assume that, independently on the number of wheels, they are displaced in a symmetric fashion with respect to the mid of the track.

Cosinusoidal Pressure Distribution

In the following, the second pressure distribution function reported in Figure 3.6 will be studied, in order to point out the pros and cons of this specific function. In partic-

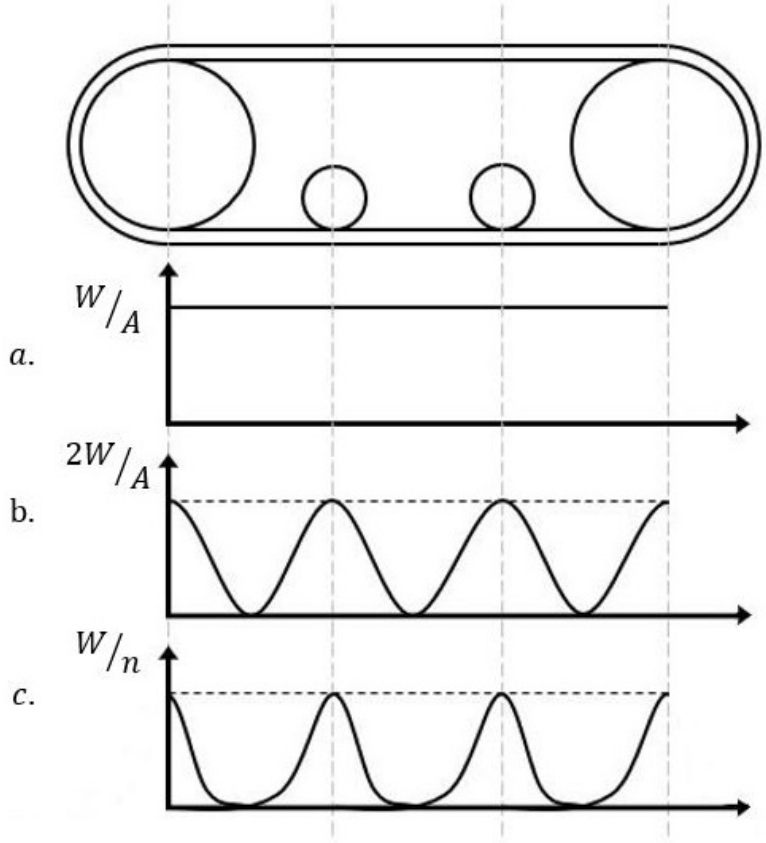


Figure 3.6: Different kind of Normal Pressure Distribution. (a.) Uniform, (b.) Sinusoidal, (c.) Gaussian.

ular, the sinusoidal pressure distribution function proposed by [70] can be expressed as

$$\sigma(y) = \frac{m_h g}{A}\left[1 + \cos\left(\frac{2n\pi y}{L}\right)\right] \tag{3.18}$$

where $m_h g$ represents half of the vehicle weight[1], A the contact surface between the track and ground, $n = N - 1$ and N is the number of roadwheels along the track, y is the coordinate along the length of the track L.
Notice that, the pressure is considered to be just function of the y-coordinate only and not of the x-coordinate of the track (i.e. the pressure is considered uniform for all the points with the same y), as shown in Figure 3.7.

[1] In this context, for sake of simplicity, only one track is considered, but the analysis can be straightforwardly extended to both of them.

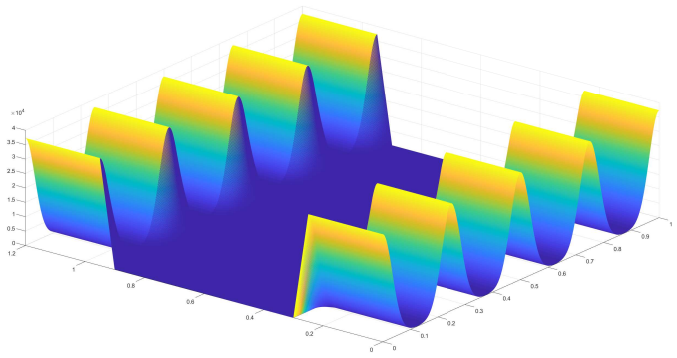


Figure 3.7: Normal Pressure Distribution maps onto track contact surface.

Focusing, now, the attention onto the structure of the (3.18) it is possible to see that a *cosinusoidal* function is considered and its values are shifted up by 1, this makes it always positive, which make sense as a negative pressure, in this context, is considered meaningless; it has been chosen a $\cos(y)$ function instead of a $\sin(y)$ function in order to have one of the peaks at the very beginning of the track since the first touching point is considered to be the central point of the first roadwheel in the track.
Figure 3.8 shows that changing N and, in turn n, the number of peaks of the function changes, but also the "frequency" of the cosinusoidal function varies, making each sinusoid narrower. This, in turn, suggests to model the system as it had a smaller size of track roller.

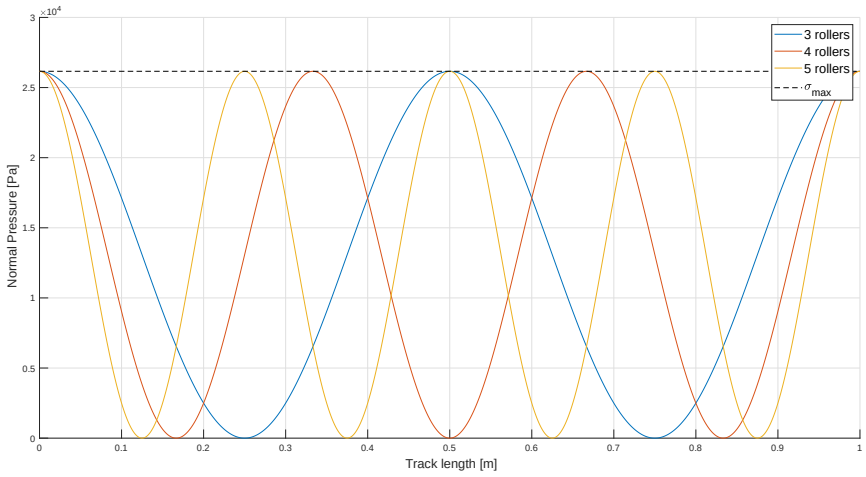


Figure 3.8: Cosinusoidal NPD functions changing the number of rollers.

As Figure 3.8 shows, increasing the number of roadwheels, "automatically" makes

the width of the sinusoid decreasing, therefore there is no parameter related to the size of the roadwheels in the track, furthermore, peak values are not dependent on the number of roadwheels present, hence independently on the latter number, it is considered to always have the same pressure peak value.
Another important property of the ideal pressure distribution function is that the integral of the function over the total track length has to be equal to half of the overall weight, feature that is easily proven by the following integral

$$\iint_A \frac{m_h g}{A}\left[1+\cos\left(\frac{2n\pi y}{L}\right)\right]dA = \frac{m_h g}{A}\left\{W\int_0^L\left[1+\cos\left(\frac{2n\pi y}{L}\right)\right]dy\right\} \tag{3.19}$$

where W is the track width. Developing further the (3.19) it comes out that

$$\begin{aligned}\frac{m_h g}{A}W\left\{\int_0^L 1 \quad dy+\int_0^L\cos\left(\frac{2n\pi y}{L}\right)dy\right\} &= \frac{m_h g}{A}W\left\{\Big[y\Big]_0^L+\frac{L}{2n\pi}\left[\sin\left(\frac{2n\pi y}{L}\right)\right]_0^L\right\}\\ = \frac{m_h g}{A}W\left\{\Big(L-0\Big)+\frac{L}{2n\pi}\Big(\sin(2n\pi)-\sin(0)\Big)\right\} &= \frac{m_h g}{A}WL = m_h g\end{aligned} \tag{3.20}$$

The result obtained above shows that the function considered is an eligible function for describing the NPD under the track.

Having defined and analyzed the expression of the pressure distribution, it's now useful to study the effect of this pressure function into the resistive moment equation. Taking into consideration a reduced version of the (3.14) in which, for sake of simplicity, the *cohesion* term is neglected, and then, expressing the pressure as function of the position, it follows that

$$\tau = \sigma(y)\mu\big(1-e^{-j/K}\big) = \left(\frac{m_h g}{A}\left[1+\cos\left(\frac{2n\pi y}{L}\right)\right]\right)\mu\Big(1-e^{-j/K}\Big). \tag{3.21}$$

Equation (3.21) is inserted into the computation of the forces acting on the two tracks by obtaining

$$\begin{aligned} dF_o &= \tau_o dA = \sigma_o(y)\mu\big(1-e^{-j_o/K}\big)dA && \textit{(on outside track)}\\ dF_i &= \tau_i dA = \sigma_i(y)\mu\big(1-e^{-j_i/K}\big)dA && \textit{(on inside track)}\end{aligned} \tag{3.22}$$

where τ_o,τ_i are respectively the shear stress of the outer and inner track, $\sigma_o(y), \sigma_i(y)$ are the modified pressure relatively of the outer and inner track.
It follows that, including the (3.22) in (3.15) and (3.16) the following resistive moments acting on the two tracks are obtained:

$$M_{r_o} = -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0}\int_{-W/2}^{W/2} y_1\sigma_o(y)(y_1+\left(L/2+c_y-s_0\right))\mu\left(1-e^{-j_o/K}\right)\cos\delta_1 dx_1 dy_1 \quad (3.23)$$

$$M_{r_i} = -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0}\int_{-W/2}^{W/2} y_2\sigma_i(y)(y_2+\left(L/2+c_y-s_0\right))\mu\left(1-e^{-j_i/K}\right)\cos\delta_2 dx_2 dy_2 \quad (3.24)$$

Notice that, in (3.23) and (3.24), the pressure distribution functions are expressed in terms of $y_1+\left(L/2+c_y-s_0\right)$ and $y_2+\left(L/2+c_y-s_0\right)$ due to the fact that the pressure distribution function defined in (3.18) has coordinates going from 0 to L whilst the integration variables are defined within the interval $\left]-L/2+c_y-s_0, L/2+c_y-s_0\right[$
The pros characterizing this Cosinusoidal NPD (CNPD) function can be summarized as follows: (i) it is dependent on the number of roadwheels; (ii) it is symmetric with respect to the track mid point; (iii) the integral over the track contact surface gives the weight. On the other hand, the cons can be listed as: (i) no parameters related to the size of the roadwheels; (ii) peak value not dependent on the number of roadwheels.

3.4.1 Gaussian Pressure Distribution

Considering the pros and cons outlined above, it can be noticed that the pressure distribution defined by the cosinusoidal function can be somehow enhanced, for instance considering the sum of n gaussian functions centered in the mid point of the roadwheels.

$$\sigma(x) = \sum_{i=1}^{N} \frac{m_h g}{nA}\frac{1}{r_w\sqrt{2\pi}} e^{-\frac{(x-l_i)^2}{2\left(r_w L\right)^2}} \qquad \textit{with} \quad l_i = \frac{L}{n}(i-1) \quad (3.25)$$

where N is the number of roadwheels in the track, r_w is a parameter related to the radius of the roadwheel and l_i represents the mid point of the wheels of the track.

Note that (3.25) is composed by the sum of normalized gaussian functions, whose integral can be expressed as

$$\iint_A \frac{m_h g}{nA} \frac{1}{r_w \sqrt{2\pi}} e^{-\frac{(y-l_i)^2}{2(r_w L)^2}} dA = \frac{m_h g}{n}. \tag{3.26}$$

Therefore, the integral of the sum of these N functions gives the overall weight $(m_h g)$[1]. Despite the pressure distribution defined before, (3.25) shows peak values that change depending on the total number of roadwheels; featuring lower peak values in case of higher number of wheels and vice versa.
In addition, there is a parameter (r_w) that is related to the radius of the wheels, giving the chance to analyse the presence of a different number of wheels, keeping constant the size of these ones.

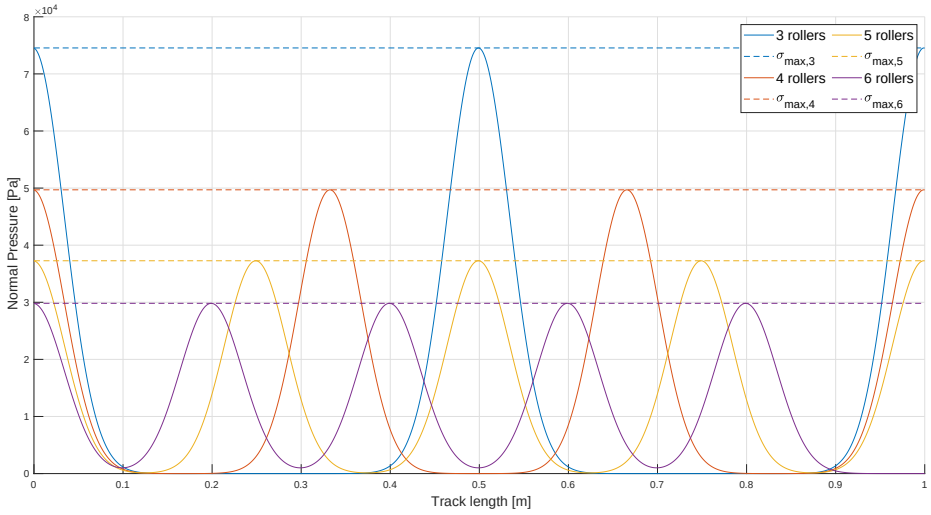


Figure 3.9: Gaussian pressure distribution functions changing the number of rollers.

Figure 3.9 shows all the features listed before of the Gaussian NPD (GNPD):

- Increasing the number of wheels, the peak value under each of those decreases

[1]Notice that, the correct form should be $\iint_A \frac{m_h g}{nA} \frac{1}{r_w \sqrt{2\pi}} e^{-\frac{(y-l_i)^2}{2(r_w L)^2}} dA \approx \frac{m_h g}{n}$, since the equality sign is not correct, as it would be true only if $y \in (-\infty, \infty)$. For sake of simplicity we consider the equal sign, since for small values of N the error generated is quite small. More about this will be discussed later.

- The width of the "bell", representing the radius of the wheel, can be kept constant varying the number of wheels

Figure 3.10 highlights an important behavior that it is possible to map thanks to the chosen Normal Pressure Distribution function. It is intuitive that a wider roadwheel will spread pressure in a larger area and it can happen that the resulting pressure of a given point along the track length could be affected by more than one wheel. Therefore, even points not directly in contact with the wheels feature pressure different from zero, and this is an important effect that could not be mapped with the previous pressure distribution function. In fact, the second part of Figure 3.10 shows that, due to the deformation of the terrain and flexibility of the tracks, there exist pressure also in points not in contact with the roadwheels.

This feature can also be exploited in order to map soft ground, in which, due to the failure of the terrain, the part of the track characterizing a higher level of pressure sinks into the ground and in turn the track deforms as shown in Figure 3.10.

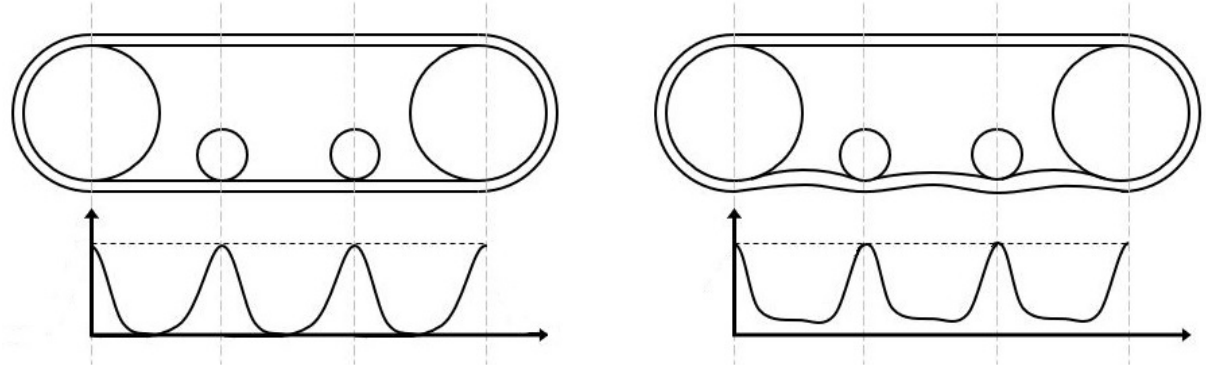


Figure 3.10: Gaussian Pressure Distributions in case of hard and soft ground.

Therefore the (3.25) can be modified in order to add a term related to the firmness of the terrain (k_f), as follows

$$\sigma(y) = \sum_{i=1}^{N} \frac{m_h g}{nA} \frac{1}{k_f r_w \sqrt{2\pi}} e^{-\frac{(y-l_i)^2}{2(k_f r_w)^2}} \qquad \textit{with} \quad l_i = \frac{L}{n}(i-1). \tag{3.27}$$

Comparing the Sinusoidal and the Gaussian Pressure Distributions, it comes out that

the one we proposed solves the problem of peak values, which are now related to the number of rollers. Then, it also introduces a parameter related to the dimensions of track wheels, and it allows one to take into consideration the deformable nature of the terrain and of the track. In the following section we will leverage the knowledge derived by modeling the dynamics of tracked skid-steered vehicle to define a model-based design tool.

3.5 Model-based design procedure

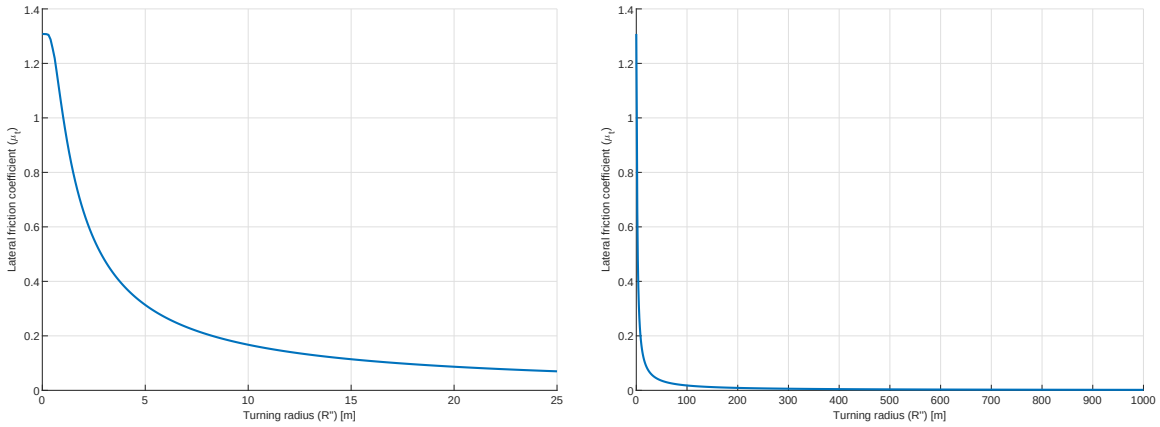
The design procedure of a tracked UGV for agricultural application can be driven and optimized leveraging the analysis carried out in the previous section. In particular, it is interesting to study the effects on μ_t of design parameters such as:

- Track to track distance ($2d$)
- Length and Width of the tracks (L, W)
- Number and radius of the track rollers (N, r_w)

For the time being we do not consider the effects of parameters such as R'', V_G, Ω_z, since they can be taken into consideration afterward, while planning optimal trajectories in terms of energy-efficient motion planning.
It is anyway, interesting to validate the dependency of M_r on the turning radius, as reported by the plots in Figure 3.11, in particular it can be noticed that, as R'' increases μ_t decreases, approaching zero with the turning radius going to infinity, as highlighted in Figure 3.11b. This remark shows that the new model depends on turning radius, and it tends to the simpler case of straight motion, where $M_r = 0$.
To plot Figure 3.11, and, in general, to exploit the analysis performed in previous sections (3.15) has to be computed, which is not possible via analytical methods. Therefore, we solved it via numerical integration along the two dimensions of the track contact surface, achieving

$$\begin{aligned} M_{r_{o,i}} &= -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} f(x_{1,2}, y_{1,2}) dx_{1,2} dy_{1,2} \\ &\approx -\sum_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \sum_{-W/2}^{W/2} f(\Delta L, \Delta W) \Delta A \end{aligned} \tag{3.28}$$

with $\Delta A = \Delta L \Delta W$ denoting finite discretization of the contact surface, while ΔL and ΔW are the unitary elements of length and width, respectively.



(a) μ_t as function of small turning radii. (b) μ_t as function of large turning radii.

Figure 3.11: μ_t as function of turning radii R''.

3.5.1 Vehicle Geometry and Size

The first step to design a tracked UGV concerns the sizes of the vehicle itself, namely its mass m and the distance between the two tracks d.
Keeping the vehicle mass as small as possible mitigates the ruts depth on the ground caused by vehicle passage, as extensively discussed in Chapter 1, as well as reducing the overall resistive torque, since the mass defines the normal pressure which, in turns, generates friction. Accordingly to this, and to the applications to be performed by an autonomous vehicle for agriculture, the ideal mass, m, ranges approximately between 350 to 1000 kg[1].
Figure 3.12 shows how the design parameter μ_t changes accordingly to both the turning radius (R'') and half the track to track distance (d). On the other hand, we observe that larger d leads to lower values of μ_t (in case of small R''). We observe that, for each d, the largest μ_t (and in turn M_r) is reached when the turning radius equals d. This peak is given by the fact that, in these cases, one of the two tracks is completely blocked, producing the highest level of slippage among the different values of R''.
Considering the plot and the above analysis, the optimal design would lead to a large d, however we have to bear in mind that the robot has to work within orchard lanes, which impose an upper bound on d. In addition, the right choice of the vehicle width has also to ensure enough margin for maneuvers within lanes.

[1]Notice that the actual prototype weighs 600 kg (850 kg with the liquid tank completely full).

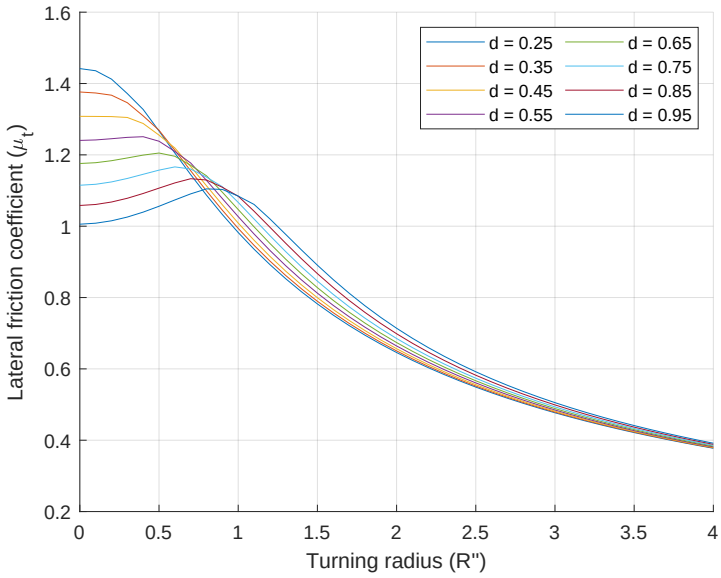
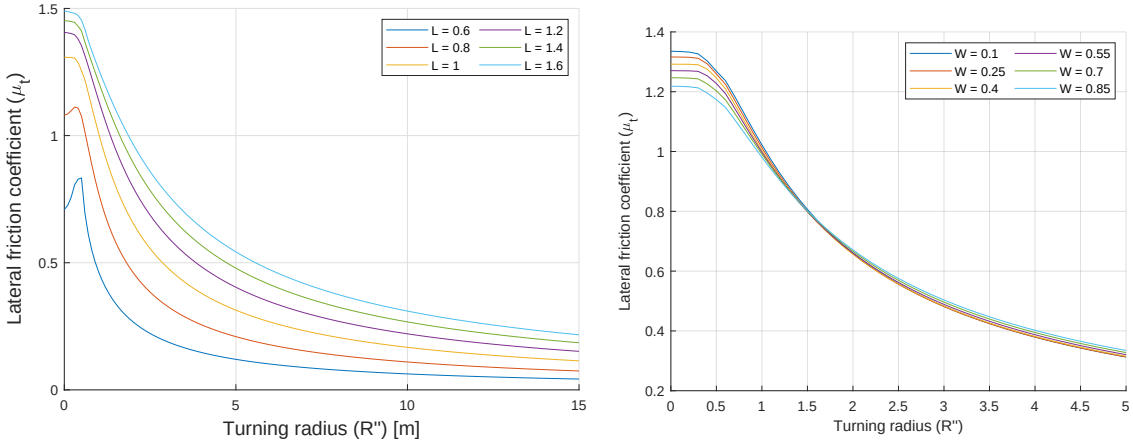


Figure 3.12: Variation of μ_t as function of different values of d.

3.5.2 Track design



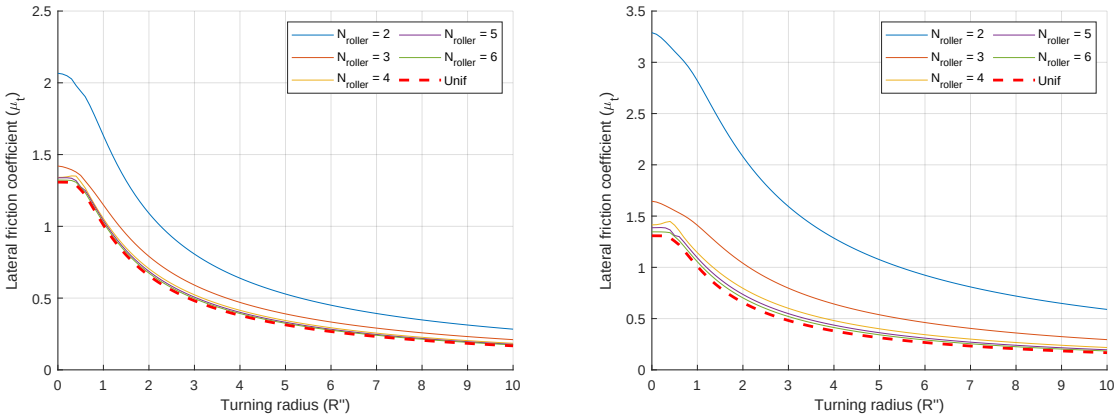
(a) μ_t as function of track length L.

(b) μ_t as function of track width W.

Figure 3.13: Variation of μ_t as function of different values of L and W.

The length of the track is an important parameter since it defines, along with d, the "static stability" of the robot. Therefore, the longer the track the more stable the robot is. On the other hand, as Figure 3.13a highlights, the longer the track, the higher μ_t and, in turn, the motor force/torque. In sizing L, it must be considered also that the robot has to be big enough to host the required tools and components to carry

out its tasks.
Figure 3.13b shows that large variations of W produce small variations in μ_t. This allows the designer for large decision margin. At the same time, if the problem of soil compaction is of critical relevance, as it can happen in orchard applications, the larger the track is the better it is. It then follows that, once again, the final application affects design choices.



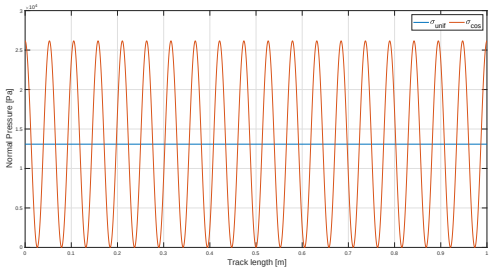
(a) μ_t as function of N, considering sinusoidal NPD.

(b) μ_t as function of N, considering gaussian NPD.

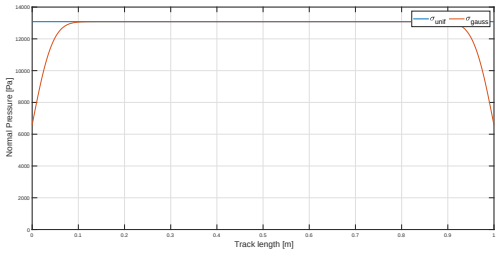
Figure 3.14: Variation of μ_t as function of different roller number, N. The r_w coefficient was fixed at $r_w = 0.075$m.

In the end, Figure 3.14[1] shows that, considering non-uniform NPD, the higher the number of rollers the more μ_t resembles the one defined by uniform NPD. In particular, Figure 3.14 compares the values of μ_t obtained using CNPD, namely Figure 3.14a, and the ones derived considering the proposed GNPD, Figure 3.14b. Observe that, for small numbers of rollers, the CNPD function provides smaller values of lateral friction coefficient if compared to the gaussian one. The reason for this phenomenon can be found looking at Figure 3.8 and Figure 3.9. In contrast to the CNPD, the peak values of the GNPD change accordingly to the number of wheels. Therefore, for small values of N, higher peaks are present at a large distance from the center of the track, thus the forces generated by these pressure peaks feature large arms, resulting in larger M_{r_o}, M_{r_i}. This, in turn leads to bigger values of the later friction coefficient.
Another remarkable difference between the two NPD functions lies in how they ap-

[1]The actual relation between r_w and the effective roller radius is still under investigation. Similarly, we are performing tests aimed at establishing a relation between k_f, ground parameters and the elasticity of the track.



(a) Cosinusoidal NPD, $N = 20$.



(b) Gaussian NPD, $N = 20$.

Figure 3.15: Variation of μ_t as function of different roller number, N. The r_w coefficient was fixed at $r_w = 0.075$m.

proximate uniform pressure distribution as the number of rollers increases, this can be observed in Figure 3.15. The CNPD, in Figure 3.15a, keeps the sinusoidal shape independently on the number of rollers, producing peaks and troughs, which does not represent properly the real behavior. On the other hand, GNPD better approximates the uniform NPD, as shown in Figure 3.15b, except for the lateral terminals. These "rounded edges" of the GNPD can be attributed to the fact that (3.26) holds as far as the integral is computed within the interval $(-\infty, \infty)$, and as long as the number of the gaussians is small the approximation is good even if the integral is computed only spanning the length of the track. As N increases the error introduced is no longer negligible and then it generates the mismatching shown in Figure 3.15b. Nonetheless, the error generate in the integration of the weight is about 0.71% when $N = 20$, and about 1.05% when $N = 25$. Moreover, high values of N are unlikely to be obtained in practice, as the roller would require to either overlap each other or to be infinitesimally small[1]. In addition, notice that already with small values of N the resultant μ_t is a good approximation of the ideal (uniform) one, as shown in Figure 3.14b.

The simulations presented in this section numerically validate the proposed normal pressure distribution function. Since it tends to the uniform one as the number of rollers are increased. Nevertheless, experimental tests are currently being performed, in order to fit real pressure data with the model proposed.

[1] In this case the error introduced is even smaller, given that smaller radii reflect in smaller r_w (variance of the GNPD).

3.6 Experimental model validation

This section aims at validating the dynamic model presented before, comparing real forces read from motor encoders with their estimates. To this purpose, we performed different tests while moving along circular trajectories, achieved by sending constant reference speed to the motors.
We recorded raw data from the GNSS receiver, in order to record the 3D position of the vehicle, and raw data received by the motor telemetry (motor torques). As all the equations described before refer to track forces, the quantity measured are then expressed in terms of track linear forces. To obtain linear forces from torques, we need to define two quantities:

1. $r_{sprocket}$ - the radius of the driving wheel of the track, which is the sprocket directly connected to the outlet of the gearbox, which is $r_{sprocket} = 0.18m$
2. k_{gb} - is the reduction coefficient generated by the planetary gearbox connected to the motor shaft. In our case $k_{gb} = 32$.

From this, the track force F can be computed starting from the motor torque τ measured by the motor controller, as follows

$$F = k_{gb} \left(\frac{\tau}{r_{sprocket}} \right). \tag{3.29}$$

As shown in Figure 3.16, each test was repeated twice, clockwise (CW) and counterclockwise (CCW), with the intent of reducing the effect of asymmetries[1] in the experimental platform. We performed tests with three different turning radii: $R_1'' = 2m$, $R_2'' = 4m$ and $R_3'' = 7m$. Notice that, the ones claimed here are the ideal turning radii, computed starting from the kinematics of a differential wheeled robot [73], which does not take into account track skidding.
To implement the dynamic model, the following soil parameters were considered: $f_r = 0.3$, $\mu = 0.35$, $c = 3.45kPa$ and $K = 0.05m$. Accordingly to [1], these data represent *Heavy Clay Terrain*, which is the one that best describes the soil of our test field.
The left-hand side of Figure 3.16 shows planar trajectories performed during tests with the three nominal radii, while the right-hand side of Figure 3.16 compares the motor forces recorded from the telemetry with our estimates. The latter comparisons

[1] It is unlikely to consider a perfectly symmetric real robot. Therefore, it can happen that the resistive forces, both due to ground interaction and to internal frictions, are different from one side to the other.

show that the model estimates in good approximation real motor forces. More details about RMS estimation errors can be found in Table 3.1. It reports an overall good approximation of the real data, with bigger deviation from real values (in percentage) when the forces have small absolute values, as it happens for small turning radii. As described at the beginning of this chapter and in Appendix A, small turning radii lead the inner track to produce forces close to $0N$, where the relevance of measurement noises increases, reducing the *goodness* of the estimation. Anyway, this is quite normal as the Signal-to-Noise Ratio (SNR) is quite small, in particular in the case of F_i while negotiating curves with $R'' = 2m$, the SNR is slightly bigger than 0 dB, meaning that the signal magnitude is comparable with the amplitude of the noise oscillations. Therefore, strongly affecting the quality of the estimation in terms of RMSE. We kept these results anyway, to emphasize that to evaluate the model we used raw data without any kind of filtering.

R''		RMSE [N]	RMSE %	SNR [dB]	RMSE_{av} [N]	RMSE_{av} %
2 m	F_o	114.7	5.2 %	24.07	28.31	1.3 %
	F_i	91.9	80.7 %	0.91	21.15	18.7 %
4 m	F_o	98.2	5.4 %	24.03	40.04	2.2 %
	F_i	88.2	17.2 %	14.62	70.13	13.5 %
7 m	F_o	88.9	6 %	23.02	65.27	4.4 %
	F_i	83.6	10 %	20.49	70.8	8.6 %

Table 3.1: Root Mean Square Error between real and estimated track forces.

The last two columns of Table 3.1 highlight that, as far as mean values of the measurements are considered, the accuracy of the estimate is good even in those cases featuring small SNRs. This shows that the model developed, along with the proposed normal pressure distribution function, properly describes the dynamic behavior of this kind of vehicles, providing an excellent tool to exploit during the design procedure.

3.7 Conclusions and Future developments

In this chapter we defined a new function modeling the normal pressure distribution underneath tracks. The gaussian-based model proposed allows one to improve the state of the art, taking into consideration the radius of the rollers within the track, bet-

ter approximating the ideal (uniform) pressure distribution as the number of rollers are increased.
Then, we leveraged the new pressure distribution function, together with the derived dynamic model of skid-steering vehicles, to describe the dependency of track forces on both the negotiated trajectory and the different design parameters. Thanks to the latter dependency, it was possible to obtain a model-based design tool driving the sizing of tracked vehicles, therefore minimizing the power consumption.
In the end, we performed experimental tests in order to gather data about track forces required to negotiate constant-radius circular trajectories. We used these data to validate the models derived, resulting in good estimation performances.

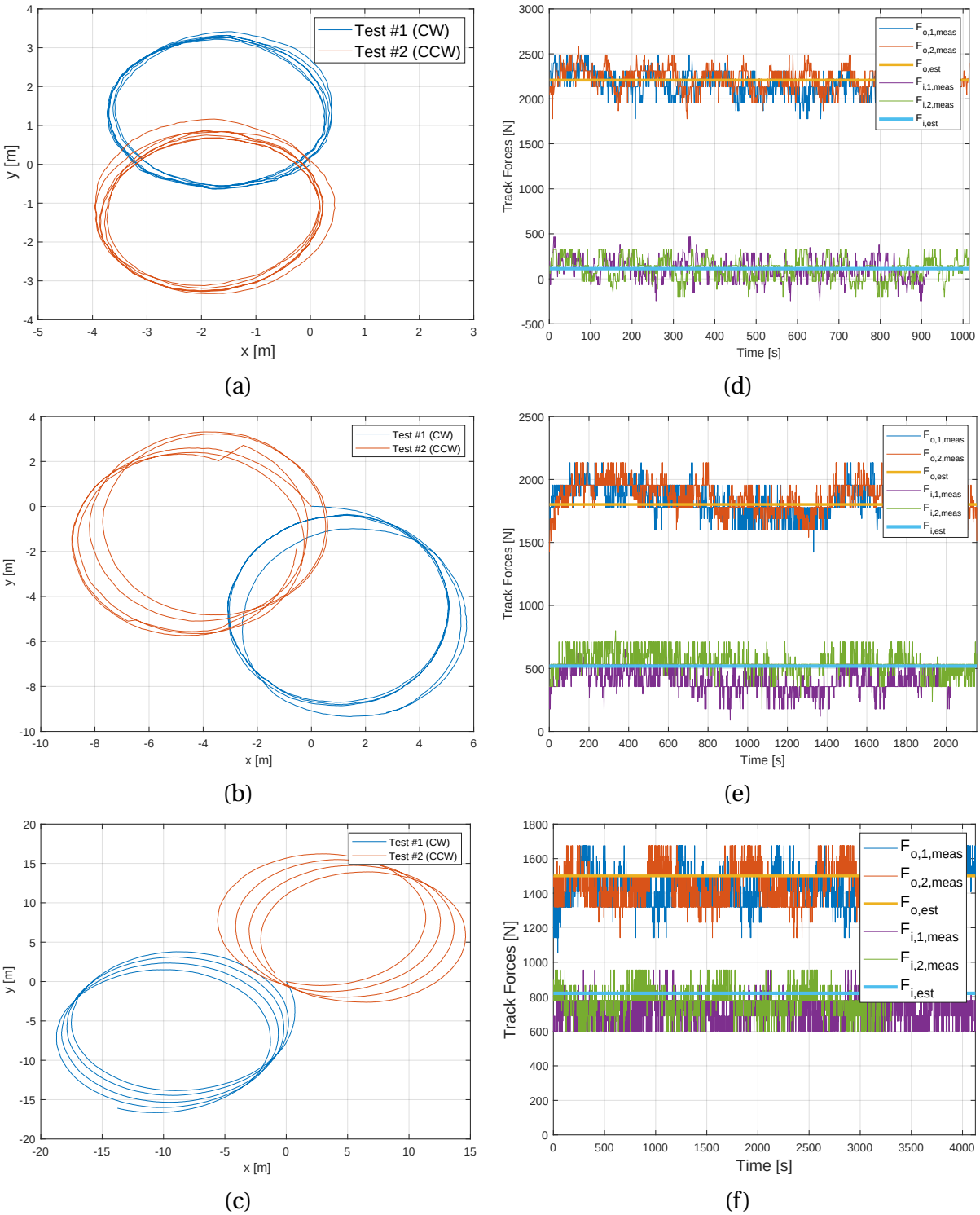


Figure 3.16: Comparison between real data collected during turning motion tests and estimates obtained by the proposed model.

4

Autonomous Navigation and Control

In this chapter, a detailed description of the autonomous localization and navigation algorithms implemented is given. Then, a position control for skid-steering vehicles, based on a novel adaptive observer for slip estimation is shown.

4.1 Introduction

This chapter is composed by two main parts, the first one recalls our works [52, 48, 74], and it describes how localization and navigation inside and outside orchards have been implemented. While the second part, is an extension of our work [65] showing the proposed design of an adaptive observer for slip estimation in Skid Steering Vehicles (SSVs). This is followed by the design of a position controller with slip compensation, based on the estimates computed by the adaptive observer.

As mentioned in [48, 52, 74], navigation in farming environment is mainly divided in two scenarios, independently on whether orchard, greenhouses or open-field tasks are concerned:

- **Open-field** - in which the robot has to move from its depot-station toward the field into which the mission has to be carried out, or viceversa (i.e. from the field to the base station), once the mission is completed or suspended. In this

scenario, the autonomous vehicle has to travel through the farm along field alleyways, avoiding drain trenches and any kind of possible obstacles.

- **In-row** - during this phase, the UGV has to travel along a straight line trajectories, either following plant or tree rows, as smoothly as possible, keeping at the same time a fixed longitudinal velocity. Once at the end of the lane, it has to switch to the following one and then back again along the working straight trajectory.

In this chapter, we show how we implemented open-field navigation and how we improved state-of-the-art autonomous navigation in *in-row* scenario. In particular, the use of sensor fusion algorithm is described, with special focus on the in-row scenario, where line detection algorithms are considered as a sensor within the context of data-fusion. Along with this, we also show how we dealt with trajectory planning, path following and localization.

4.2 Sensor Fusion and Localization

In Chapter 2, we have listed the sensors the UGV is equipped with, and most of them have the purpose of endowing the robot with localization and autonomous navigation capabilities. It comes straightforwardly that being able to define the local and global pose of a robot is the first and most strict requirement, this is why, in order to achieve autonomous navigation, we tackled this problem as first.
Localization is based on an Extended Kalman Filter (EKF), [75, 76], used to provide accurate pose estimation by merging all the available sensors: motor encoders, GPS, IMU readings and LiDAR pointclouds. Nonetheless, as described also in [77, 16, 78, 48] it is not advisable to completely rely on GPS positioning within orchard and greenhouses, since tree canopy, as well as metallic frame/poles and wires can obstruct and affect reliability of the GPS signals, therefore, relative localization with respect to trees is an effective and GPS-free method to achieve *in-row* navigation. On the other hand, in open-field scenario, GPS is almost unavoidable, and LiDAR can result of secondary importance as it is not ensured the presence of obstacles with respect to which define the robot relative pose. From these considerations the idea of an EKF that "changes" the set of available sensors accordingly to the navigation scenario arose:

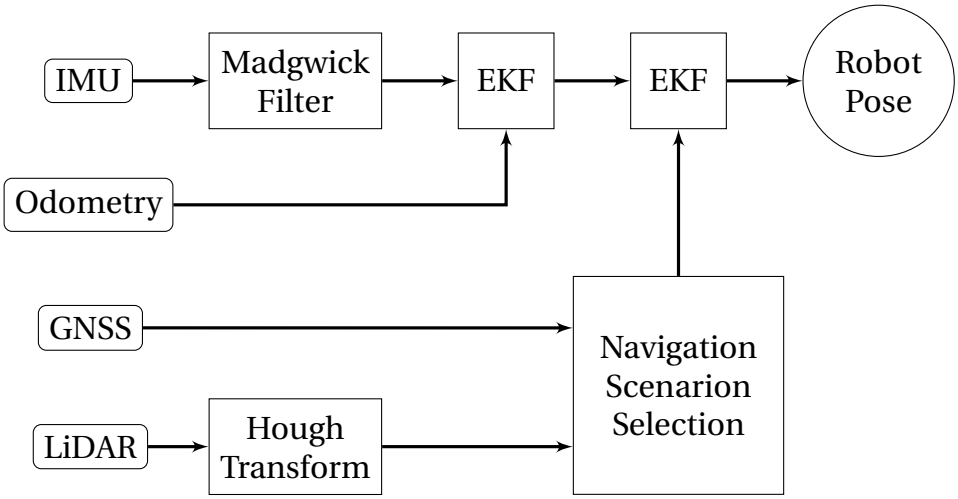


Figure 4.1: Robot pose estimation scheme.

- **Open-field navigation** - used to exit from parking place and reach the orchard in a reliable way. This scenario requires the vehicle to strongly rely on GPS positioning, using the LiDAR for obstacle detection and avoidance.
- **Orchard navigation** - used to perform working activities inside the orchard. For this navigation context, the GPS is not used and the LiDAR measurement are processed to provide relative localization with respect to the tree rows, mainly exploiting the Hough Transform algorithm.

Figure 4.1 shows the logical scheme representing the localization algorithm of the robot. In particular, notice that the scheme presents two EKF blocks: the first one, which is invariant to the navigation scenario, merges together IMU data (filtered by a Madgwick Filter [79, 80]) and motor odometry, while the second one changes the set of data to fuse, based on the navigation scenario considered (open-field or in-row navigation). We also want to highlight the *Hough Transform* block, this embeds the line estimation algorithm used during navigation within greenhouse or orchard rows.
As mentioned in Chapter 2, navigation, localization and trajectory planning algorithms are coded and developed using the Robotic Operating System (ROS).

4.2.1 Open-field navigation

For open-field navigation, the robot has been setup to exploit the state-of-the-art ROS Navigation Stack [81]. The idea behind this navigation paradigm is to model the

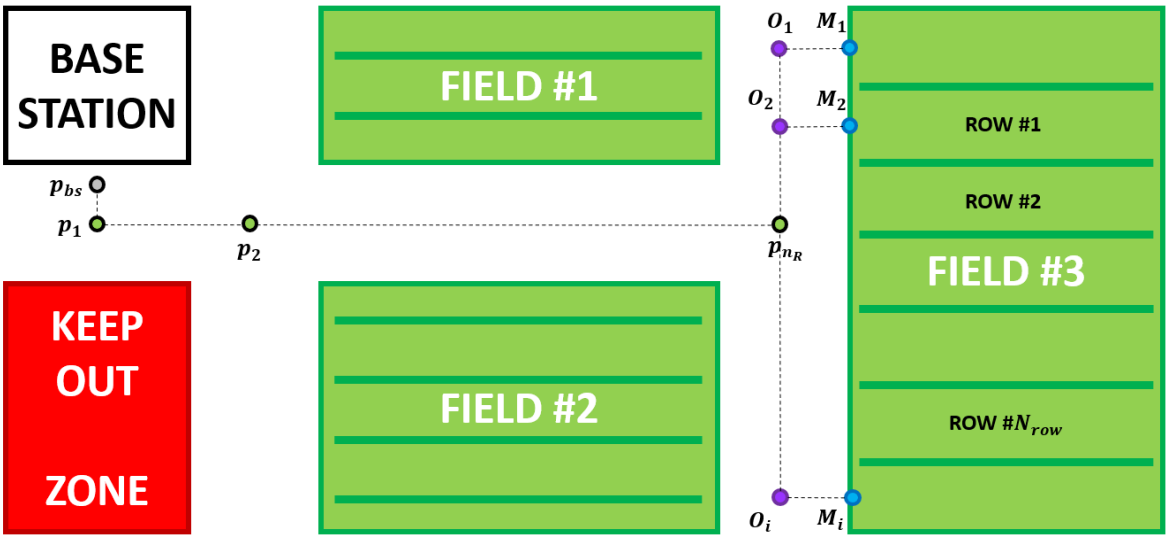
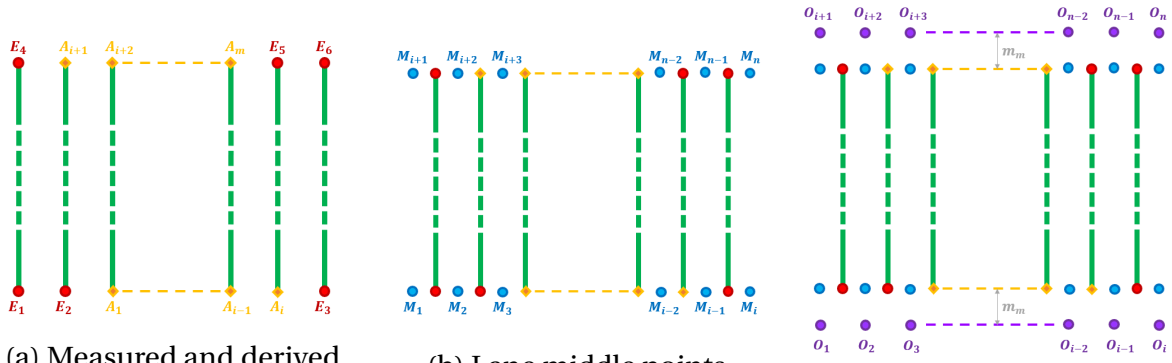


Figure 4.2: Waypoints used for openfield trajectory planning.

farm and its alleyways as a structured environment with a known map, either available a-priori or obtained via recognition activities. The map generated, can then be enriched using a multi-layer map approach, which allows us to overlap real obstacles detected by the 3D laser scanner and virtual obstacles, generating forbidden zones, used for instance to represent drain trenches which are hardly detectable by the LiDAR[1]. The localization filter described in [82] is used to fuse motor encoders and IMU readings via the first-stage EKF, estimate which is, in turn, merged with the GPS signal coming out from the Navigation Scenario Selector (NSS). The resultant pose estimate is then used in order to plan the most efficient trajectory toward the entry point of the orchard.

Figure 4.2 shows the set of waypoints used to define the trajectory to follow in order to reach the target work field. The first waypoint, fixed immediately outside the *base station*, features a predefined and fixed GPS position (p_{BS}), which ensures that satellite reception is not occluded by the roof of the depot or by the building itself. On the other hand, the last waypoint (the one closest to the target field) (p_f) is used to safely define the approaching maneuvers of the robot to the orchard/greenhouse row to work in. All the other waypoints required to define the proper "macro-path" of the robot from the base station to the field are selected among a set of predefined routes (either defined by the user or planned autonomously by the mission planner, as described in 2). We have decided to follow this procedure since it allows to define

[1]If drain trenches are not mowed regularly, they can visually result not different from tall grass.



(a) Measured and derived row end points.

(b) Lane middle points.

(c) Outer middle points.

Figure 4.3: Waypoints used to define each working field.

high-level trajectories along farm-alleyways, the latter is a crucial point since it allows to follow the most efficient way even at the first deployment in a new farm.
The last waypoint of the route mentioned above (p_f) corresponds to one of the purple points shown in 4.3c. These points have been derived starting from a field-mapping procedure which can be outlined as follows:

1. Measure the GPS position of the 6 row ends as shown in Figure 4.3a, namely $\mathbf{E_{wp}} \coloneqq \{E_1, E_2, ..., E_6\}$
2. Define the total number of tree rows in the field N_{row}.
3. The set of waypoints $\mathbf{A_{wp}} \coloneqq \{A_1, A_2, ..., A_m\}$ is automatically computed, where $m = 2(N_{row} - 3)$. In this way the systems estimates all the initial and terminal GPS positions of each single row in the field, as it can be seen in Figure 4.3a.
4. $\mathbf{M_{wp}} \coloneqq \{M_1, M_2, ..., M_{2n}\}$ is then computed, where $n = N_{row} + 1$. This set of point defines the entry points of each lane in the field. Considering as a "field lane" also each one of the outer lane at the sides of the field. Notice that each $M_i \in \mathbf{M_{wp}}$ lies on the line passing through the two closest point belonging to $\mathbf{M_{wp}} \cup \mathbf{A_{wp}} \cup \mathbf{E_{wp}}$, as also depicted in Figure 4.3b.
5. $\mathbf{O_{wp}} \coloneqq \{O_1, O_2, ..., O_{2n}\}$ is in the end computed. This set of point has the aim of bridging *open-feild* and *in-row* navigation, as they are meant to allow the robot to properly adjust its pose before entering the target lane $p_f \in \mathbf{M_{wp}}$. As shown in Figure 4.3c, the waypoints belonging to this set lie along two lines which are parallel to the two end-of-row lines distant from them by m_m a maneuverability margin defined taking into account the frontal implement attached to the vehicle, so to allow the robot to safely approach p_f.

Figure 4.4 shows how the procedure outlined above works also for non-rectangular fields, and also in presence of non-perfect GPS measurements (as it is in real applications).

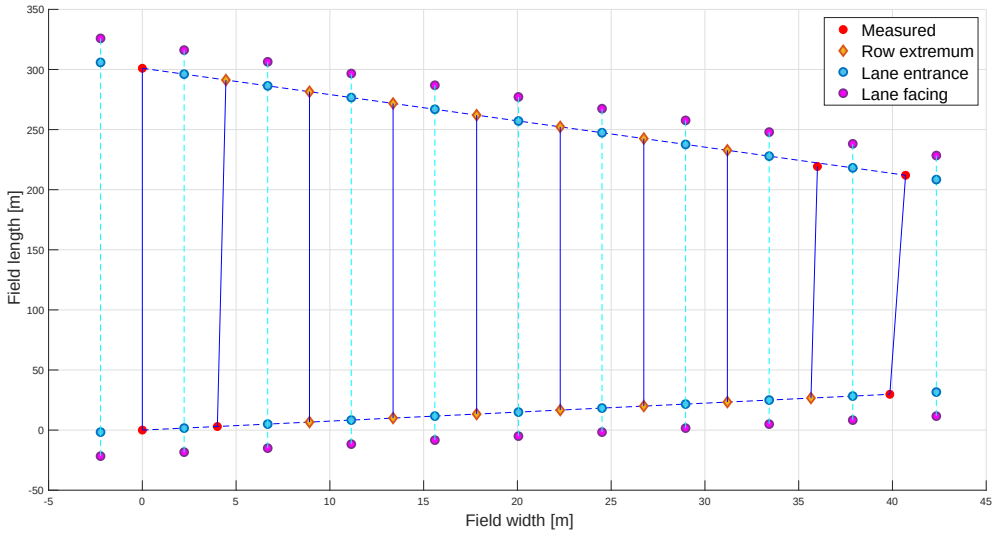


Figure 4.4: Field waypoints definition in case of measurements errors.

It follows from what has been said above that the route from the base station to the i-th row of the target field is generated by stacking sequences of waypoints. For instance, consider to take as target field target field $\#3$ in Figure 4.2, and to aim at entering the row $\#1$ from the left hand side ($p_f = M_2 \in \mathbf{M_{wp}}$). Then, the route $\{R_{bs,f3,r1}\}$ can be expressed as

$$\{R_{bs,f3,r1}\} \coloneqq \{p_{bs}, p_1, p_2, ..., p_{n_R}, O_2, M_2\} \tag{4.1}$$

where $p_1, p_2, ..., p_{n_R}$ are waypoints defining the most efficient route toward field $\#3$. In addition, it is important to notice that p_{n_R} lies on the line interpolating $\{O_1, O_2, ..., O_i\}$. This procedure is flexible in terms of which row the robot needs to enter as only the last two waypoints are directly associated to the desired row. In addition, if the UGV requires to suspend the mission while traversing the j-th row (for instance if the water tank is empty), in order to compute the route toward the base station starting from the corresponding M_j point, it is enough to remove from $\{R_{bs,f3,r1}\}$ the last two waypoints and append the proper outer-line waypoints from $\mathbf{O_{wp}}$ and $\mathbf{M_{wp}}$, then follow

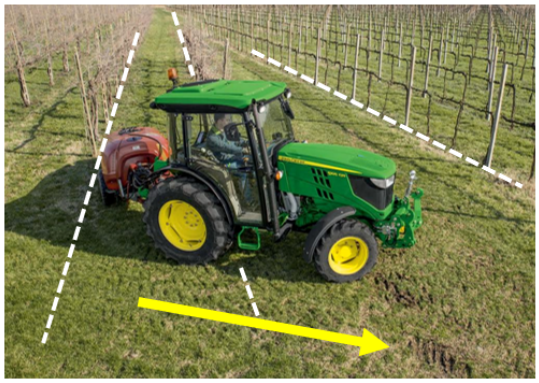
the route defined from right to left

$$\{R_{rj,f3,bs}\} \coloneqq \{M_j, O_j, p_{n_R}, p_{n_R-1}, ...p_2, p_1, p_{bs}\}\,. \tag{4.2}$$

Once the mission planner has defined the sequence of waypoint leading to the desired orchard/greenhouse row, the online planner leverages standard graph optimization algorithms, such as A* [83], [84] to interpolate them and then reach the target position.
In general, during openfield navigation, the LiDAR sensor is used to enable obstacle avoidance capabilities, this information are then used both to update the waypoint interpolation trajectory as well as to refine the local map. Then, ss soon as the target goal waypoint (orchard) has been reached, the controller can switch to the orchard navigation algorithm.

4.3 In-row navigation

For what concerns the in-row navigation scenario, the localization of the vehicle follows the same structure as depicted in Figure 4.1 and already described. The main difference between this navigation scenario and the previous one is that the in-row localization does not fuse GPS measurements, rather, as shown in Figure 4.1 it considers as inputs the processed LiDAR measurements coming from the *Hough Transform* block. Anyway, the GPS is available onboard, it can be used for other purposes, e.g. triggering end-of-row detection algorithm as described in [48].
Both orchards and greenhouses are typically semi-structured environments, organized in several parallel rows of trees/plants aligned one to each other and with known dimensions. This is an information which makes no sense to neglect during the development of navigation and localization algorithms for autonomous vehicles designed to work within such environments. In particular, differently from open-field scenario (either for motion from one point of the farm to another or for performing work in fields such as plowing or seeding), relative position is more relevant than absolute. Namely, moving along an orchard lane and then switching to the following one can be effectively done GPS-free, as shown in [77, 16, 78, 48], furthermore, global positioning sensors do not provide any information about obstacles of any sort. In line with this, we decided to refine the pose estimation of the vehicle (i.e. sec-



(a) Orchard rows modeled as lines.

(b) Navigation along lane center line.

Figure 4.5: Orchard rows structure and navigation paradigm.

ond stage EKF in Figure 4.1) in this navigation scenario using the LiDAR information rather than GPS.

In-row scenario can be divided into two sub-scenarios:

- *Along-the-row* - This is the main part of the in-row scenario and it describe the navigation along orchard/greenhouse rows. Keeping a fixed lateral distance from the plant rows, and moving along the lane at a constant speed.
- *Row-change* - This navigation phase requires the robot to be able to detect that the current row is finished and then to plan the trajectory toward the following one.

Both along-the-row and row-change navigation rely on the relative position measured processing data read by the LiDAR. As shown in Figure 4.5, orchard rows, as well as greenhouse rows, can be modeled as straight lines, therefore LiDAR measurements are used in order to estimate the lines which best model the tree rows surrounding the vehicle.
In the following we provide a description of the line detection techniques implemented, then it is explained how to merge these within a sensor fusion context.

4.3.1 Hough Transform line detection

As most of the real field test were carried out within orchards, in the following we will mainly refer to orchard rows, but, as the structure of greenhouses are really similar to orchards', all the following considerations can be extended also to greenhouses.

In line with what has been done in [77], we implemented the line detection algorithm named Hough Transform (HT) [85]. The HT is a feature extraction technique that is widely used in image analysis, computer vision and image processing, as it features a good robustness with respect to noise in the images and presence of outliers [86]. This techniques was firstly applied as line detector, [85], but it also works with other known shape, such as circles and ellipses, [87], as long as their mathematical model is known. The HT was then extended also as shape detector for any arbitrary shape, assuming the name of Generalized Hough Transform, [88]. The fundamental steps of this algorithm are introduced here, as they allow the reader to better understand the proposed optimization. Notice that the algorithm described in the following was implemented in custom ROS nodes within the High Level PC.

As the HT is meant to work with images, 3D LiDAR pointclouds are processed and filtered in order to down-sample them, so to reduce the computational burden of the algorithm, and then projected in 2D pointclouds. The resultant 2D pointclouds can be considered as images and therefore the HT can be applied to them.
As the "standard" HT is conceived as a line detector, the output of the algorithm will be a pair of parameters identifying a line on a plane. For sake of clarity, in the first place, we will describe the working principle of the HT considering the Cartesian representation of a line:

$$y - mx - q = 0. \tag{4.3}$$

where (x, y) are the coordinates of each single point belonging to the 2D pointcloud and (m, q) are the parameters describing a line, namely, the angular coefficient and the intercept, respectively, thus the HT will return the pair $(\bar{m}, \bar{q})$ describing the line sought.
The HT converts each point (x, y) in the image space into lines in a parametric 2D space (the *Hough Space*)[1], as shown in Figure 4.6.
If two points in the image plane lies on the same line, the corresponding lines in the Hough Space will intersect in a point, and this goes on if more points lies on the same line.
The parametric space is then discretized (quantized), as reported in Figure 4.7, generating a grid (or a matrix), after this, for each cell the number of intersections are counted, as shown in Figure 4.8. It follows that the highest peak in the discretized

[1] Points in the image space are converted into lines in the Hough Space if the cartesian representation is considered, while, if polar representation is considered, points are converted into sinusoids.

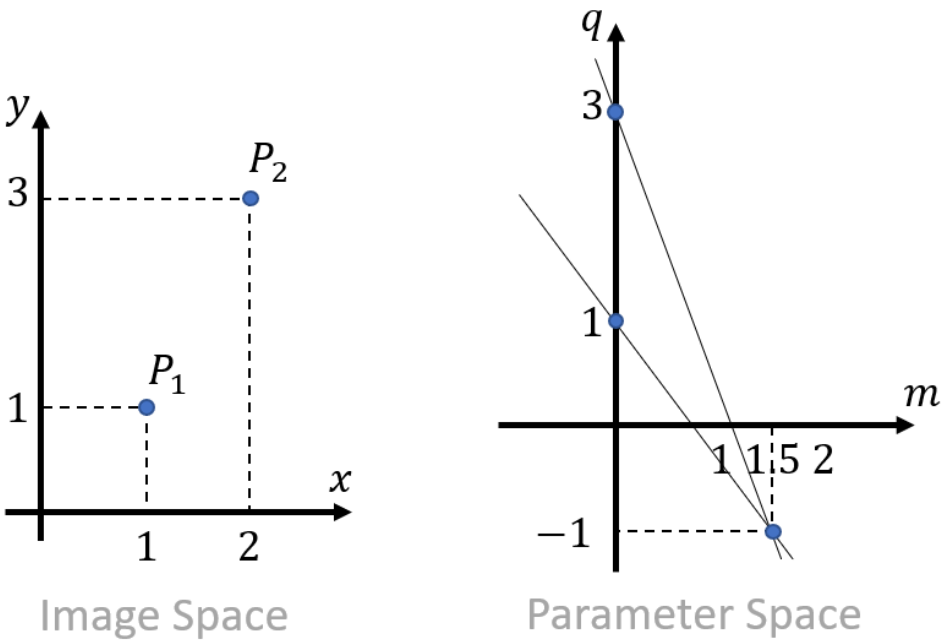


Figure 4.6: Two points in the image space and their associated lines in the parametric plane (Hough Space).

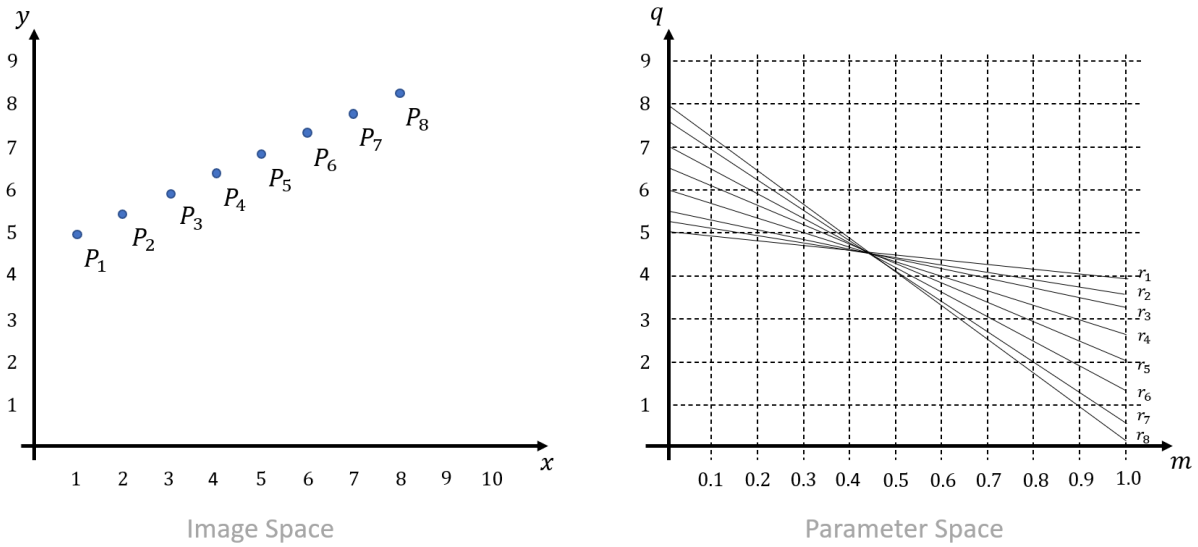


Figure 4.7: Image points belonging to the same line generate intersecting lines in the Hough Space, where the coordinates of the intersection correspond to the line parameter in the image space.

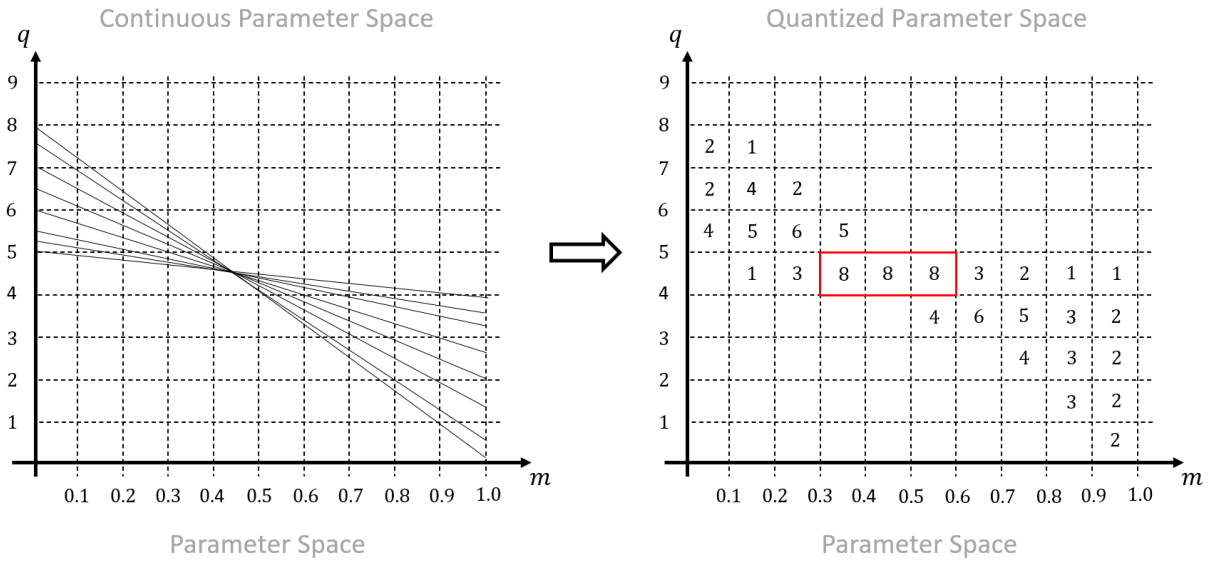
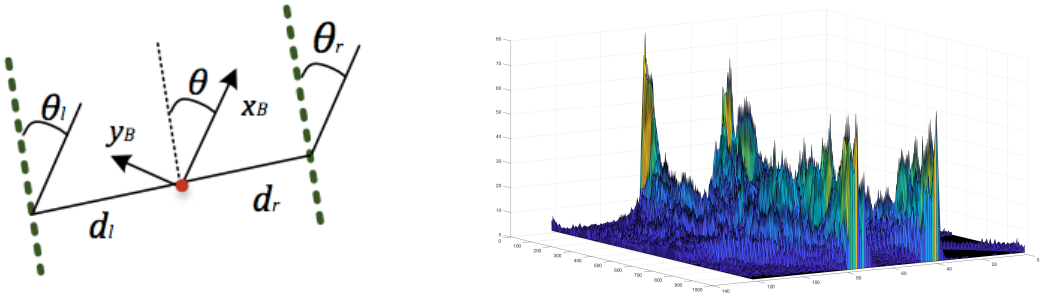


Figure 4.8: Discretization (quantization) of the Hough Space reduces the problem of finding a solution in a continuous 2-dimensional space to a maximum search on a discretized 2D array.

Hough Space (HS) is associated with the line interpolating the largest number of image points. Therefore, the line detection problem is reduced to a 2D maximum search, [89].
Similar reasoning can be applied to the case in which the polar representation of a line is considered

$$d = x\cos\theta + y\sin\theta. \tag{4.4}$$

Where d, θ are the polar parameters of the line on the plane, thus the HT algorithm returns as output $(\bar{d}, \bar{\theta})$.
If the HT is run within the orchard lane, the algorithm should be modified in order to look for two lines rather than only one. Notice that, the LiDAR is mounted on the vehicle in such a way that the heading of the sensor reference frame and the UGV's are aligned. Allowing us to assume that the output of the algorithm $(\bar{d}_r, \bar{\theta}_r), (\bar{d}_l, \bar{\theta}_l)$, describing the right and left rows, respectively, corresponds to the relative orientation of the vehicle with respect to the orchard lane, as well as the distances from the two sides $\bar{d}_r, \bar{d}_l$, as shown in Figure C.5a. Figure C.5b shows the hough space matrix generated by a 2D pointclouds taken within an orchard lane, highlighting the presence of the two distinct peaks.
As the algorithm looks for two different lines modeling the two orchard rows, it is possible to assume that these lines must be parallel to each other, and apart from one



(a) HT lines estimates.

(b) Navigation along lane center line.

Figure 4.9: Orchard rows structure and navigation paradigm.

another by about the lane width. This redundant conditions can be used in order to enhance the efficiency of the 2D search as well as increase the level of robustness of the algorithm. In fact, one of the contributions of our work [48], was to show that it is possible to optimize the computational complexity of the HT algorithm for line detection, and at the same time further improve its reliability and robustness.

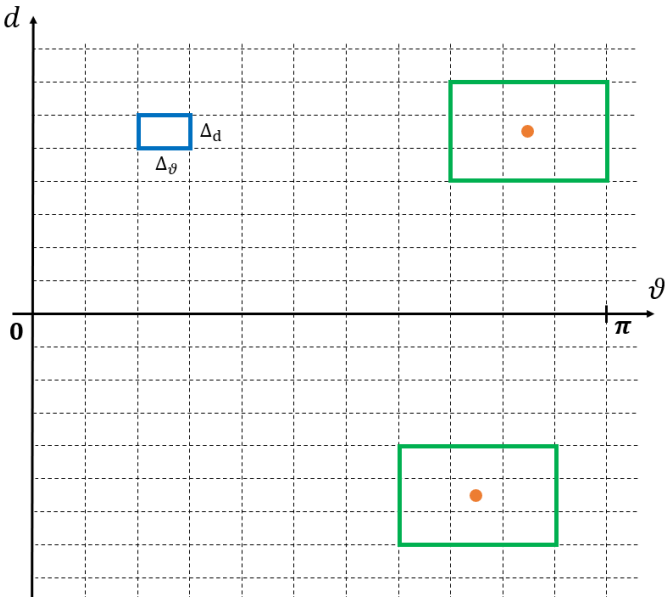


Figure 4.10: Search area (green) around last solution found (orange points) for the improved algorithm.

Since the lane structure is preserved along the whole row length, after the first iterations on the whole Hough space, the search grid can be reduced, asking the algorithm to find lines that are “close” to the previous solutions (i.e. the lines do not change so much when the rover is moving along the row), as represented in Figure 4.10. There-

fore, the solutions for the new pointcloud frame is not obtained by scanning the entire matrix, but only the cells that are adjacent to the previous solutions.
This improvement is also meant to increase the robustness to misdetections caused by missing trees along the row, since in this case one of the two lines would be strongly different from the previous ones, but the reduced search span automatically filters out those outliers votes. Therefore, it becomes important to properly tune the parameters of the neighborhood sizes, in order to not lose sensitivity to variations in the real angles, and at the same time to not widen the search space too much, reducing the advantage in terms of computational load.

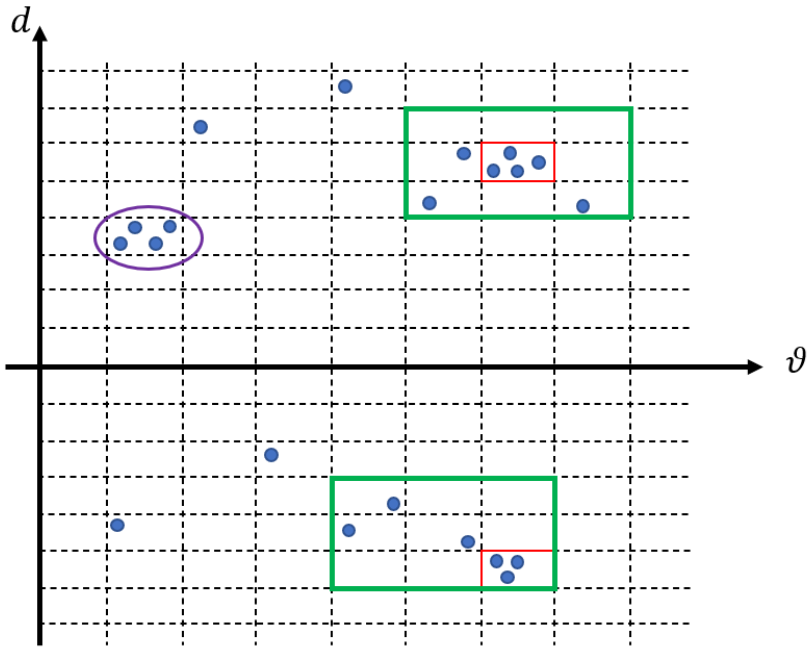


Figure 4.11: Outliers are automatically neglected, as the search space is bounded within a neighborhood of the previous solution.

From what has been said above, it is possible to consider the estimate obtained by the two-parallel-line HT as a sensor returning two distances d_r, d_l and one angle θ_H which is considered to be the average between the two values θ_r, θ_l, remarking even more that the two lines have to be parallel each other.

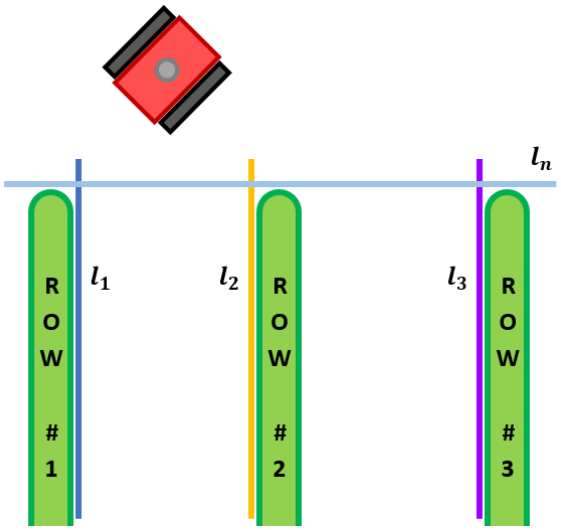


Figure 4.12: Lines detected during row-changing maneuvers.

4.3.2 Row-change

The same line detection approach can be also used to estimate the relative position of the robot with respect to the end of the row during row-changing maneuvers.
Once the robot exited the lane it was working in, it will be able to detect the four lines depicted in Figure 4.12: l_1, l_2 modeling the rows of trees in the orchard lane just completed, l_3 generated by the new tree rows the robot is moving toward, plus l_n which is the virtual line passing through all the end-of-row points. In terms of HT, the main difference between this scenario and the previous one, is that in this case the algorithm has to look for three parallel lines and one orthogonal to all the others. Therefore, the algorithm is initialized using the last output of the 2-line HT (HT-2), and the search space is polarized toward those angles respecting parallelism (green area in Figure 4.13a) and orthogonality constraints (blue area in Figure 4.13a), namely with an angle $\theta_n = \theta_H - \pi/2$.
Figure 4.13b, then, shows the four solutions retrieved by the 4-line HT (HT-4) algorithm. As also shown by Figure 4.14, the heading of the vehicle, is equivalent to the angle with respect to the l_n line (θ_n), it then follows, that the considered output of the line recognition algorithm in this phase is considered to be just d_n, θ_n.

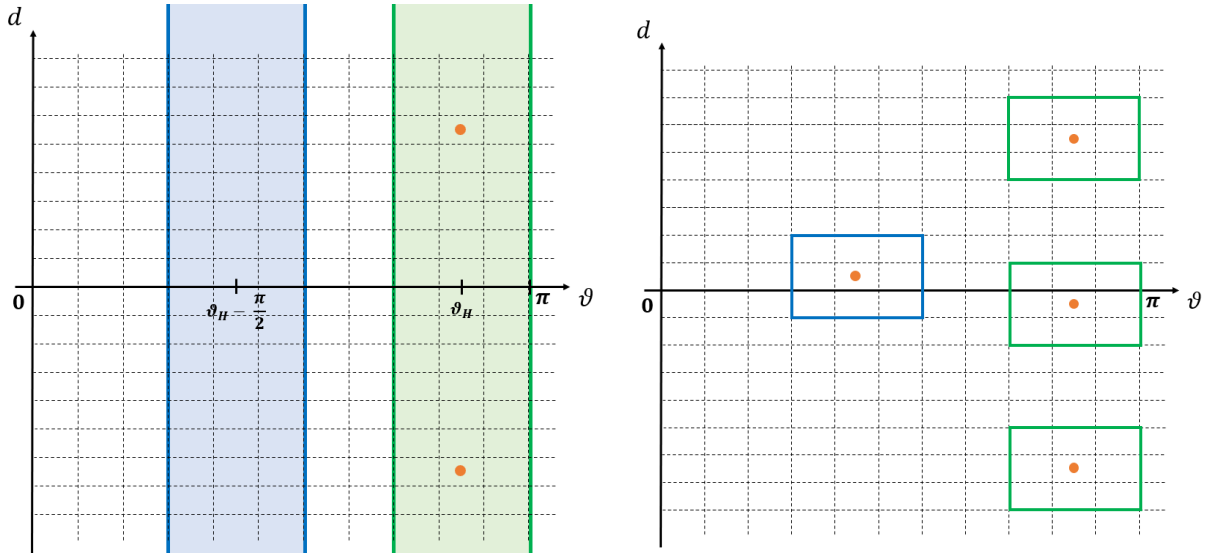


Figure 4.13: Hough space search zones considering parallelism and orthogonality constraints.

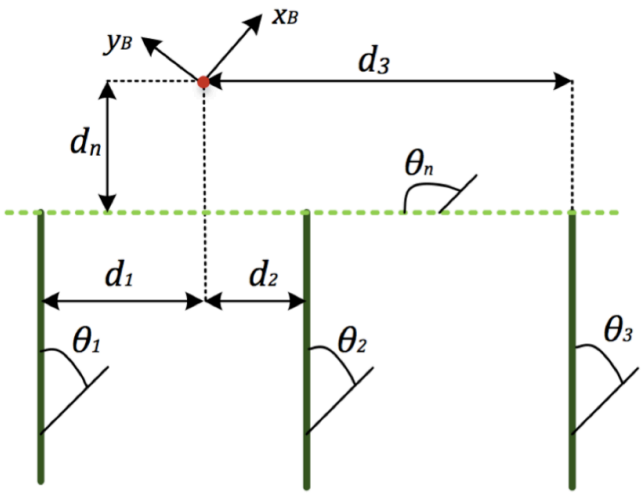
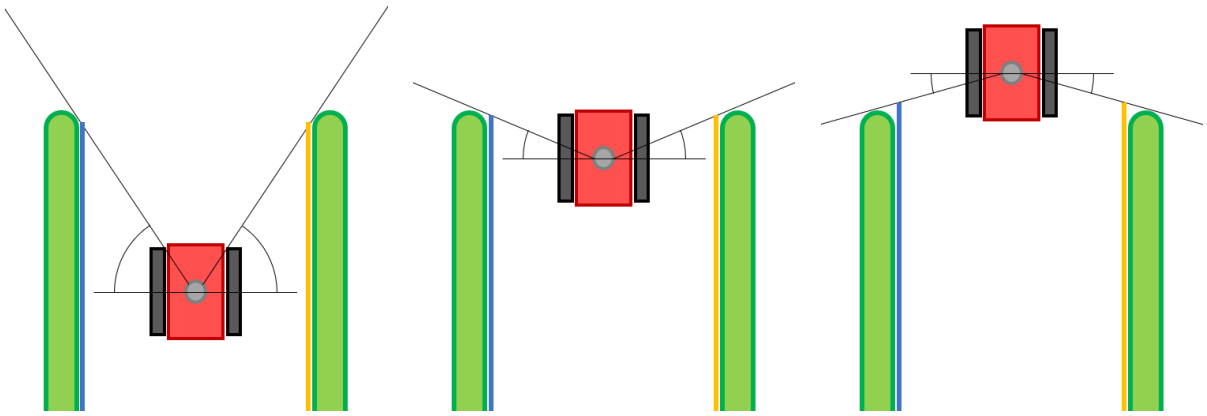


Figure 4.14: Row-change lines angles and distances.

4.3.3 Marker-free end-of-the-row detection

So far, we have described how to process 3D pointclouds in order to obtain a reliable measure of the relative pose of the robot with respect to the surrounding environment during in-row navigation scenario. Now, we have to define the rationale behind the switching procedure, from HT-2 to HT-4, and again HT-2 once the turning



(a) (b) (c)

Figure 4.15: End-of-row detection based on later points angle.

maneuver has been completed. The robot has, somehow, to detect that the lane it was traveling is finishing and then switch from the HT-2 to the HT-4.
The main idea is to process once again the 2D pointclouds and check, as also reported in Figure 4.15, whether the robot is still "seeing" the rows in front of itself or the ones sensed lie only behind it. If the latter is the case, then the robot is exiting the orchard lane, and then the HT-4 can be initialized and started, .
It can anyway happen that, due to missing trees on both sides of the robot, that the end-of-row detection is triggered even though the UGV is still far from the end of the lane. To avoid this, Bergerman's group [16] placed optical markers at the end of each row. Since visual markers are inherently prone to deterioration in outdoor applications, and the farmers themselves spray products along the whole lane possibly reducing their visibility, in [48, 74] we propose a solution not relying on any external visual aid. As mentioned in Section 4.2.1, and shown in Figure 4.3 and Figure 4.4, for open-field navigation, the shape of the different orchard fields is available. Therefore, it is possible to use the computed set of waypoints (in particular those belonging to M_{wp}) in order to retrieve the length of each lane, then use the position estimate provided by the EKF, to trigger the end-of-row recognition, only when the traveled distance along the lane is close to the total length. This solution can also be applied in GPS-free scenarios thanks to the modular structure of the proposed EKF algorithm, depicted also in Figure 4.1.

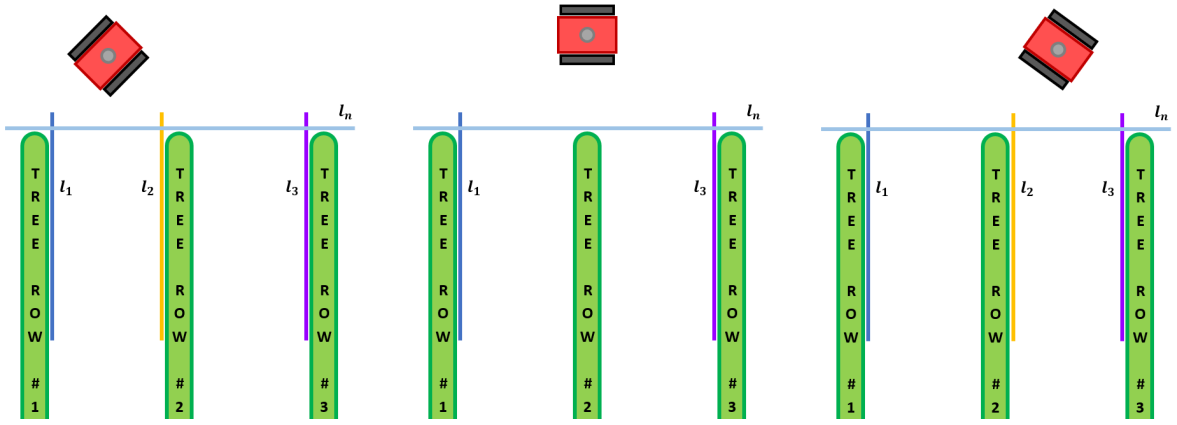


Figure 4.16: Line detection during row-changing maneuver.

4.3.4 In-row trajectory planning

As far as trajectory planning during in-row navigation is concerned, the two possible sub-scenarios need to satisfy the requirements imposed by the farming tasks that have to be performed inside the orchard. In particular, notice that orchard missions feature a repetitive pattern in which one has to travel along the field lane, then switch to the following one, and so on. Therefore, two kind of trajectories can be defined:

- *Straight motion* - navigating the current orchard row at a constant speed;
- *Turning motion* - switching from the current row to the next one at constant speed and turning radius.

While for straight motion constant speed is required for properly performing tasks (e.g uniformly spraying pesticides), for turning motion the most efficient trajectory for SSVs is given by circular motion with constant angular speed and radius. In [16], it is shown that different techniques can be adopted for lane-switching manoeuvres when car-like vehicles are considered, since turning radius is bounded by their steering capabilities. Differently, SSVs allow one to perform turning manoeuvres with small turning radius (even zero).
It is interesting to point out how the turning radius of the circular trajectories is computed. HT-4 allows us to obtain an estimate of the lane the robot is exiting and the one it has to enter, in particular the three parallel lines l_1, l_2, l_3 can be exploited in order to compute intersection points with l_n, where the most interesting point seems to

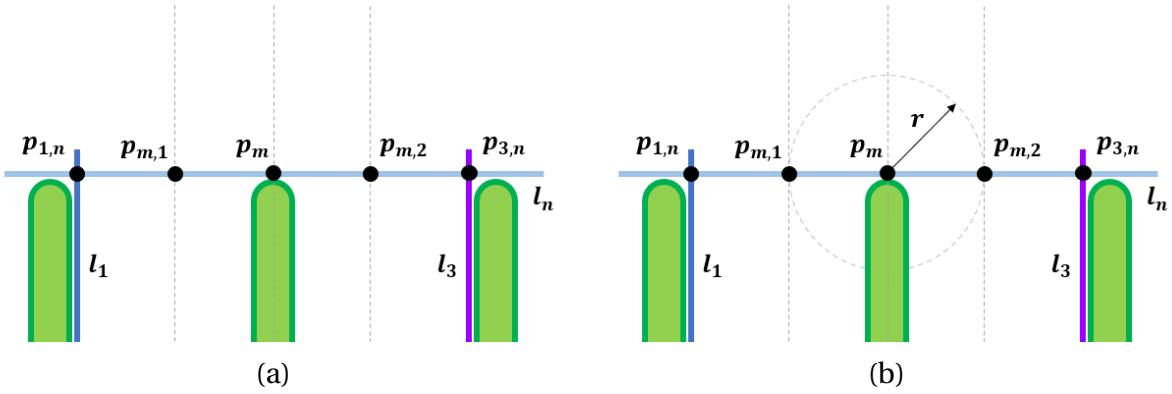


Figure 4.17: Circular trajectory planning: definition of the center and radius of rotation.

be the intersection between l_2 and l_n. The latter point is a good candidate for a *pivot* point about which the robot can rotate. The main drawback of this choice is shown in Figure 4.16:

- In Figure 4.16a the robot detects l_1, l_2, l_3, respectively on right side of row 1, on the left side of row 2 and on the left side of row 3.
- In Figure 4.16b the robot detects l_1 on right side of row 1 and l_3 on the left side of row 3. The line l_2 is not detected, as around that position the LiDAR only sees the "head" of the row 2, therefore it cannot identify a line from that set of points.
- In Figure 4.16c the robot detects l_1, l_2, l_3, respectively on right side of row 1, on the right side of row 2 and on the left side of row 3.

From this, it follows that the estimate of l_2 can be detected on either the two sides of row 2, or even not detected at all, accordingly to the position of the robot (and in turn of the LiDAR) along the circular trajectory. On the other hand, the estimates of l_1, l_3 are steady enough to be used for this purpose, i.e. we can compute $p_{1,n}$, the intersection point between l_1, l_n, and $p_{3,n}$, the intersection point between l_3, l_n. As both of them, by definition, lie on l_n, it is straightforward to compute the distance between them and define p_m, the middle point between them along l_n, as shown in 4.17. From there, it is possible to define the points $p_{m,1}, p_{m,2}$, which are defined as the midpoints, respectively, of the segments $\overline{p_{1,n}p_m}$ and $\overline{p_m p_{3,n}}$. Starting from this, it is possible to define the desired turning radius $r = d(p_{m,1}, p_m) = d(p_m, p_{m,2})$.

4.3.5 HT-2 and HT-4 as sensors for the EKF

We already mentioned that the "Hough Transform" block in Figure 4.1 provides information to be fused together with the other sensors within the EKF for pose estimation. So, to be able to compare the HT-2 and the HT-4 data with the other sensors, we need to express their outputs with respect to some reference frame.
In Figure 4.18 it is possible to see $\mathcal{F}_F(O_F)$, which is a global reference frame, whose origin corresponds to $E_1 \in \mathbf{E_{wp}}$, thus its global position is known. In a similar way, $\mathcal{F}_{R_1}(O_{R_1})$ is a reference frame oriented along the orchard lane, in this way the outputs of HT-2, (d_l, d_r, θ_H), are expressed accordingly to the notation considered during the analysis done in Section 4.3.1 (see Figure C.5a). Notice that in this case O_{R_1} corresponds to $M_2 \in \mathbf{M_{wp}}$, this correspondence changes accordingly to the lane the robot is traveling through and the direction of motion. Similar reasoning can be applied for HT-4, if a reference frame, centered in the pivoting point is considered, the output of this algorithm (d_n, θ_n) is compliant with the notation in Section 4.3.2 (see Figure 4.14). Once again, the position of the origin of this reference frame is known as, in this case, $O_{R_{1,2}}$ is equivalent to $A_{i+1} \in \mathbf{A_{wp}}$.
It is also important to notice that all the transformation between the considered reference frames are constant and known as the the sets $\mathbf{E_{wp}}, \mathbf{M_{wp}}, \mathbf{A_{wp}}$ are always available.

4.4 Adaptive slip-observer

So far we have discussed about localization and trajectory planning, for both the navigation scenarios. Then, to complete the discussion about autonomous navigation capabilities of the robot, we need to describe how the motors of the UGV are controlled in order to properly track and follow the described trajectories in presence of slip.

For this purpose, we recall the kinematic model of a tracked robot described in Chapter 3. In particular, as far as planar motion is concerned, a SSV behave as a Differential Wheeled Robot (DWR) with the addition of slip. As already said in Chapter Chapter 3and in [65], this difference makes the problem of path following for tracked vehicles more complex, and it motivated several research activities aiming to design al-

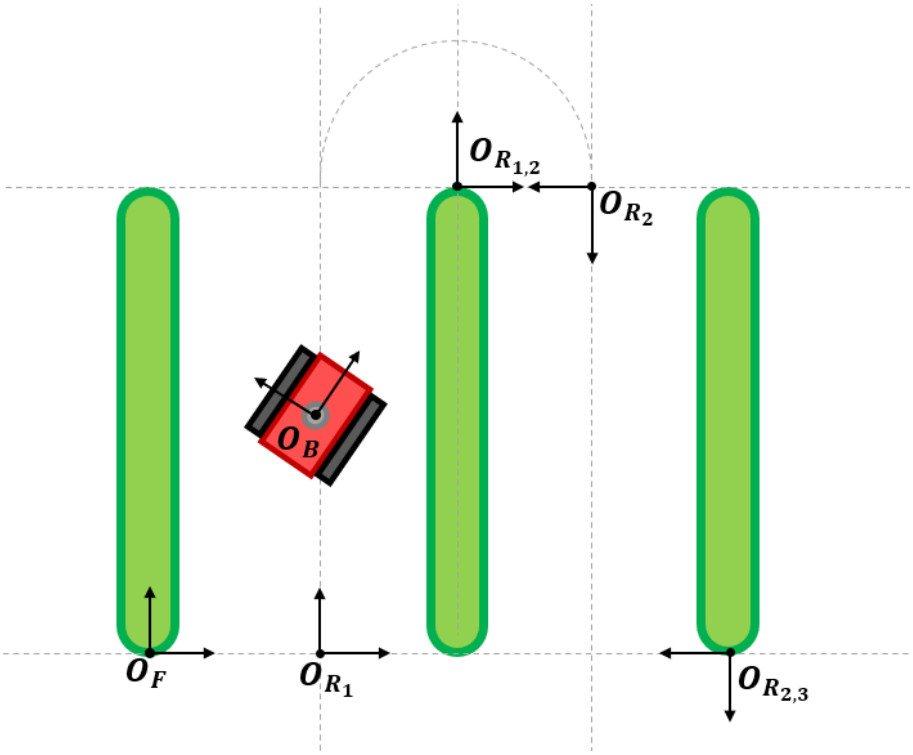


Figure 4.18: Reference frames considered during in-row navigation.

gorithms for estimating/compensating this effect. In particular, the kinematic slip has been modeled as a constant plus an additive zero-mean stochastic process and then estimated, using Kalman Filters in [90, 91, 92, 93] or adaptively in [94]. On the other hand, [62, 63] studied the relation between the slip and track speeds, remarking the dependence of the slip on track velocities, turning radius and soil parameters. However, these relations work only when the turning radius is kept small enough. Differently from all the other approaches, [95], [96] were driven by empirical observation and data fitting, leading to the definition of ad hoc slip models tailored for the particular platform-terrain combination considered.
Focusing on adaptive approaches, [94] models the slip as a brownian random walk, estimating it through recursive least squares and then using it to adapt the control action considering a DWR-based reference model, following a Model Reference Adaptive Control (MRAC) approach. While, [97] designs an adaptive feedback linearization control for the dynamic model of a wheeled SSV, differently from all the works mentioned before. We observe that, in line with classic adaptive control literature, both the [94] and [97] share the need of Persistency of Excitation (PE) conditions in order to have convergence of the slip estimates to the true slip coefficients, without,

anyway, discussing what this would mean in terms of mobile robot trajectories.

Notation Before starting the discussion it is useful to integrate the general notation used in this thesis, with some, more specific terms. In particular, throughout the rest of this chapter, we denote the $\mathcal{L}_p$ norm of a measurable function $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, as $\|\phi\|_p := (\int_{t_0}^{\infty} \|\phi(t)\|^p \, dt)^{1/p}$ for $p \in [1, \infty)$, while we denote as $\|\phi\|_\infty := \text{ess sup}_{t \geq t_0} \|\phi(t)\|$ the $\mathcal{L}_\infty$ norm of ϕ. For $c_r > 0$, B_{c_r} denotes the open ball $B_{c_r} := \{x \in \mathbb{R}^n : \|x\| < c_r\}$. A solution to the differential equation $\dot{x} = f(t, x)$ at time t with initial conditions $(t_0, x_0) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^n$ is denoted as $(x(t, t_0, x_0))$, or simply $x(t)$. For the function $V : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}$, we define $\dot{V}_{(\#)}(t, x) := (\partial V / \partial t) + (\partial V / \partial x) f(t, x)$ where $(\#)$ denotes the equation number that labels equation $\dot{x} = f(t, x)$.

DWR are very well known vehicles, and many works are present in literature on path following for this kind of robots. Our goal is to control the tracked vehicle using standard control techniques for DWR such as [98], and then correct the resulting control action accordingly to slip estimates. In fact, the difference between the kinematics of DWRs, (3.2), and SSVs ,(3.3), is given by the presence of $i_R(t), i_L(t) \in (-1, 1)$ that are the time-varying slip coefficients associated with the right and left track, respectively. Notice that in (3.3) the real inputs driving the system are $(1 - i_R)v_R$ and $(1 - i_L)v_L$, resulting in a variation of the efficiency of the control inputs v_L, v_R. By letting

$$\begin{cases} \eta_R & := 1 - i_R, \qquad 0 < \eta_R < 2 \\ \eta_L & := 1 - i_L, \qquad 0 < \eta_L < 2 \end{cases} \tag{4.5}$$

the overall kinematic model (3.1) is rewritten as

$$\begin{aligned} \dot{x} &= \frac{\eta_R \cos\theta}{2} v_R + \frac{\eta_L \cos\theta}{2} v_L \\ \dot{y} &= \frac{\eta_R \sin\theta}{2} v_R + \frac{\eta_L \sin\theta}{2} v_L \\ \dot{\theta} &= \frac{\eta_R}{2d} v_R - \frac{\eta_L}{2d} v_L \, , \end{aligned} \tag{4.6}$$

which is a system with control inputs v_R, v_L, and exogenous (uncontrollable) inputs η_R, η_L.

4.4.1 Observation problem statement

The following general problem is considered.

Problem 1 *Considering system* (4.6)*, the problem at hand is to design a dynamic system with inputs* x, y, θ, v_R, v_L*, producing as output the estimates* $\hat{\eta}_R$*,* $\hat{\eta}_L$ *asymptotically converging to the true coefficients* η_R *and* η_L*, respectively.*

This problem is hard to be solved as it would require a model relating the intrinsically time-varying slip coefficients to the system states and inputs, as well as the time-varying (and practically impossible to be modeled or measured) terrain parameters. Unfortunately, to the best of our knowledge, this model is not available.
Nevertheless, the application framework given by Orchard Precision Agriculture (OPA), in which most of the farming tasks require to navigate the orchard at constant speed, leads to simplifying assumptions which make the problem more tractable.
Recalling what has been said in Section 4.3.4, orchard works feature a repetitive pattern in which one has to travel along the field lane, then switch to the following one, and so on. Therefore, it is possible to define two different navigation scenarios:

- *Straight motion* - orchard row traveled at a constant speed;
- *Turning motion* - switching from one row to the following one at constant speed and turning radius.

While for straight motion constant speed is required for properly performing tasks (e.g uniformly spraying pesticides), for turning motion it is simply the most efficient trajectory for SSVs.
Notice that each scenario requires the robot to move at a constant V_G and Ω_Z, which are obtained, according to (3.2), from constant values of v_R, v_L. From (3.4), it is possible to claim that when v_i is constant then also i is constant. On the other hand, provided that constant track velocities are given by constant track forces, (3.5) requires the terrain to be homogeneous in order to obtain constant coefficients, which is a reasonable assumption in OPA. Then, this application framework allows one to rewrite the coefficients $i_R(t)$ and $i_L(t)$ as constants, namely $i_R(t) = i_R$ and $i_L(t) = i_L$.
We want to comment already at this point that this considerations are validated by

all the experimental results we performed on the field (as thoroughly described in Section 4.4.3). Formally, the problem is considered under the following assumption.

Assumption 1 *The linear velocities v_R, v_L in (4.6) are constant and positive. Moreover, the soil is homogeneous, therefore also i_R, i_L are constant.*

Notice that considering the problem at hand under Assumption 1 means that we are considering both OPA-related scenarios. In fact, during straight motion we have constant and positive $v_R = v_L$, while, during turning motion, we still have constant and positive $v_R \neq v_L$.

4.4.2 Observer design

Assuming that x, y, θ, v_R, v_L are available for measurement, the proposed estimator is the *adaptive identity observer*:

$$\begin{aligned}
\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{y}} \\ \dot{\hat{\theta}} \end{bmatrix} &= \begin{bmatrix} \frac{\cos\theta}{2} v_R & \frac{\cos\theta}{2} v_L \\ \frac{\sin\theta}{2} v_R & \frac{\sin\theta}{2} v_L \\ \frac{1}{2d} v_R & -\frac{1}{2d} v_L \end{bmatrix} \begin{bmatrix} \hat{\eta}_R \\ \hat{\eta}_L \end{bmatrix} + \underbrace{\begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix}}_{L} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} \\
\begin{bmatrix} \dot{\hat{\eta}}_R \\ \dot{\hat{\eta}}_L \end{bmatrix} &= \underbrace{\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}}_{\Lambda} \begin{bmatrix} \frac{\cos\theta}{2} v_R & \frac{\sin\theta}{2} v_R & \frac{1}{2d} v_R \\ \frac{\cos\theta}{2} v_L & \frac{\sin\theta}{2} v_L & -\frac{1}{2d} v_L \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}
\end{aligned} \tag{4.7}$$

where the state $\begin{bmatrix} \hat{x} & \hat{y} & \hat{\theta} & \hat{\eta}_R & \hat{\eta}_L \end{bmatrix}^\top \in \mathbb{R}^5$ contains the estimates of $x, y, \theta, \eta_R, \eta_L$ in (4.6), the output-injection matrix $L \in \mathbb{R}^{3\times 3}$ and the adaptive-gain matrix $\Lambda \in \mathbb{R}^{2\times 2}$ are design parameters to be chosen, and we define the *observation error*

$$e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \\ \theta - \hat{\theta} \end{bmatrix}. \tag{4.8}$$

The properties of the estimator are presented in the following theorem, which represents the main result of [65].

Theorem 1 *Under Assumption 1, the problem of slip coefficient estimation is solved by the adaptive observer (4.7) with arbitrary initial conditions, and with L and Λ positive*

definite. In particular, the estimator guarantees uniform exponential convergence of the estimates to the true slip coefficients, as time goes to infinity: $\hat{\eta}_R \to \eta_R$, $\hat{\eta}_L \to \eta_L$.

Proof: It is convenient to change coordinates by considering the *observation error* (4.8) and, in addition, the *estimation error*

$$\tilde{\eta} = \begin{bmatrix} \tilde{\eta}_R \\ \tilde{\eta}_L \end{bmatrix} = \begin{bmatrix} \hat{\eta}_R - \eta_R \\ \hat{\eta}_L - \eta_L \end{bmatrix}. \tag{4.9}$$

System (4.7) in the new error coordinates becomes

$$\dot{\varepsilon} = \begin{bmatrix} \dot{e} \\ \dot{\tilde{\eta}} \end{bmatrix} = \begin{bmatrix} -L & \Phi^\top(t) \\ -\Lambda\Phi(t) & 0 \end{bmatrix} \begin{bmatrix} e \\ \tilde{\eta} \end{bmatrix} = A(t,\varepsilon) \tag{4.10}$$

where, in line with standard adaptive control notation, we call *regressor* the matrix

$$\Phi(t) = \begin{bmatrix} -\frac{\cos\theta(t)}{2} v_R & -\frac{\sin\theta(t)}{2} v_R & -\frac{1}{2d} v_R \\ -\frac{\cos\theta(t)}{2} v_L & -\frac{\sin\theta(t)}{2} v_L & \frac{1}{2d} v_L \end{bmatrix}. \tag{4.11}$$

System (4.10) is almost standard in adaptive control literature, with the unusual feature of having a regressor being a matrix, instead of just a vector. In order to conclude uniform global asymptotic stability of the origin for the error system (we will say the error system is 0-UGAS), which is equivalent to solving the problem at hand, we rely on [99], which presents results that are general enough to treat our problem. The results we want to use from [99] are Corollary 1 and Theorem 2. These results require two assumptions, called A1 and A2, to hold, and require the pair (Φ, A) to be uniformly persistently exciting (u-PE).

A1 [99] *There exist* $\gamma_1 \in \mathcal{K}_\infty$, *a locally bounded function* $\gamma_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *and a positive definite continuous function* $\gamma_3 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, *such that for all* (t_0, ε_0), *all corresponding solutions of* (4.10) *satisfy*

$$\begin{aligned} \|\varepsilon\|_\infty &\leq \gamma_1(\|\varepsilon_0\|) \\ \|\gamma_3(\|e\|)\|_1 &\leq \gamma_2(\|\varepsilon_0\|). \end{aligned} \tag{4.12}$$

From [99, Remark 4] we know that A1 is satisfied if, for instance, there exists a locally

Lipschitz function V and two functions $\underline{\alpha}, \overline{\alpha} \in \mathcal{K}_\infty$ such that

$$\begin{aligned} \underline{\alpha}(\|\varepsilon\|) \leq V(t,\varepsilon) \leq \overline{\alpha}(\|\varepsilon\|) \\ \dot{V}_{(4.10)}(t,\varepsilon) \leq -\gamma_3(\|e\|). \end{aligned} \tag{4.13}$$

In order to prove this, we consider the quadratic Lyapunov function candidate

$$V(\varepsilon) = \frac{1}{2}\left(e^\top L^{-1} e + \tilde{\eta}^\top \Lambda^{-1} \tilde{\eta}\right) \tag{4.14}$$

which is a valid candidate by choosing $l_1 > 0$, $l_2 > 0$, $l_3 > 0$, $\lambda_1 > 0$, $\lambda_2 > 0$, as stated in Theorem 1. Now, its time derivative along the error system trajectories is

$$\begin{aligned} \dot{V}_{(4.10)}(\varepsilon) &= -e^\top e + e^\top \Phi^\top \tilde{\eta} - \tilde{\eta}^\top \Phi e \\ &= -\|e(t)\|^2 = -\gamma_3(\|e\|) \end{aligned} \tag{4.15}$$

which is negative semi-definite. In addition, each entry of the regressor $\Phi(t)$ is bounded with bounded time derivative (or equivalently, bounded and globally Lipschitz). Therefore, by La Salle/Yoshizawa [100, Theorem 8.4], system (4.10) is 0-UGS and

$$\lim_{t\to\infty} \|e(t)\|^2 = 0 \tag{4.16}$$

resulting in $e(t) \to 0$, satisfying A1. In fact, we can introduce $c_m, c_M > 0$ such that

$$\underline{\alpha}(\|\varepsilon\|) = c_m^2 \|\varepsilon\|^2 \leq V(\varepsilon) \leq c_M^2 \|\varepsilon\|^2 = \overline{\alpha}(\|\varepsilon\|). \tag{4.17}$$

Moreover, being $V(\varepsilon)$ a positive non-increasing function which is bounded from below by 0, we conclude that it has a limit

$$\lim_{t\to\infty} V(t) = V_\infty. \tag{4.18}$$

Integrating (4.15) from t_0 to ∞, it results

$$\int_{t_0}^{\infty} \|e(\tau)\|^2 \, d\tau = -\int_{t_0}^{\infty} \dot{V}(\tau) \, d\tau = V(\varepsilon_0) - V_\infty \tag{4.19}$$

and we have

$$V(\varepsilon_0) - V_\infty \leq V(\varepsilon_0) \leq c_M^2 \|\varepsilon_0\|^2 = \gamma_2(\|\varepsilon_0\|). \tag{4.20}$$

Also,

$$c_m^2\|\varepsilon\|_\infty^2 \leq c_m^2\|\varepsilon(t)\|^2 \leq V(\varepsilon(t)) \leq V(\varepsilon_0) \leq c_M^2\|\varepsilon_0\|^2 \tag{4.21}$$

from which $\|\varepsilon\|_\infty \leq (c_M/c_m)\|\varepsilon_0\|$. Thus (4.12) holds with $\gamma_1(s) = (c_M/c_m)s$, $\gamma_2(s) = c_M^2 s^2$ and $\gamma_3(s) = s^2$.

A2 [99] *Each entry of* Φ *is locally Lipschitz, and there exist non-decreasing functions* $\rho_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $(i = 1, 2, 3)$ *such that, for almost all* $(t, \varepsilon) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^5$

$$\begin{aligned} \max\{\|Le\|, \|\Lambda\Phi e\|\} &\leq \rho_1(\|\varepsilon\|)\|e\| \\ \|\Phi\| &\leq \rho_2(\|\varepsilon\|) \\ \left\|\frac{\partial\Phi}{\partial t}\right\| &\leq \rho_3(\|\varepsilon\|). \end{aligned} \tag{4.22}$$

As already mentioned, each entry of our regressor is bounded and globally Lipschitz, therefore A2 trivially holds with some constants ρ_1, ρ_2, ρ_3.

Definition: u-PE [99] *The pair* (Φ, A) *is called uniformly persistently exciting (u-PE) if, for each* $c_r > 0$*, there exist* $\mu, T > 0$*, such that, for all* $(t_0, \varepsilon_0) \in \mathbb{R}_{\geq 0} \times B_{c_r}$*, all corresponding solutions satisfy*

$$\int_t^{t+T} \Phi(\tau)\Phi^\top(\tau)\, d\tau \geq \mu I \quad \forall t \geq t_0. \tag{4.23}$$

In general it is difficult to check the u-PE condition since it must be valid for all possible solutions, clearly impossible to be known a priori. However, the proposed design leads to

$$\Phi\Phi^T = \begin{bmatrix} \frac{v_R^2}{4}\left(1 + \frac{1}{d^2}\right) & \frac{v_R v_L}{4}\left(1 - \frac{1}{d^2}\right) \\ \frac{v_R v_L}{4}\left(1 - \frac{1}{d^2}\right) & \frac{v_L^2}{4}\left(1 + \frac{1}{d^2}\right) \end{bmatrix} \tag{4.24}$$

which is positive definite for any positive v_R, v_L. In particular, it is even constant and positive definite under Assumption 1, and, as a consequence, (4.23) is directly satisfied in both scenarios. Under A1, A2 and u-PE we conclude that the error system (4.10) is 0-UGAS [99, Corollary 1].

Moreover, by [99, Theorem 2] we conclude Uniform Global Exponential Stability of the Origin (0-UGES) as: (i) $\Phi(t)$ is independent of ε; (ii) $\gamma_1(s)$, $\gamma_2(s)$, $\gamma_3(s)$ are proportional to s, s^2, and s^2, respectively, for all s. ■

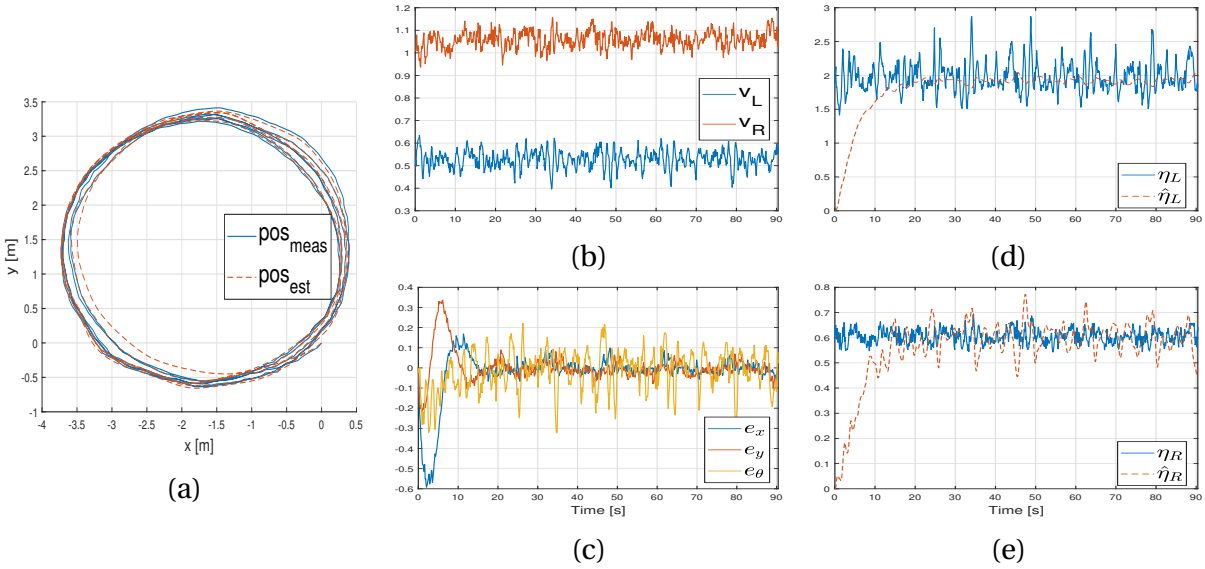
Remark 1 *It is well known in adaptive control literature that the extra requirement given by the u-PE property does not only guarantee convergence to the true parameters, but it also guarantees robustness for the overall error system* (4.10) *with respect to additive perturbations (i.e. considering* $\dot{\varepsilon} = A(t,\varepsilon) + d(t)$*). In particular, when the u-PE contributes to conclude uniform global asymptotic stability, the overall error system is "totally stable", which means it is robust against "small" non-vanishing perturbations [99, Equation (13)]. In our case, as u-PE induces uniform global exponential stability, the error system is robust with respect to arbitrarily large non-vanishing perturbations as the system is globally input-to-state stable [100, Lemma 4.6]. This property derives from a converse Lyapunov theorem [100, Theorem 4.14]. Robustness of the proposed adaptive observer is shown in Section 4.4.3 as we have real noisy measurements.*
It is important to remark that the vast majority of other designs present in the literature, explicitly or implicitly assume the slip coefficients to be constant (resulting in a zero-error result) or slowly time-varying (resulting in a small-error "practical" result). Our work is perfectly in line with these results and, in addition: (i) we discuss how only with constant track velocities and homogeneous soil it makes sense to consider constant slip coefficients; (ii) we formally prove that for any positive constant track velocities we have u-PE and related robustness. Notice that with other design choices the straight motion could result to be not u-PE [94].

Remark 2 *In many cases, adaptive approaches are not appealing for robotics applications as, in order to have robustness (which is induced by the u-PE property), the robot needs to perform some kind of oscillatory and, in general, nonsmooth trajectories in order to obtain sufficiently rich signals. In contrast to this, with our design we have u-PE even for constant inputs* v_R, v_L*, resulting in smooth trajectories, which makes the designed estimator appealing for OPA applications. This fact is highlighted in the simulations of Section 4.4.3.*

Remark 3 *The design of an identity observer on top of the adaptive laws may seem redundant and not needed as we are measuring the whole state. We want to explain why this is not the case and an extra structure is required. The intuition behind the proposed design comes from noticing that* (4.6) *is linear in the parameters to be found:*

$$\begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^\top = -\Phi^\top \begin{bmatrix} \eta_R & \eta_L \end{bmatrix}^\top$$

where, as already emphasized by Assumption 1, we have that the unknown parameters



(a) (b) (c) (d) (e)

Figure 4.19: Real data collected during turning motion tests.

η_R, η_L *are constant. This is a desirable scenario for designing adaptive laws. However, as we do not have direct measurements of* $\dot{x}, \dot{y}, \dot{\theta}$*, we cannot design estimation laws for the unknown parameters without designing any extra dynamical system. This reasoning is what justified our choice to design an identity observer to be combined with the estimation laws, resulting in what we called adaptive identity observer. This choice resulted to be very convenient both in order to bring the error system in the standard form* (4.10)*, and in terms of u-PE requirement, as already emphasized in Remark 2.*

4.4.3 Observer design validation

To enrich the analysis with real data, we performed experimental tests, measuring x, y using a high precision GNSS sensor, which embeds also a compass, from which the heading angle θ was retrieved. Notice that we decided to use raw-data measurements, without any filtering, so as to highlight robustness of the proposed design.
To compare the estimates of slip coefficients coming from (4.7) with ground truth values, we used the (3.4) as follows:

$$\begin{aligned} i_R &= 1 - \frac{V_{t,R}}{v_R} = 1 - \frac{V_{G,m} + d\Omega_{z,m}}{v_{R,m}} \\ i_L &= 1 - \frac{V_{t,L}}{v_L} = 1 - \frac{V_{G,m} - d\Omega_{z,m}}{v_{L,m}} \end{aligned} \tag{4.25}$$

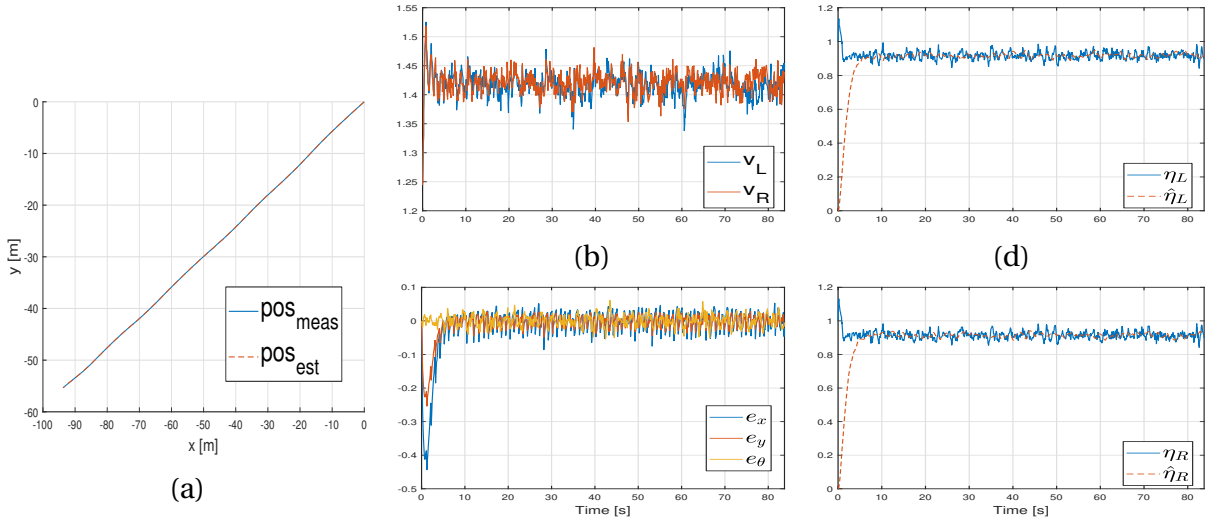


Figure 4.20: Real data collected during straight motion tests.

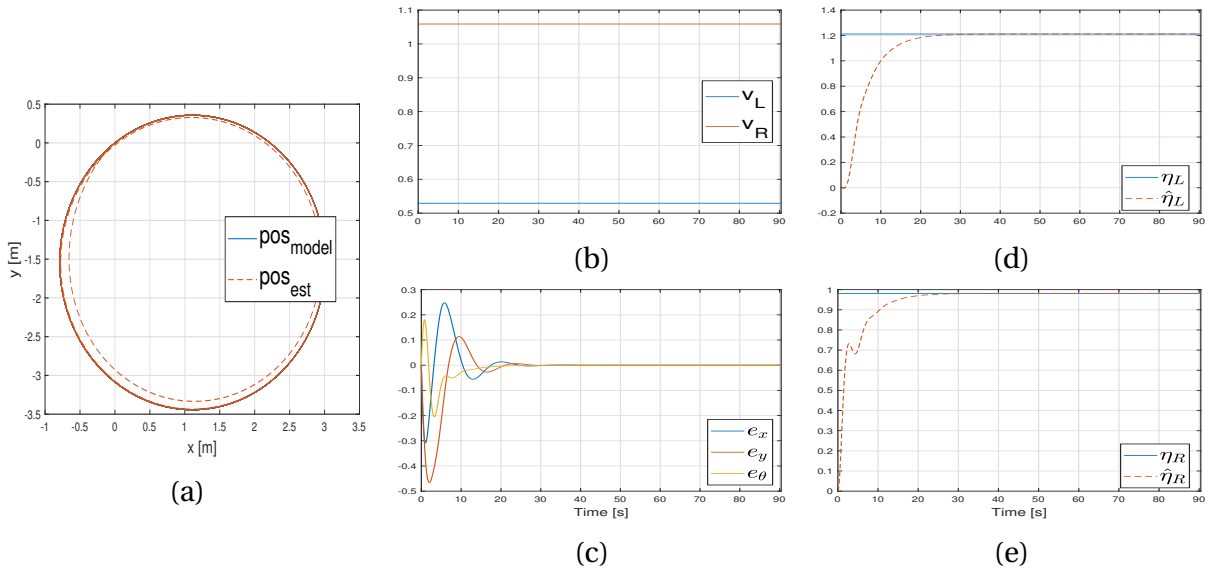


Figure 4.21: Results of the numerical simulation for turning motion.

in which, $V_{t,R}, V_{t,L}$ are the right and left track true velocities, respectively. The subscript m identifies the measured quantities: $V_{G,m}$ from the GPS, $\Omega_{z,m}$ from IMU data (gyroscope) and $v_{R,m}, v_{L,m}$ collected from motor encoders. Since we performed these tests in open-field scenario, it was possible to rely on GNSS data, but it is important to mention that this was done only for comparison purposes since, as mentioned above, these data are not always reliable in the considered practical application (inside orchard rows).

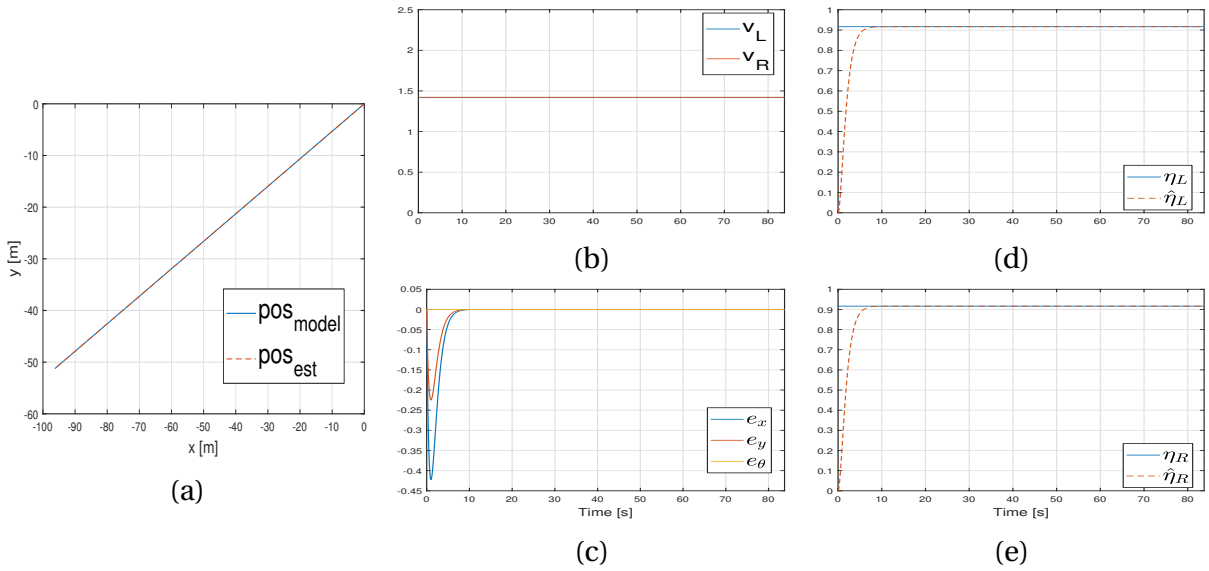


Figure 4.22: Results of the numerical simulation for straight motion.

To test the performance of (4.7) in both the navigation scenarios described above, we performed different on-field tests, whose results are shown in Figure 4.19 and Figure 4.20. In each one of the cases, we imposed constant reference speeds to the two motors, equal to each other during straight motion and different during turning motion tests. In fact, Figure 4.19.b, Figure 4.20.b show track speeds, highlighting the noisy nature of measurements collected and used to feed the observer. As expected, the observation errors tend to zero, as reported indeed in Figure 4.19.c and Figure 4.20.c. This is also understandable qualitatively from Figure 4.19.a and Figure 4.20.a, which compare the real trajectories recorded during motion, and the ones estimated by the proposed observer.

As expected from Remark 1, Figure 4.19.d and Figure 4.19.e, as well as Figure 4.20.d and Figure 4.20.e, show the robust convergence of the estimated slip coefficients $\hat{\eta}_L, \hat{\eta}_R$, to the real ones, η_L, η_R, approximated using (4.5) and (4.25).

It is also important to mention that the good performance obtained during experimental tests are not given by specifically-tuned observer parameters, rather by the structural properties described and proved in Section 4.4.2. In fact, both L, Λ have been considered as identity matrices (this also explains the convergence time), since our aim is to show the effectiveness of the proposed approach rather than fine tuning parameters for some particular application.

In the end, Figure 4.21 and Figure 4.22 point out what said in Remark 2, in noise-free cases. The convergence of the slip coefficients to real values is asymptotic and not just practical as before, remarking once again u-PE of the trajectories. To show this, we fed the observer with mean values obtained by the tests shown in the cases above, achieving asymptotic convergence both in turning and straight motion scenarios.

4.5 Position control

In this section we want to design a position control for a generic tracked vehicle. In particular, we want to keep it as simple as possible, indeed we want to show that using well-known *input-output feedback linearization* position control for Differential Wheeled Robots (DWRs), it is possible to control the position of Skid-Steering Vehicles (SSVs), simply adding a slip compensation stage based on the estimates obtained by the adaptive observer proposed in Section 4.4.
The design of the controller will be developed accordingly to the following steps:

1. Design the position controller for DWRs (therefore without slip) based on well-known and tested control paradigm, [98];
2. Use the estimates of slip coefficients to compensate the control commands generated by the controller developed in the previous step and apply the new commands to the SSV, then verify the feasibility of the control design.

The basic control scheme envisioned is shown in Figure 4.23.

4.5.1 DWR and SSW kinematics

Before starting the synthesis of the controller, it is convenient to recall some of the definition given in Chapter 3, in particular, the kinematic model of a generic non-holonomic vehicle (such as DWRs and SSVs) can be described as

$$\begin{aligned} \dot{x} &= V_G \cos\theta \\ \dot{y} &= V_G \sin\theta \\ \dot{\theta} &= \Omega_z. \end{aligned} \tag{4.26}$$

Where the linear and angular velocities of the vehicle V_G, Ω_z can be considered as the two control inputs of the system ($u_1 = V_G, u_2 = \Omega_z$), since they are shared by both DWR and SSV. Then, we want to find the map between these control inputs and the

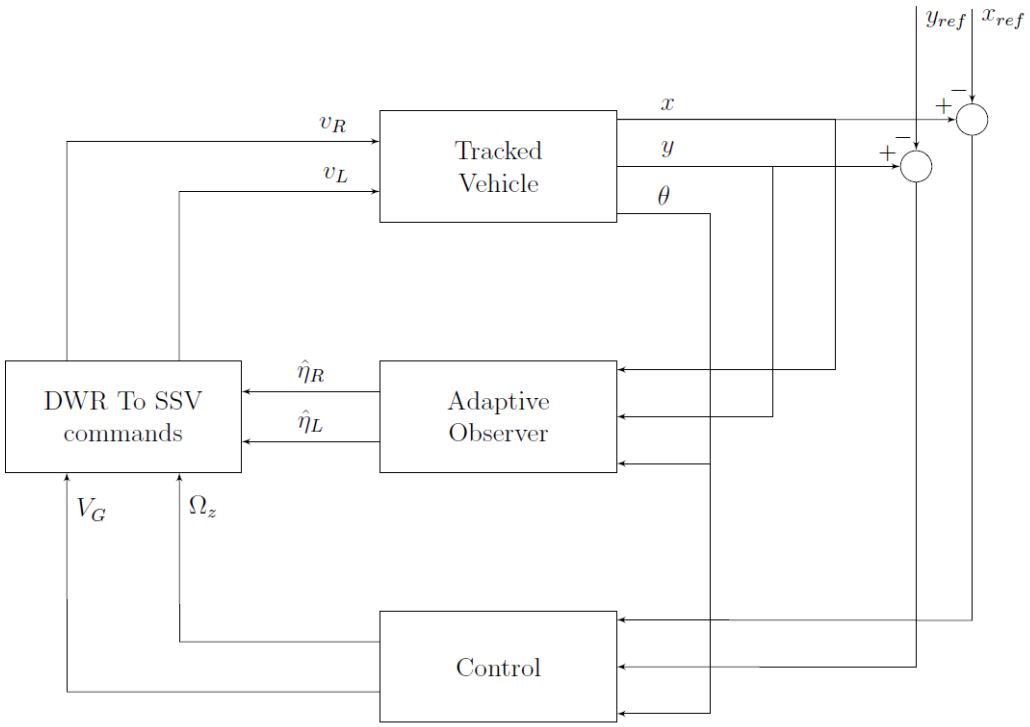


Figure 4.23: Position control scheme implemented with slip compensation.

real inputs used to steer the vehicle, namely the linear velocities of the wheels/tracks, v_R, v_L. In the case of DWRs, we have that

$$\begin{aligned} V_G &= \frac{v_R + v_L}{2} \\ \Omega_z &= \frac{v_R - v_L}{2d} \end{aligned} \quad \Rightarrow \quad \begin{aligned} v_R &= V_G + d\Omega_z = u_1 + du_2 \\ v_L &= V_G - d\Omega_z = u_1 - du_2. \end{aligned} \tag{4.27}$$

Whereas for SSVs, the desired map is given by

$$\begin{aligned} V_G &= \frac{v_R\eta_R + v_L\eta_L}{2} \\ \Omega_z &= \frac{v_R\eta_R - v_L\eta_L}{2d} \end{aligned} \quad \Rightarrow \quad \begin{aligned} v_R &= \frac{V_G + d\Omega_z}{\eta_R} = \frac{u_1 + du_2}{\eta_R} \\ v_L &= \frac{V_G - d\Omega_z}{\eta_L} = \frac{u_1 - du_2}{\eta_L}. \end{aligned} \tag{4.28}$$

Looking at the (4.27) and (4.28), it is even more evident the interpretation of slip coefficients as efficiency terms we gave in the previous sections. The relations found allow us to implement the desired control scheme using standard techniques for the definition of u_1, u_2 assuming the vehicle to be a DWR. Then remap them accordingly to the measure/estimated slip into the real commands v_R, v_L.

4.5.2 DWR position controller

In line with what has been said before, the system to be controlled is the (4.26), in particular we are interested in controlling its 2D position, thus, considering as outputs $y_1 = x - x_{ref}$ and $y_2 = y - y_{ref}$. Where x_{ref}, y_{ref} represent the constant reference position coordinates[1]. Therefore, the system can be rewritten as

$$\begin{aligned} \dot{x} &= u_1 \cos\theta \\ \dot{y} &= u_1 \sin\theta \\ \dot{\theta} &= u_2 \\ y_1 &= x - x_{ref} \\ y_2 &= y - y_{ref}. \end{aligned} \tag{4.29}$$

The reader is referred to the Appendix B or to [100, 101, 102] for the general formulation of concepts that we are going to use extensively in the following, such as *relative degree, decoupling matrix, and normal form*. Given the form of the system (4.29), it is possible to consider it as a Multi-Input Multi-Output (MIMO) system with the same number of inputs and outputs (*Squared MIMO system*). It is now possible to make use of the [102, Proposition 5.3.1], which states

Proposition 1 *Consider a multivariable nonlinear system with m inputs and outputs (squared MIMO system)*

$$\begin{aligned} \dot{x} &= f(x) + g_1(x)u_1 + \cdots + g_m(x)u_m \\ y_1 &= h_1(x) \\ &\cdots \\ y_m &= h_m(x). \end{aligned} \tag{4.30}$$

*The Noninteracting Control Problem (*input-output decoupling *problem) is solvable if and only if the $m \times m$ decoupling matrix $A(x)$ is nonsingular.*

[1] Even if [98] shows that this control structure is valid also for time-varying reference positions, we decided to keep them constant since the main focus of this control design is to have is as simple as possible. We will discuss later on about the trajectories used to feed this controller.

Then, we are interested in defining the relative degree of each output of (4.29), which straightforwardly results in

$$y_1 = x - x_{ref} \quad \Rightarrow \quad \dot{y}_1 = u_1 \cos\theta \quad \Rightarrow \quad r_1 = 1 \quad \forall\theta \neq \left(\frac{\pi}{2} + k\pi\right), \forall k \in \mathbb{Z} \tag{4.31}$$

and

$$y_2 = y - y_{ref} \quad \Rightarrow \quad \dot{y}_2 = u_1 \sin\theta \quad \Rightarrow \quad r_2 = 1 \quad \forall\theta \neq (k\pi), \forall k \in \mathbb{Z}. \tag{4.32}$$

Where r_1, r_2 are respectively the relative degrees of y_1, y_2. More in general, considering m outputs it is possible to write

Remark 4

$$\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = N(x) + A(x) \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \tag{4.33}$$

where $y_i^{(r_i)}$ represents the r_i-th derivative of the output y_i, we call r_i the relative degree of the y_i.

It follows from (4.29) and the Proposition 1 that we can rewrite the (4.31) and (4.32) in matrix form as

$$\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \end{bmatrix} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{N(x)} + \underbrace{\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{bmatrix}}_{A(x)} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{4.34}$$

Then, being $A(x)$ singular, we can state that the system (4.29) cannot be input-output decoupled by feedback linearization, which is in line with [103, 98] which state that the kinematic model of nonholonomic vehicles cannot be linearized by static state feedback. Indeed, it is now necessary to extend the states of (4.29), for instance we

can consider

$$\begin{aligned}
\dot{x} &= u_1 \cos\theta \\
\dot{y} &= u_1 \sin\theta \\
\dot{\theta} &= u_2 \\
\dot{u}_1 &= w_1 \\
y_1 &= x - x_{ref} \\
y_2 &= y - y_{ref}.
\end{aligned} \tag{4.35}$$

Following the same procedure shown before in (4.31) and (4.32), the derivation of the relative degree of y_1 becomes

$$\begin{aligned}
y_1 &= x - x_{ref} \\
\dot{y}_1 &= \dot{x} = u_1 \cos\theta \\
\ddot{y}_1 &= \dot{u}_1 \cos\theta - u_1 \dot{\theta} \sin\theta \\
&= w_1 \cos\theta - u_1 u_2 \sin\theta
\end{aligned} \quad \Rightarrow \quad r_1 = 2 \quad \forall u_1 \neq 0, \tag{4.36}$$

while for y_2

$$\begin{aligned}
y_2 &= y - y_{ref} \\
\dot{y}_2 &= \dot{y} = u_1 \sin\theta \\
\ddot{y}_1 &= \dot{u}_1 \sin\theta + u_1 \dot{\theta} \cos\theta \\
&= w_1 \cos\theta + u_1 u_2 \cos\theta
\end{aligned} \quad \Rightarrow \quad r_2 = 2 \quad \forall u_1 \neq 0. \tag{4.37}$$

We can then rewrite the output derivative vector as

$$\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \end{bmatrix} = \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{N(x)} + \underbrace{\begin{bmatrix} \cos\theta & -u_1 \sin\theta \\ \sin\theta & u_1 \cos\theta \end{bmatrix}}_{A(x)} \begin{bmatrix} w_1 \\ u_2 \end{bmatrix} \tag{4.38}$$

In this case, it can be easily proven that, the matrix $A(x)$ is nonsingular $\forall u_1 \neq 0$, which is consistent with the assumptions for which the Proposition 1 holds, meaning that the extended robot kinematic (4.35) can be input-output decoupled via feedback linearization. It is now possible to recall the *standard noninteractive feedback* Corollary from [102]:

Corollary 1 *When $A(x)$ is nonsingular, we can use an input of the form*

$$u = -A^{-1}(x)\,N(x) + A^{-1}(x)\,v \tag{4.39}$$

which solves the Nonintercating Control Problem. [102] refers to this input as to the standard noninteractive feedback. *It then results that*

$$\begin{bmatrix} y_1^{(r_1)} = v_1 \\ y_2^{(r_2)} = v_2 \\ \vdots \\ y_m^{(r_m)} = v_m \end{bmatrix}. \tag{4.40}$$

From Corollary 1, we can derive the linearizing inputs for the (4.35) system as

$$u = \begin{bmatrix} w_1 \\ u_2 \end{bmatrix} = \underbrace{-A^{-1}(x)\,N(x)}_{=0} + A^{-1}(x)\,v = \begin{bmatrix} \cos\theta & \sin\theta \\ -\dfrac{\sin\theta}{u_1} & \dfrac{\cos\theta}{u_1} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \tag{4.41}$$

At this point of the design, we can select two coordinates changes $\Phi_1(x,y,\theta,u_1)$, $\Phi_2(x,y,\theta,u_1)$ allowing us to properly define the new control inputs v_1, v_2. Namely, $z_1 = \Phi_1(x,y,\theta,u_1)$ and $z_2 = \Phi_2(x,y,\theta,u_1)$.

$$z_1 = \begin{bmatrix} \xi_{11} \\ \xi_{12} \\ \sigma_{11} \\ \sigma_{12} \end{bmatrix} = \underbrace{\begin{bmatrix} h_1(x) = x - x_{ref} = y_1 \\ L_f h_1(x) = u_1 \cos\theta = \dot{y}_1 \\ ? \\ ? \end{bmatrix}}_{\Phi_1(x,y,\theta,u_1)} \qquad z_2 = \begin{bmatrix} \xi_{21} \\ \xi_{22} \\ \sigma_{21} \\ \sigma_{22} \end{bmatrix} = \underbrace{\begin{bmatrix} h_2(y) = y - y_{ref} = y_2 \\ L_f h_2(x) = u_1 \sin\theta = \dot{y}_2 \\ ? \\ ? \end{bmatrix}}_{\Phi_2(x,y,\theta,u_1)} \tag{4.42}$$

It then follows that the evolution of the z_1, z_2 systems can be written as:

$$\dot{z}_1 = \begin{bmatrix} \dot{\xi}_{11} \\ \dot{\xi}_{12} \\ \dot{\sigma}_{11} \\ \dot{\sigma}_{12} \end{bmatrix} = \begin{bmatrix} \xi_{12} \\ v_1 = -k_{11}\xi_{11} - k_{12}\xi_{12} \\ ? \\ ? \end{bmatrix} \qquad \dot{z}_2 = \begin{bmatrix} \dot{\xi}_{21} \\ \dot{\xi}_{22} \\ \dot{\sigma}_{21} \\ \dot{\sigma}_{22} \end{bmatrix} = \begin{bmatrix} \xi_{22} \\ v_2 = -k_{21}\xi_{21} - k_{22}\xi_{22} \\ ? \\ ? \end{bmatrix} \tag{4.43}$$

Where $k_{11}, k_{12}, k_{21}, k_{22} \in \mathbb{R}$ are the controller gains to be tuned.
In order to study the stability of the system in the new coordinates, we have to check the stability of the *zero dynamics*, $\sigma_{11}, \sigma_{12}, \sigma_{21}, \sigma_{22}$. In particular, if we select the coor-

dinate transform matrices Φ_1, Φ_2 such that

$$\begin{aligned} \xi_{11} &= x - x_{ref} & \xi_{21} &= y - y_{ref} \\ \xi_{12} &= u_1 \cos\theta & \xi_{22} &= u_1 \sin\theta \\ \sigma_{11} &= y - y_{ref} & \sigma_{21} &= x - x_{ref} \\ \sigma_{12} &= u_1 \sin\theta & \sigma_{22} &= u_1 \cos\theta. \end{aligned} \tag{4.44}$$

Then, the gradients of the matrices Φ_1, Φ_2 turn out to be

$$\frac{\partial \Phi_1(x)}{\partial x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -u_1 \sin\theta & \cos\theta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_1 \cos\theta & \sin\theta \end{bmatrix} \quad \frac{\partial \Phi_2(x)}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & u_1 \cos\theta & \sin\theta \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -u_1 \sin\theta & \cos\theta \end{bmatrix} \tag{4.45}$$

which are full rank $\forall u_1 \neq 0$, meaning that the candidates coordinate change matrices Φ_1, Φ_2 are proper change of coordinates and in particular, in the new coordinates the evolution of the system can be written as:

$$\begin{cases} \dot{\xi}_{11} &= \xi_{12} \\ \dot{\xi}_{12} &= v_1 = -k_{11}\xi_{11} - k_{12}\xi_{12} \\ \dot{\sigma}_{11} &= \sigma_{12} \\ \dot{\sigma}_{12} &= v_2 = -k_{21}\sigma_{12} - k_{22}\sigma_{12} \end{cases} \quad \begin{cases} \dot{\xi}_{21} &= \xi_{12} \\ \dot{\xi}_{22} &= v_2 = -k_{21}\xi_{21} - k_{22}\xi_{22} \\ \dot{\sigma}_{21} &= \sigma_{22} \\ \dot{\sigma}_{22} &= v_1 = -k_{11}\sigma_{21} - k_{12}\sigma_{22} \end{cases} \tag{4.46}$$

It is possible to claim that the system (4.46) is Hurwitz (including its *zero dynamics*) if $k_{11} > 0, k_{12} > 0, k_{21} > 0, k_{22} > 0$.

Remark 5 *As long as $u_1 \neq 0$, we can safely use the controller obtained in (4.46), which guarantees asymptotically stable zero dynamics by means of input-output decoupling via feedback linearization.*

Figure 4.24 shows the interconnection of the observer (4.7) and the design Input-Output Feedback Linearized position control used to regulate the tracked vehicle kinematics.

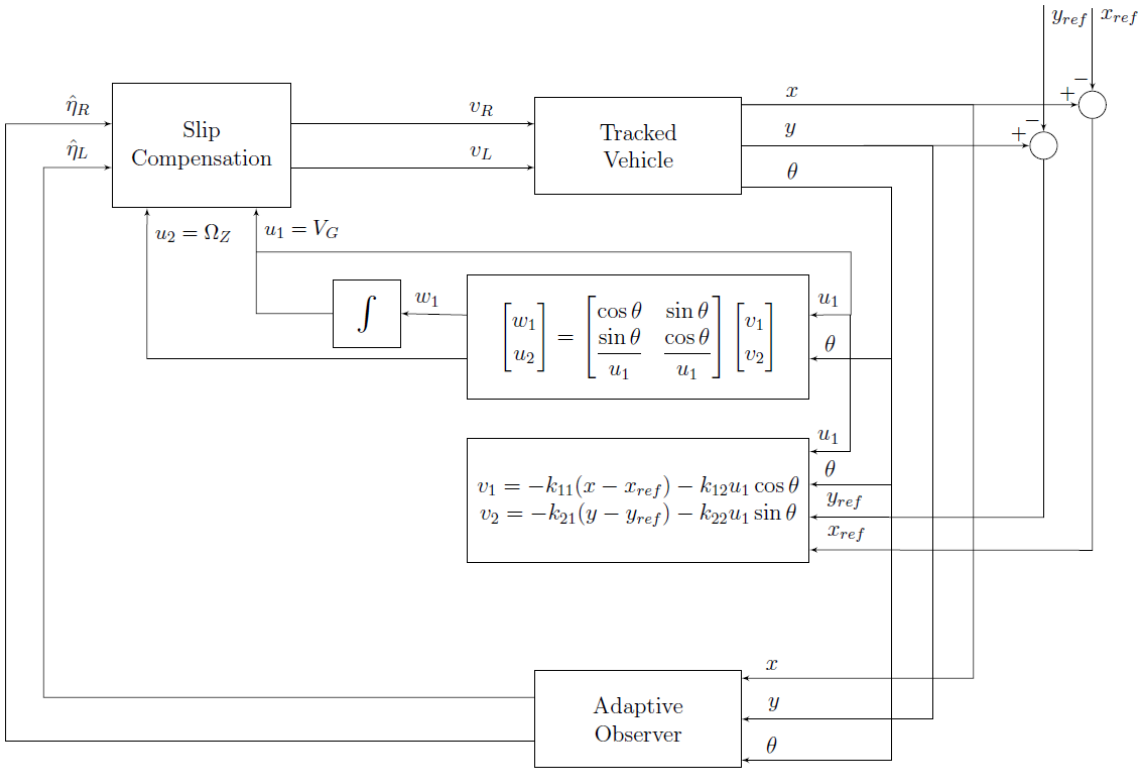


Figure 4.24: Final control scheme with Adaptive Slip compensation

4.5.3 Simulation results

To conclude this section, we briefly want to show a simulation comparison between position control of a tracked vehicle with and without slip[1] compensation, using the control paradigm introduced above.
In particular, Figure 4.25a shows the tracking performance of the controller, in which the control gains were set as $k_{11} = 0.5, k_{12} = 0.75, k_{21} = 0.5, k_{22} = 0.75$. The red dashed line is the case in which the slip compensation is turned off, showing worst performances in path following, on the other hand, the blue dashed line shows that with the slip compensation enabled, the position error is reduced remarkably.
Figure 4.25c and Figure 4.25b shows how the the adaptive observer is capable of properly estimate the slip coefficients. Notice that the trajectory to be tracked does not satisfy completely the Assumption 1, which requires constant values for track velocities v_R, v_L, and in turn of η_R, η_L. Anyway, we designed the trajectory as a sequence

[1]For simulation purposes, as there is no model relating V_G, Ω_z, v_R, v_L with slip under tracks, we fitted data recorder in real tests. We want to point out that these values are not reliable, as extensively discussed throughout the whole thesis; we only used that to have something to start with.

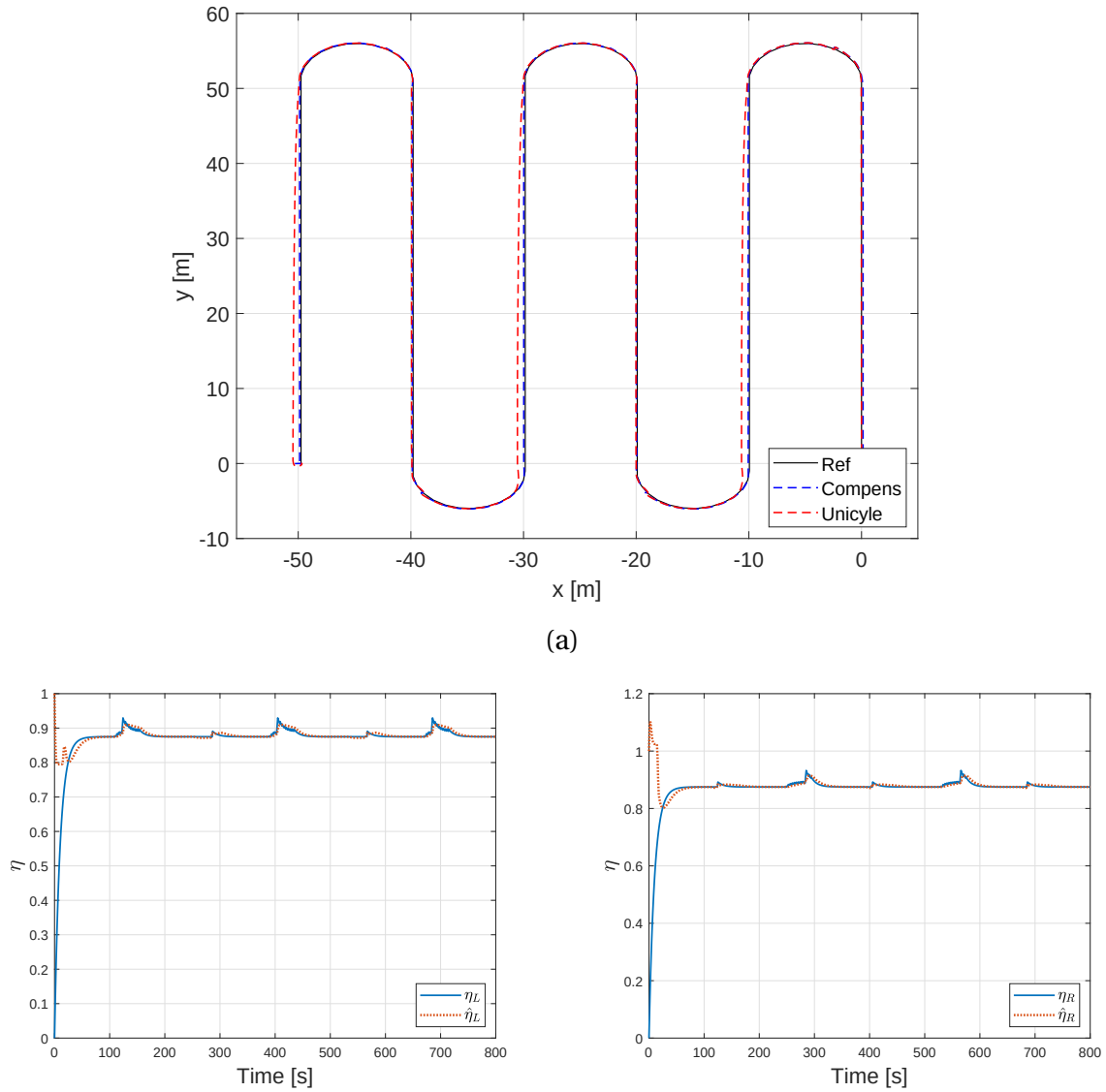


Figure 4.25: Results of the numerical simulation for straight motion.

of waypoints in such a way that the trajectory is *piece-wise* constant, then, as soon as the vehicle approaches the goal waypoint, the trajectory generator switches to the following one, ensuring $u_1 \neq 0$. The final result can be considered as a *slowly time-varying* reference trajectory. We want to stress the fact that the results shown here are only meant to experimentally simulate the interconnection between the observer and the position controller.

4.6 Conclusions

In the first part of this chapter, we presented our solution for the problem of localization and autonomous navigation both inside and outside the orchard. In particular, we showed how a 2-stage extended Kalman filter allows the robot to estimate its global position reliably. In particular, we pointed out how we used information about the environment (trees are organized in straight rows), to achieve GPS-free global localization inside the orchard.
We also shown an optimization of the line detection algorithm *Hough Transform*, which allows the robot to determine its relative position with respect to the surrounding orchard rows in a faster and more robust way. A similar approach is also used to plan the trajectory to follow in order to switch from one orchard lane to the following.

In the second part of the chapter, we designed an adaptive observer for estimation of the slip coefficients affecting the dynamics of tracked vehicles. Thanks to the particular structure of the proposed observer, we are able to prove exponential convergence to the real values of the coefficients. Then, we leverage the estimates in order to design a position control for tracked vehicle using standard techniques for the slip-free kinematic model. Simulation tests shown improvement of the tracking performance when the compensation of the slip is activated.

5 Experiments

In this chapter, real on-field experimental results are reported, validating both the mechatronic design of the robot developed and the proposed navigation algorithms.

5.1 Introduction

The aim of the experimental activities described in this chapter is the validation of the different subsystems described throughout this dissertation. In particular, these tests can be are meant to assess the performance of the most important features of the robot, focusing on efficiency and failures. The different experiments we performed can be divided in the following categories:

- *Mechatronics* - These experiments were carried out in order to asses the robustness of the mechanical components and frame, as well as the proper functioning of the customized implements. Moreover, we verified the communication, integration and reliability of all the electric and electronic components. In addition, we also investigated the power consumption of the locomotion system in different conditions, in order to obtain an estimate of the battery autonomy in terms of working time.
- *Logical structure* - This set of tests are meant to validate the reliability of the

basic hardware and software components of the High and Low Level, as well as verifying the robustness of both *horizontal* and *vertical* communication[1].

- *Model validation* - Experimental tests are required in order to validate the proposed dynamic model presented in Chapter 3. Anyway, for sake of readability, the results of these experiments have been already shown at the end of Chapter 3.

- *Navigation algorithm* - We tested the autonomous navigation capabilities of the robot, in particular we verified the line detection performance during in-row navigation, along with the capability to track the desired trajectory during this navigation scenario.

5.2 Mechatronic system test

5.2.1 Brief history

The prototype currently available at CASY laboratory is the result of more than four years of work of the *Agricultural Robotics Group* lead by professor Lorenzo Marconi[2]. The current is the last one of three previous versions developed and incrementally upgraded[3] in order to correct the observed flaws. In this section we are going to review the main features of the previous versions.

Rover Mark 1

The vehicle shown in Figure 5.1 is the first version of the agricultural UGV developed in our laboratory. It is compliant with all the design concepts described in Chapter 1and it already presents some of the features we kept in its latest version, i.e. the track subsystem (track, electric gear-motor, encoder and parking brakes), the combustion engine and the four-bar linkage[4].
On the other hand, as it is possible to see in Figure 5.1, the mechanical frame hosts a

[1] With the term *horizontal*, we identify the communication between components of the same logical level, whilst *vertical* is used to indicate the relation between High and Low level.
[2] https://www.unibo.it/sitoweb/lorenzo.marconi/en
[3] With *incremental* we mean that we did not assembled several robots, rather we upgraded the one available. This was done to avoid repeating design mistakes before extensively testing each component of the robot.
[4] In the following we will also discuss how to upgrade these components

(a)

(c)

(b)

(d)

Figure 5.1: Rover Mark 1 in different configurations: (a),(c) bare bone configuration; (b) equipped with a mower, and (d) equipped with a sprayer.

fiberglass water tank for spraying purposes, which is no more present in the actual vehicle. In line with the design concepts of compactness and integration, a pre-shaped tank was not allowing us to reach the level of customization required to keep the same footprint of the vehicle, but adding several other components such us water pumps and bigger batteries.

The *Rover Mark 1* was also featuring two lateral boxes hosting lead-acid batteries providing power to the locomotion system. Inside these boxes the motor controllers *Elife-Drive HR*, [104], were installed with outward fans for cooling purposes, visible in Figure 5.1b. Both the set of lead-acid batteries and the Elife motor controllers were replaced as lead batteries feature lower *specific energy* (Wh/kg) than lithium-ion batteries, [105].

This prototype was mainly used to verify the mechanical design and robustness of the frame, as well as the viability of the design concepts. We tested it for more than 100

hours during which the robot was equipped either with the mower (Figure 5.1b) and the sprayer (Figure 5.1d). In addition, it allowed our research group to collect data about power absorption and main critical aspects of the design, namely, the batteries did not guaranteed the desired performance and motor the controllers were not giving the robot the capability to recover kinetic energy during braking.

Rover Mark 2

The analysis of the data collected during experimental test with Mark 1, led us to change the batteries supplying power to the vehicle, switching to lithium-ion (LiFePO4) technologies, which guarantees better specific energy levels, as well as discharge characteristics which fitted better the applications requirements, i.e. the voltage drop rate during discharge is smaller than the one of lead-acid batteries, therefore it allows higher current demand for longer time periods [106]. As mentioned before, along with batteries we also upgraded the motor controllers, as the one used before resulted a little undersized for applications on sloped terrains.
The new lithium-ion battery and motor controllers installed on the *Rover Mark 2* are the same actually used in our lastest version described in Chapter 2.

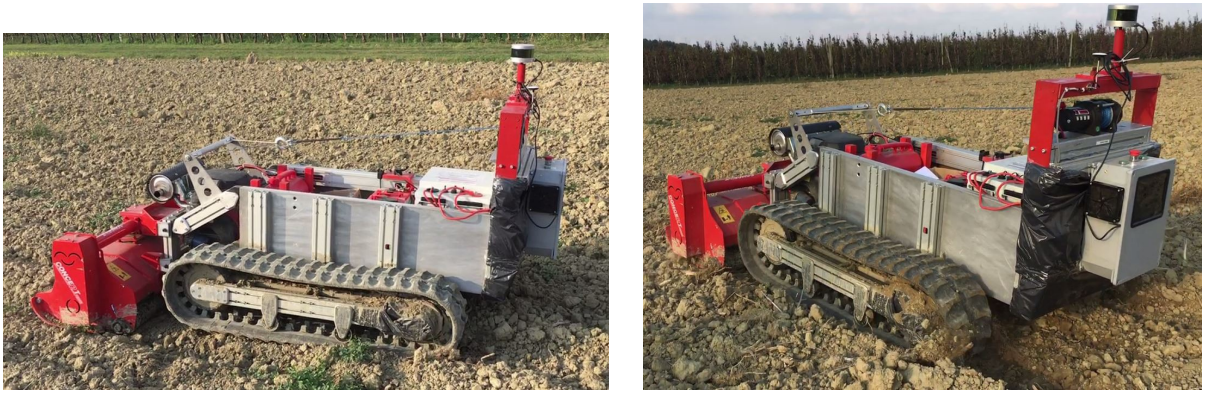
(a) (b)

Figure 5.2: Rover Mark 2 in different configurations: (a) bare bone configuration; (b) equipped with a mower.

Figure 5.2 shows how the second version of the CASY-Rover (Rover Mark 2), notice that along with the li-ion battery, there still is a lead-acid battery, which was used for the ignition of the combustion engine and for powering the winch. It is also easily

noticeable the absence of the water tank, which was not mounted on this vehicle as this version was particularly meant for testing navigation capabilities of the enriched sensor suite (i.e. GNSS receiver, 3D LiDAR, IMU and motor telemetry[1]), in fact the mechanical structure is really similar to the previous one except for the absence of lateral boxes which were removed.

Stress test on loose soil In order to stess test the locomotion subsystem and the mechanical structure, we design special experiments on loose and soft soil, which is not the typical terrain which this prototype is meant to work on, but it allowed us to let the mower act as a frontal plow/scraper, increasing the force required to move the robot forward. Figure 5.3 and the video[2] show the robot during the tests described.



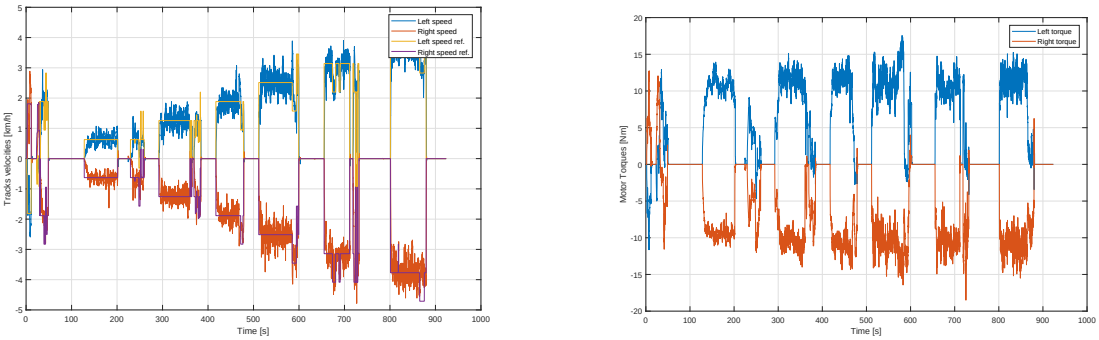
(a) (b)

Figure 5.3: Experimental setup of the stress tests on loose terrain.

As depicted in Figure 5.4, the robot is controlled sending step-wise speed references to each of the two motors, we used steps in order to leave idle times (about 30s) between each test, making the collected data to be easily read. After each idle time we increased the reference command, ranging from 0.6 km/h to 3.8 km/h, Figure 5.4a shows a comparison between the reference speeds sent to the motors and the real velocities measured from the encoders. On the other hand, Figure 5.4b shows the plot of the motor torques. It interestingly shows that they reach values of 15 Nm and are roughly constant among different-velocity tests.

[1] Motor encoders were the same of the previous version. Moreover, the sensor suite of this prototype is the same present on the latest version.

[2] https://youtu.be/5kembWpUYdM



(a) (b)

Figure 5.4: Motors telemetry data about motor velocities and torques.

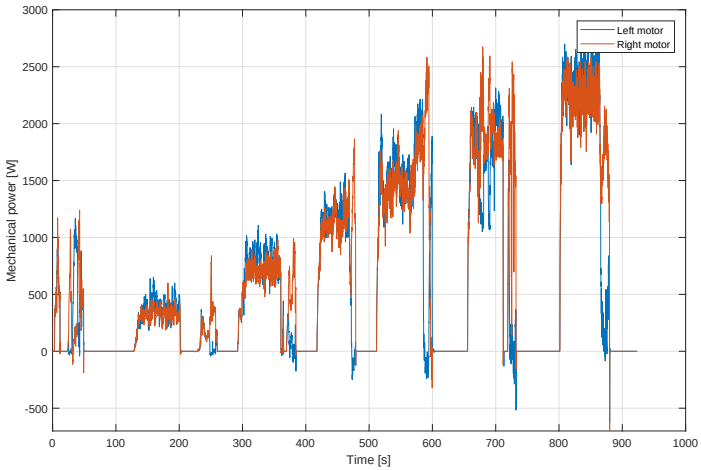


Figure 5.5: Mechanical powers absorbed by the robot during stress tests

Figure 5.5 shows the mechanical power evolution during the whole experiment, and the power values are almost the same for each of the two tracks, as expected since moving straight on the same terrain conditions. The power goes from 0.5 kW to 2.5 kW, for *each* motor. These values are within the safety limits of the motors, but way above the nominal values (1.5 kW), meaning that the robot can keep this working conditions only for a limited period of time. The results are anyway good, as this application is very far from the ones the robot is designed for (working on orchard/-greenhouses terrains), meaning that these results do not limit the expected application, as typically it is less demanding in terms of locomotion power. In line with this analysis, Figure 5.6 reports an increase in motors temperature of about 15 °C, just

during this sequence of tests (almost 900s), while motor controllers' is almost constant. This effect is given by two reasons, on one side they feature different cooling conditions, as motors are embodied in the track frame, while the motor controllers are fixed on a metal sheet which increases the dissipation effect. The motors are working really close to their maximum, affecting heat dissipation. On the other hand, motor controllers have been over-sized for the nominal power of our motors.

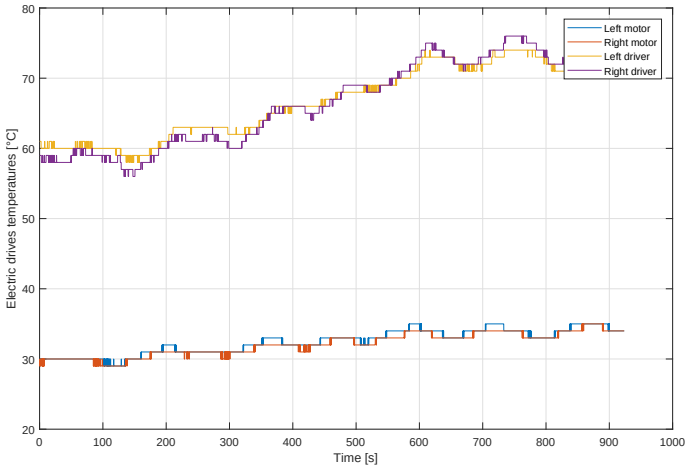
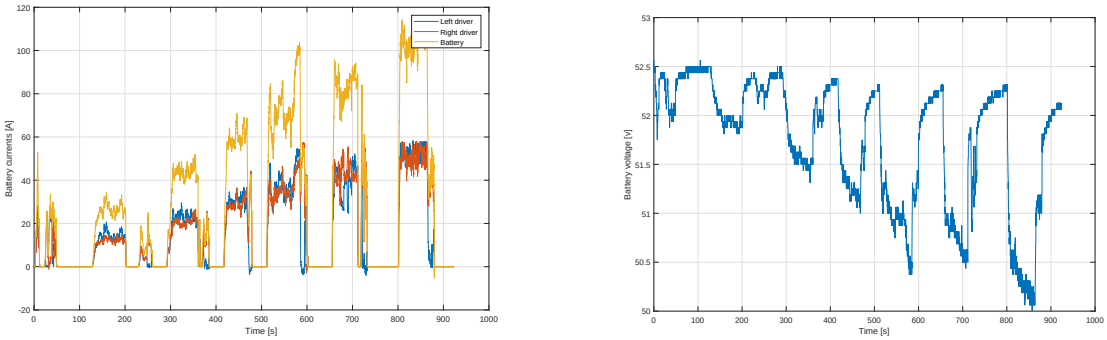


Figure 5.6: Drivers temperatures during stress test

Now, we want to take a closer look to the power flowing from and to the battery. Figure 5.7a shows the *DC* currents absorbed by left and right controllers alone and their sum, i.e. the whole current coming from the battery. We encourage the reader to notice the negative peaks of currents, corresponding to the instants in which speed references are suddenly set to zero and the robot has to brake for a while, recovering part of the kinetic energy, injecting it into the battery. This effect is more evident at high speeds, when the braking action is stronger.
Then the battery voltage during the experiment is proposed in Figure 5.7b, the maximum voltage drop is about 2.5 Volts (4.6 %) at discharge current about 0.85C, in line with the specifications of this type of batteries.
Summing up, the mechanical frame of the vehicle was well designed and robust enough to work even on heavy duty terrains, such as the one described in this test. Nevertheless, the locomotion system resulted to be used far above its nominal conditions, this aspect will be investigated further in nominal working conditions. We want to point out that this set of tests is relevant even if they were performed on the previous prototype, as the whole locomotion system (track subsystem, motor controller and



(a) (b)

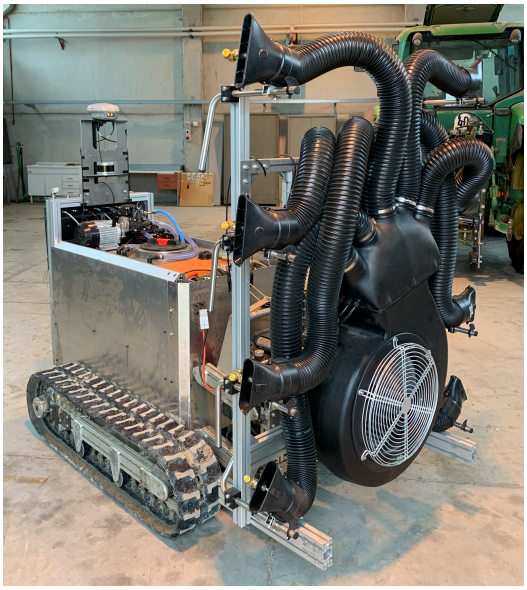
Figure 5.7: Current absorbed during stress tests and consequent battery voltage drops.

battery) is deployed also in current version.

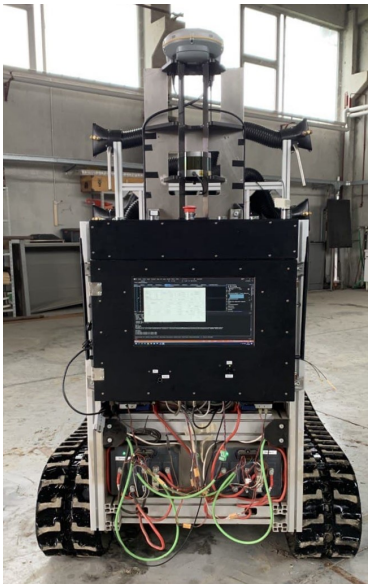
Rover Mark 3

The results of all the tests performed with Mark 2 suggested that the use of non-industrial electric/electronic components was not ensuring a reliable behavior of the system. Therefore, as extensively described in Chapter 2, the latest version of the robot is based onto industrial-standard components and communication protocols. In addition, the mechanical frame was then modified in order to host the custom-shaped water tank mentioned in Chapter 2.

Endurance tests in nominal working conditions In line with the analysis carried out in the previous paragraph, we performed tests about power consumption in nominal working conditions using the Mark 3 prototype (the latest version) shown in Figure 5.8. In fact, as shown by GPS logs in Figure 5.9a, we traveled among orchard rows, following an eight-shaped trajectory, in order to have the same number of left and right curves (i.e. stressing each motor homogeneously). It is also important to mention that we considered the worst-case scenario, in which the water tank is completely full, this allowed us to stress the vehicle during nominal working scenarios, carrying the maximum payload (total weight of the robot is about 950 kg). The overall duration of the test was about 40 minutes, during which the robot was driven at a constant linear speed along the orchard lanes (about 5 km/h) and circular trajectories (row-change maneuvers) with fixed radius and angular velocity.



(a)



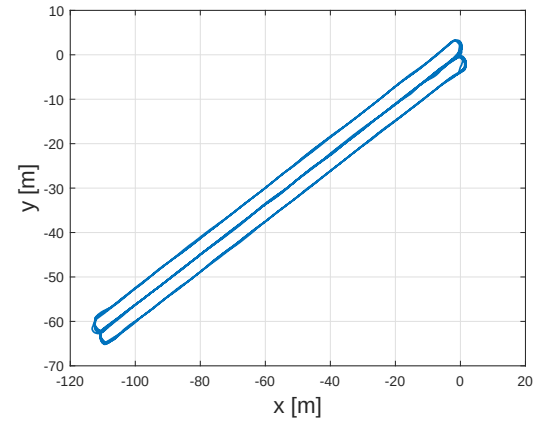
(b)

Figure 5.8: Rover Mark 3 (a.k.a Surus Prime), the latest and current version.

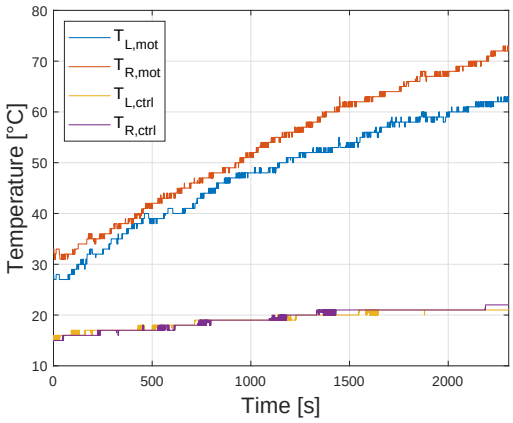
The first result we want to highlight is the fact that in this scenario the motors are used within their nominal capabilities. In fact Figure 5.9b shows that the mechanical power required is less than 1.5 kW per motor (nominal power of the motors), meaning that there is some margin for increasing further the robot speed during its tasks. The second important result comes from the data shown in Figure 5.9d, which represents the instantaneous and mean electric power required for our working conditions. This gives important data about robot autonomy, and in turn, about the sizing of the battery capacity.
Integrating over time the mean electric power consumption $\bar{P}_{el}$, it is possible to compute the energy consumed per hour and, in turn, determine the battery autonomy in terms of time. Let's consider the integration interval of one hour time, i.e. $t_0 = 0$, $t_f = 3600s$, then the time integral of $\bar{P}_{el}$ is

$$\int_{t_0}^{t_f} \bar{P}_{el}\,\mathrm{d}t = \int_{0}^{3600} 1765\,\mathrm{d}t = 6.355 \times 10^6\,J = 1.765\,kWh. \tag{5.1}$$

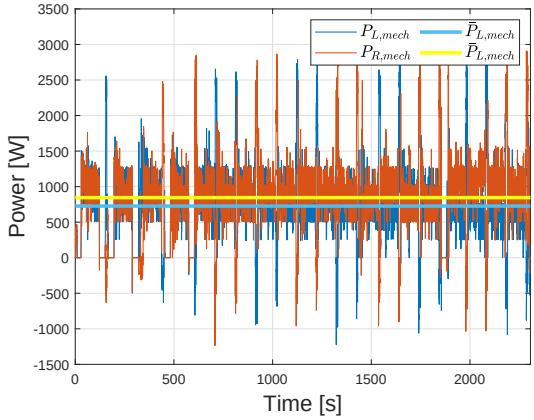
Then, considering the nominal capacity of the battery (6.3 KWh), we can conclude that the autonomy of the robot is expected to be about 3.5 hours. It is important to notice that this test was performed considering the worst case scenario in terms of robot weight, therefore the outcome of this analysis is inherently conservative, any-



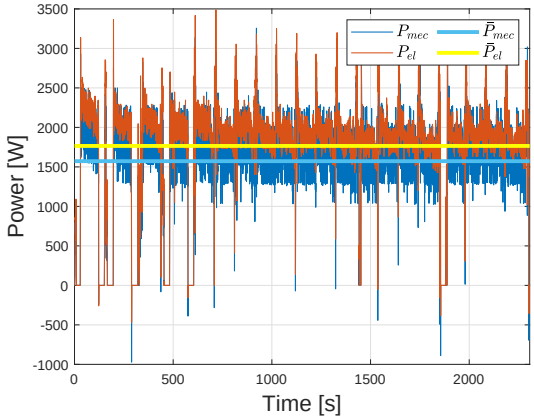
(a) GPS data recorded.



(c) Motors and controllers temperature.



(b) Motors mechanical power.



(d) Electrical and mechanical power.

Figure 5.9: Data collected during Nominal-working-conditions tests.

way it provides useful preliminary information about battery autonomy.
This discussion concludes the experimental results concerning mechatronics aspects of the design. In the next section, coming outs about autonomous navigation will be presented and discussed.

5.3 In-row navigation tests

The experimental results described in this section are meant to validate both the along-row and row-change navigation. Special focus will be given to the line estimation in each scenario. In particular, the reader is referred to this video[1] for specific

[1] https://youtu.be/wfRHI3WsAA0

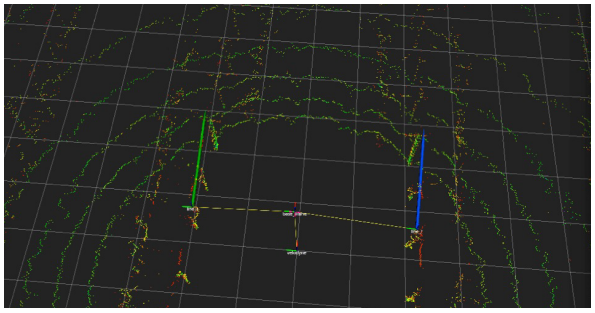
insights on the final results achieved, as well as to our works [48, 74]. It is also important to emphasize once again that the project is still ongoing and that part of the results shown in the following were achieve with the Mark 2, while some other, after the hardware and software upgrade, using the Mark 3. In fact, both pictures and videos reporting the tests results show both the two versions.

5.3.1 Along-row scenario

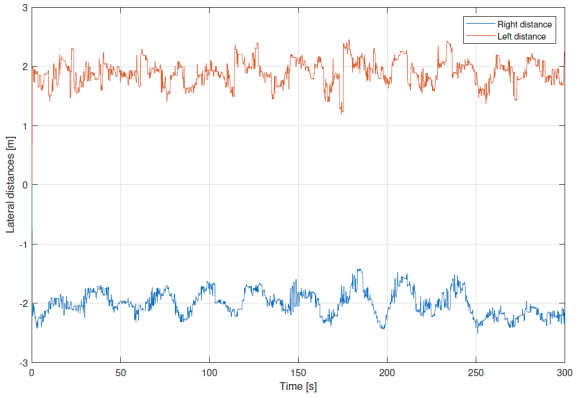
In these navigation tests, the robot starts at the beginning of one row inside the orchard, with the aim of performing autonomous navigation along rows, keeping the center line of the orchard lane, as described in Figure 5.10. The test finishes when the robot detects the end of the row, in compliance with the conditions described in Chapter 4.

Figure 5.10: Surus Prime during in-row navigation.

Figure 5.11a shows a screen of the onboard computer in which the real-time line estimates are displayed. Whereas, Figure 5.11b shows the lateral distances between the robot and the row lines estimated by the localization algorithm. The right distance is negative, accordingly to the body reference frame, where y_B-axis is pointing towards the left. The two estimated distances (d_L, d_R) feature a mean value of about 2 m which is constant during the entire row length, meaning that the desired lateral distance kept by the navigation system is actually correct. Moreover, the distances sum is about $4\,m$, which is consistent with the real orchard lane width.
It is important to mention that the oscillations in d_L, d_R are not only given by "un-



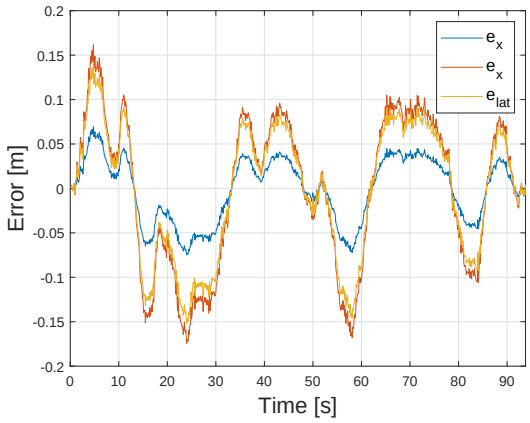
(a) Line estimation along-row.



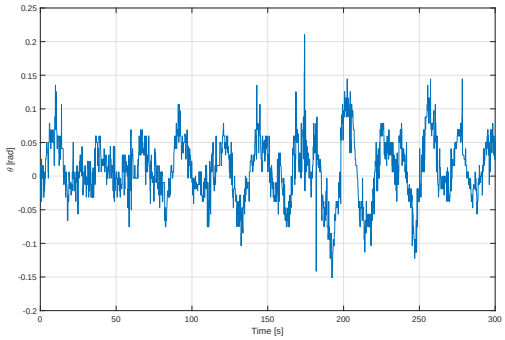
(b) Estimated later distance along-row.

Figure 5.11: Experimental results of the 2-line Hough Transform.

certainties" of the line estimation algorithm, rather it reflects *trees* and vegetation shape. Detecting foliage and possibly spurious invasive branches of vegetation, while performing autonomous navigation, is generally a good feature as the operation become aware of vegetation and branches that may protrude significantly from the main trunks.



(a) Positioning error.



(b) Heading error.

Figure 5.12: Experimental results of the 2-line Hough Transform in terms of navigation error.

Within a *sensor fusion* context, it is important to do not filter out meaningful oscillations, since they can represent protruding branches that the robot might try to avoid, therefore, the proper weights/values in the EKF covariance matrices have to be selected.

Figure 5.12a shows the position error computed by the control system with respect

to a straight constant speed trajectory along the orchard lane. It interesting to notice that the peaks present in the e_x signal correspond also to the peaks along the y direction. The reason for this is that the controller tends to increase the robot angular velocity to properly align with the reference trajectory, this results in temporary reduction of the linear body velocity, affecting the tracking along the x axis.
For evaluation sake, the error is computed with respect to the ideal center line resulting from interpolation of the GPS points corresponding to the entry and exit points of this lane (i.e. $m_i, m_{n+i} \in \mathbf{M_{wp}}$ as explained in Section 4.2.1). In fact, the reference trajectory is generated keeping the same lateral distance from each side of the robot, then the oscillations can either be generated by noise in line estimates or given by branch detection and avoidance. Nevertheless, the error is bounded and below 5% of the lane width. In addition, Figure 5.12b represents the heading angle with respect to the reference line orientation, from which it is possible to observe small peak values of the angle, about 0.1 rad (i.e. 5°), which is actually a very limited error.

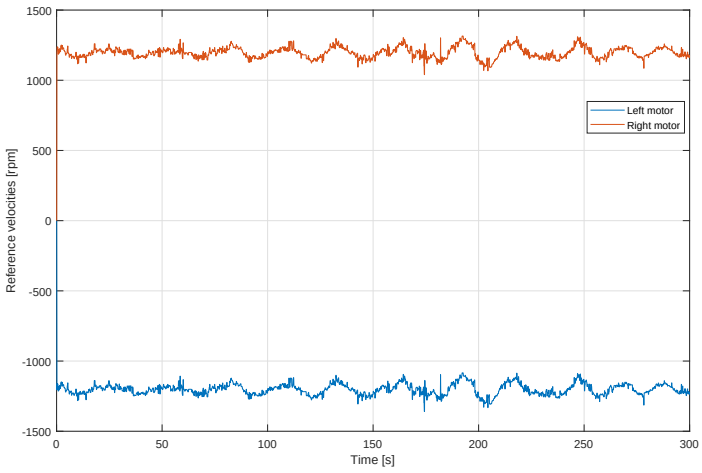


Figure 5.13: Motor reference velocities during autonomous navigation

Figure 5.13 shows the motor rotating speed during this test, notice that the resulting signals feature zero mean oscillations which is actually correct, indicating that the robot is navigating parallel to the rows, while the noise presence is due to all the factors described above. The overall result is a constant value of translation speed V_G, as it was highly expected due to the nature of the reference trajectory.

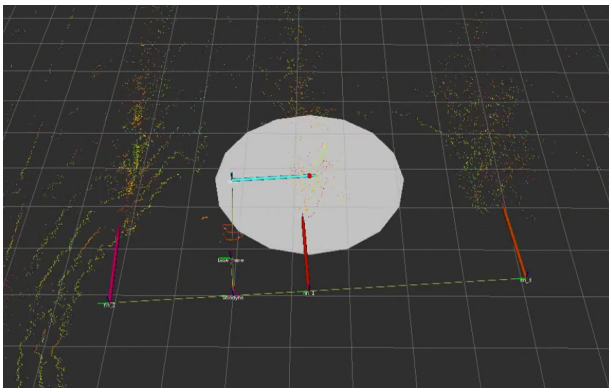
5.3.2 Row-change scenario

Position-based autonomous row change, have not been tested yet, on the other hand, we tested extensively the row-change navigation based only on the 4-line Hough Transform algorithm. For this reason, in the following we will present the results of the autonomous row change carried out using a velocity based controller, in which the only control input is the body angular velocity Ω_z, as V_G is kept constant. The input of this controller is the distance error between the desired turning radius, computed as described in Section 4.3.2, and the actual distance of the robot from the turning pivot point. The second part of the video[1] shows how the robot behaves during this maneuver.

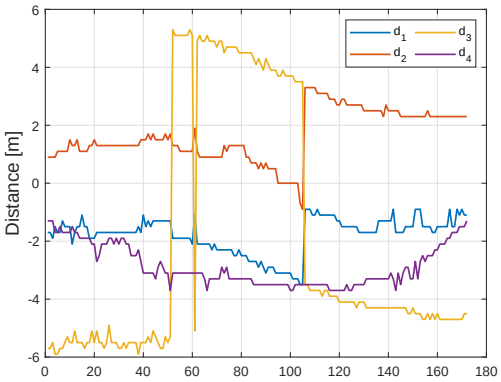
Figure 5.14: Image from a localization experiment, taken during a row change.

In order to give an idea about the real test conditions, Figure 5.14 shows a picture taken during some of the following experiments using Mark 3, in a moment after the robot exited the row and is in the middle of the row-change maneuver.
Figure 5.15a reports an *RVIZ* screenshot from the ROS environment, showing the estimates of the four lines detected during the current turning maneuver toward right: the three row lines, parallel to each other (purple, red and dark orange), plus the line orthogonal to the others (light blue). In the same screenshot it is also possible to identify a white circle, representing the ideal turning radius computed starting from the estimated lines, whose center (the red dot) is defined as explained in previous

[1] https://youtu.be/wfRHI3WsAA0?t=59



(a) Line estimation during row-change.



(b) Estimated line distances during row-change.

Figure 5.15: Experimental results of the 4-line Hough Transform.

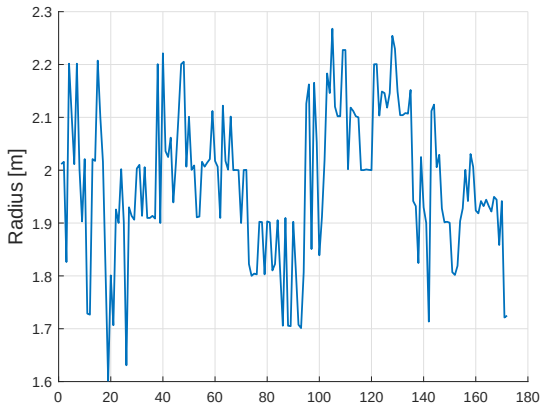
sections.

On the other hand, Figure 5.15b shows the relative distance of the robot from each of the four lines. It is interesting to notice how the distances between the robot and l_1, l_2, l_3 change. In particular, the lateral distances from l_1, l_2, respectively the blue and red line in Figure 5.15b, vary following a sinusoidal fashion, as expected by the circular trajectory imposed to the robot. In addition, it can be noticed that both at the beginning and at the very end of the maneuver the robot detects two lines at negative distance (i.e. on its right side), this verifies that the turn negotiated was toward the right hand side as also shown in the video and in Figure 5.15a. Moreover, it is interesting to observe that the colors assigned to the estimated lines are not casual. Throughout the execution of the maneuvers, the yellow-color is always assigned to the most distant row, namely, the one belonging to the lane the robot is aiming to enter and then the one modeling the lane the vehicle was coming from. This validate the algorithm capability to *label* each line, allowing us to better exploit the search-space reduction paradigm introduced in Section 4.3.2.

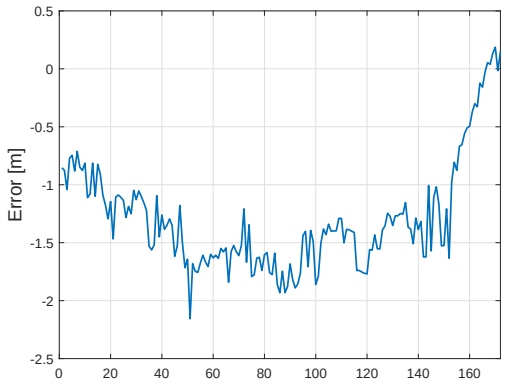
In fact, we want to highlight that in these example the reduced search space is playing a fundamental role. Up to a point in which without it, the standard algorithm could not estimate reliably these four lines, in certain cases represented by a few points in the pointcloud.

Now, we want to discuss further about estimation of the turning radius and position of the pivoting point, as it is used in position-based control for circular trajectory

planning purposes, and in this preliminary (velocity) controller directly to define the distance error.



(a) Turning radius estimate.



(b) Position error with respect to pivoting point.

Figure 5.16: Experimental results of the 4-line Hough Transform.

In line with this, Figure 5.16a shows the resultant values of the turning radius computed by the algorithm, and it is worthy to notice that the mean value of the radius estimates is about 1.95 m, really consistent with the row width[1].
In the end, we want to point out the error signal used to feed the velocity-based controller, reported in Figure 5.16b. In particular, that the initial values is quite big as, in order to detect the end of the row, the robot has to exit the current orchard lane of about one meter, then the row-change controller starts and it requires the error to increase almost up to 2 meters before becoming strong enough to steer the error to zero, this would require more fine tuning of the controller, but we didn't do it as we started the development of Mark 3 already with the idea of moving to position-based control. Autonomous navigation experiments are still ongoing. In fact, after the mechatronic refactoring of the Mark 3, and the implementation of the new pose estimation and control algorithms, we are still collecting data in order to present fully autonomous navigation of the robot.

[1]Ideally, the circular trajectory should have the diameter described by two adjacent $m_i, m_{i+1} \in \mathbf{M_{wp}}$ points, and its center in the $O_i \in O_{wp}$ point lying in between of the latter to points.

5.4 Gazebo simulator

In parallel to mechatronics upgrade between Mark 2 and Mark 3, we also started the development of a simulation environment aimed at recreating the farming context the robot will encounter during its real-life tasks. It goes without saying that the main intent of this simulator is to speedup the debugging process of navigation and localization algorithms. In addition, we decided to move from MATLAB simulation environment to a 3D one, such as Gazebo, [107], in order to reduce even more the gap between simulation and real world, since Gazebo allows an higher level of simulation for outdoor environments. Figure 5.17 shows the main features of the environment generated.

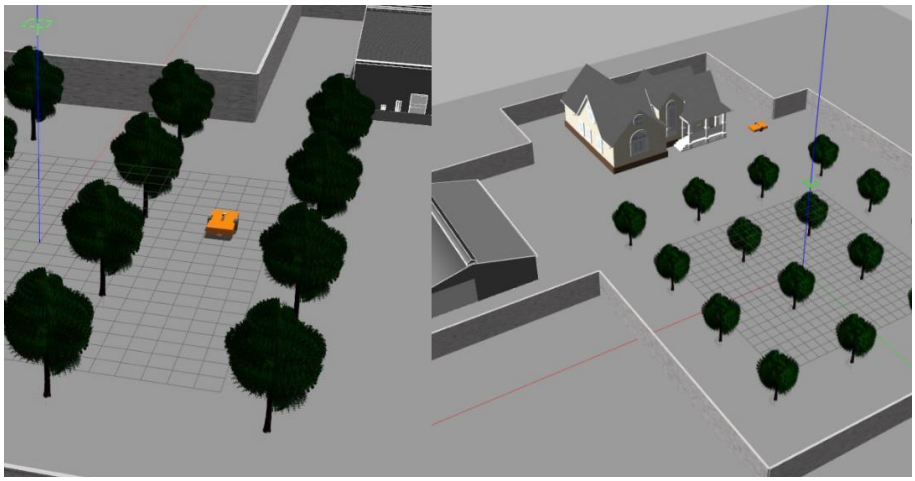


Figure 5.17: Gazebo simulation showing the robotic platform inside a mockup virtual orchard

Indeed, as shown in the video[1], we tested open-field navigation within this environment and then moved it to the real platform. We will not discuss any further about this navigation algorithm as it mainly relies on open source ROS packages, and we already described in details their customization for our application in Section 4.2.1. In addition, reliability tests are still ongoing, therefore we have no additional data to show.
At last, Figure 5.18 shows line estimation algorithm tested within the virtual orchard lane.

[1] https://youtu.be/wfRHI3WsAA0?t=14

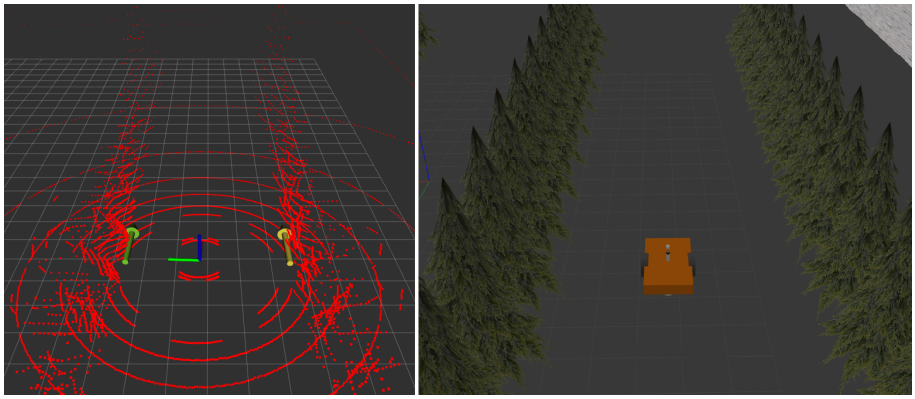


Figure 5.18: Gazebo simulation showing the robot inside orchard lane estimating lines.

5.5 Conclusions

In this chapter, we presented the experimental activities performed on-field, as well as the virtual environment created on Gazebo. The tests described have been organized in order to get data and validate all the robot features or components. The outcomes are quite positive, both for what concerns the mechatronic subsystem and the autonomous navigation algorithms. In fact, the results show that the mechatronic design of the robot is suited for the irregular and rough environment typical of agricultural context, while managing an onboard tool, as well as to properly accomplish orchard farming tasks. For what concerns autonomous navigation, tests demonstrate the capabilities of the robot to localize and drive itself both outside and inside orchard rows. Nevertheless, further systematic tests are yet required to validate each navigation scenario described in Chapter 4, hence we will discuss the current and future steps and developments in Chapter 6.

6

Conclusions and Future Directions

This chapter concludes the dissertation, resuming the results achieved during the PhD research activities. In the end, new possible challenges and improvements for the project are presented and discussed.

This thesis presented the most relevant aspects of the design and implementation of a multi-purpose robot for precision agriculture in orchards and greenhouses. We described the main design concepts pursued during the development, focusing in particular on the advantages presented by our solution, in terms of **soil compaction, flexibility and modularity**.
The designed robot is capable of performing autonomously farming tasks accordingly to the tool equipped, as well as supporting human operators. The insights provided about the mechatronic design, allowed us to better highlight the advantages brought in by integrating the implements directly into the robot frame, if compared to standard tractors carrying or pulling tools. In Chapter 2, we also described how the user can remotely control the robot, or define new missions thanks to the Human Machine Interface developed.

Then, in Chapter 3, the discussion continued with an analysis of the kinematic and dynamic behavior of tracked vehicle, with particular emphasis on the influence of skidding effects on the torque required to negotiate circular trajectories. Based on

this, we derived a modeling tool capable of relating motor torque (or track force) with trajectory and design parameters, allowing one to evaluate design choices with the awareness of their effects on power consumption. We also enriched the state-of-the-art dynamic modeling, proposing a **novel simplified pressure distribution function**, describing with higher accuracy the effects of normal weight on the ground underneath the tracks. This part was then concluded with the report of the experimental results, comparing real data collected during turning maneuvers with the estimates provided by the proposed models. The two quantities fitted with good accuracy, even though in some cases the poor Signal-to-Noise-Ratio produced high RMSEs, nevertheless, when data mean values were considered, they resulted only 5-10% away from the estimated values.

In the first part of Chapter 4, we discussed our solution for localization within agricultural environments, reporting the structure of the sensor fusion algorithm, highlighting its variable sensors set, namely the EKF relies on different sensor sets depending on the navigation scenarios (i.e. open-field or in-row). We also presented the navigation stack, endowing the UGV with the capability of autonomously navigating from a resting position to the orchard and then to perform its mission inside it. Then we remarked how the **structured property of orchards can improve in-row navigation**, thanks to relative position estimation provided by the improved Hough Transform line recognition.
In the second part of Chapter 4, we presented an **adaptive observer for slip estimation for tracked skid-steering vehicles**. We formally proved uniform global exponential stability of the error system, resulting in robustness with respect to additive perturbations in terms of input-to-state stability. Through two different experimental tests, straight and turning motion, we validated the claims obtained via analytical methods, showing the good performance of the proposed approach. Then we used this estimator in synergy with a standard position controller for slip-free differential robots, compensating the control actions accordingly to the slip estimates.

Chapter 5 reports the **on-field experimental tests** performed with our robot. From them, the first result obtained is the validation of the whole mechatronic design: the robot behaved properly, showing that the communication glitches that were present in the previous versions have been solved, thanks to the adoption of industrial standard components and communication protocols. Moreover, the experimental tests

validated the autonomous navigation paradigm proposed, showing good robot performance in terms of trajectory tracking during in-row navigation. Even though for open-field scenario we have only simulation results they are promising as well, as the Gazebo simulations run the same software structure running on the real robot.

One final remark, assembling and designing a complex system such as an autonomous robot, which involves the integration of several components from different domains (i.e. mechanics, electric-electronics and computer science) requires the acquisition of many skills and specific know-how which in most of the cases cannot be reached by theoretical studies. Even if it cannot be estimated or plotted, I consider this knowledge to be one of the most important results achieved during the development of this project.

6.1 Future directions

As mentioned throughout the thesis discussion, experimental tests are currently ongoing. It then follows that the very next steps will be the real implementation of **open-field navigation**, as well as completing one global mission, starting from the base station, reaching the field, performing the mission task, and returning home. In parallel, we also intend to extend the dynamic modeling to **sloped-ground fields**.

A part from these really-short-term objectives, there are plenty of possibilities in this fascinating research field, from the addition of new tools in order to increase the versatility of the robot, to the development of obstacle detection and recognition algorithms, to endow the robot with the capability of distinguishing the farmer who has to be followed from the bins that has to be avoided.
With the arrival of new battery technologies, the idea of a **full-electric robot** is no longer unrealistic. Indeed, the design of a new robot, offers the perfect chance to directly test the model-based design tool developed. Allowing us to take also advantage from the actual prototype flaws, in particular it showed some limitations while moving at high speed, in fact, vibrations transmitted to the main body due to the irregular terrain are not beneficial for the onboard equipment. Then, a **suspension system** aimed at dampening these vibrations will be one of the *must-have* of a future prototype. Furthermore, the prospect of having more than one robot available opens many other research topics related to **cooperative and distributed robotics**,

increasing even more the appeal and effectiveness of such robots.

Appendix

A

Dynamic Analysis

In this appendix we report the integral calculation of track inertia which is not considered in Chapter 3, as the analysis is limited to the steady state case. As well as the complete version of the analysis carried out in Section 3.3, what follows is mostly taken from [1].

A.1 Motor Dynamics

In a similar way of the two-wheeled robot case, the overall dynamics of tracked vehicles depends on the thrust generated by the two motors connected to each of the two actuated wheels, as shown in Figure A.2. Therefore, to infer the dynamic equations of the whole vehicle body it is required to analyze the forces and torques acting on each of the two motors.

Considering the Drive-chain of the track (either the left or the right one), three basic

Figure A.1: Skid-steering Dual-Drive tracked vehicle drive chain.

elements can be identified and modeled:

- ***Motor*** **-** it generates the torque τ_{m} which balances the resisting torques and accelerating the track connected to the motor sprocket;
- ***Gearbox*** **-** it reduces the turning speed and increasing the available torque, the gear ratio k_{τ} is chosen depending on the motor and the payload of the application;
- ***Equivalent Inertia*** **-** it models the whole moving mass and turning parts, seen from each of the two motors, as an equivalent inertia which opposes to the acceleration imposed by the two motors;
- ***Resistive Forces*** τ_{r} **-** it takes into account the interaction between tracks and ground, as well as the interaction between different mechanical parts of the body and other elements.

Thus the following equation properly represents the dynamics of a generic motor with payload attached

$$J_{eq}\dot{\omega} = \tau_m - \tau_r. \tag{A.1}$$

A.1.1 Equivalent Inertia

As previously mentioned, to estimate the forces acting on the motor a fundamental step is given by the computation of the inertia "sensed" by the latter. In general, to perform inertia equivalence two different ways can be followed:

1. *Geometric Approach* - Considering the geometric relations between the mass connected to the motor and the motor rotational axis, the masses and inertia of the body can be expressed as an equivalent inertia. Even if effective this procedure requires a high computational burden and it is not recommended for complex bodies.

2. *Kinetic Energy approach* - For each mass/inertia component of the body the corresponding kinetic energy is computed, then all these energy components are summed up to compose the overall kinetic energy of the body. This procedures results in an equation relating all inertial elements to an equivalent one, with the expression of the velocities of each body object as function of the velocity of the motor considered. This approach is faster than the previous one but it requires the possibility of expressing the velocities of the body components in terms of some basic speeds.

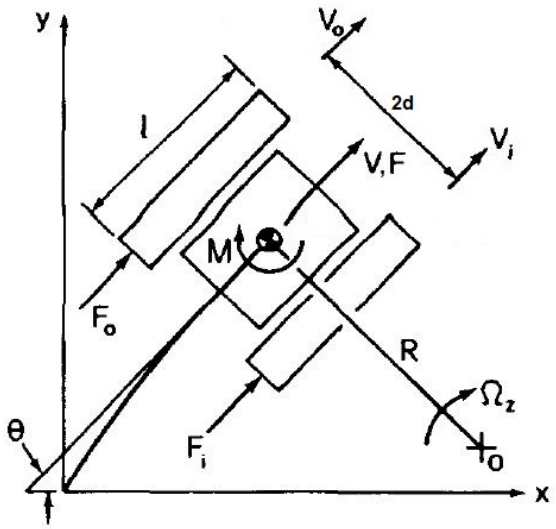


Figure A.2: Velocities and forces acting on a Skid-steering tracked vehicle, [1].

As shown in Figure A.2, forces and velocities feature two different subscripts: F_o, v_o and F_i, v_i, referring respectively to forces and velocities of the Outer and Inner track. The position described by the subscripts is referred to the turning motion, indicating therefore which is further from the *Instantaneous Center of Rotation (ICR)* and which one is closer.
Carrying out the dynamic analysis of a tracked vehicle, two different situations can be identified:

- Straight motion
- Turning motion

During the development of the dynamics study, particular focus will be given to turning scenario, due to the fact that straight path can be considered as a restriction of the previous scenario.
Let's now consider the above-listed elements composing the (A.1). In developing the mechanical model, the computation of the equivalent inertia of the bodies and masses linked to the motor has been performed by means of the Kinetic Energy approach.

Motor load Equivalent Inertia

The total inertia opposing to the motor rotation is composed by all those elements that are attached to the motor shaft or directly connected to the sprocket:

1. Motor shaft inertia moment
2. Gearbox inertia moment
3. Sprockets inertia moments
4. Inertia moment of the track

$$J_{mot,tot} = J_{mot} + J_{gearbox} + 2J_{sprocket} + J_{eq,track} \tag{A.2}$$

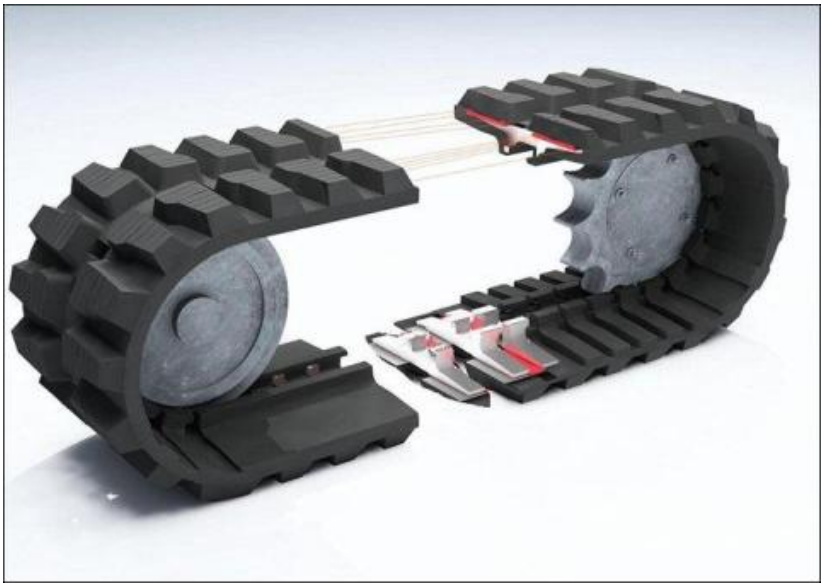


Figure A.3: Track section and sprocket.

All inertial components in (A.2) come directly from datasheets except for J_{track}, therefore it has to be computed and to do this the kinetic energy of the track will be considered. Assume to split the whole track into infinitesimal elements and to assign to each of those their respective velocity.
In addition, looking at the geometric structure of the track, four different regions/-sections can be identified, shown in Figure A.4, and these regions contain elements with the same velocity:

(1) Semi-circular section of the track

(2) Straight section of the track

(3) Semi-circular section of the track

(4) Straight section of the track [1]

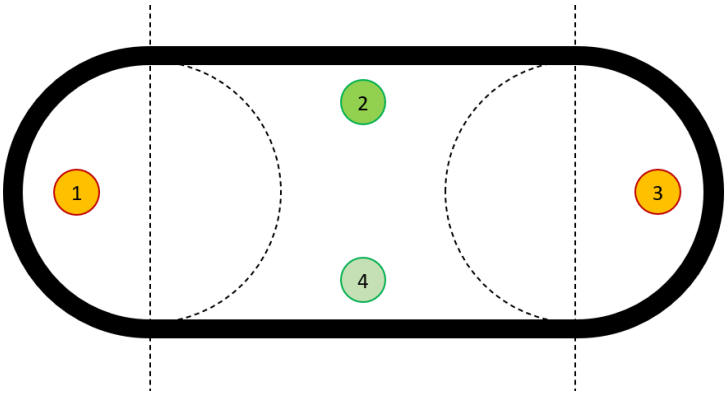


Figure A.4: Velocity sections of the track.

As previously said, the kinetic energy of each infinitesimal element will be computed, but to perform this the mass of each of those "particles" has to be identified; assume therefore to have the mass of the whole track m_{track} described by an uniform volumetric density ρ

$$\rho = \frac{m_{track}}{V_{track}}, \tag{A.3}$$

where V_{track} is the overall volume of the track, that can be computed as

$$V_{track} = V_1 + V_2 + V_3 + V_4 \tag{A.4}$$

where V_i is the volume corresponding to the i-*th* section of the track, with $i \in \{1,2,3,4\}$. Those volumes can be expressed as function of the parameters reported in Figure A.5, where L is the length of the track-section in contact with the ground or anyway the straight part of it, R is the radius of the sprocket, W the width of the track and T is the thickness of the track

$$V_1 = V_3 = \frac{\pi}{2}\left[(R+T)^2 - R^2\right]W \quad \textit{and} \quad V_2 = V_4 = TLW, \tag{A.5}$$

[1] Notice that, the velocity of this part of the track (v_4) is zero with respect to a frame fixed to the ground, but in carrying out the computation of kinetic energy the velocity of the different parts of the track has to be considered with respect to a frame fixed to the motor, therefore $v_4 = v_2 \neq 0$.

leading to

$$V_{track} = 2V_1 + 2V_2 = \left[(2TL) + \left(\pi\left((R+T)^2 - R^2\right)\right)\right]W. \tag{A.6}$$

From the above result, the density can be expressed as

$$\rho = \frac{m_{track}}{\left[(2TL) + (\pi((R+T)^2 - R^2))\right]W} \quad \Rightarrow \quad dm = \rho dV \tag{A.7}$$

where dm and dV are respectively the mass and volume of the infinitesimal element considered.
Having those two quantities it becomes possible to express the kinetic energy of a generic infinitesimal element

$$dK = \frac{1}{2}v^2 dm \tag{A.8}$$

where v represents the velocity of the infinitesimal element of the track.

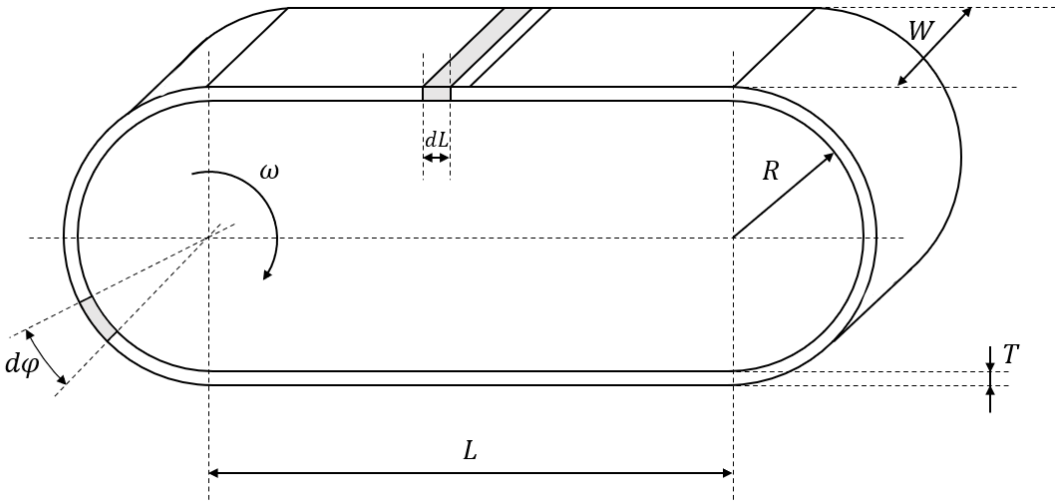


Figure A.5: Integration sections of the track.

As depicted in Figure A.4, it is possible to identify 4 different regions in which elements are defined by the same velocity, namely

$$v_2 = v_4 = \omega R \qquad \textit{and} \qquad \omega_1 = \omega_3 = \omega \tag{A.9}$$

where ω is the turning velocity of the motor; notice that the velocities related to sections 1 and 3 are considered as angular velocities, because their tangential velocities

v depend on the distance from the rotating center r.
The kinetic energy the whole region can be computed through the integral of the kinetic energy of each single mass element, over the whole the mass of the specific region

$$\begin{aligned} K_2 = K_4 = \frac{1}{2}v^2 \int_{m_2} dm = \frac{1}{2}(\omega R)^2 \int_{m_2} dm = \frac{1}{2}\omega^2 R^2 \rho \int_{V_2} dV = \\ = \frac{1}{2}\omega^2 R^2 \rho \int_0^L TW dl = \frac{1}{2}\rho R^2 TWL\omega^2 \end{aligned} \tag{A.10}$$

$$\begin{aligned} K_1 = K_3 = \frac{1}{2}v^2 \int_{m_2} dm = \frac{1}{2}(\omega r)^2 \int_{m_1} dm = \frac{1}{2}\omega^2 \rho \int_{V_1} dV = \\ = \frac{1}{2}\omega^2 R^2 \rho \int_0^\pi d\varphi \int_0^W dh \int_R^{R+T} rr^2 dr = \frac{1}{2}\omega^2 R^2 \rho \pi W \frac{\left[(R+T)^4 - R^4\right]}{4} = \\ = \frac{1}{2}\frac{\pi\rho W}{4}\left[(R+T)^4 - R^4\right]\omega^2 \end{aligned} \tag{A.11}$$

Considering the (A.10), (A.11), the overall kinematic energy can be derived as

$$K_1 + K_2 + K_3 + K_4 = \frac{1}{2}J_{eq,track}\omega^2 \tag{A.12}$$

which can be expanded as

$$\begin{aligned} \frac{1}{2} \quad \frac{\pi\rho W}{4}\left[(R+T)^4 - R^4\right] \quad \omega^2 \quad + \quad \frac{1}{2} \quad \rho R^2 TWL \quad \omega^2 + \\ + \quad \frac{1}{2} \quad \frac{\pi\rho W}{4}\left[(R+T)^4 - R^4\right] \quad \omega^2 \quad + \quad \frac{1}{2} \quad \rho R^2 TWL \quad \omega^2 = \\ = \quad \frac{1}{2} \quad J_{eq,track} \quad \omega^2. \end{aligned} \tag{A.13}$$

Then, the (A.13) can be rewritten in a more compact form as

$$\frac{1}{2} \quad \left\{ 2\left(\frac{\pi\rho W}{4}\left[(R+T)^4 - R^4\right]\right) + 2\rho R^2 TWL \right\} \quad \omega^2 = \quad \frac{1}{2} \quad J_{eq,track} \quad \omega^2, \tag{A.14}$$

from which it folloes

$$J_{eq,track} = 2\left(\frac{\pi\rho W}{4}\left[(R+T)^4 - R^4\right]\right) + 2\rho R^2 TWL = \\ = \rho W\left(\frac{\pi}{2}\left[(R+T)^4 - R^4\right] + 2R^2 TL\right). \tag{A.15}$$

Substituting the expression of the density reported in (A.7) into the previous equation, the final expression of the equivalent inertia can be derived as

$$J_{eq,track} = m_{track}\frac{\left(\frac{\pi}{2}\left[(R+T)^4 - R^4\right] + 2TLR^2\right)}{\left[(\pi((R+T)^2 - R^2)) + (2TL)\right]}. \tag{A.16}$$

Optional farming tools and additional masses

Since the mobile robot introduced at the beginning of this thesis is designed to work and operate within farming environments, it will be required to be equipped with different types of working tools designed to carry out specific purposes such as mowing the orchard grass, spraying the plants and the like.
From a mechanical point of view those additional equipment can be considered as extra masses and elements modifying the inertia moments of the vehicle, since they are usually rigidly connected to the main vehicle body. These new terms can be added, for sake of completeness and generality, to the inertia expression as general mass and inertia terms that can even result to be time varying, producing:

- **mass variation and time-varying[1] mass**
- **possible COG shift** due to the variation of mass in some part of the vehicle body
- **variation of the inertia tensor**

Let thus define

$$M(t) = m + m_{tool}(t) \quad \textit{and} \quad \mathbf{J_{new}}(t) = \begin{bmatrix} J_{new,x}(t) & J_{new,xy}(t) & J_{new,xz}(t) \\ J_{new,xy}(t) & J_{new,y}(t) & J_{new,yz}(t) \\ J_{new,xz}(t) & J_{new,yz}(t) & J_{new,z}(t) \end{bmatrix} \tag{A.17}$$

[1]The mass can change with time passing by due to the releasing of part of the "body matter", *(e.g. during spraying, the tank containing the fluid will empty)*

where $m_{tool}(t)$ is the (possibly time varying) mass of the farming tool, $\mathbf{J_{new}}(t)$ represents the overall inertia matrix of the vehicle equipped with an implement, whose main and cross elements represent the inertia terms of the vehicle body.

A.2 Resistive and driving forces [1]

As already mentioned, the motion of tracked Skid Steering Vehicles (SSVs) depends on the thrusts generated by the two tracks. In particular, we call F_o the force generated by the track placed furthest from the center of rotation (outer) and F_i the one closest to it (inner). F_o and F_i have to counterbalance the resultant resisting force R_{tot}, the moment of turning resistance M_r exerted by the ground on the track.
For the sake of clear exposition we will consider the problem of describing SSVs dynamics with an incremental approach. Therefore, we will neglect the centrifugal force for the time being, then the behavior of the vehicle can be described as follows

$$m\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} = m\frac{\mathrm{d}V_G}{\mathrm{d}t} = F_o + F_i - R_{tot} \tag{A.18}$$

$$J_z\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = J_z\frac{\mathrm{d}\Omega_z}{\mathrm{d}t} = (F_o - F_i)d - M_r \tag{A.19}$$

where s is the translational displacement of the Center of Gravity (COG) of the vehicle, θ the angular displacement of the vehicle, V_G the linear velocity of the COG, Ω_z the angular velocity about the axis normal to the 2D motion plane and d is half the distance between the center lines of the two tracks. Considering a steady-state conditions, linear and angular acceleration are no longer present (hence the track forces only have to balance the resistive forces)

$$F_o + F_i - R_{tot} = 0 \qquad \text{and} \qquad (F_o - F_i)d - M_r = 0 \tag{A.20}$$

From the (A.20), the forces acting either on the outer or on the inner track can be expressed:

$$F_o = \frac{R_{tot}}{2} + \frac{M_r}{2d} = \frac{f_r mg}{2} + \beta\omega_o + \frac{M_r}{2d} \tag{A.21}$$

$$F_i = \frac{R_{tot}}{2} - \frac{M_r}{2d} = \frac{f_r mg}{2} + \beta\omega_i - \frac{M_r}{2d} \tag{A.22}$$

where f_r is the rolling friction coefficient describing the resistance with respect to rolling motion of the tracks along the longitudinal direction, β is the viscous friction coefficient that is due to the damping effects of the mechanisms connected to the motors and g represents the gravitational acceleration.
As previously discussed, M_r represents the resistive turning moment and it is given by the track shearing of the ground during skid-steering, therefore, this term can be considered as dependent on (i) the footprint of the track on the ground, (ii) the contact pressure distribution and in turns on the weight of the mobile robot and (iii) terrain parameters. Furthermore, if the normal pressure is uniformly distributed along the track, the lateral resistance per unit length R_L of the track can be expressed as

$$R_L = \mu_t \frac{mg}{2L} \tag{A.23}$$

where μ_t is the coefficient of lateral resistance and L is the contact length of each of the two tracks.
Here above, it has been mentioned that the resistive moment M_r depends also on the features of the terrain traveled by the tracks (this dependency on soil characteristics is embodied within μ_t) as well as on the tracks layout. The lateral forces acting on tracks may also depend on the turning radius (this dependency will be further investigated in Section A.3), by the way, for the time being μ_t will be considered as constant.

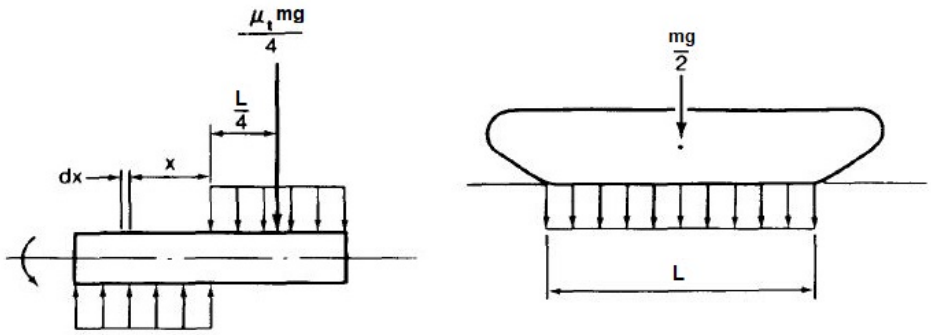


Figure A.6: Lateral resistive forces and normal pressure distribution, [1].

Figure A.6 highlights that the resultant moment of the lateral resistance about the centers of the two tracks can be expressed as

$$M_r = 4\frac{mg\mu_t}{2L}\int_0^{L/2} x dx = \mu_t \frac{mgL}{4}. \tag{A.24}$$

Therefore, (A.21) and (A.22) can be now rewritten as

$$F_o = \frac{f_r mg}{2} + \beta\omega_o + \frac{\mu_t mgL}{4 \cdot 2d} \quad \approx \quad \frac{mg}{2}\left(f_r + \frac{\mu_t L}{4d}\right) \tag{A.25}$$

$$F_i = \frac{f_r mg}{2} + \beta\omega_i - \frac{\mu_t mgL}{4 \cdot 2d} \quad \approx \quad \frac{mg}{2}\left(f_r - \frac{\mu_t L}{4d}\right) \tag{A.26}$$

The parts within brackets in (A.26) points out the fact that if $f_r < \mu_t L/4d$ the force required by the inner motor (F_i) will be negative, that means that a braking effect is required to obtain the desired turning motion. Notice that, in (A.25) and (A.26) we dropped the viscous friction terms, as they are not relevant for the purpose of this analysis. We are going to add them only in the equations at the end of each section.

A.2.1 Centrifugal forces

During turning motion, centrifugal forces should be taken into consideration, not due to the high velocity at which the turn is accomplished, but due to the high mass and weight characterizing the vehicle.

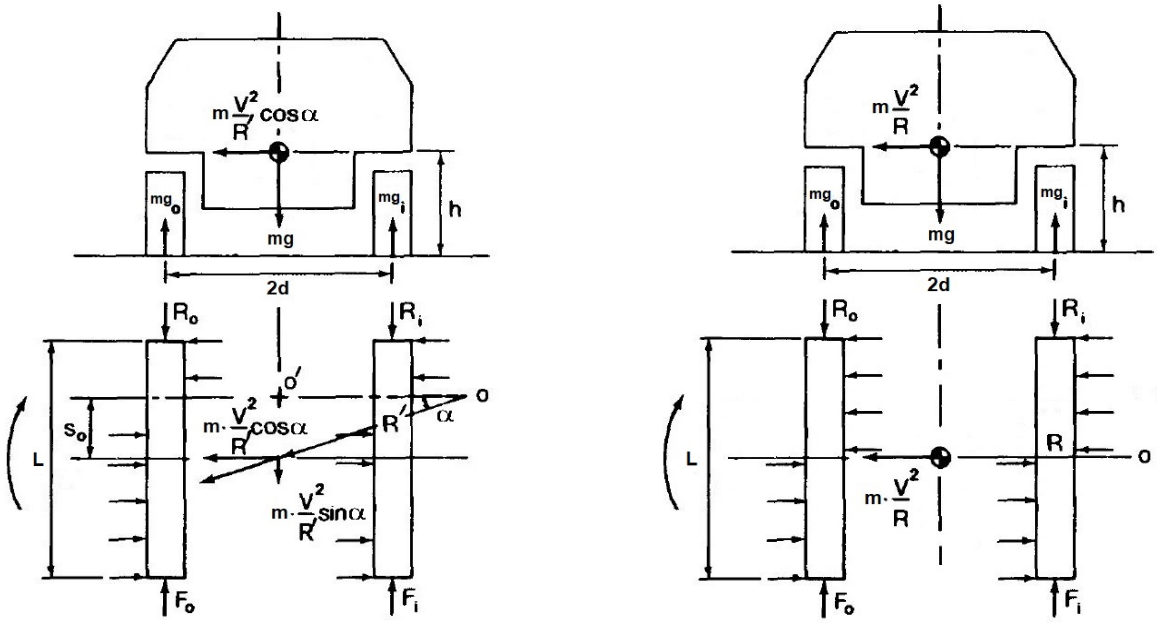


Figure A.7: Centrifugal forces acting on the vehicle.

Considering planar steady-state turning, to achieve equilibrium in the lateral direction, the resultant lateral friction force exerted by the ground on the track must be

equal to the centrifugal forces, as shown in Figure A.7.
In fact, Figure A.7 shows that if uniform normal pressure distribution is considered along the track length, and the lateral friction coefficient μ_t is considered constant, to satisfy the equilibrium condition on lateral direction the ICR must be shifted by a distance s_0 in front of the cross center line. Then s_0 can be computed from (A.20) adding the centrifugal force as

$$\begin{gathered}\left(\frac{L}{2}+s_0\right)\frac{\mu_t mg}{L}-\left(\frac{L}{2}-s_0\right)\frac{\mu_t mg}{L}=m\frac{V_G^2}{R'}\cos\alpha \\ s_0=\frac{LV_G^2}{2\mu_t gR'}\cos\alpha=\frac{La_y}{2\mu_t g}\cos\alpha\end{gathered} \tag{A.27}$$

where R' is the resultant turning radius considering the shifted ICR, and a_y is the lateral acceleration of the COG of the vehicle, $a_y = V_G^2/R'$. In addition, α represents the lateral slipping angle, and since it is mostly given by the magnitude of V_G, we can assume α to be almost zero, which leads to

$$\alpha \approx 0 \quad \Rightarrow \quad \cos\alpha \approx 1 \quad \text{and} \quad \sin\alpha \approx 0. \tag{A.28}$$

Therefore, the centrifugal forces reduces to the lateral component only, leading to the force diagram depicted in Figure A.7. The effects of centrifugal forces become relevant when the weight (or normal pressure distribution) is considered, indeed the force $ma_y = mV_G^2/R'$ is directed toward one side of the vehicle hull, and it generates a moment with respect to the ground plane, this torques is defined by the force itself and by the distance between the COG and the terrain (since the force is considered to be applied directly on the COG)

$$\tau_{a_y} = ma_y \times h = ma_y h. \tag{A.29}$$

Where h is the distance of the *COG* from the ground plane, and with $\times$ we indicate the cross product between vectors. As previously said, the presence of this torque will affect the normal pressure on the two tracks since this torque generates two forces equal in magnitude but opposed in direction, acting on the two crawlers, as shown in Figure A.8. In fact, the forces $F_{a_y,o}$ and $F_{a_y,i}$ can be expressed as:

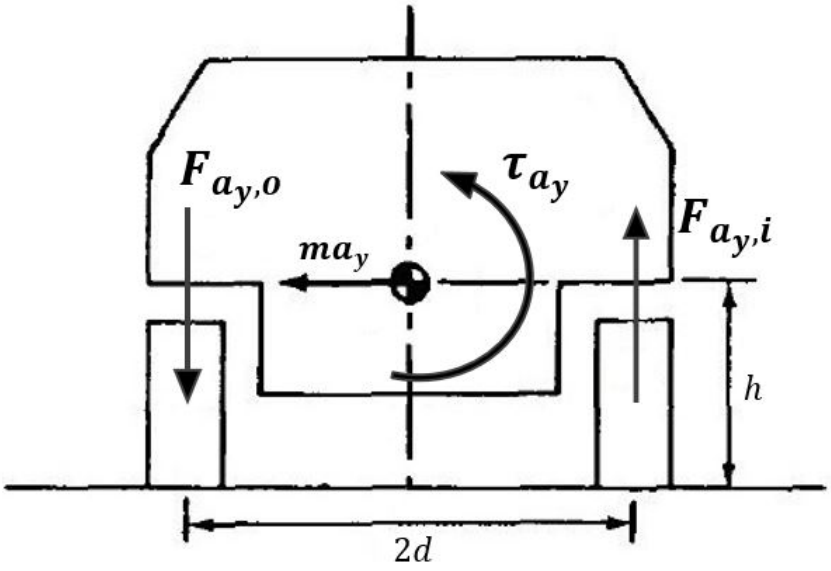


Figure A.8: Centrifugal torque and forces affecting the normal pressure.

$$\begin{aligned} F_{a_y,o} &= \frac{\tau_{a_y}}{2d} = \frac{ma_y h}{2d} = h\frac{mV_G^2}{2dR'} \\ F_{a_y,i} &= -\frac{\tau_{a_y}}{2d} = -\frac{ma_y h}{2d} = -h\frac{mV_G^2}{2dR'}. \end{aligned} \tag{A.30}$$

Therefore, (A.21) and (A.22) can be rewritten in order to embed equations in (A.30), achieving a final formulation of F_o and F_i taking into consideration both resisting and centrifugal forces:

$$F_o = \left(\frac{mg}{2} + h\frac{mV_G^2}{2dR'}\right) f_r + \beta\omega_o + \mu_t \frac{mgL}{4\cdot 2d} = \left(\frac{mg}{2} + h\frac{ma_y h}{2d}\right) f_r + \beta\omega_o + \mu_t \frac{mgL}{4\cdot 2d} \tag{A.31}$$

$$F_i = \left(\frac{mg}{2} - h\frac{mV_G^2}{2dR'}\right) f_r + \beta\omega_i - \mu_t \frac{mgL}{4\cdot 2d} = \left(\frac{mg}{2} - h\frac{ma_y h}{2d}\right) f_r + \beta\omega_i - \mu_t \frac{mgL}{4\cdot 2d}. \tag{A.32}$$

A.3 Lateral Friction Coefficient

If the equations obtained for motors required forces (A.25) and (A.26) are carefully observed it can be seen that there is no relation between the force required to move the vehicle and the turning radius R' or the velocities of the two tracks, leading to a constant value of motor torque independently on the turning radius or speeds. Nev-

ertheless, experimental observations showed a strong dependency of F_o and F_i on the turning radius R', therefore it is natural to look for a dependency of the force equation to turning radius and track velocities. As we are mainly focused on the effect of the turning radius on M_r, we assume to keep V_G small enough to neglect the centrifugal forces. This allows us to better focus on the dependency of F_o, F_i on R', keeping the equations lighter to be read.

The analysis carried out so far has been developed for the first time by Steeds [66], but he has made some strong assumptions that make the model less realistic, in particular Steed considers that the shear stress developed on the track-ground interface is obeying the Coulomb law of friction. This friction law implies that the resultant shear stress on a track elements acts in a direction opposite to that of the relative motion between the track element and the ground. It also assumes that the shear stress reaches its maximum value instantly, as soon as a small relative movement between the crawler and the terrain is initiated; by the way experimental tests shows that the shear stress developed on the track-ground interface is dependent on the shear displacement, as stated and developed by Wong in [68] and [1].
Following Wong's analysis, the relation between motor forces and turning radius is derived; in particular, the term that has been identified to take care of this dependency is the moment of lateral resistance, which will keep the same structure, but the lateral friction coefficient will become variable rather than fixed, depending on the aforementioned parameters

$$\begin{aligned} M_r = \mu_t \frac{mgL}{4} \quad &\longrightarrow \quad M_r(R', \omega_o, \omega_i) = \mu_t(R', \omega_o, \omega_i)\frac{mgL}{4} \\ &\iff \mu_t = \mu_t(R', \omega_o, \omega_i). \end{aligned} \tag{A.33}$$

As previously said Wong's analysis is based on some assumptions:

1. The ground is firm.
2. Shear stress at a given point on the track-ground interface during a turn depend on the shear displacement at that point, as shown in Figure A.9.
3. The direction of the shear stress at a point on the track-ground interface is opposite to that of sliding velocity between the track and the ground at that point.
4. The lateral component of the shear stress constitutes the lateral resistance of

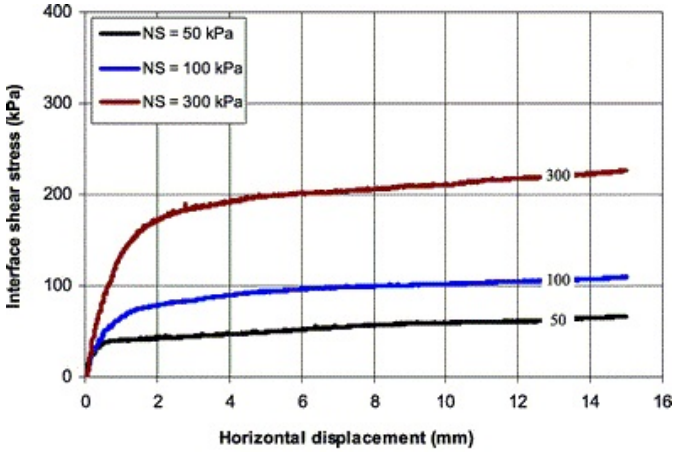


Figure A.9: Relation between shear displacement and shear stress, [108].

the track, that in turns defines M_r

A.3.1 Shear Displacement Analysis

To consider the shear stress, it is required to analyze the development of shear displacement between track and ground.
Consider a tracked vehicle, with track width W, to be in a steady-state turn about O, as shown in Figure A.10. Let, now, O_1 be the origin of a reference frame $\mathcal{F}_1(O_1, X_1, Y_1)$ fixed to, and moving with, the vehicle body (hull) and located on the longitudinal centerline of the outside track at a distance s_0 from the COG of the vehicle; as mentioned in previous sections it is now possible to prove that in some cases s_0 can be neglected, nevertheless in the picture and throughout the following analysis is reported for sake of generality and completeness. Looking at Figure A.10 and to Figure A.11, it is possible to conclude that the absolute velocity $V_{O_1 y_1}$ of O_1 in the Y_1 direction can be expressed by:

$$V_{O_1 y_1} = \big(R'' + d + c_x\big)\Omega_z \tag{A.34}$$

where R'' is the lateral distance between the ICR O and the COG, as shown in Figure A.10, c_x is the lateral distance between the *COG* and the longitudinal centerline of the vehicle hull. R'' can be computed as

$$R'' = R' \cos\alpha = \sqrt{R'^2 - s_0^2} \tag{A.35}$$

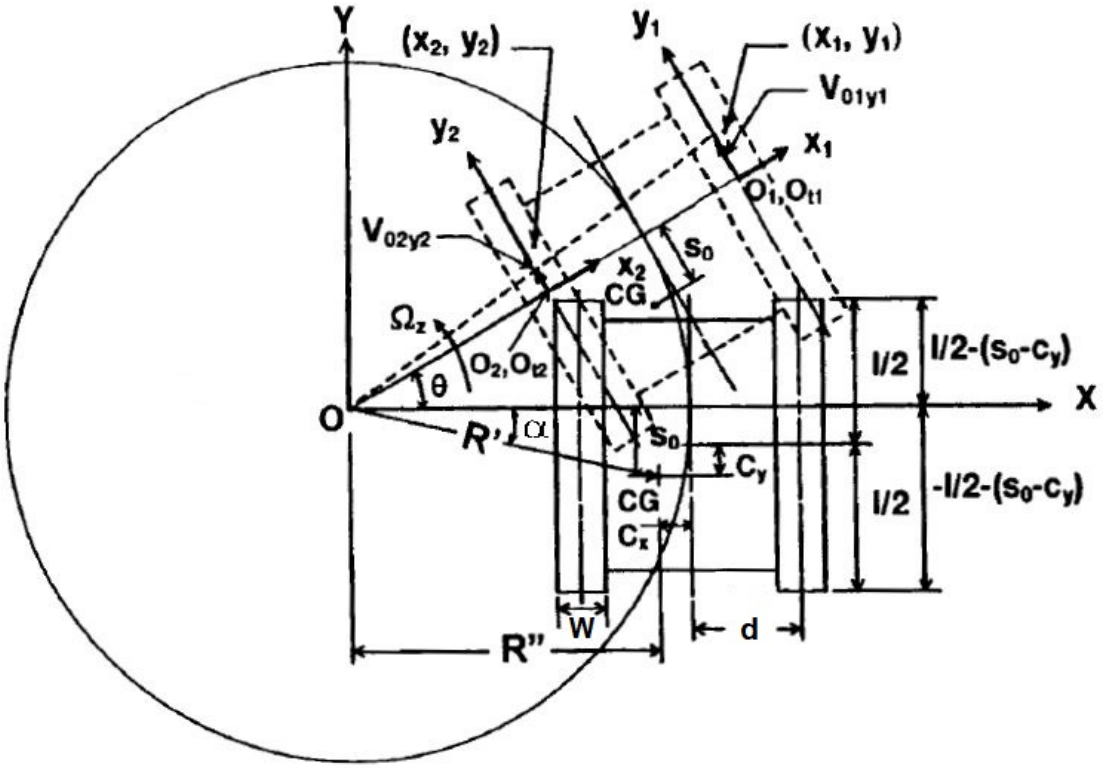


Figure A.10: Relation between shear displacement and shear stress, [1].

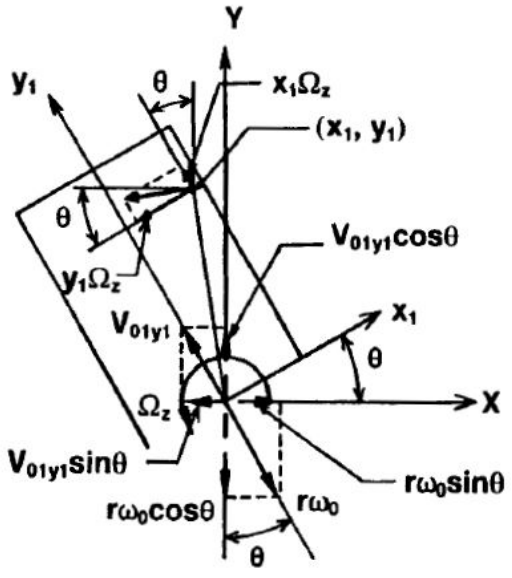


Figure A.11: Slippage velocities of different points of the track, [1].

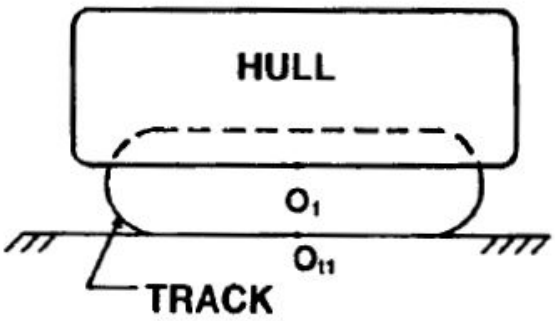


Figure A.12: Position of the origin of $\mathcal{F}_1$ and its projection on the contact plane, [1].

which can be approximated considering the (A.28)

$$R'' = R' \cos\alpha \quad \Rightarrow \quad R'' \approx R' \quad \Longleftrightarrow \quad \alpha \approx 0. \tag{A.36}$$

Referring to Figure A.12, let's consider a point o_{t_1} on the external part of the track, in contact with the ground, coincident with O_1, it has a relative velocity V_{t_1,O_1} with respect to O_1 that can be expressed as

$$V_{t_1,O_1} = r\omega_o. \tag{A.37}$$

notice that, without loss of generality, the considered track is the outer one since, as shown in Figure A.10, the $\mathcal{F}_1$ reference frame is attached to the outer track, but this analysis can be as well performed considering the inner track, and in turn $\mathcal{F}_2(O_2, X_2, Y_2)$. As a result, the sliding velocity $V_{t_1 j}$ (the subscript j is used to refer to shear displacement) of the point o_{t_1} on the ground along the longitudinal direction of the outside track is

$$V_{t_1 j} = V_{O_1 y_1} - r\omega_o. \tag{A.38}$$

Consider now an arbitrary point defined by (x_1, y_1) on the outside track in the section touching the ground. Since the track is rotating together with the vehicle around the *ICV* (O) at an angular velocity of Ω_z the relative velocity components of point (x_1, y_1) with respect to o_{t_1} in the longitudinal and lateral directions of the track are given by $x_1\Omega_z$ and $y_1\Omega_z$, respectively.

Based on this, the components of the sliding velocity V_{j_o} of the point (x_1, y_1) onto the outside track in contact with the ground, can be expressed in the inertial reference

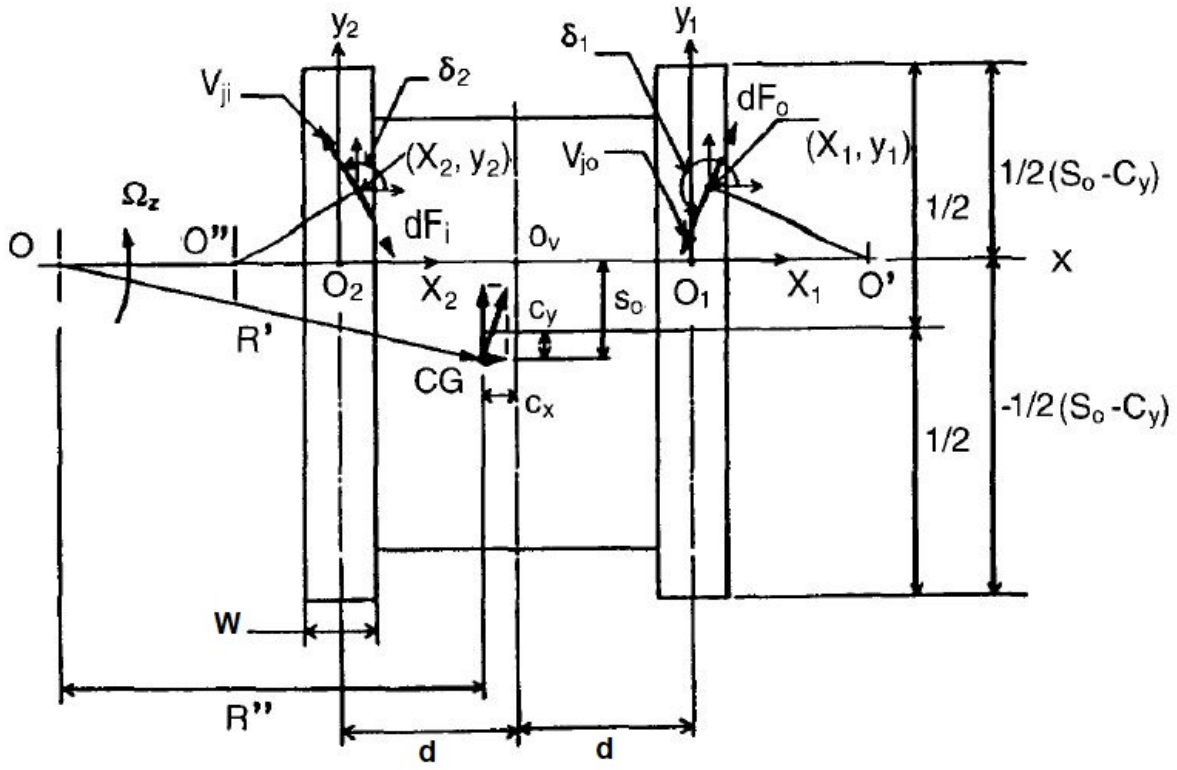


Figure A.13: Sliding velocities of the track during a steady-state turn, [1].

frame $\mathcal{F}_0(O, X_0, Y_0)$, as represented in Figure A.11 and Figure A.13, by

$$\begin{aligned} V_{j_{X_o}} &= -V_{O_1 y_1} \sin\theta + r\omega_o \sin\theta - x_1\Omega_z \sin\theta - y_1\Omega_z \cos\theta \\ &= -\big[\big(R'' + d + c_x + x_1\big)\Omega_z - r\omega_o\big] \sin\theta - y_1\Omega_z \cos\theta \end{aligned} \tag{A.39}$$

$$\begin{aligned} V_{j_{Y_o}} &= V_{O_1 y_1} \cos\theta - r\omega_o \cos\theta + x_1\Omega_z \cos\theta - y_1\Omega_z \sin\theta \\ &= \big[\big(R'' + d + c_x + x_1\big)\Omega_z - r\omega_o\big] \cos\theta - y_1\Omega_z \sin\theta. \end{aligned} \tag{A.40}$$

As mentioned by Steed in [66], it is important to notice that O' and O'' in Figure A.13 are the ICRs for the parts belonging to the outside and inside tracks in contact with and sliding on the ground.

In Figure A.10 and Figure A.11 the angle θ is used to represent the angular displacement of the vehicle and can be determined by the integration of the angular velocity of the vehicle Ω_z over the time t that the point (x_1, y_1) takes to travel from the initial point of contact to the one at the front of the track, refer to Figure A.14 (at

$y_1 = L/2 + c_y - s_0$), and it can be expressed as

$$\theta = \int_0^t \Omega_z dt = \Omega_z t \tag{A.41}$$

where t can be computed as

$$t = \int_0^t dt = \int_{y_1}^{L/2+c_y-s_0} \frac{dy_1}{r\omega_o} = \frac{L/2 + c_y - s_0 - y_1}{r\omega_o}. \tag{A.42}$$

In (A.42), c_y is the longitudinal distance between the COG and the lateral centerline of the vehicle hull.

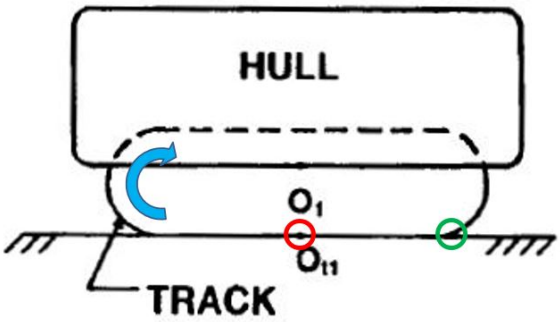


Figure A.14: The red-circled point represents the initial contact point while the green-circled one the contact at the front of the track.

Having so far computed the sliding velocities expressed in terms of $\mathcal{F}_0$, it is possible to define the displacement j_{X_o}, j_{Y_o} at the point (x_1, y_1) on the outside track in contact with the ground, along the X_0 and Y_0 directions as

$$\begin{aligned}
j_{X_o} &= \int_0^t V_{j_{X_o}} dt = \\
&= \int_{y_1}^{L/2+c_y-s_0} \left\{ -\left[\left(R'' + d + c_x + x_1\right)\Omega_z - r\omega_o \right] \sin\theta - y_1 \Omega_z \cos\theta \right\} \frac{dy_1}{r\omega_o} = \\
&= \left(R'' + d + c_x + x_1\right) \left\{ \cos\left[\frac{\left(L/2 + c_y - s_0 - y_1\right)\Omega_z}{r\omega_o} \right] - 1 \right\} + \\
&- y_1 \sin\left[\frac{\left(L/2 + c_y - s_0 - y_1\right)\Omega_z}{r\omega_o} \right],
\end{aligned} \tag{A.43}$$

$$
\begin{aligned}
j_{Y_o} &= \int_0^t V_{j_{Y_o}} dt = \\
&= \int_{y_1}^{L/2+c_y-s_0} \left\{ \left[\big(R'' + d + c_x + x_1\big)\Omega_z - r\omega_o \right] \cos\theta - y_1 \Omega_z \sin\theta \right\} \frac{dy_1}{r\omega_o} = \\
&= \big(R'' + d + c_x + x_1\big) \sin\left[\frac{\big(L/2 + c_y - s_0 - y_1\big)\Omega_z}{r\omega_o} \right] - \left(\frac{L}{2} + c_y - s_0 \right) + \\
&+ y_1 \cos\left[\frac{\big(L/2 + c_y - s_0 - y_1\big)\Omega_z}{r\omega_o} \right].
\end{aligned}
\tag{A.44}
$$

Now, the resultant shear displacement of the point (x_1, y_1) can be computed as

$$
j_o = \sqrt{j_{X_o}^2 + j_{Y_o}^2}. \tag{A.45}
$$

Similar approach, applied for points on the outer track, can be applied for those on the inner one, this time with respect to $\mathcal{F}_2$. Therefore, the absolute velocity $V_{O_2 y_2}$ of O_2 in the y_2 direction can be expressed as

$$
V_{O_2 y_2} = \big(R'' - d + c_x\big)\Omega_z \tag{A.46}
$$

Similarly to the steps described before

$$
\begin{aligned}
V_{j_{X_i}} &= -V_{O_2 y_2} \sin\theta + r\omega_i \sin\theta - x_2 \Omega_z \sin\theta - y_2 \Omega_z \cos\theta \\
&= -\big[\big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i \big] \sin\theta - y_2 \Omega_z \cos\theta,
\end{aligned}
\tag{A.47}
$$

$$
\begin{aligned}
V_{j_{Y_i}} &= V_{O_2 y_2} \cos\theta - r\omega_i \cos\theta + x_2 \Omega_z \cos\theta - y_2 \Omega_z \sin\theta \\
&= \big[\big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i \big] \cos\theta - y_2 \Omega_z \sin\theta
\end{aligned}
\tag{A.48}
$$

In this case the contact time t that the generic point (x_2, y_2) takes to travel from the initial contact point to the contact point at the front of track is

$$
t = \frac{L/2 + c_y - s_0 - y_2}{r\omega_i}. \tag{A.49}
$$

Therefore the shear displacement along the to inertial axes X and Y can be determined as follow:

$$\begin{aligned} j_{X_i} &= \int_0^t V_{j_{X_i}} dt = \\ &= \int_{y_2}^{L/2+c_y-s_0} \left\{ -\Big[\big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i\Big] \sin\theta - y_2\Omega_z \cos\theta \right\} \frac{dy_2}{r\omega_i} = \\ &= \big(R'' - d + c_x + x_2\big) \left\{ \cos\left[\frac{\big(L/2 + c_y - s_0 - y_2\big)\Omega_z}{r\omega_i} \right] - 1 \right\} + \\ &- y_2 \sin\left[\frac{\big(L/2 + c_y - s_0 - y_2\big)\Omega_z}{r\omega_i} \right] \end{aligned} \tag{A.50}$$

$$\begin{aligned} j_{Y_i} &= \int_0^t V_{j_{Y_i}} dt = \\ &= \int_{y_2}^{L/2+c_y-s_0} \left\{ \Big[\big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i\Big] \cos\theta - y_2\Omega_z \sin\theta \right\} \frac{dy_2}{r\omega_i} = \\ &= \big(R'' - d + c_x + x_2\big) \sin\left[\frac{\big(L/2 + c_y - s_0 - y_2\big)\Omega_z}{r\omega_i} \right] - \left(\frac{L}{2} + c_y - s_0 \right) + \\ &+ y_2 \cos\left[\frac{\big(L/2 + c_y - s_0 - y_2\big)\Omega_z}{r\omega_i} \right] \end{aligned} \tag{A.51}$$

Finally, the resultant shear displacement of the point (x_2, y_2) can be computed as

$$j_i = \sqrt{j_{X_i}^2 + j_{Y_i}^2}. \tag{A.52}$$

A.3.2 Dynamics in Steady-state turn

In this section an alternative way to compute the turning resistive forces are developed starting from the shear elements analysis carried out in the previous sections. As previously mentioned, the shear stress developed at a given point on the track-ground interface is dependent on the shear displacement at that point. As stated before, the typology of the terrain and the conditions of the latter influences the relationship between shear stress and shear displacement, relation that can be expressed as

$$\tau = \sigma \tan\phi\big(1 - e^{-j/K}\big) = \sigma\mu\big(1 - e^{-j/K}\big). \tag{A.53}$$

Where σ represents the normal pressure, μ the coefficient of friction between the track, that can also be defined as $\mu = \tan\phi$, ϕ is the angle of internal shearing resistance of the ground material, j the shear displacement and K is the shear deformation modulus of the soil.
The assumptions done by Wong during the development of his analysis, which have been reported at the very beginning of this sections, can be relaxed. In particular the first one, since it can result extremely limiting to consider for tracked vehicles to travel only onto firm ground. Into his work, Al-Milli [69] proved that it is possible to extend Wong's analysis also to soft terrains, namely, adding a term related to the internal cohesion of the material (c) to the (A.53):

$$\tau = \big(c + \sigma\tan\phi\big)\big(1 - e^{-j/K}\big) = \big(c + \sigma\mu\big)\big(1 - e^{-j/K}\big). \tag{A.54}$$

Considering the infinitely small area of the tracks in contact with the ground, the shear forces developed on these elements can be expressed as

$$\begin{aligned} dF_o &= \tau_o dA = \big(c + \sigma_o\mu\big)\big(1 - e^{-j_o/K}\big)dA && \textit{(on outside track)} \\ dF_i &= \tau_i dA = \big(c + \sigma_i\mu\big)\big(1 - e^{-j_i/K}\big)dA && \textit{(on inside track)} \end{aligned} \tag{A.55}$$

where τ_o and τ_i are the shear stresses, j_o and j_i shear displacements and σ_o and σ_i the the normal pressures of the elements of the outside and inside tracks, respectively. In addition, it can be noticed that the two pressures are inserted in the equation as constants but those ones can be easily expressed as a function of the position of the considered point in the track, therefore, even the assumption made about the uniformity of pressure distribution can be extended to other cases in which non uniform normal pressure distribution is considered, as described in details in Chapter 3, Section 3.4.
Another observation is proposed to the reader, it has been said that Steeds [66] considered the shear force between tracks and ground to obey Coulomb's friction law, this is equivalent to consider the K parameter to be equal to zero, causing the full frictional force to be reached as soon as an infinitesimal shear displacement takes place.
Figure A.15 shows that the longitudinal forces F_{y_o} and F_{y_i} acting respectively on the

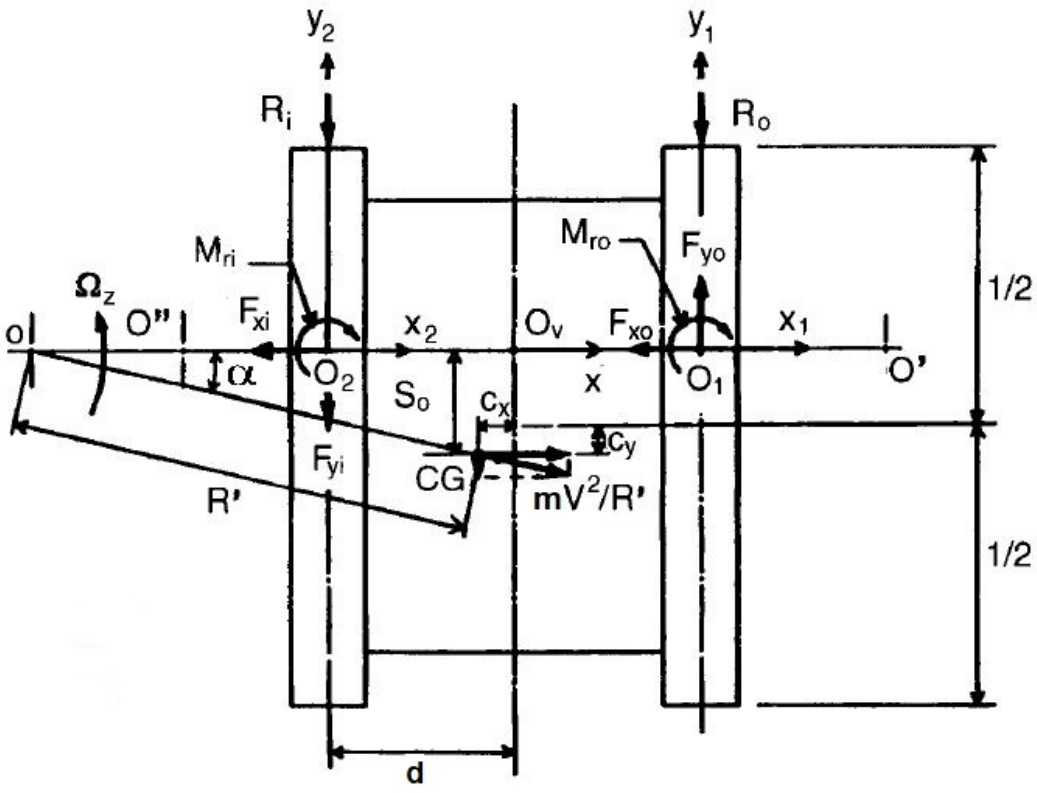


Figure A.15: Forces and moments acting on a tracked vehicle during a steady-state turn, [1].

outside and inside tracks can be expressed as

$$\begin{aligned} F_{y_o} &= \int dF_o \sin\left(\pi + \delta_1\right) \\ &= -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} \left(c + \sigma_o \mu\right)\left(1 - e^{-j_o/K}\right) \sin\delta_1 dx_1 dy_1 \end{aligned} \tag{A.56}$$

$$\begin{aligned} F_{y_i} &= \int dF_i \sin\left(\pi + \delta_2\right) \\ &= -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} \left(c + \sigma_i \mu\right)\left(1 - e^{-j_i/K}\right) \sin\delta_2 dx_2 dy_2 \end{aligned} \tag{A.57}$$

where δ_1 and δ_2, shown also in Figure A.15, are the angles between the resultant sliding velocities of the points on the outside and inside tracks and the lateral directions of the tracks (i.e. x_1 and x_2 axes), respectively. It is also worthy to notice that follow-

ing Coulomb's friction law, the shear force acting on the track will be in the opposite direction of the resultant sliding velocity (i.e. $\pi + \delta_{1,2}$). Then, the lateral forces F_{x_o} and F_{x_i} acting respectively on the outer and inner tracks are given by:

$$\begin{aligned} F_{x_o} &= \int dF_o \cos\left(\pi + \delta_1\right) \\ &= -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} \left(c + \sigma_o \mu\right)\left(1 - e^{-j_o/K}\right) \cos\delta_1 dx_1 dy_1 \end{aligned} \tag{A.58}$$

$$\begin{aligned} F_{x_i} &= \int dF_i \cos\left(\pi + \delta_2\right) \\ &= -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} \left(c + \sigma_i \mu\right)\left(1 - e^{-j_i/K}\right) \cos\delta_2 dx_2 dy_2. \end{aligned} \tag{A.59}$$

From (A.56) and (A.57) the moments M_{l_o} and M_{l_i}, generated by the longitudinal shear forces with respect to the point O_v can be expressed as

$$M_{l_o} = -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} \left(d + x_1\right)\left(c + \sigma_o \mu\right)\left(1 - e^{-j_o/K}\right) \sin\delta_1 dx_1 dy_1 \tag{A.60}$$

$$M_{l_i} = -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} \left(d - x_1\right)\left(c + \sigma_i \mu\right)\left(1 - e^{-j_i/K}\right) \sin\delta_2 dx_2 dy_2. \tag{A.61}$$

While, starting from (A.58) and (A.59), the moments M_{r_o} and M_{r_i} generated by the lateral shear forces with respect to O_1 and O_2 can be expressed as

$$M_{r_o} = -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} y_1\left(c + \sigma_o \mu\right)\left(1 - e^{-j_o/K}\right) \cos\delta_1 dx_1 dy_1 \tag{A.62}$$

$$M_{r_i} = -\int_{-L/2+c_y-s_0}^{L/2+c_y-s_0} \int_{-W/2}^{W/2} y_2\left(c + \sigma_i \mu\right)\left(1 - e^{-j_i/K}\right) \cos\delta_2 dx_2 dy_2. \tag{A.63}$$

To determine all the terms appearing in the latter equations δ_1, δ_2 are still to be identified. To do this, the longitudinal sliding velocities $V_{j_{y_o}}$, $V_{j_{y_i}}$ of the elements respectively on the outside and inside tracks with respect to the moving reference frames $\mathcal{F}_1$

and $\mathcal{F}_2$, are fist derived as (in line to what reported in Figure A.15)

$$V_{j_{y_o}} = \big(R'' + d + c_x + x_1\big)\Omega_z - r\omega_o, \tag{A.64}$$

$$V_{j_{y_i}} = \big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i. \tag{A.65}$$

Whilst the sliding velocities $V_{j_{x_o}}$ and $V_{j_{x_i}}$ of the track elements on the outer and inner tracks respectively, can be expressed as

$$V_{j_{x_o}} = -y_1\Omega_z, \tag{A.66}$$

$$V_{j_{x_i}} = -y_2\Omega_z. \tag{A.67}$$

Therefore, it follows that

$$\sin\delta_1 = \frac{V_{j_{y_o}}}{\sqrt{V_{j_{x_o}}^2 + V_{j_{y_o}}^2}} = \frac{\big(R'' + d + c_x + x_1\big)\Omega_z - r\omega_o}{\sqrt{\Big[\big(R'' + d + c_x + x_1\big)\Omega_z - r\omega_o\Big]^2 + \big(y_1\Omega_z\big)^2}} \tag{A.68}$$

$$\sin\delta_2 = \frac{V_{j_{y_i}}}{\sqrt{V_{j_{x_i}}^2 + V_{j_{y_i}}^2}} = \frac{\big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i}{\sqrt{\Big[\big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i\Big]^2 + \big(y_2\Omega_z\big)^2}} \tag{A.69}$$

$$\cos\delta_1 = \frac{V_{j_{x_o}}}{\sqrt{V_{j_{x_o}}^2 + V_{j_{y_o}}^2}} = \frac{-y_1\Omega_z}{\sqrt{\Big[\big(R'' + d + c_x + x_1\big)\Omega_z - r\omega_o\Big]^2 + \big(y_1\Omega_z\big)^2}} \tag{A.70}$$

$$\cos\delta_2 = \frac{V_{j_{x_i}}}{\sqrt{V_{j_{x_i}}^2 + V_{j_{y_i}}^2}} = \frac{-y_2\Omega_z}{\sqrt{\Big[\big(R'' - d + c_x + x_2\big)\Omega_z - r\omega_i\Big]^2 + \big(y_2\Omega_z\big)^2}} \tag{A.71}$$

It is important to notice that substituting (A.68-71) into (A.56-59) and (A.60-63) the resulting forces and moments become functions of the theoretical speeds $r\omega_o$ and $r\omega_i$ and of the turning radius R''.
Therefore, considering all the other parameters as given data, it is possible to com-

pute the overall resistive moment M_r using M_{r_o} and M_{r_i}, which are dependent on the desired parameters, i.e. $r\omega_o$,$r\omega_i$ and R''. Putting altogether we obtain

$$M_r = M_{r_o} + M_{r_i}, \tag{A.72}$$

from which it is possible to find an expression for the lateral friction coefficient μ_t that is dependent on the type of turn the robot is going to initiate

$$M_r = \mu_t \left(\frac{g_{z_B}}{2} - h\frac{g_{y_B}}{2d} \right) \frac{mL}{4d} \quad \Rightarrow \quad \mu_t \left(\frac{g_{z_B}}{2} - h\frac{g_{y_B}}{2d} \right) \frac{mL}{4d} = M_{r_o} + M_{r_i} \tag{A.73}$$

$$\mu_t = \mu_t(\omega_o, \omega_i, R'') = \frac{4d\big(M_{r_o} + M_{r_i}\big)}{mL} \left(\frac{g_{z_B}}{2} - h\frac{g_{y_B}}{2d} \right)^{-1}. \tag{A.74}$$

The reader is encouraged to notice that this analysis embeds in M_r, and in turn μ_t, also a generic position for the COG that, as seen previously, can be an important parameter in the specific study case due to the eventual presence of attached tools that modify the mass distribution of the vehicle.

A.4 Conclusions

In this chapter, an analytical and detailed mechanical study has been reported, describing the main elements governing dynamics of a generic tracked skid steering vehicle. The dynamic model derived here enriches the analysis performed in Chapter 3, in particular, it provides the background knowledge required to understand how we embedded in a single parameter (μ_t) the (i) vehicle parameters, (ii) soil features, and (iii) trajectory parameters.

B

Complements of System Theory

In this appendix, we recall basic definitions of nonlinear system theory.

B.1 Single-input single-output nonlinear systems

Definition 1 ([101] - *Lie derivative*) *Consider the n-dimensional system, Single-input Single-output (SISO) system*

$$\dot{x} = f(x) + g(x)u, \qquad y = h(x) \tag{B.1}$$

where f, g and h are sufficiently smooth in a domain $D \subset \mathbb{R}^n$. The mappings $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^n$ are called vector fields *on D. The derivative $\dot{y}$ is given by*

$$\dot{y} = \frac{\partial h}{\partial x}\left[f(x) + g(x)u\right] \coloneqq L_f h(x) + L_g h(x)u \tag{B.2}$$

where

$$L_f h(x) = \frac{\partial h}{\partial x} f(x) \tag{B.3}$$

*is called the **Lie Derivative** of h with respect to f or along f.*

Definition 2 ([101] - *Relative degree*) *The nonlinear system (B.1) has **relative degree***

ρ, $1 \leq \rho \leq n$, in an open set $\mathcal{R} \subset D$ if, for all $x \in \mathcal{R}$,

$$L_g L_f^{i-1} h(x) = 0, \quad for \quad i = 1, 2, \ldots, \rho - 1; \qquad L_g L_f^{\rho-1} h(x) \neq 0 \tag{B.4}$$

Remark 6 ([102]) *If a system has relative degree ρ, then its input-output map can be converted into a chain of integrators $y^{(\rho)} = v$ by the state feedback control*

$$u = \frac{1}{L_g L_f^{\rho-1} h(x)} \left(-L_f^{\rho} h(x) + v \right) \tag{B.5}$$

Definition 3 ([101] - *Coordinate change*) *We define the change of coordinates $z = T(x)$ such that the new state z can be partitioned into a ρ-dimensional vector ξ and $(n-\rho)$-dimensional vector σ, where the components of ξ comprises the output and its derivatives up to $y^{(\rho-1)}$, while σ satisfies a differential equation whose right-hand side depends on σ and ξ, but not on u. We take ξ as in the linear case:*

$$\xi = \begin{bmatrix} y \\ y^{(1)} \\ \vdots \\ y^{(\rho-1)} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{\rho-1} h(x) \end{bmatrix}. \tag{B.6}$$

When $\rho = n$, there is no σ variable and the change of variables is given by

$$z = T(x) = \begin{bmatrix} h(x) & L_f h(x) & \ldots & L_f^{n-1} h(x) \end{bmatrix}^{\top} \tag{B.7}$$

When $\rho < n$, the change of variables is taken as

$$z = T(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{n-\rho}(x) \\ h(x) \\ \vdots \\ L_f^{\rho-1} h(x) \end{bmatrix} := \begin{bmatrix} \phi(x) \\ \mathcal{H}(x) \end{bmatrix} := \begin{bmatrix} \sigma \\ \xi \end{bmatrix} \tag{B.8}$$

where ϕ_1 to $\phi_{n-\rho}$ are chosen such that $T(x)$ is a diffeomorphism on a domain $D_x \subset \mathcal{R}$ and

$$\frac{\partial \phi_i}{\partial x} g(x) = 0, \qquad \textit{for } 1 \leq i \leq n - \rho, \quad \forall x \in D_x \tag{B.9}$$

so that the the u term cancels out in

$$\dot{\sigma} = \frac{\partial \phi_i}{\partial x} \left[f(x) + g(x)u \right]. \tag{B.10}$$

Definition 4 ([101] - *Normal form*) *When $\rho < n$, the change of coordinates (B.8) transforms (B.1) into*

$$\dot{\sigma} = f_0(\sigma, \xi) \tag{B.11}$$
$$\dot{\xi} = A_c \xi + B_c \left[L_f^{\rho} h(x) + L_g L_f^{\rho-1} h(x) u \right] \tag{B.12}$$
$$y = C_c \xi \tag{B.13}$$

*which is said to be in **normal form**.*

B.2 Multi-input Multi-output nonlinear systems

Consider the multivariable nonlinear system taken from [102]

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x) u_i \\ y_1 &= h_1(x) \\ &\dots \\ y_m &= h_m(x) \end{aligned} \tag{B.14}$$

in which $f(x), g_1(x), ..., g_m(x)$ are smooth vector fields, and $h_1(x), ..., h_m(x)$ smooth functions defined on an open set of $\mathbb{R}^n$. Whenever possible and convenient, these equations will be rewritten in the more condensed form

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{B.15}$$

where

$$\begin{aligned} u &= [u_1 \ \dots \ u_m]^\top \\ y &= [y_1 \ \dots \ y_m]^\top \end{aligned} \tag{B.16}$$

and where

$$\begin{aligned} g(x) &= [g_1(x) \ \dots \ g_m(x)] \\ h(x) &= [h_1(x) \ \dots \ h_m(x)]^\top \end{aligned} \tag{B.17}$$

are respectively an $n \times m$-dimensional matrix and an m-dimensional vector.

Definition 5 ([102] - *Vector relative degree*) *The nonlinear system (B.14) has **(vector) relative degree** $\{r_1, \dots, r_m\}$ at a point x^o if*
(i)

$$L_{g,j} L_f^k h_i(x) = 0 \tag{B.18}$$

for all $1 \leq j \leq m$, for all $k \leq r_i - 1$, for all $1 \leq i \leq m$, and for all x in a neighborhood of x^o;
(ii)

$$A(x) = \begin{bmatrix} L_{g,1} L_f^{r_1-1} h_1(x) & \dots & L_{g,m} L_f^{r_1-1} h_1(x) \\ L_{g,1} L_f^{r_2-1} h_2(x) & \dots & L_{g,m} L_f^{r_2-1} h_2(x) \\ \dots & \dots & \dots \\ L_{g,1} L_f^{r_m-1} h_m(x) & \dots & L_{g,m} L_f^{r_m-1} h_m(x) \end{bmatrix} \tag{B.19}$$

is nonsingular at $x = x^o$.

Remark 7 ([102]) *In the the coordinates ξ, σ, it is possible to show that the input must be set as*

$$u(t) = -\left[A\left(0, \sigma(t)\right)\right]^{-1} N\left(0, \sigma(t)\right) \tag{B.20}$$

where $A\left(0, \sigma(t)\right) = A\left(\xi = 0, \sigma(t)\right)$ and $N\left(0, \sigma(t)\right) = N\left(\xi = 0, \sigma(t)\right)$ is defined as

$$N(x) = \begin{bmatrix} L_f^{r_1-1} h_1(x) \\ L_f^{r_2-1} h_2(x) \\ \dots \\ L_f^{r_m-1} h_m(x) \end{bmatrix} \tag{B.21}$$

with x replaced by $\Phi^{-1}(\xi, \sigma)$.

Definition 6 ([101] - *Coordinate change*) *Suppose a system has a (vector) relative degree $r_1, \dots, r_m$ at x^o. Then*

$$\sum_{i=1}^{m} r_i \leq n. \tag{B.22}$$

Set, for $1 \leq i \leq m$,

$$\begin{aligned} \phi_1^i(x) &= h_i(x) \\ \phi_2^i(x) &= L_f h_i(x) \\ &\dots \\ \phi_{r_i}^i(x) &= L_f^{r_i-1} h_i(x) \end{aligned} \tag{B.23}$$

If (B.22) *is strictly less than* n, *it is always possible to find* $n-r$ *more functions* $\phi_{r+1}(x), \dots, \phi_n(x)$ *such that the mapping*

$$\Phi(x) = \begin{bmatrix} \phi_1^1(x) & \dots & \phi_{r_1}^1(x) & \dots & \phi_1^m(x) & \dots & \phi_{r_m}^m(x) & \phi_{r+1}(x) & \dots & \phi_n(x) \end{bmatrix}^\top \tag{B.24}$$

has a jacobian matrix which is nonsingular at x^o *and therefore qualifies as a local coordinates transformation in a neighborhood of* x^o. *The value at* x^o *of these additional functions can be chosen arbitrarily. Moreover, if the distribution*

$$G = \textit{span}\,(g_1, \dots, g_m) \tag{B.25}$$

is involutive near x^o, *it is always possible to choose* $\phi_{r+1}(x), \dots, \phi_n(x)$ *in such a way that*

$$L_{g,j}\phi_i(x) = 0 \tag{B.26}$$

for all $r+1 \leq i \leq n$, *for all* $1 \leq j \leq m$, *and all* x *around* x^o.

C

Event Cameras

In this appendix, an overview of the features of Dynamic Vision Sensors is provided, focusing in particular on the application implemented in order to gain confidence with this novel technology. It is important to bear in mind that, the studies carried out had the main goal of defining whether our agricultural robot would benefit or not from the addition of Event Cameras with its sensor suite.

C.1 Dynamic Vision Sensor

Standard-camera-based robotic applications have limitations where the scene objects are moving with high dynamics, current CMOS cameras provide sampling rates of the images within 15-40 Hz, [109], if the scene moves faster than that, images will result blurred and affect performance. These physical limits cannot be solved with sophisticated algorithms.
The solution is given by a novel type of sensor, called Dynamic Vision Sensor (DVS), which has an almost-negligible sensing latency. DVSs, also known as Event Cameras (ECs), [110], are bio-inspired sensors that work radically different from Standard Cameras (SCs). In fact, rather than capturing images at a fixed rate, event cameras feature "smart pixels" independent from each other, capable of detecting variation in the scene asynchronously. Each pixel transmits information (called *event*) only if it detects variation in brightness. Contrary to SCs, whose pixels are sampled syn-

chronously altogether, the output of ECs is a sequence of events, as represented in Figure C.1 and in a video[1].

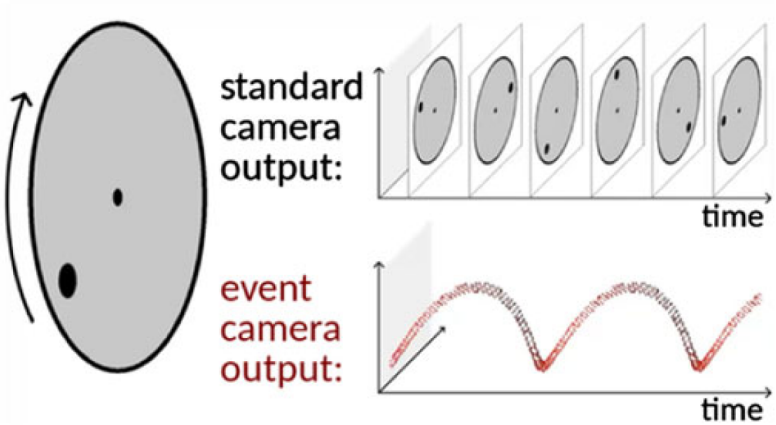


Figure C.1: Comparison between the output frames of a standard camera and the asynchronous event stream of a DVS, while both are looking at a rotating disk. Picture taken from [111].

An event e_k "fired" at time t_k is defined as a 3-tuple $e_k := (\mathbf{x}_k, t_k, p_k)$, where $\mathbf{x}_k = (x_k, y_k)^\top$ are the coordinates of the triggered pixel and $p_k \in \{-1, 1\}$ is the polarity (sign) of the brightness change.
More details about events cameras, their working principles and applications can be found in [112]. To better understand the following sections, it is important to briefly review how an event occurs.
For sake of simplicity, let's assume a noise-free DVS, then, an event e_k is triggered as soon as the brightness level L at time t_k of the pixel $\mathbf{x}$ changes more than an user-defined threshold $C > 0$. In other words, the pixel $\mathbf{x}$ generates a positive event ($p_k = +1$) at time t_k if

$$L(\mathbf{x}, t_k) - L(\mathbf{x}, t_{k-1}) > C, \tag{C.1}$$

and a negative event ($p_k = -1$) if

$$L(\mathbf{x}, t_k) - L(\mathbf{x}, t_{k-1}) < -C. \tag{C.2}$$

It is now clear the meaning of the name *Dynamic Vision Sensors* as they can be seen as motion-activated sensors, producing output only when relative motion between the camera and the scene occurs. Thanks to their asynchronous nature, the perception

[1] https://youtu.be/LauQ6LWTkxM?t=3

latency introduced by neuromorphic sensors are ideal for **high-bandwidth applications**, [113].
Apart from the high time resolution, [112] also points out that this family of sensors feature **a very high dynamic range** ($> 120dB$). The latter was one of the main reasons why this sensor was a candidate for the sensor suite of our agricultural robot, as working in poor lighting conditions (e.g. after the sunset) is one of the key requirement of our design.
In order to asses whether to add or not ECs to the robot, we decided to study their working principles more in deep, therefore we decided to develop some practical application.

C.2 Practical application

As already mentioned, this study has been carried out in order to evaluate the *Dynamic Range* of Event Cameras, as well as the interesting property of having a light weight output to process. On top of everything, we were also interested in re-using computer vision algorithm already known but apply them to different contexts.

C.2.1 Dimitrova's line tracking dualcopter

In line with the intentions listed above, a really interesting application for ECs and line detection algorithm (Hough Transform) was given by [114][1].
Figure C.2 shows the experimental setup used in [114]: a dual-motor drone, hinged in the middle point of the beam connecting the two motors, resulting in a one-Deegre-Of-Freedom system, rotation about the roll axis. The dual-motor drone (*dualcopter*) is equipped with a DAVIS240C, an EC with spatial resolution of 180x240 pixels and a temporal resolution of 1 µs, [115]. Right in front of the camera, a black-and-white rotating disk is placed in order to generate the reference attitude[2] that the dualcopter has to track.
The output of the DAVIS is sent via USB connection to an UP board [116], then processed in order to estimate the relative angle between the dualcopter and the disk. In the end, the control action is computed accordingly to a PD-controller and motor commands are sent to the flight controller. The latter, in turn, converts them for the

[1] The reader can find an explanatory video here: https://youtu.be/3nIznSMCMtc
[2] In particular the junction line between the two halves.

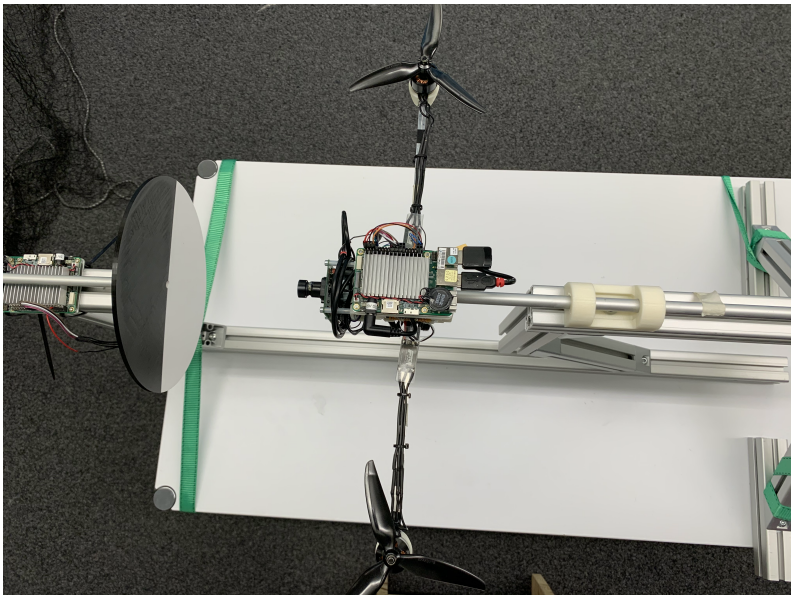


Figure C.2: Experimental setup of a dualcopter facing a rotating disk.

ESCs which apply the corresponding input voltage to the motors. Figure C.3 shows the described pipeline.

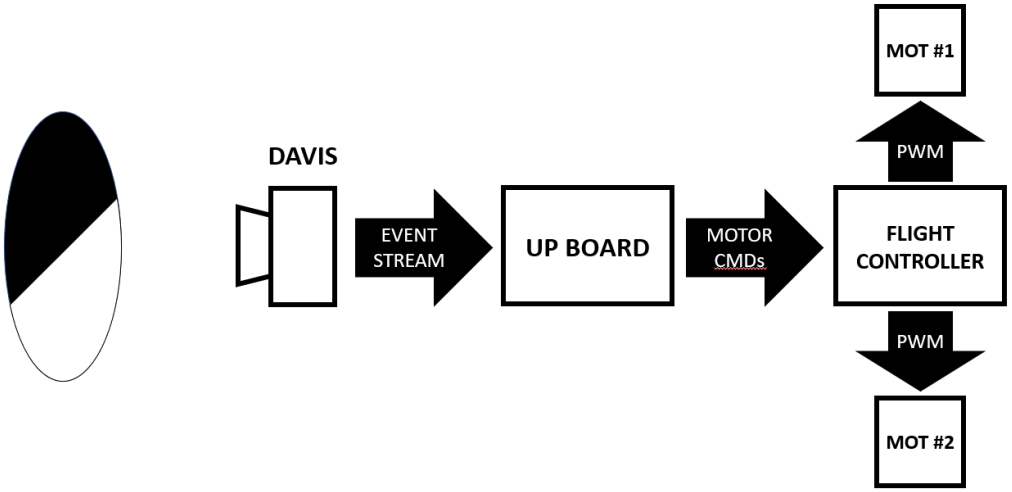
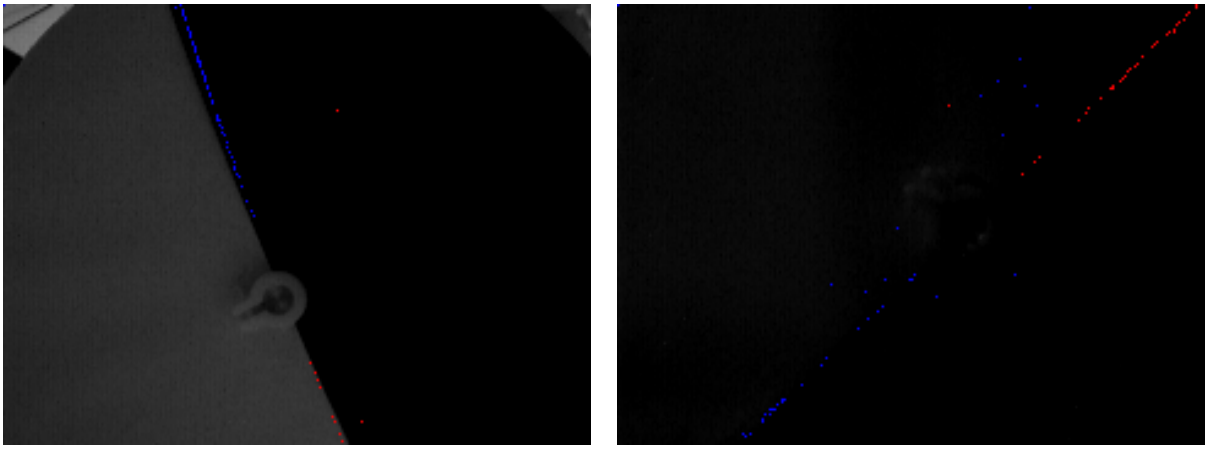


Figure C.3: Dualcopter control pipeline using the DAVIS camera.

In the end, both the disk and the dualcopter are equipped with absolute rotary encoders which measure the roll angles for ground truth.

The aim of [114] is to implement an attitude tracking closed-loop controller, using event-based visual servoing, in order to reduce the latency with respect to standard-



(a) (b)

Figure C.4: View of the disk while rotating using the DAVIS camera.

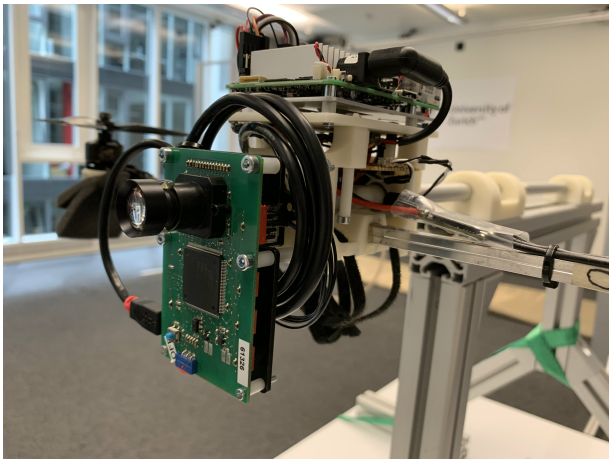
camera-based visual servoing, thus increasing the bandwidth of the overall system.

In order to estimate the attitude of the dualcopter with respect to the disk, the Hough Transform line detection algorithm has been used, more about this can be found also in Chapter 4. The idea comes from the observation of Figure C.4, where it is possible to see that, as expected, the events are all aligned along the junction between the black and while halves. It follows that, the output of the Hough Transform (HT) gives already the relative angle between the disk and the dualcopter (notice that this angle can be directly used as the attitude error feeding the PD controller). More about this can be found in [114].

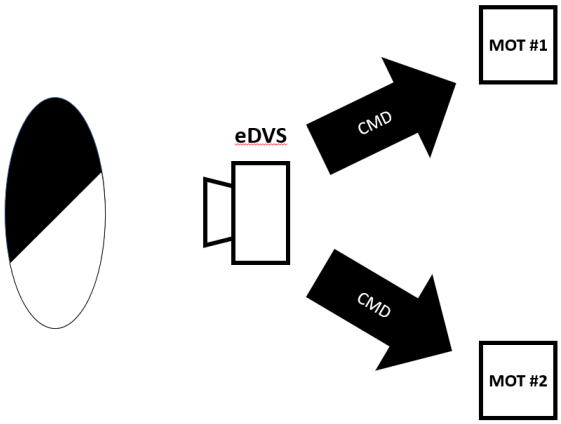
C.2.2 Our contribution

During our collaboration with Davide Scaramuzza's lab[1], we tried to reduce even further the perception latency, exploiting the eDVS (embedded DVS), [117, 118]. This event camera features a dual-core microprocessor, used both for data acquisition and processing. This board can directly generate motor commands for the ESCs, removing the latency introduced by the USB-communication present in Figure C.3 between the DAVIS and the UP-board, i.e. leading to the pipeline presented in Figure C.5.
The main challenge in the development of this project was given by the computa-

[1]This work was carried out at Davide Scaramuzza's RPG laboratory (UZH-ETH). For more information refer to http://rpg.ifi.uzh.ch/

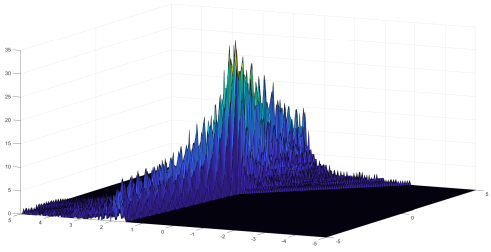


(a) Dualcopter with eDVS.

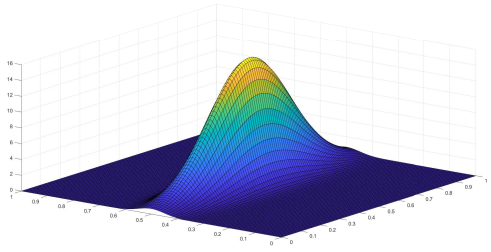


(b) eDVS control pipeline.

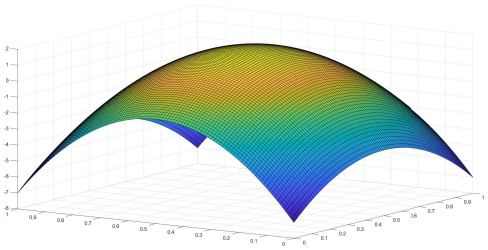
Figure C.5: Detailed view of the upgraded version of the experimental setup, and control pipeline.



(a) 2D Hough space counter.



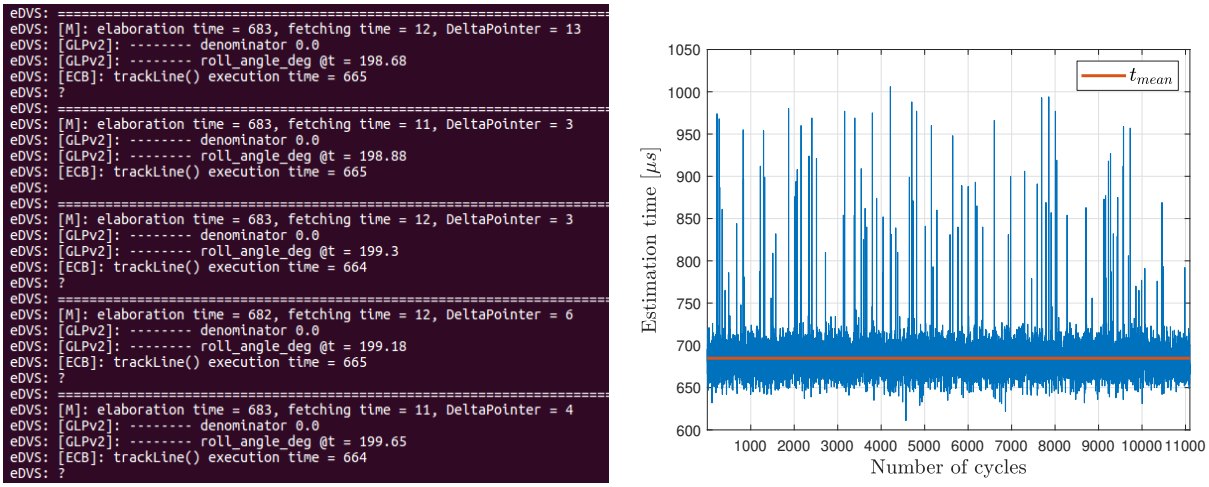
(b) Bivariate gaussian function.



(c) Logarithm of a 2D gaussian.

Figure C.6: Comparison between the 2D Hough space and the two functions used to approximate it proposed in [119].

tional power of the microcontroller (µC), which was much less performing than the microprocessor (µP) available on the UP board. In fact, running the state-of-the-art



(a) Terminal prints.

(b) Estimation time plot.

Figure C.7: Estimation time (in µs) required to execute the optimized code (CHT) on the eDVS.

code on the eDVS required more than 30 times the amount of time stated in [114], less than 700 µs. This was due to the fact that, from a computational point of view, the most expensive part of the HT algorithm is the 2D maximum search in the Hough Space, which is incredibly demanding for eDVS's µC. To improve the performance we implemented the algorithm presented in [119]. Where Conradt proposes to approximate the 2D histogram generated by the the HT[1], shown in Figure C.6a, as a bivariate gaussian function, Figure C.6b. Then, if the logarithm of the former gaussian function is computed, it is possible to notice that they both share the same maximum point, and in addition, the logarithm of a gaussian is a quadratic function. Then, Conradt defined a direct relation between the histogram generated in the Hough space by a single image point and the coefficients of the quadratic function associated to the former logarithmic function. In this way, it is possible to find a closed form solution for the coordinates of the peak value, and in turn, a closed form solution to the HT line detection problem. The optimization of the line-estimation code using Conradt's Hough Transform (CHT) reduced the estimation time[2] on the eDVS from 20 ms to an average of about 685 µs, as shown by Figure C.7.

[1]It is worthy to highlight that these approximations are adequate only if cartesian representation of a line is considered, $y = mx + q$.

[2]Time between the first new event fetched and the instant in which the CHT generates the output angle.

Final Results

The final setup was then tested and the tracking performance achieved can be seen in Figure C.8a, as well as in this video[1]. In particular, we prompt the reader to take a closer look to Figure C.8a, from which it is possible to see that, at the beginning, the tracking response is really accurate, but after $t_1 \approx 0.45 \times 10^4 ms$, the tracking error becomes steadily bigger than 180 [deg], and this increases again by other 180 [deg], after $t_2 \approx 0.6 \times 10^4 ms$. The reason for this is that the angle between the disk and the dualcopter is derived from the line angular coefficient m by means of the $atan$ function, thus, the obtained angle $\theta_{dc} \in (-\pi/2, \pi/2)$ or, by a coordinate change $\theta'_{dc} \in (0, \pi)$. This means that, when the initial acceleration of the disk is higher than the one reachable by the dualcopter, the error angle start to increase and if it becomes higher that π, the angle values wrap and reduce to a neighborhood of 0. This causes the absolute error to drift, even if the relative error is then steered to 0.
Then, if the data collected are considered piece-wise, it is possible to estimate the overall system delay, and thus define the bandwidth of the closed loop system. In particular, to do this, we used the two dataset shown in Figure C.8b and Figure C.8c. By basic system identification tools, it possible to obtain the closed-loop frequency response shown in Figure C.9b, which is faster than the one shown in Figure C.9a, with an estimated delay[2] of about 3.84 ms.
Finally, Figure C.10a shows a comparison between the rotating speed of the disk and the tracking error, as expected, the largest values of the error are present when the velocity profile of the disk undergoes steep variations, therefore high acceleration (in line to what discussed before). Figure C.10b shows that with this setup the tracking error is kept bounded even when the disk is rotated at about 3000 deg/s, even though the maximum rotating speed reached by the dualcopter is around 1900 deg/s.

C.3 Conclusions

In conclusion, we can say that the properties of Dynamic Vision Sensors are extremely suited for high-bandwidth applications, such as Micro Aerial Vehicles, in which ex-

[1] https://youtu.be/IwuPQOivsmk

[2] Notice that, the estimated delay of the system, is even smaller than the one of the encoder-driven closed-loop shown in [114]. Anyway, the comparison is not fair as those data were considered with smaller gains of the controller as it was meant as a comparison for their vision-based control.

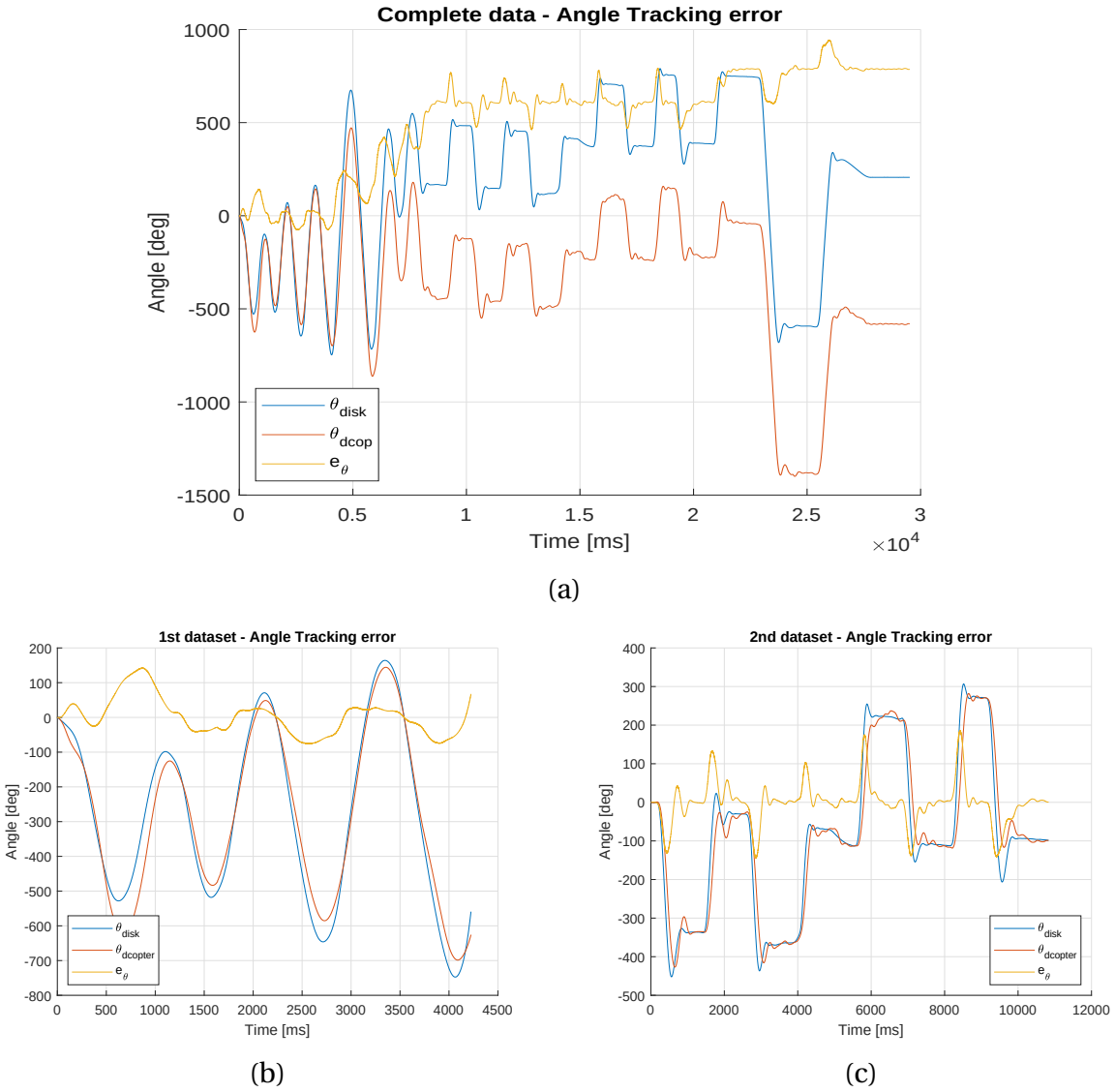
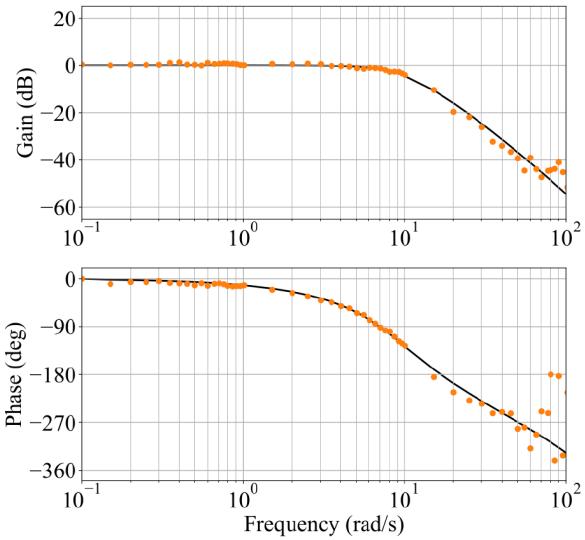


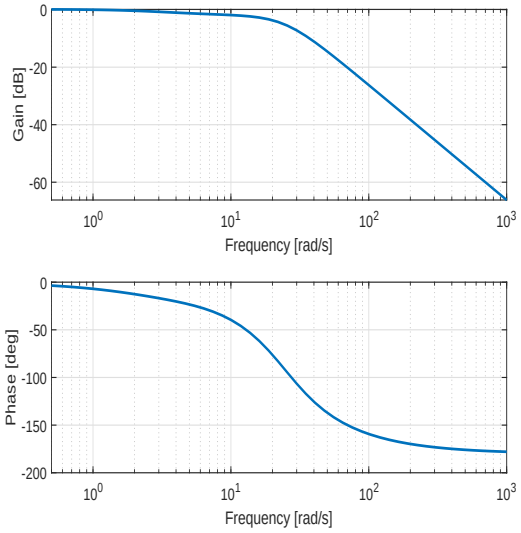
Figure C.8: Ground truth angles collected by the encoders measuring absolute attitude of the dualcopter and of the disk.

tremely low perception latency can be exploited by the dynamics of the controlled system.

On the other hand, agricultural environments are bumpy and extremely rich in terms of features (e.g. canopy leaves), resulting in tons of events triggered. It is also important to mention that the perception advantages brought in by this novel technology results in high retail prices. Thus, currently, for what concerns ground robots for agricultural applications, we consider the technology not yet ready. Nevertheless, the ex-

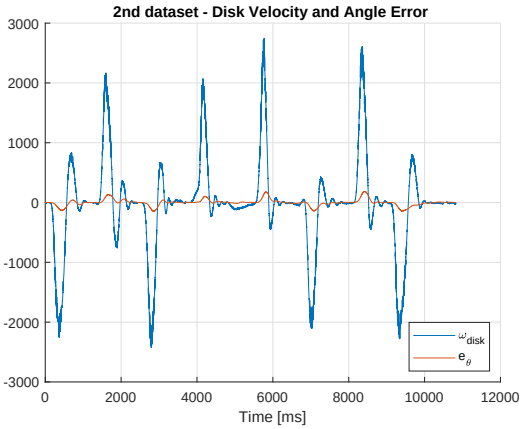


(a) Bode plots showing the frequency response of the closed-loop system using the DAVIS camera, [114].



(b) Bode plots showing the frequency response of the closed-loop system using the eDVS.

Figure C.9: Comparison of the bandwidth of the visual-driven closed-loop systems.



(a)

(b)

Figure C.10: Evaluation of the angle error as function of the rotational speed of the disk.

perience made with this kind of sensor is important, as it will push us to find new possible application fields for this technology.

Bibliography

[1] J. Y. Wong, *Theory of Ground Vehicles*, 4th ed. John Wiley & Sons, 2008.

[2] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.

[3] I. A. Azzollini, N. Mimmo, and L. Marconi, "An Extremum Seeking Approach to Search and Rescue Operations in Avalanches using ARVA," *Accepted to the 21st IFAC World Congress*, 2020.

[4] W. H. Chun and N. Papanikolopoulos, "Robot surveillance and security," in *springer Handbook of Robotics*. Springer, 2016, pp. 1605–1626.

[5] K. Yoshida, "Achievements in space robotics," *IEEE Robotics & Automation Magazine*, vol. 16, no. 4, pp. 20–28, 2009.

[6] M. Bergerman, J. Billingsley, J. Reid, and E. van Henten, "Robotics in agriculture and forestry," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1463–1492.

[7] T. Duckett, S. Pearson, S. Blackmore, B. Grieve, P. Wilson, H. Gill, A. Hunter, and I. Georgilas, *Agricultural Robotics: The Future of Robotic Agriculture*, ser. UK-RAS White Papers. UK United Kingdom: UK-RAS Network, Jun. 2018.

[8] 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*, vol. 26, no. 6-7, pp. 504–518, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20282

[9] F. A. A. Cheein and R. Carelli, "Agricultural robotics: Unmanned robotic service units in agricultural tasks," *IEEE industrial electronics magazine*, vol. 7, no. 3, pp. 48–58, 2013.

[10] D. Slaughter, D. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and electronics in agriculture*, vol. 61, no. 1, pp. 63–78, 2008.

[11] Haverkort, A.J. Ancha Srinivasan (ed), *Handbook of precision agriculture: principles and applications*. The Haworth Press Inc., 2007.

[12] F. Callegati, "Perception and Localization Techniques for Navigation in Agricultural Environment and Experimental Results," Ph.D. dissertation, University of Bologna (UNIBO), 2019.

[13] Y. Edan, D. Rogozin, T. Flash, and G. E. Miles, "Robotic melon harvesting," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 831–835, 2000.

[14] M. M. Foglia and G. Reina, "Agricultural robot for radicchio harvesting," *Journal of Field Robotics*, vol. 23, no. 6-7, pp. 363–377, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20131

[15] G. Bayar, M. Bergerman, A. B. Koku, and E. Ilhan Konukseven, "Localization and control of an autonomous orchard vehicle," *Computers and Electronics in Agriculture*, vol. 115, pp. 118–128, 2015.

[16] M. Bergerman, S. M. Maeta, J. Zhang, G. M. Freitas, B. Hamner, S. Singh, and G. Kantor, "Robot farmers: Autonomous orchard vehicles help tree fruit production," *IEEE Robotics & Automation Magazine*, vol. 22, no. 1, pp. 54–63, 2015.

[17] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. J. Sousa, "Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry," *Remote Sensing*, vol. 9, no. 11, p. 1110, 2017.

[18] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, "A review on uav-based applications for precision agriculture," *Information*, vol. 10, no. 11, p. 349, 2019.

[19] A. Walter, R. Khanna, P. Lottes, C. Stachniss, R. Siegwart, J. Nieto, and F. Liebisch, "Flourish-a robotic approach for automation in crop management," in *Proceedings of the international conference on precision agriculture (ICPA)*, 2018.

[20] M. Mammarella, L. Comba, A. Biglia, F. Dabbene, and P. Gay, "Cooperative agricultural operations of aerial and ground unmanned vehicles," in *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 2020, pp. 224–229.

[21] Cornell Chronicle - Joe Wilensky. (Last access 2020-12-26) Digital ag is Cornell's newest radical collaboration initiative. [Online]. Available: https://news.cornell.edu/stories/2019/02/digital-ag-cornells-newest-radical-collaboration-initiative

[22] J. P. Fritz. (Last access 2020-12-26) Hannibal et ses éléphants n'ont pas encore livré tous leurs secrets (in French). [Online]. Available: https://www.nouvelobs.com/histoire/20180713.OBS9671/hannibal-et-ses-elephants-n-ont-pas-encore-livre-tous-leurs-secrets.html

[23] Wikipedia. (Last access 2020-12-26) Hannibal. [Online]. Available: https://en.wikipedia.org/wiki/Hannibal#Name

[24] ——. (Last access 2020-12-26) Transformes. [Online]. Available: https://en.wikipedia.org/wiki/Transformers

[25] ——. (Last access 2020-12-26) Optimus Prime. [Online]. Available: https://en.wikipedia.org/wiki/Optimus_Prime

[26] S. Blackmore *et al.*, "New concepts in agricultural automation," in *HGCA conference*, 2009.

[27] European Parliamentary Research Service. (Last access 2020-12-26) Small Farms - Statistical Spotlight . [Online]. Available: https://www.europarl.europa.eu/RegData/bibliotheque/stspotlight/2014/140794/LDM_STS(2014)140794_REV1_EN.pdf

[28] Wikipedia. (Last access 2021-04-10) Power take-off. [Online]. Available: https://en.wikipedia.org/wiki/Power_take-off

[29] Nobili SPA. (Last access 2020-12-26) Oktopus - Sprayer. [Online]. Available: https://www.nobili.com/oktopus/s2d24efd2

[30] The Crop Site. (Last access 2021-04-10) How Long Will it Take Farmers to Take Advantage of New Technology? [Online]. Available: https://www.thecropsite.com/news/15717/how-long-will-it-take-farmers-to-take-advantage-of-new-technology/

[31] J. Engström and O. Lagnelöv, "An autonomous electric powered tractor - simulation of all operations on a swedish dairy farm," *Journal of Agricultural Science and Technology A*, vol. 8(3), pp. 182 – 187, 2018. [Online]. Available: http://www.davidpublisher.org/index.php/Home/Article/index?id=36884.html

[32] O. Lagnelöv, G. Larsson, D. Nilsson, A. Larsolle, and P.-A. Hansson, "Performance comparison of charging systems for autonomous electric field tractors using dynamic simulation," *Biosystems Engineering*, vol. 194, pp. 121 – 137, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1537511020300805

[33] Geograph. (Last access 2020-12-26) NH9508: Tractor in the mud. [Online]. Available: https://www.geograph.org.uk/photo/578481

[34] L. Jamar, M. Cavelier, and M. Lateur, "Primary scab control using a "during-infection" spray timing and the effect on fruit quality and yield in organic apple production," *Biotechnol. Agron. Soc. Environ.*, vol. 14, no. 3, pp. 423–439, 2010.

[35] Vulcano SRL. (Last access 2020-12-26) Georgia - Trailed Sprayers. [Online]. Available: http://www.vulcanoatomizzatori.it/ENG/modelli/georgia.html

[36] Nobili SPA. (Last access 2020-12-26) Euro T - Sprayer. [Online]. Available: https://www.nobili.com/euro-t/s1200724c

[37] Martignani SRL. (Last access 2020-12-26) Pull-type Whirlwind M612 (PTO) Sprayers. [Online]. Available: https://www.martignani.com/en/pull-type-whirlwind-m612-pto-sprayers

[38] New Holland Agriculture. (Last access 2020-12-26) T4 F/N/V Tractors. [Online]. Available: https://agriculture.newholland.com/eu/en-uk/equipment/products/agricultural-tractors/t4fnv/specifications

[39] Deere & Company. (Last access 2020-12-26) 5G Series Speciality Tractors. [Online]. Available: https://www.deere.co.uk/assets/publications/index.html?id=9cdbf62e#21

[40] L. Marconi, D. Mengoli, A. Sala, and R. Tazzari, "Veicolo agricolo per agricoltura di precisione," Italian Patent Application 102 021 000 023 093. 07 September 2021. (Submitted).

[41] ——, "Sistema di sollevamento di un veicolo agricolo per agricoltura di precisione e relativo veicolo agricolo," Italian Patent Application 102 021 000 023 096. 07 September 2021. (Submitted).

[42] Honda. (Last access 2021-05-02) GX690. [Online]. Available: https://www.honda-engines-eu.com/documents/10912/16022/TS_GX690

[43] L. Marconi, D. Mengoli, A. Sala, and R. Tazzari, "Veicolo elettrico cingolato," Italian Patent Application 102 021 000 023 099. 07 September 2021. (Submitted).

[44] Sevcon Inc. (Last access 2020-12-26) Sevcon Gen4™. [Online]. Available: http://www.sevcon.com/products/low-voltage-controllers/gen4/

[45] Stanford Research Institute (SRI). (Last access 2020-12-26) Shakey. [Online]. Available: http://www.ai.sri.com/shakey/

[46] N. J. Nilsson, "Shakey The Robot," AI Center, SRI International, Tech. Rep., Apr 1984.

[47] F. Michaud and M. Nicolescu, "Behavior-based Systems," in *Springer Handbook of Robotics*. Springer, 2016, pp. 307–328.

[48] D. Mengoli, R. Tazzari, and L. Marconi, "Autonomous robotic platform for precision orchard management: Architecture and software perspective," in *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 2020, pp. 303–308.

[49] Arduino.cc. (Last access 2020-12-26) Arduino Nano Every. [Online]. Available: https://store.arduino.cc/arduino-nano-every

[50] Trimble Inc. (Last access 2020-12-26) Trimble R8s. [Online]. Available: https://geospatial.trimble.com/products-and-solutions/trimble-r8s

[51] Velodyne LiDAR®. (Last access 2020-12-26) Velodyne Puck™ VLP-16. [Online]. Available: https://velodynelidar.com/products/puck/

[52] F. Callegati, A. Samorì, R. Tazzari, N. Mimmo, and L. Marconi, "Autonomous Tracked Agricultural UGV Configuration and Navigation Experimental Results," in *Workshop on Small UAVs for Precision Agriculture*, 2018.

[53] ROS.org. (Last access 2021-05-02) ROS. [Online]. Available: https://www.ros.org/about-ros/

[54] Wikipedia. (Last access 2021-05-02) Robot Operating System. [Online]. Available: https://en.wikipedia.org/wiki/Robot_Operating_System

[55] D. Project. (Last access 2020-12-26) MAVLink Developer Guide. [Online]. Available: https://mavlink.io/en/

[56] Cincoze Co., Ltd. (Last access 2020-12-26) Cincoze DI-1000-i7. [Online]. Available: https://www.cincoze.com/goods_info.php?id=227

[57] Intel® Corporation. (Last access 2020-12-26) Intel® NUC. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/81164/intel-nuc-kit-d54250wykh.html

[58] Wikipedia. (Last access 2020-12-26) R2-D2. [Online]. Available: https://en.wikipedia.org/wiki/R2-D2

[59] ——. (Last access 2020-12-26) Technology Readiness Level. [Online]. Available: https://en.wikipedia.org/wiki/Technology_readiness_level

[60] R. Tazzari, D. Mengoli, and L. Marconi, "Design concept and modelling of a tracked ugv for orchard precision agriculture," in *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 2020, pp. 207–212.

[61] M. G. Bekker, "Off-the-road locomotion," *Research and development in terramechanics*, 1960.

[62] K. Nagatani, D. Endo, and K. Yoshida, "Improvement of the odometry accuracy of a crawler vehicle with consideration of slippage," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2752–2757.

[63] D. Endo, Y. Okada, K. Nagatani, and K. Yoshida, "Path following control for tracked vehicles based on slip-compensating odometry," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2871–2876.

[64] T. Zou, J. Angeles, and F. Hassani, "Dynamic modeling and trajectory tracking control of unmanned tracked vehicles," *Robotics and Autonomous Systems*, vol. 110, pp. 102 – 111, 2018.

[65] R. Tazzari, I. Azzollini, and L. Marconi, "An Adaptive Observer approach to Slip Estimation for Agricultural Tracked Vehicles," in *European Control Conference (ECC)*, 2021. Accepted.

[66] W. Steeds, "Tracked vehicles," *Automobile Engineer*, vol. 40, p. 526, 1950.

[67] M. G. Bekker, "Theory of land locomotion," 1956.

[68] J. Wong and C. Chiang, "A general theory for skid steering of tracked vehicles on firm ground," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 215, no. 3, pp. 343–355, 2001.

[69] S. Al-Milli, L. D. Seneviratne, and K. Althoefer, "Track–terrain modelling and traversability prediction for tracked vehicles on soft terrain," *Journal of Terramechanics*, vol. 47, no. 3, pp. 151–160, 2010.

[70] B. Wills, "The measurement of soil shear strength and deformation moduli and a comparison of the actual and theoretical performance of family of rigid tracks," *J Agric. Engng Res.*, vol. 8, no. 2, pp. 115–131, 1963.

[71] T. Keller and J. Arvidsson, "A model for prediction of vertical stress distribution near the soil surface below rubber-tracked undercarriage systems fitted on agricultural vehicles," *Soil and Tillage Research*, vol. 155, pp. 116–123, 2016.

[72] W. Liu and K. Cheng, "An analytical model for predicting ground pressure under a rigid-flexible tracked vehicle on soft ground," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[73] C. Samson, P. Morin, and R. Lenain, "Modeling and control of wheeled mobile robots," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1235–1266.

[74] L. M. Roberto Tazzari, Dario Mengoli, "Design and navigation concepts of an ugv for orchard management," in *2020 I-RIM Conference*. I-RIM, 2020, pp. 167–168.

[75] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. [Online]. Available: https://doi.org/10.1115/1.3662552

[76] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, 2004.

[77] M. Bergerman, S. Singh, and B. Hamner, "Results with autonomous vehicles operating in specialty crops," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1829–1835.

[78] A. Costley and R. Christensen, "Landmark aided gps-denied navigation for orchards and vineyards," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 987–995.

[79] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[80] x-io Technologies. (Last access 2020-12-26) Open source IMU and AHRS algorithms. [Online]. Available: https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/

[81] ROS.org. (Last access 2020-12-26) Ros navigation stack. [Online]. Available: http://wiki.ros.org/navigation

[82] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Intelligent autonomous systems 13*. Springer, 2016, pp. 335–348.

[83] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 300–307.

[84] ROS.org. (Last access 2020-12-26) Ros global planner. [Online]. Available: http://wiki.ros.org/global/_planner

[85] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[86] A. S. Hassanein, S. Mohammad, M. Sameer, and M. E. Ragab, "A Survey on Hough Transform, Theory, Techniques and Applications," *ArXiv*, vol. abs/1502.02160, 2015.

[87] S. J. K. Pedersen, "Circular Hough Transform," in *Encyclopedia of Biometrics*, 2009.

[88] D. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111 – 122, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0031320381900091

[89] L. G. Shapiro and G. C. Stockman, *Computer Vision*, 1st ed. USA: Prentice Hall PTR, 2001.

[90] B. Zhou, Y. Peng, and J. Han, "UKF based estimation and tracking control of nonholonomic mobile robots with slipping," in *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2007, pp. 2058–2063.

[91] T. M. Dar and R. G. Longoria, "Slip estimation for small-scale robotic tracked vehicles," in *Proceedings of the 2010 American Control Conference*, 2010, pp. 6816–6821.

[92] F. Rogers-Marcovitz, N. Seegmiller, and A. Kelly, "Continuous vehicle slip model identification on changing terrains," in *Proceedings of RSS 2012 Workshop on Long-term Operation of Autonomous Robotic Systems in Changing Environments*, July 2012.

[93] B. Sebastian and P. Ben-Tzvi, "Active Disturbance Rejection Control for Handling Slip in Tracked Vehicle Locomotion," *Journal of Mechanisms and Robotics*, vol. 11, no. 2, 02 2019.

[94] M. Burke, "Path-following control of a velocity constrained tracked vehicle incorporating adaptive slip estimation," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 05 2012, pp. 97–102.

[95] S. A. A. Moosavian and A. Kalantari, "Experimental slip estimation for exact kinematics modeling and control of a tracked mobile robot," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 95–100.

[96] V. Rajagopalan, Ç. Meriçli, and A. Kelly, "Slip-aware model predictive optimal control for path following," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4585–4590.

[97] J. Yi, D. Song, J. Zhang, and Z. Goodwin, "Adaptive trajectory tracking control of skid-steered mobile robots," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2605–2610.

[98] A. De Luca, G. Oriolo, and M. Vendittelli, "Stabilization of the unicycle via dynamic feedback linearization," *IFAC Proceedings Volumes*, vol. 33, no. 27, pp. 687–692, 2000.

[99] E. Panteley, A. Loria, and A. Teel, "Relaxed persistency of excitation for uniform asymptotic stability," *IEEE Transactions on Automatic Control*, vol. 46, no. 12, pp. 1874–1886, 2001.

[100] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[101] ——, *Nonlinear Control*. Pearson Higher Ed, 2014.

[102] A. Isidori, *Nonlinear control systems*. Springer Science & Business Media, 1995.

[103] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion planning and control*. Springer, 1998, pp. 171–253.

[104] Elife International S.r.l. (Last access 2020-12-26) Elife-Drive - HR series. [Online]. Available: https://www.elifeinternational.com/it/elife-drive/elife-drive-h

[105] J. X. Weinert, A. F. Burke, and X. Wei, "Lead-acid and lithium-ion batteries for the chinese electric bike market and implications on future technology advancement," *Journal of Power Sources*, vol. 172, no. 2, pp. 938 – 945, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775307010294

[106] A. C. Hua and B. Z. Syue, "Charge and discharge characteristics of lead-acid battery and lifepo4 battery," in *The 2010 International Power Electronics Conference - ECCE ASIA -*, 2010, pp. 1478–1483.

[107] Open Source Robotics Foundation. (Last access 2020-12-26) Gazebo - Robot simulation made easy. [Online]. Available: http://gazebosim.org/

[108] M. A. Hossain and J. H. Yin, "Behavior of a Pressure-Grouted Soil-Cement Interface in Direct Shear Tests," *International Journal of Geomechanics*, vol. 14, no. 1, pp. 101–109, 2014. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29GM.1943-5622.0000301

[109] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-dof pose tracking for high-speed maneuvers," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2761–2768.

[110] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128$\times$ 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[111] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "EKLT: Asynchronous Photometric Feature Tracking Using Events and Frames," *International Journal of Computer Vision*, vol. 128, p. 601–618, 2020.

[112] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[113] D. Falanga, S. Kim, and D. Scaramuzza, "How Fast Is Too Fast? The Role of Perception Latency in High-Speed Sense and Avoid," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1884–1891, 2019.

[114] R. S. Dimitrova, M. Gehrig, D. Brescianini, and D. Scaramuzza, "Towards Low-Latency High-Bandwidth Control of Quadrotors using Event Cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[115] iniVation. (Last access 2020-12-26) User Guide - DAVIS240. [Online]. Available: https://inivation.github.io/inivation-docs/Hardware%20user%20guides/User_guide_-_DAVIS240.html

[116] Intel® Corporation. (Last access 2020-12-26) UP board Specifications. [Online]. Available: https://up-board.org/up/specifications/

[117] iniVation. (Last access 2020-12-26) User Guide - eDVS4337 (embedded Dynamic Vision Sensor). [Online]. Available: https://inivation.github.io/inivation-docs/Hardware%20user%20guides/User_guide_-_eDVS_and_PushBot.html

[118] N. Waniek, J. Biedermann, and J. Conradt, "Cooperative SLAM on small mobile robots," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 1810–1815.

[119] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of AER dynamic vision sensors," in *2009 IEEE International Symposium on Circuits and Systems*, 2009, pp. 781–784.

List of Figures