POLITECNICO DI TORINO

Co-simulation of an autonomous driving rover for agro-industrial applications

Master's Degree Thesis in Mechatronics Engineering

Student:
GABRIELE IEVA

Tutors:
Prof. AURELIO SOMÀ
DOTT. SALVATORE MARTELLI
ING. FRANCESCO MOCERA

ACCADEMIC YEAR 2023-2024

Index

List of Figures

List of Tables

1. INTRODUCTION

In the last years, the field of mobile robotics and autonomous vehicles has seen an exponential growth, driven by the need for automation in various sectors, including agriculture, manufacturing, logistics and defence. This expansion is due to the progress of advanced technologies and control systems, which have enabled the development of vehicles capable of operating in complex and dynamic environments without direct human intervention.

The design and optimization of these types of rovers have significant challenges, particularly in accurately modeling their dynamics and controlling motion.

Simulation represents a fundamental tool for addressing these challenges because it is possible to create detailed models that simulate vehicle behaviour in real world conditions.

The use of robotics in agriculture has gained increasing attention in recent times. This trend as already stated in previous studies [1] is driven by several factors that are transforming the agricultural sector globally. The growing demand for products, due to the increase in the world population, requires more efficient and sustainable production methods. Agricultural robots offer a solution to this problem, allowing to increase productivity and optimize the use of resources, such as water, fertilizers and pesticides.

Modern agriculture faces significant challenges related to the lack of manpower, especially in rural areas. Robots, in this context, represent a fundamental resource, useful to carrying out repetitive and physically demanding tasks with greater precision and without the need for breaks. This is useful for reducing operating costs.

The introduction of robots in agriculture is also motivated by the need for more sustainable and environmentally friendly practices. Robotic technologies, in fact, allow for more precise monitoring of crops, helping to reduce the use of chemicals and the environmental impact of agricultural activities. For example, robots can be programmed to identify and treat only the necessary areas, avoiding spraying pesticides on the entire crop. Robots equipped with advanced sensors and technological tools can collect and analyze a large amount of information, providing farmers with a clear and detailed view of the status of their crops. However, the implementation of autonomous rovers requires careful calibration and validation of the models, to ensure that the vehicle can operate safely and reliably in variable and unstructured environments.

This thesis aims to develop an accurate dynamic model of a differential traction rover, using a combination of kinematic and dynamic simulations trough MATLAB/Simulink® and MSC/Adams® software. One of the motivations behind this work is to achieve a more realistic modeling of the autonomous driven vehicle through co-simulation. The second motivation is to create a simulation platform that allows for the development and thorough testing of control logic in a virtual environment, before to implementation on a physical prototype.

One of the key aspects of this work is the integration of differential steering control, which allows the rover to follow predetermined paths with high precision. This control is achieved through a path following algorithm implemented in Simulink, which adjust the speed of the wheels based on the waypoints generated by the path planning in MATLAB. The ability to accurately follow a path is crucial for many applications, including autonomous navigations in agricultural environments, where the terrain can present unexpected obstacles and surface variations.

The main motivation of this thesis is to develop an integrated simulation framework that can be used to design, test and optimize differential drive rover for autonomous applications in agriculture environments. This work will not only contribute to the understanding of the dynamics of autonomous vehicles but will also provide practical tools for developing innovative solutions for more applications systems.

The structure of this thesis is designed to guide the reader through the research and development process, starting with a general overview and progressing to technical details and results.

The Introduction establishes the context and motivation for the research, highlighting the importance of autonomous vehicle simulation and the use of co-simulation between Simulink and Adams. It addresses the challenges in autonomous driving, especially for differential-drive vehicles, and outlines the specific goals of the thesis aimed at overcoming these challenges. This chapter also highlights the personal and academic motivations behind the choice of this topic, underlining its practical and theoretical significance.

The state of the art chapter reviews the current landscape in three key areas: emerging trends in vehicle dynamics, autonomous driving technologies and path tracking algorithms, and the integration of simulation and multi-body software for vehicle dynamics simulation. This chapter highlights relevant developments, studies, and advantages of the co-simulation approach.

The Materials and Methods chapter details the design and implementation process, starting with the algorithm specifications, which outline the rationale behind the vehicle motion models and the technical goals of the project. It then explains the development of the kinematic model in Simulink, followed by the integration of the dynamic model in Adams to enable co-simulations that validate the rover performance in virtual environments.

In the Results and Discussion chapter, the simulation results are presented and analyzed. The Results section compares the rover performance in different scenarios, while the Discussion interprets these results, evaluating the effectiveness of the co-simulation approach and addressing any discrepancies between expected and actual results.

The thesis ends with a Conclusion chapter that summarizes the main results, discusses the implications of the research and evaluates the achieved goals. It also offers suggestions for future research and development in the field of autonomous vehicle simulation.

2. STATE OF THE ART

The agricultural industry is constantly evolving with the necessity to develop more safer, efficient, and environmentally friendly vehicles.

The simulation in this context is crucial on the development process, allowing to study dynamic behaviour of vehicles in virtual conditions to have more information before moving to the physical prototyping phase. An aspect that is critical to this simulation is the ability to model in accurate way the vehicle dynamics in various road and environmental conditions including driving situations like manoeuvres and non-ideal driving conditions. This integration of multi-physics simulation software like Simulink and Adams offers a complete solution to do and enabling a detailed simulation of differential drive vehicle in this case.

The combined use of these kind of software allow to create complex models that can be used to simulate a wide scenario. In this work the vehicle is a self-driving agricultural rover that must operate in vineyards and orchards. In addition, the integration of these software enables analysis of vehicle performance allowing to optimize vehicle parameters and dynamic behaviour.

Through a combination of modeling on Simulink and rigid body dynamics analysis on Adams, this work will contribute, with data analysis, the understanding and advancement of vehicular simulation methodologies to develop environmentally friendly vehicles.

In the last years, the simulation of mechatronics machines using co-simulation has seen progress. These developments have been driven by a combination of technological improvements, growing the industrial needs and more awareness on the accuracy of these studies.

The integration between simulation and multi-body software allows to use dynamic system modeling capabilities of Simulink with the advanced mechanical simulation of Adams. This permits to obtain more detailed and accurate analyses of vehicle dynamics, including not only forces and torques, but also inertial behaviour, for example the tyre and the vehicle inertia. Recently more studies have shown that including accurate models can significantly improve the accuracy of simulation.

Nowadays the computing power permits real-time simulations allowing dynamic vehicle models to be tested and validated in virtual environment before their real word implementation. This permits a reduction of costs and lower development times, but there is also more safety.

The integration of Artificial Intelligence and Machine Learning it was crucial in vehicle simulation recently. These technologies are used to improve prediction of the vehicle dynamics and help to build advanced control systems. Machine learning algorithms can be implemented to analyse more data simulations at the same time and identify characteristics that can be not evident with traditional analysis.

In the last years artificial intelligence (AI) technologies have become fundamental for the developing of vehicle systems and, specially, in autonomous vehicles. The study of these technologies is particularly focused in areas, such as perception, sensor data fusion, path planning and following that are important for the safety and obtaining an efficient autonomous drive. Motion planning is an active research domain that involves more actions to achieve goal, using sensor data fusion to detect obstacles and relevant information.

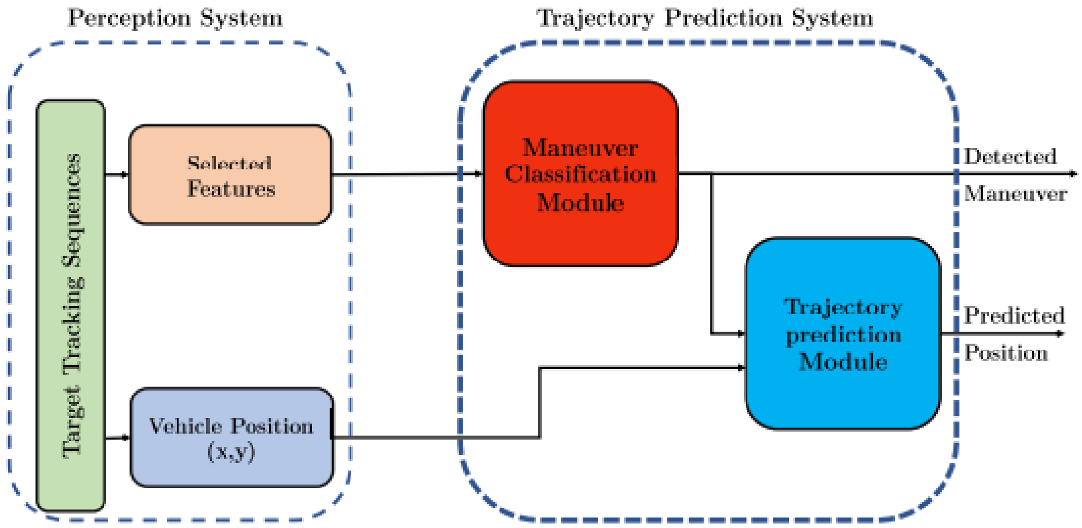


Figure 1: Perception system integrated with Trajectory prediction system.

Benterki [2] describes a hybrid model for trajectory prediction indicated in figure 1 based on a combination of manoeuvre classification by Artificial Neural Network (ANN) and trajectory prediction by Long Short-Term Memory (LSTM). This considers the ability of neural networks to learn human behaviour patterns to give more accuracy at the movement.

These studies shown how AI technologies into vehicle simulation can significantly improve the ability to predict and react at dynamic behaviours and make safer vehicles.

For the study of testing and improvement of autonomous vehicles, real-time simulations are becoming more important, especially integrating hardware-in-the-loop techniques (HIL). The real-time simulations permit to test and validate control algorithms using virtual environments before implementing them in real word, with the aim to reducing risks and costs associated with real prototyping. Recently there were improvements in computational context that have permitted to simulate models that runs at real-world speed.

Real-time simulation and hardware-in-the-loop (HIL) testing are key tools for the development and validation of control systems in autonomous vehicle. Software these days are complicated, and for this reason, they require too many tests to achieve high level of quality and safety.

According with the analysis in the article [3], a fleet of 100 vehicles operating 24 hours a day, 365 days a year at 40km/h would require 17 billion test kilometers and would take 518 years to validate the software with a confidence of 95%, considering the fatality rate better than 20% with respect to the actual rate. To reduce costs and time, HIL simulation is used to complement in-vehicle testing. In autonomous drive, the path following controls are essential to obtain the lateral control of the vehicle.

The objectives of the research [3] were three. The first was to implement a real-time version of the lateral path-following controller to add lateral capability to a HIL configuration based on longitudinal control of the powertrain. The second was to validate the path following capability of a lateral controller. The final point was to understand the real-time behaviour and sensitivity on the lateral controller using simulations with different inertia, environmental conditions, speed, load position exc.

These studies demonstrate the importance of real-time simulation and HIL testing for the validation of autonomous vehicle control systems making possible complicated scenarios.

The integration of detailed multibody models related to control models allows to simulate mechanical dynamics and the control strategies that govern them. This cooperation between two different software is useful in particularly for developing and testing different advanced driver assistance systems (ADAS) and control algorithms for autonomous drive. This allow to understand how control systems influence vehicle dynamics in real scenarios, and with this method there are improvement in the precision and accuracy of the designed model.

The integration of multibody and control model with the co-simulation represents a very important improvement in the field of vehicular simulation, allowing the advantages of different techniques of modeling. This approach is useful for analyzing complex models and predict vehicle performances.

An example of this methodology shown in the article [4]. In this work, the co-simulation of a full 3D vehicular model developed in MSC Adams with a 1D transmission model developed in LMS AMESim. The aim is to investigate vehicle NVH (Noise, Vibration, Harshness) error states associated with both hybrid and non-hybrid drive systems. AMESim job's is to generate engine, damper and transmission excitations, and is capable to model the control strategy for powertrain component operations. Every mechanical component as engine block, engine mounts, suspension axle, axle shaft, wheels and body are modelled in Adams and imported as functional unit into AMESim.

Another study concerns the use of semi-active magnetorheological (MR) shock absorbers to predict the dynamic response of a vehicle. The used approach has the job to exploit the capabilities of multibody system (MBS) code and a mathematical simulation code, integrating MBS vehicular models with selected models of semi-active shock absorbers and their controllers. More MBS vehicle models are developed using MSC.visualNastran and linked to three local two-state control algorithms and two global controllers developed in MATLAB/Simulink. The control strategies are implemented in the vehicle model using an MR damper model derived from experimental test data. Road inputs, bumps/potholes and random holes excitation, and the tire model are implemented also in MATLAB/Simulink.

An example of integration of multibody and control models is a study on fuzzy control-based active front steering. A dynamic model of the active steering system with dual planetary gear mechanism and the complete vehicle model were constructed using Adams. The strategy of fuzzy control considers the lateral slip angle of the center of mass and the yaw rate of the vehicle as the input of the fuzzy control rules, and the output is the engine angle in active steering system. The co-simulation of the control system is obtained by the exchange of data between ADAMS and MATLAB, demonstrating that the fuzzy control method can significantly improve the lateral stability of the vehicle.

These studies demonstrate how multibody, and control co-simulation can be an optimal way to optimize and validate complex model systems. It gives improvements in the ability to predict and decrease errors issues and develop a safer system.

In this period the interest of sustainability and energy efficiency is increased and has led a development of simulation models that consider the environmental impact of different vehicle configurations and the application of operational strategies. There are many studies that consider that, in particularly in agriculture, that shows the real impact of agricultural machinery and the way to follow to contain these effects.

The use of advanced technologies is transforming the agriculture, making agricultural practices more precise and sustainable. In particular, the internet of things (IoT) and unmanned aerial vehicles (UAVs) are revolutionizing the way farmers monitor and manage their crops. These innovations allow for an efficient use of resources improving quality and quantity of crops.

In the article [7] there is a study that explores the integration of IoT in the agriculture, analyzing how sensors and other advanced technologies can be used to monitor crops in real time. The article shows that farmers can use sensors to detect soil health, moisture, temperature, and other critical parameters, allowing scheduled interventions. An example is a sensor system can alert the farmer to a nutrient deficiency in the soil, allowing fertilizer to be applied more precisely.

The article discusses also of the use of UAVs for crop monitoring. These drones can fly over fields, capturing high-resolution images for analyze and identify problems. The use of UAVs allows large areas to be covered quickly and with great precision with respect to traditional methods.

The research shows how these technologies can be integrated into agricultural vehicles, to optimize field operations. For example, tractors with IoT devices can collect and analyze data in real time, automatically adjusting specific working conditions. This approach could improve overall energy efficiency.

Autonomous driving represents one of most important innovations in automotive sector. The aim is developments of vehicles capable of operating without human intervention, using a combination of sensors, algorithms, and powerful control systems. There is shown the core technologies for autonomous driving, path following algorithms, and the challenges and solutions that there are in this field. Understanding these arguments is important to develop safer and efficient vehicles.

This field is based on more components that work in synergy to allow the vehicle to perceive its surroundings, make decisions and navigate autonomously. These components include sensors such lidar, radar and cameras, which provide real-time data on road conditions and obstacles. The algorithm path has a key role in processing this data, allowing the vehicle recognize objects, predict behaviours, and plan routes. Furthermore, control systems, ensures that algorithm decisions are executed precisely. This part will examine each of these technologies in detail, particularly their applications in autonomous vehicles.

The autonomous vehicles (AVs) are equipped with advanced systems that allow complete automation driving. The article [8] explain that the market for self-driving vehicles is rapidly expanding: the number of vehicles with some level of automation is expected to grow from 31 million by 2024. Although the market suffered a decline of 3% in 2020 due to Covid-19 pandemic, it was a growth of 60% between 2020 and 2023. Advanced electronics, computing and communications technologies have favoured the development of various technologies for autonomous vehicles. However, autonomous driving systems face significant challenges, such as driving safely in adverse weather conditions and interactions with pedestrians and other vehicles.

One of the main benefits of autonomous driving is the promise of significantly increased safety. Most accidents are caused by human error, such as distraction or fatigue. Autonomous vehicles, on the other hand, do not get distracted or tired, making them potentially much safer. Additionally, autonomous vehicles can communicate with each other, exchanging data that are useful for the job at hand.

In addition to safety, autonomous driving also offers advantages in terms of efficiency and sustainability. Autonomous vehicles can optimize routes, reduce fuel consumption and limit greenhouse gas emissions. They could also revolutionize logistics and transportation, reducing delivery costs and making public transportation more accessible and efficient.

However, the extended adoption of autonomous driving is not without challenges. The technology still must overcome some technical obstacles, such as the ability to operate in adverse weather conditions or in complex, unstructured environments. There are also legal and ethical issues to consider, such as responsibility in the event of an accident and the impact on the labour market, especially in the transportation and logistics sectors.

Sensors are fundamental devices in AVs because they detect events in the environment, after there is transformation of these in numerical values, and then they are processed. They are divided into two main categories: internal (proprioceptive) sensors, which record the dynamic state of the vehicle, and external (exteroceptive) sensors, which perceive information from the external environment. Proprioceptive sensors include IMUs, encoders and GNSS receivers, while exteroceptive sensors include cameras, radars, LIDARs, and ultrasonic sensors. These are very important for managing the movements of the AV and for environmental perception, vehicle localization and route planning. Below there is an explanation of the most important devices:

- **Camera** is the most known method to observing the environment. These devices produce sharp images by detecting light emitted from the surrounding environment. The operation of cameras is based on a photosensitive (image) plane that captures light through a lens placed in front of the sensor. This process allows for the generation of high-resolution images that can be used to detect moving and stationary obstacles in the field of view.
- **LiDAR** (Light Detection and Ranging) is a remote sensing technology that uses pulses of laser light to measure distances and create detailed maps of environment. LiDAR has been widely used in aerospace and topographic mapping applications. In Autonomous vehicles, LiDAR emits pulses of light that reflect off surrounding objects. This process generates a cloud of data points (PCD) which represents the environment three dimensionally.
- **Radar** (Radio Detection and Ranging) is the technology that uses electromagnetic waves to detect and locate objects. Radar emits electromagnetic waves that reflect off objects and return to sensor. Analyzing

these reflections, the radar can determine the distance, speed, and direction of detected objects.

Despite technological advances, autonomous vehicle sensors need to face several challenges. One of the main difficulties is designing reliable and robust sensors to ensure precise and accurate measurements. The problem that wireless sensor needs to face is for example the selection of appropriate hardware, network calibration, and data synchronization. LiDAR sensors do not provide colour information, requiring data to be fused with other sensor using advanced algorithms. Additionally, radars, while effective in adverse weather conditions, have lower resolution than cameras, limiting their ability to recognize objects on detail. Adverse weather conditions like snow, fog and rain, can negatively affect the operations of all sensors, reducing their effectiveness.

The Path Following is an essential topic to ensuring that an autonomous vehicle can follow a predefined trajectory accurately and safely. These algorithms use data from vehicle sensors to determine the current position and compare it to the desired route, making real-time corrections to keep the vehicle on the right path. Common method includes PID control, model predictive control (MPC), and techniques that consider geometrical formulas. The problem of motion planning and control of mobile robots is an important research area, with big relevance in applications. The studies founded in the literature are the following.

Adaptive motion control is a technique designed to enhance the accuracy of a robot's movements by reducing system motion errors through learning from position measurements. In the article [5], an adaptive motion controller is proposed to improve the precision of path following. This approach utilizes a control model that dynamically adjusts to changes in operating conditions, thereby improving the robot's ability to follow a predetermined path with greater accuracy. The adaptive controller is designed to correct errors in real-time, continuously optimizing the robot’s trajectory. This type of controller is particularly beneficial in dynamic environments where conditions can change rapidly, such as in areas with varying terrain or unpredictable obstacles. By learning from the robot's performance and adjusting as needed, adaptive motion control systems ensure that the robot can maintain its intended path more reliably.

The unicycle model is a simple yet effective framework for understanding and controlling the motion of mobile robots. In the article [11], a unicycle model based on Lyapunov theory is presented, which is capable of steering, planning routes, and navigating in a 2D plane. This model primarily considers the kinematics of the robot, simplifying the control problem by focusing on the relationship between the robot's position, velocity, and orientation. The kinematic equations of the unicycle model are:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{1}$$

The parameter v is the linear velocity of the unicycle model, ω is the angular velocity, and θ is the orientation angle of the robot. By applying these equations, the unicycle model can effectively steer the robot along a desired path. The backstepping control technique is used to design controllers that ensure the stability of the robot's motion. This method provides a systematic way to construct control laws that drive the robot to follow the desired trajectory.

Inverse kinematics is a fundamental technique in robotics used to determine the necessary joint movements to achieve a desired end position. In the article [12], an inverse kinematic model is implemented to directly control a two-wheeled robot, allowing it to reach any desired position by calculating the necessary speed for both wheels. The inverse kinematic model is based on the following equations:

$$\begin{cases} vl = 2 * R * v - (\omega * L)/(2 * R) \\ vr = 2 * R * v + (\omega * L)/(2 * R) \end{cases} \tag{2}$$

where vl and vr are the speeds of the left and right wheels, v is the linear speed of the robot, ω is the yaw speed, L is the track and R is the radius of the wheels. This model allows precise control over the robot's movements by relating the wheel speeds to the desired trajectory. By adjusting the speeds of the wheels, the robot can navigate to any target position with high accuracy.

The article [13], had proposed the use of PID control for tracking reference paths of a differentially driven mobile robot. The kinematic and dynamic modelling of the robot was presented, with both the master and slave controllers configured as PIDs. The control parameters obtained from the simulation were optimized for hardware implementation using PCB-mounted potentiometers. The cascade control structure is based on two control levels: the master controller, which regulates the orientation of the robot, and the slave controllers, which control the angular speeds of the wheel. The kinematic equations of the robot are:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = (vr - vl)/L \end{cases} \tag{3}$$

The parameters vr and vl are the speed of the right and left wheels respectively, and L is the axial distance between the wheels. The master controller receives the desired orientation of the robot as input and generates set points for the slave controllers. Slave controllers regulate the input to the DC motors to maintain the desired angular speed of the wheels. This approach improves the dynamic response of the system and the robot's ability to follow complex paths with precision.

Collision Avoidance in Path Following argument is a widely used for manipulators and mobile robotics. In the article [8] collision avoidance techniques are classified into two main categories that are perception and action. Regarding the perception category, several sensors receive environmental information to detect obstacles. In contrast, the action category has four main methodologies: geometric, force field, optimized, and sense of avoid. The strategies of collision avoidance can be further divided into artificial intelligence methods, repulsive artificial potential functions, motion planning, repulsive vector fields, and biologically inspired algorithms. One of the first works addressing collision avoidance was developed using artificial potential field methodology for obstacle avoidance for both manipulators and mobile robots. This topic is an open field in mobile robotics, for example Ramírez-Neria et al., in their article [14] propose a control law for collision avoidance between two mobile robots. The attractive control law uses a coordinate change with a reference point on the front of the robot, a commonly implemented method to control differentially driven mobile robots to avoid singularities in the control law. The repulsive control law uses the dynamics of distance and angle formation between robots, with the aim of avoiding collisions within a region of radius d_{ij} as indicated in figure 2.

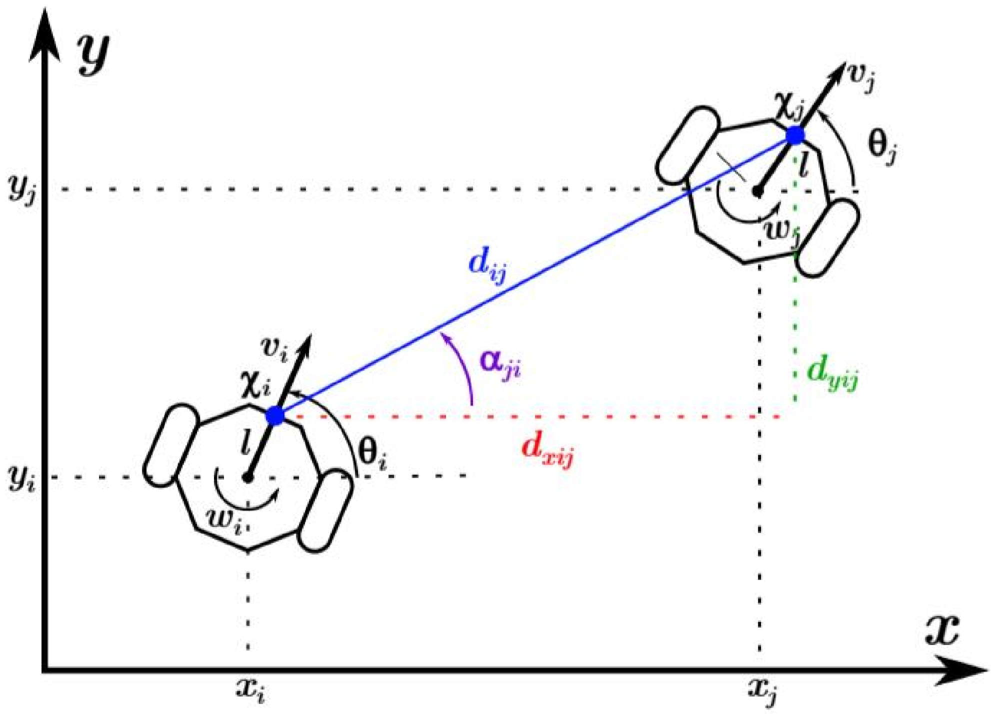


Figure 2: Kinematic model.

The proposed control law has two modes: the first uses negative feedback to maintain a stable distance between the robots and a desired angle, while the second uses positive feedback to produce a counterclockwise angular motion that moves the robot away from each other, allowing them to avoid a collision.

The path-following control is a key issue in controlling the movement of autonomous vehicles, including unmanned aerial vehicles, wheeled land vehicles, and autonomous underwater vehicles. The main goal of path-following control is to design control laws that allow a mobile robot to follow a precisely planned geometric path. Unlike tracking control that follows a time reference, path-following algorithms focus on handling implicit functions or timeless parametrization of geometric paths.

In article [11], is introduced a new approach for path-following control planar applicable to nonholonomic systems. Nonholonomic systems, such as wheeled mobile robots, are characterized by motion constraints that cannot be solved simply by integrating velocities. This makes the control of such systems particularly complex. In their study, Song et al. propose an innovative solution that exploits a combinations of advanced control techniques to address the challenges posed by non-holonomic constraints. They introduce a vector field-based method to generate direction commands that guide the robot along the desired path. This method involves building a vector field around the planned path, where each vector indicates the direction in which the robot must move to stay on the path. One of the many innovations of this approach is the ability to manage complex paths and adapt to different path and adapt to different path geometries, such as straight lines, circular orbits, or combinations of both. This method is important for its robustness against disturbances, such as wind, which can influence the robot's trajectory. The path following control based on vector fields proposed by Song et al offers: Flexibility because can be applied to a variety of geometric paths without the need of controller redesigned, Robustness because the system is able to maintain the path even in the presence of external disturbances, and ease implementation because its uses algorithms that can be implemented in real time, making it suitable for practical applications.

One of the practical implementations of path-following algorithms, is the Pure Pursuit algorithm. Pure Pursuit is a geometrical path-following method that is straightforward and effective for following curved paths. The Pure Pursuit algorithm as shown in figure 3 uses the current position of the vehicle and a look-ahead point on the desired path to determine the required steering angle. This look-ahead point is a fixed distance ahead of the vehicle on the path, and the steering angle is calculated to minimize the distance between the vehicle and this point.

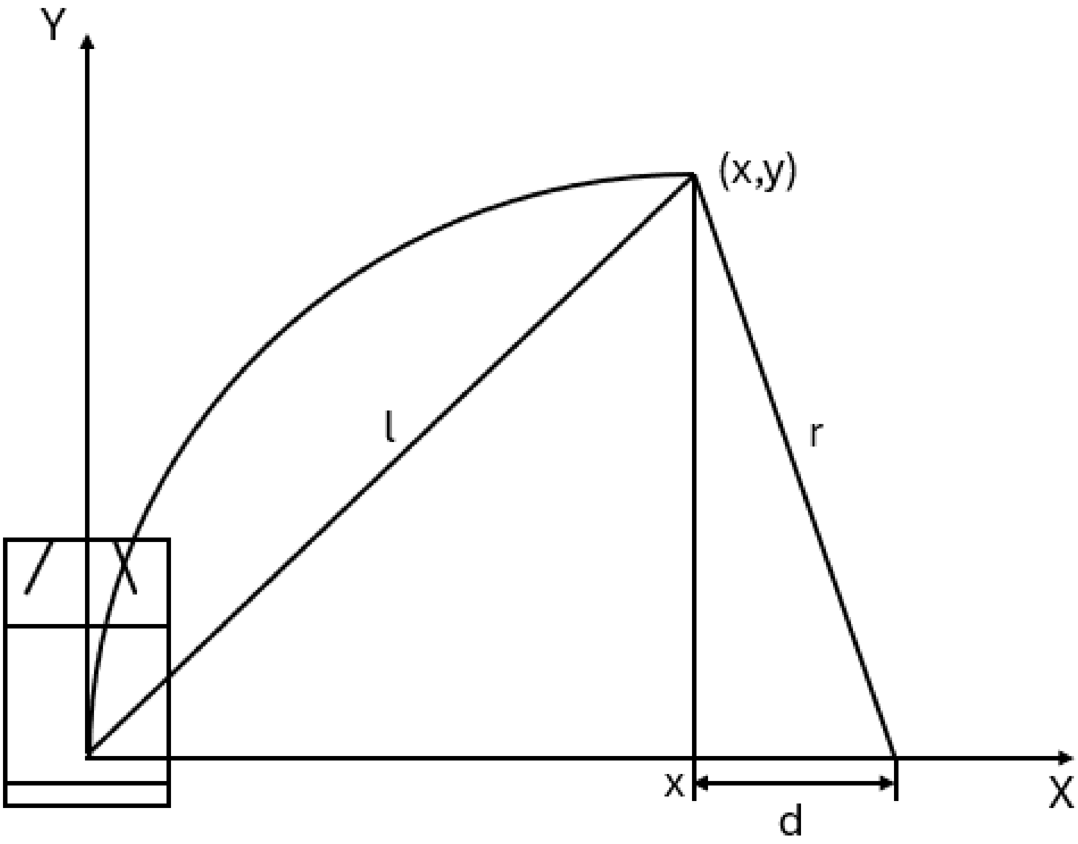


Figure 3: Geometric scheme of Pure pursuit.

The algorithm works as follows:

- **Determine Current Position and Orientation:** The vehicle's current position and orientation are obtained from sensors or a simulation model.
- **Select Look-Ahead Point:** A point on the path at a fixed distance ahead of the current position is selected as the look-ahead point.
- **Calculate Steering Angle:** The steering angle required to move towards the look-ahead point is calculated based on the geometric relationship between the current position and the look-ahead point.
- **Update Vehicle Controls:** The calculated steering angle is applied to the vehicle's steering mechanism to adjust its trajectory towards the look-ahead point.

The Pure Pursuit algorithm in MATLAB is implemented using functions that continuously update the look-ahead point and the steering angle as the vehicle moves along the path. This approach ensures that the vehicle can smoothly follow curved paths and adjust its direction as needed to stay on track. The simplicity and effectiveness of the Pure Pursuit algorithm make it a popular choice for path-following tasks in autonomous vehicle projects.

The vehicle autonomy in agriculture presents unique and complex challenges that require innovative solutions to improve the efficiency and accuracy of agricultural operations. These challenges are related to the unique characteristics of agricultural environments, which include the presence of natural obstacles, terrain variability and the need for precise and safe operations in the absence of GPS. For this situation there is an examination of challenges and emerging solutions through the recent developments in Unmanned Aerial Vehicles (UAV) technology drawing from the article [15], but applicable to all autonomous vehicles.

Challenges in Agriculture:

- **Sensor and Autonomy Limitation:** Commercial agricultural instruments are advanced but are often designed for less complex work. Most of these systems depends on GPS for long range operations, and this make them less suitable for complex agricultural environments where the GPS signal can be weak or absent.

- **Sensor configuration:** Sensors must perform a dual role that are ensuring autonomy and collecting work specific data. Commonly used sensors include cameras, IMUs (Inertial Measurement Units), LIDAR, and GNSS (Global Navigation Satellite Systems). However, each sensor has advantages and limitations. For example, cameras and IMUs are lightweight and energy efficient, but suffer from scaling ambiguity and require accurate calibration. In contrast 3D LiDAR's offer rich information but are heavy and expensive.

- **Obstacle detection Problems:** Obstacles such as thin branches are difficult to detect and can cause accidents if not avoided. The limitation of payload and computing capacity makes it difficult to solve adding more sensors.

Advanced Solutions for Autonomous Driving:

- **Using Data Driven Approaches:** Deep learning techniques are gaining ground in precision agriculture, enabling the detection and segmentation of objects such as fruits and trees from 2D images and 3D LiDAR data. These approaches significantly improve detection accuracy, making them preferable for complex, large scale agriculture environments.

- **Semantic Mapping:** Semantic mapping adds semantic concepts to geometric maps, creating a more efficient representation for human interaction and planning. This approach not only facilitates the creation of detailed and accurate maps, but also improves the estimation of the robot's position, reducing ambiguity and improving drift correction and loop closure, that are critical elements in agricultural environments.

- **Improved Localization Systems:** Localization and mapping based on semantic features improve the accuracy of robot position estimation. Using semantic objects as local descriptors reduces ambiguity and improves drive correction, where traditional geometric features may give low efficiency.

- **Intuitive User Interfaces:** A good and intuitive user interface is essential to regulate access to complex systems in agriculture. Making these technologies accessible to farmers through user friendly interfaces is a very important challenge.

Recent advantages in sensors, sensing methods, and semantic mapping techniques are opening new possibilities for precision agriculture. With the continued development of these technologies, autonomous vehicles promise to revolutionize agricultural practices, increasing efficiency and sustainability.

Co-simulation represents an advanced and highly efficient technique for developing complex mechatronic systems, integrating various subsystems such as mechanics, actuators, sensors, and control systems. The main goal of this method is to enable the design, simulation and testing of virtual prototypes using specialized software, significantly reducing the time and resources required to the development.

Co-simulation is based in the integration of dynamic models of mechanical systems with control models. A representative example of this methodology is the combined use of multi-body and control logic programming software such as Adams and MATLAB/SIMULINK.

The integration of these two software occurs through a continuous exchange of data between dynamic model and the control system. This two-way interaction allows to test and optimize system behaviour realistically and accurately, this improving the reliability of results.

A significant example of this technique is the work of T. Brezina, Z. Hadas and J. Vetiska [16] which talks about the simulation of a 3 degrees of freedom manipulator. In this scenario, the mechanical model of the manipulator is built in Adams View, including details such as bodies, joints, motors, and applied forces. The Adams dynamic model is then imported into Simulink using the Adams/Controls Toolkit, which allows to define the state variables of the model and connect them to the control system developed in Simulink. During co-simulation, Adams calculates the dynamic behaviour of the manipulator in each simulation step, while Simulink manages the control of the manipulator's DC motors. Model outputs from Adams, such as the position and speed of the motors, are provided to Simulink, which in turn processes the control signals and sends drive voltages to the motors. This iterative process continues until the simulation is complete, allowing to analyze and optimize the performance of the overall system.

The advantages are:

- **Accuracy:** Considers the mechanical dynamics and performance of the control system in detail.
- **Efficiency:** Facilitates optimization of system parameters through iterative simulations.
- **Flexibility:** Permits to test different operational scenarios and system configurations without the need to build physical prototypes.

This method represents a powerful way to developing mechatronics systems, combining the precision of dynamic simulation with the flexibility of modelling and control. This technique significantly improves the efficiency of the design process, reducing the time and costs associated with developing physical prototypes and enabling a more complete evaluation of system performance.

The mechatronic system of an autonomous vehicle with differential drive involves a sophisticated integration of various software tools, in particular Simulink and Adams as shown in figure 4. This integration is essential for effective design, simulation and control of vehicle operations, ensuring that both control algorithms and mechanical components work harmoniously.

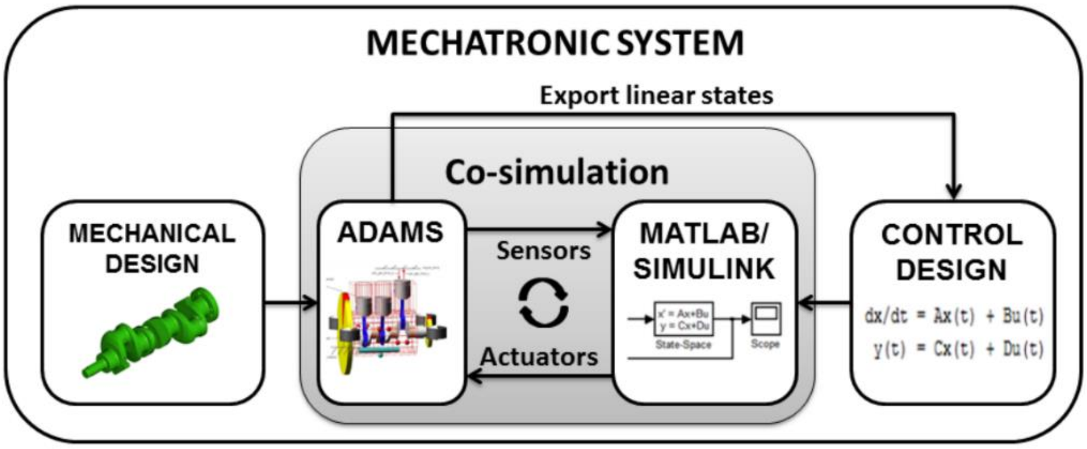


Figure 4: Co-simulation of mechatronic system.

The control algorithms are based on accurate and realistic representations of the vehicle's mechanical behaviour. In Simulink, there is developing and refining of the control algorithms that govern vehicle operations. Simulink's powerful simulation capabilities permit to test these control strategies in a virtual environment, adjusting as needed to improve performance.

Once control algorithm is refined in Simulink, control signals (such as actuator commands) are sent to Adams. This feedback loop ensures that mechanical components respond appropriately to control inputs and that any discrepancies can be identified and corrected. This continuous data exchange and continuous refinement process between Simulink and Adams ensures that the autonomous vehicle operates efficiently, safely and reliably, with the control systems and mechanical components working in perfect harmony.

Improving the efficiency of hybrid vehicles has been a significant focus of research and development, leveraging advanced simulation techniques to optimize performance and fuel economy. One of the pioneering applications of Simulink-Adams co-simulation was in the study and development of hybrid vehicles, aimed at improving their fuel efficiency. In Brian Su-Ming's research [17], a hybrid model based on the Honda Integrated Motor Assist (IMA) architecture was proposed. This model incorporates an electric motor that complements the torque of the internal combustion engine, thus improving the overall performance and efficiency of the vehicle.

In this co-simulation framework, MATLAB/Simulink is used to model powertrain components, including the engine, electric motor, battery, and energy management systems. The role of Simulink is crucial as it manages the power management controller, responsible for managing the power distribution between the combustion engine and the electric motor. This controller makes real-time decisions to optimize the use of the electric motor during different driving conditions, such as acceleration, cruising, and braking.

Adams, on the other hand, is employed to model the mechanical components of the vehicle. This includes detailed simulations of vehicle dynamics, such as the response of the suspension system, the interaction between the tires and the road surface and the overall behaviour of the vehicle in various operational scenarios. By providing a detailed and realistic representation of the mechanical aspects of the vehicle, Adams give to accurately predict how the vehicle will behave in real-world conditions. In figure 5 is represented a conceptual example of software integration.

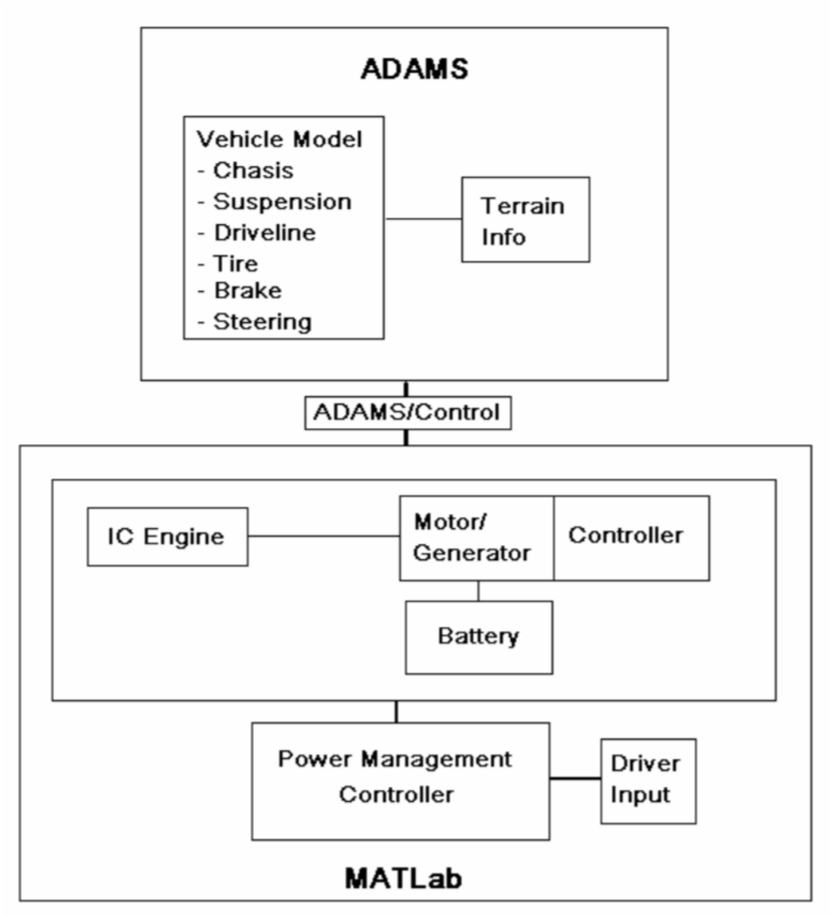


Figure 5: Adams-Matlab integration.

The integration between Simulink and Adams is facilitated by a control design framework that enables seamless data exchange between the two platforms. In this co-simulation setup, the vehicle model created in Adams is connected to the power management controller designed in Simulink. The Adams mechanical model simulates the physical behaviour of the vehicle, including forces, torques and motions, while the Simulink power management controller adjusts power distribution based on these dynamics.

One of the significant findings of Brian Su-Ming's research was the demonstration of a substantial improvement in fuel economy using regenerative braking and intelligent powertrain control. Regenerative braking is a process in which the vehicle's kinetic energy, usually lost as heat during braking, is instead captured and converted into electrical energy, which is then stored in the battery. This stored energy can later be used to power the electric motor, reducing demand on the combustion engine and thus improving fuel efficiency.

The intelligent powertrain control developed in Simulink considers various factors such as the state of charge of the battery, current driving conditions and desired performance characteristics. By optimizing the use of both the combustion engine and the electric motor, the power management controller ensures that the vehicle operates as fuel-efficiently as possible.

Through the combined use of Simulink for the control systems and Adams for the mechanical simulation, the researchers were able to create a highly accurate and effective model of a hybrid vehicle. This model not only demonstrated the potential for significant fuel savings, but also provided a solid framework for the further development and optimization of hybrid vehicle technologies. The co-simulation approach enabled comprehensive testing and validation of different control strategies and mechanical designs, leading to a deeper understanding of how to improve the efficiency and performance of hybrid vehicles.

Research conducted by Li Shengqin and He Le [18] explored the application of Simulink-Adams co-simulation to improve vehicle stability control. This study sowed in figure 6 used the strengths of Adams/Car and Simulink to create a comprehensive simulation environment that accurately models and controls vehicle dynamics.

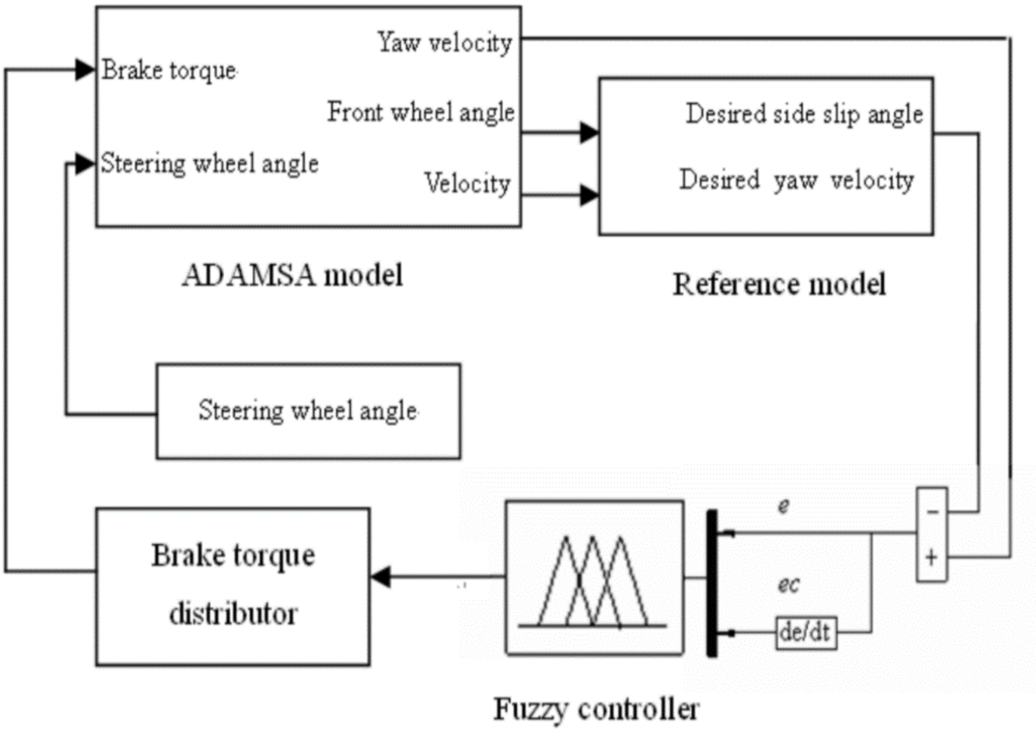


Figure 6: Block scheme with closed loop strategy and feedback Brake Torque.

In this research, Adams/Car was employed to simulate the dynamic behaviour of the vehicle. Simulink, on the other hand, was used to implement advanced control strategies, focusing on vehicle stability control. The control system designed in Simulink featured a fuzzy controller. This makes them particularly suitable for applications such as vehicle stability control, where system behaviour can be unpredictable and highly dynamic.

The co-simulation setup involved creating a closed-loop system in which vehicle dynamics modeled in Adams/Car were continuously monitored and adjusted by control strategies implemented in Simulink. Simulink's fuzzy controller processed this data in real time to determine necessary changes to electronic control systems, such as the electronic stability program (ESP).

ESP is an advanced electronic control system designed to improve vehicle stability by detecting and reducing loss of traction. When the fuzzy controller senses that the vehicle is about to lose stability, it can apply brakes to individual wheels and reduce engine power as needed to help the driver maintain control. The fuzzy controller's ability to respond quickly and effectively to changing conditions makes it a powerful tool for improving vehicle safety during emergency maneuvers.

The co-simulation approach allowed for extensive testing and validation of control strategies under a wide range of conditions, ensuring that the vehicle could maintain stability even during extreme manoeuvres.

In the research conducted by Liu S. L. and Wang Y. [19], Simulink-Adams interaction was applied to simulate the behaviour of differential traction vehicles. This approach shown in figure 7 aimed to better understand the dynamic behaviour of such vehicles under different road surfaces and driving conditions. The integration of Simulink and Adams allows to create a complete simulation environment in which both mechanical dynamics and control logic can be precisely modeled and analyzed.

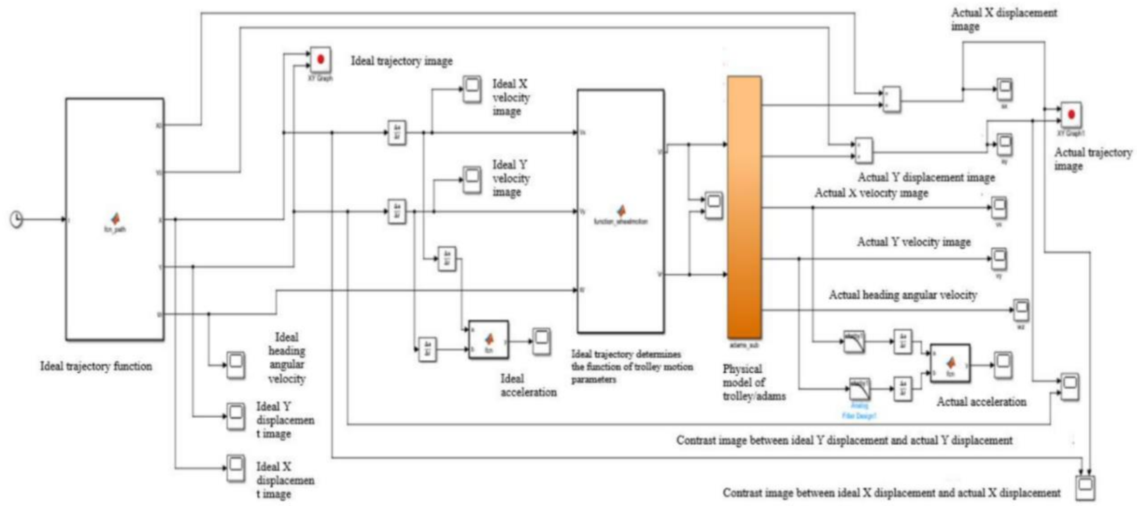


Figure 7: Block scheme with open loop strategy.

The interaction between Simulink and Adams in this research involves a block plant setup where data is continuously exchanged between the two platforms. The dynamic behaviour of the vehicle, simulated in Adams, provides real-time feedback to the control algorithms in Simulink. This feedback loop allows the control system to make precise adjustments based on the vehicle's current state and predefined trajectory.

One of the key aspects of this research is the use of advanced control logic to manage vehicle traction and stability. By simulating the vehicle's differential traction, the researchers were able to explore how different control strategies affect vehicle performance. This includes understanding how the vehicle manages various driving conditions, such as sharp turns, sudden stops and changes in road surface friction.

The results demonstrated that the Simulink-Adams co-simulation method could realistically simulate vehicle behaviour on various trajectories and road surfaces. This approach provided valuable information on the relationship between the mechanical dynamics of the vehicle and the control logic used to manage its behaviour.

Overall, the work of Liu S. L. and Wang Y. highlights the effectiveness of using Simulink-Adams co-simulation to study and improve the behaviour of differential-wheel drive vehicles.

In research conducted by Xu Tao and colleagues [20], the integration of Simulink and Adams was used to develop and test stability control systems for heavy articulated vehicles. This study aimed to manage and improve vehicle dynamics during driving, focusing on the unique challenges posed by these large and complex vehicles. The scheme used is shown in figure 8.

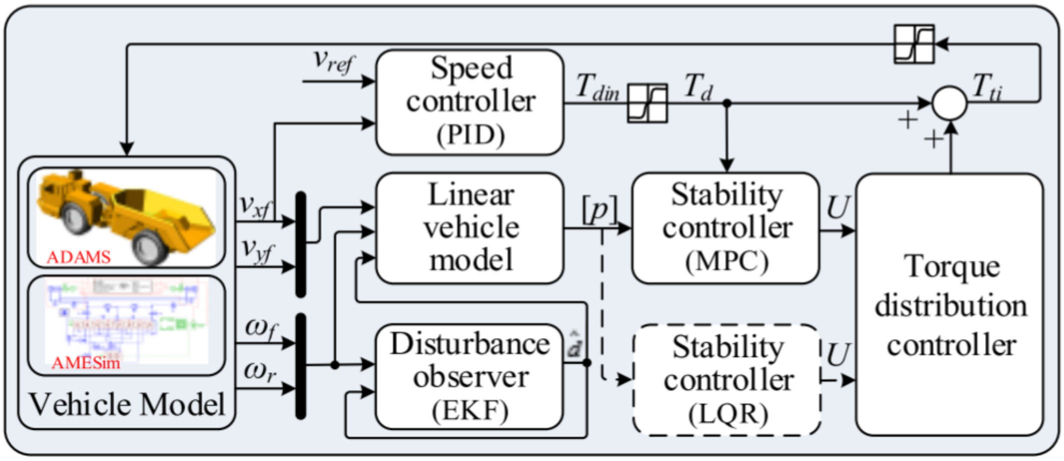


Figure 8: Block scheme with closed loop strategy and feedback Torque.

The interaction between Simulink and Adams in this research involved a co-simulation setup where data was continuously exchanged between the mechanical model in Adams and the control algorithms in Simulink. The dynamic behaviour of the vehicle simulated in Adams provided real-time feedback to the control system implemented in Simulink. This feedback loop allowed the control strategies to make precise changes based on the current state of the vehicle and the external conditions encountered.

Model predictive control (MPC) played a central role in this control system. MPC is an advanced method that uses a vehicle model to predict future states and optimize control inputs accordingly. By anticipating how the vehicle will respond to different inputs, MPC can adjust steering, throttle and braking in real time to maintain stability and follow the desired trajectory. This is particularly important for heavy articulated vehicles, where the dynamic interactions between tractor and trailer can be complex and challenging to manage.

The use of extended Kalman filters (EKFs) as observers was critical to accurately estimate the vehicle state in real time. EKFs can filter out noise and provide reliable estimates of variables such as position, velocity and yaw rate. These estimates are essential for control algorithms to work properly, as they rely on accurate data to make informed decisions.

The linear quadratic regulator (LQR) was another key component of the control system. LQR is a method used to design optimal controllers that minimize a cost function, which typically involves deviation from a desired trajectory and the effort required to control the vehicle. By integrating LQR with MPC and EKF, the researchers were able to create a control system that not only kept the vehicle stable but also optimized its performance in terms of energy efficiency and responsiveness.

Research demonstrated that the Simulink-Adams co-simulation method could significantly improve vehicle stability and reduce yaw motion oscillation in response to external disturbances. This comprehensive approach allowed for extensive testing and validation of control strategies, ensuring that the vehicle could maintain stability even in challenging scenarios.

Research [21] focused on dynamic vehicle control by implementing fuzzy controllers on virtual vehicles modeled using the Simulink-Adams co-simulation environment. This study highlights the potential of co-simulation to improve vehicle stability, particularly in challenging driving situations, through advanced control strategies such as differential brake control. The figure 9 show the scheme used for this work.

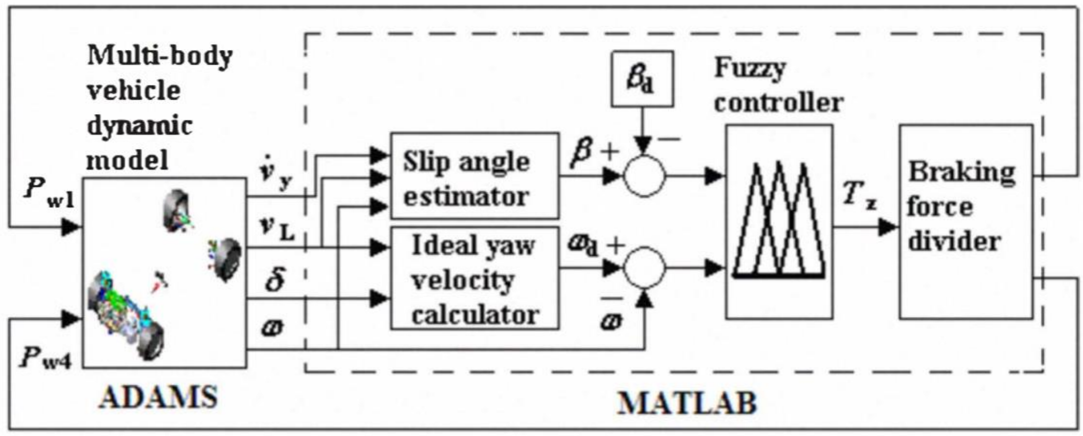


Figure 9: Block scheme with closed loop strategy and feedback brake inputs.

Simulink, based on a fuzzy controller, was used to implement the control system. Fuzzy controllers are particularly suited to managing complex, nonlinear systems with a high degree of uncertainty, making them ideal for dynamic vehicle control. Unlike traditional controllers that rely on precise mathematical models, fuzzy controllers use a set of linguistic rules to make decisions based on input data, allowing them to manage the inherent unpredictability of vehicle dynamics.

The co-simulation setup involved a closed-loop system where data was continuously exchanged between the mechanical model in Adams and the control algorithms in Simulink. In this closed-loop configuration, the dynamic behaviour of the vehicle simulated in Adams provided real-time feedback to the fuzzy controller in Simulink. The fuzzy controller processed this feedback to adjust the vehicle's braking inputs, focusing specifically on differential brake control to improve stability.

Differential brake control involves applying different braking forces to each wheel to help maintain vehicle stability, especially during extreme driving maneuvers such as sharp turns or sudden stops. By precisely controlling the braking force at each wheel, the controller can mitigate the risk of skidding and loss of control, ensuring the vehicle remains stable even in difficult conditions.

The data exchange interface between Simulink and Adams played a crucial role in this setup, enabling seamless communication and integration between detailed mechanical simulations and advanced control strategies. This interface allowed the fuzzy controller to receive real-time data on the state of the vehicle, such as speed, acceleration, and yaw rate, and to use this information to make immediate changes to the braking system.

Research has shown that implementing fuzzy controllers in a Simulink-Adams co-simulation environment could significantly improve vehicle stability in extreme driving situations. The fuzzy controller's ability to handle complex and unpredictable dynamics, combined with Adams' realistic mechanical modeling, provided a robust framework for optimizing vehicle control systems.

The co-simulation project Simulink/Adams for an autonomous vehicle with differential traction fits perfectly into this line of research, benefiting from the experiences and methodologies developed in the past. By integrating the abilities of both software, the project aims to create a virtual model that can be used to test and optimize the control system and vehicle dynamics in a realistic and accurate manner. These applications demonstrate how co-simulation can significantly improve the efficiency of the design process, reducing the time and costs associated with the development of physical prototypes and enabling a more complete evaluation of system performance.

3. MATERIALS AND METHODS

The design part of this thesis represents the center of the work carried out for the development and simulation of a small-sized agricultural rover with differential traction. This project was born with the aim of exploiting advanced simulation technologies to improve the efficiency and effectiveness of autonomous vehicles in the agricultural sector.

The MATLAB algorithm is based on two parts that are path planning and path following. The path planning is focused on the generation of waypoints which allow the rover to follow a predefined path within a 40x40 meter map. These waypoints are critical to ensuring the rover can navigate between rows of agricultural crops precisely and efficiently. Once the path planning was studied and understood in MATLAB, attention shifted to designing the path following part, with the goal of creating a closed-loop control system. This system takes waypoints, from the workspace, as input and moves the rover's kinematic model along the path, continuously correcting the trajectory through yaw adjustment.

To build the Simulink model it was necessary to start from the kinematic equations [1] of an idealized vehicle with differential traction. These equations describe the dynamic behaviour of the rover and have been implemented in a block diagram on Simulink. The tracking algorithm was translated into a function block on Simulink that receives as input the rover's current position, orientation and waypoints. As output, the algorithm generates the rover's absolute velocity and the necessary direction corrections.

These outputs are then used by the wheel differential block, which converts absolute velocity and direction corrections into angular velocities for the rover's left and right wheels (ul and ur). These angular velocities feed the rover's kinematic model, closing the feedback loop and continuously updating the rover's position and orientation.

Once the kinematic model was completed, the next step was to integrate this system with a three-dimensional multibody model of the rover developed on Adams. Co-simulation between Simulink and Adams permits to combine the dynamic systems modeling power of Simulink with the advanced mechanical simulation capabilities of Adams. The multibody model on Adams represents a simplified but accurate version of the rover, with editable wheel-ground contact parameters and default environmental settings to facilitate simulation. Adams generates a Simulink block that simulates the physical behaviour of the rover, receiving the angular velocities of the wheels as input and returning the position in xy coordinates and the orientation of the rover as output. The integration of the Adams block into the Simulink system allowed the kinematic model to be replaced with a more realistic physical model, maintaining the closed loop structure. A co-simulation was then performed to analyze the position data and verify the accuracy of the path followed by the rover.

One of the main goals of this project is to ensure that the co-simulated system follows waypoints as accurately as possible. To achieve this, it was critical to ensure that the Simulink model was completely independent of the MATLAB code, except for the generation of initial waypoints.

3.1 Algorithm Specifications

The rover to be simulated is a compact and manoeuvrable vehicle, with dimensions of 1.5 meters long and 1 meter wide. These measurements allow the rover to easily navigate the narrow rows of an agricultural plantation and are shown in figure 10. The rover's wheels, all identical, have a diameter of 0.4 meters and a width of 0.1 meters. The wheelbase, that is the distance between the front and rear axles, is 1.1 metres, while the distance between the wheel-to-ground contact points of the wheels on the same axle is 0.9 metres. The wheels are fixedly mounted to the rover chassis, allowing for differential traction movements. This means that the rover can rotate on itself, an essential characteristic for operating in confined spaces and for performing precise maneuvers while follow the route. The rover is designed not only to navigate autonomously, but also to perform environmental viewing functions and to carry small loads, making it versatile for various agricultural applications.

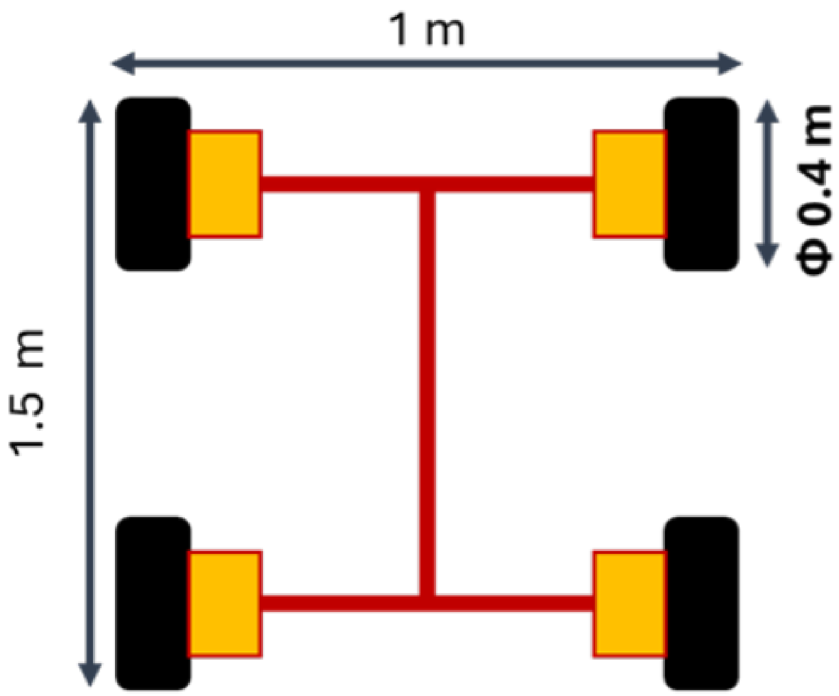


Figure 10: Dimensions of rover. [22]

The simulation environment is generated using a Matlab code according to the procedure described in [22], which models an area of 40x40 meters representative of an agricultural field indicated in figure 11. Within this map there are 4 parallel lines. Between the rows there are fixed plant obstacles that are repeated 3 times each 2 meters wide and 20 meters long. These rows are spaced 10 meters apart, creating corridors through which the rover must navigate. The fruit plants, represented as obstacles, are positioned in parallel rows within these rows. The terrain conditions are ideal for simplifying the simulation, allowing to focus on the dynamic performance of the rover without having to consider variables such as terrain resistance or weather conditions.

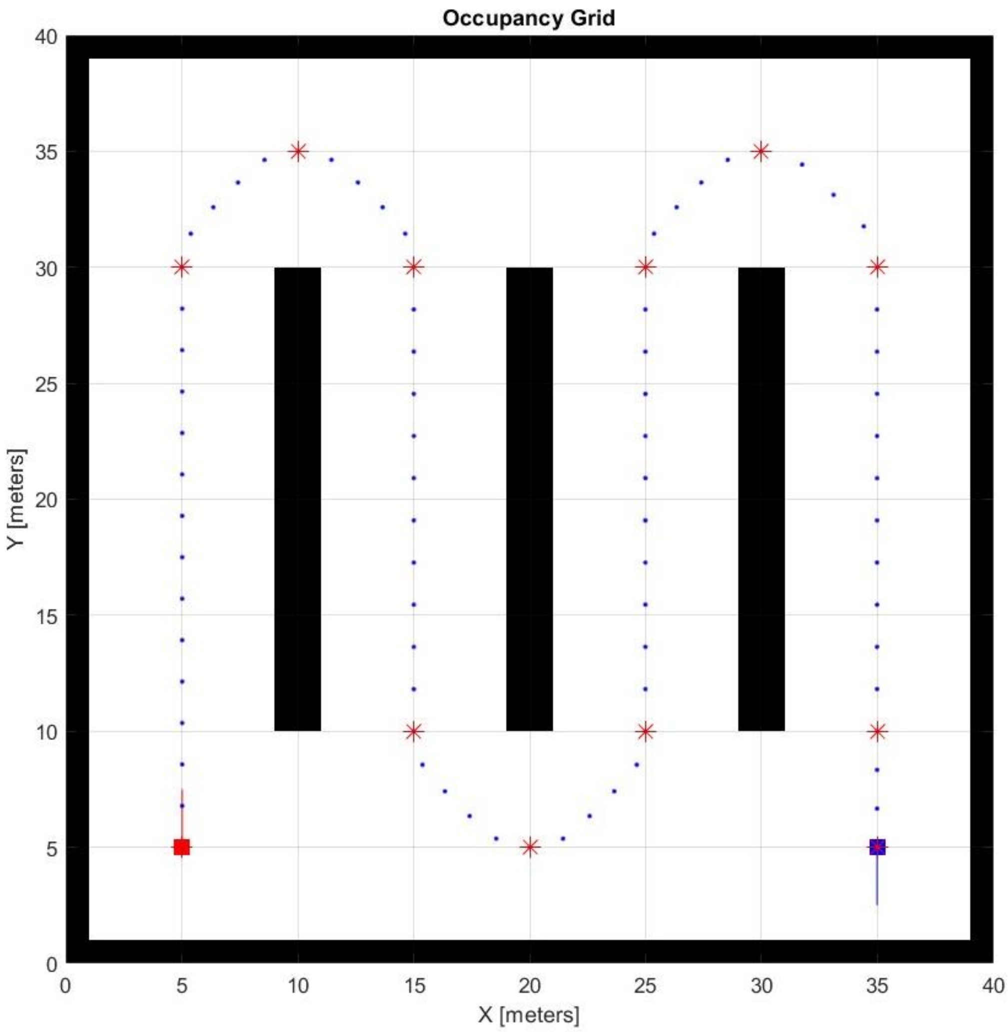


Figure 11: Map with waypoints.

The rover's path tracking system uses a Simulink algorithm that operates according to "point-2-point" logic. The vehicle must subsequently reach each waypoint along the predefined route. Each waypoint represents a specific point that the rover must pass through to complete its journey. When the rover reaches the target waypoint, located in the center of a circle with a predetermined radius (waypoint radius), the algorithm recognizes that the waypoint has been reached and updates the target to the next waypoint. This iterative process continues until the rover has traveled the entire route defined by the waypoints. The scheme of the algorithm is below in figure 12.

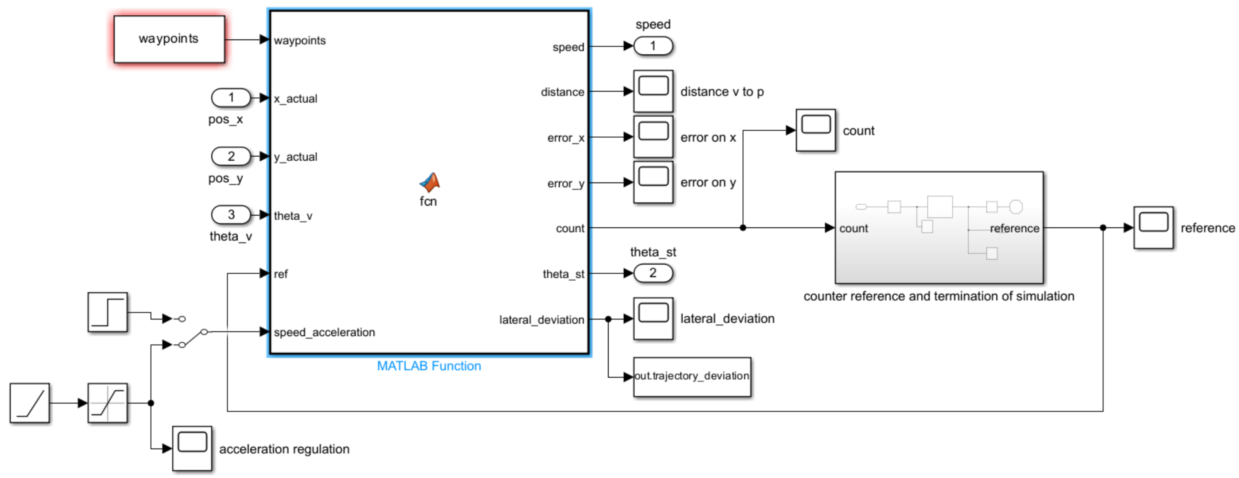


Figure 12: Algorithm block scheme.

The algorithm inputs are critical to the proper functioning of the rover's navigation system. In this section are described each of them and their role in the context of the simulation in detail.

Waypoints represent reference points that the rover must follow along its route. These points are initially generated via an algorithm in Matlab, which defines a series of coordinates (*x,y*) distributed across the entire map. Once generated, waypoints are transferred to Simulink as a data array. This matrix provides the exact locations that the rover must reach in sequence, allowing you to define a planned and specific route for the simulation.

The current coordinates of the rover (*x_actual* and *y_actual*) are dynamically generated by the model used for the simulation. The coordinates are updated at each time interval of the simulation (sampling time), ensuring that the simulation remains in a closed-loop context. This continuous update allows the system to monitor the rover's current position and adapt its actions in real time.

The theta_v parameter represents the current orientation of the vehicle expressed in radians with respect to the reference system imposed. Like the coordinates, this parameter is generated directly by the simulation model and updated at each sampling instant. Orientation is crucial for determining the direction of the rover's movement and for calculating necessary corrections along the way. Maintaining a continuous update of theta_v helps ensure that the rover accurately follows the desired trajectory.

The reference index is a parameter that indicates the last waypoint reached by the rover during the simulation. This index is continuously updated every time the rover reaches the destination waypoint. Updating the ref parameter is critical to keeping the rover oriented towards the next waypoint in the sequence, ensuring that the route is followed correctly and in order.

Acceleration rate is a gain parameter to tune the initial movement of the rover. This parameter is controlled using a ramp source block in combination with a saturation block in Simulink. This approach allows modeling both the acceleration gain and the maximum speed that the rover can reach. Adjusting the acceleration rate is critical to avoiding sudden changes at the start of the simulation, ensuring a smooth start and consistent motion throughout the path.

Matlab algorithms generate waypoints through a series of calculations and trajectory definitions. Once created, these waypoints are converted into a coordinate matrix which is then imported into Simulink. This transfer is essential to synchronize the planned route with the simulation model, ensuring that the rover follows the defined path.

The algorithm inputs are designed to provide precise, real-time control of the rover's position, orientation and speed. These inputs work together to ensure that the rover follows the predefined path accurately, dynamically adapting to simulation conditions.

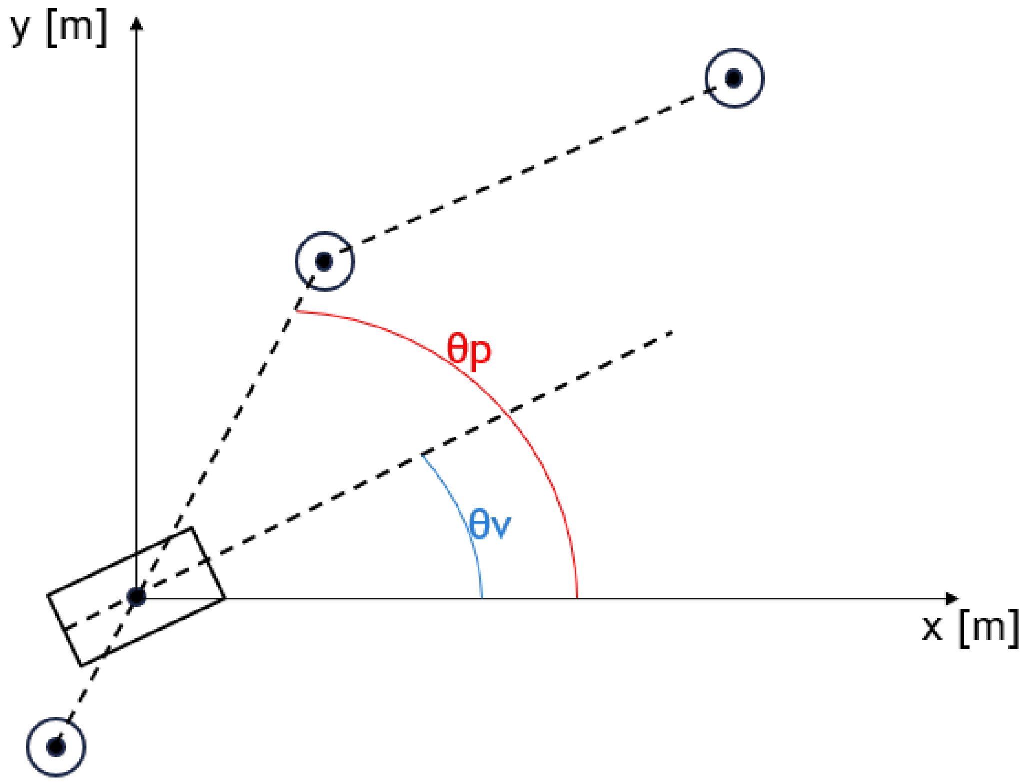


Figure 13: Steering parameters.

From figure 13, the mathematical relationship used to calculate θ_steering is given by the formula:

$$\theta_{steering} = Kp * (\theta p \ - \theta v) \tag{4}$$

Where:

- θ_steering is the steering angle that must be applied.
- Kp is the proportional gain, a parameter that determines the amount of correction.
- θp is the target angle that the rover must follow to reach the next waypoint.
- θv is the current angle of the vehicle with respect to the reference coordinates.

The waypoint selection criterion is based on the dynamic update of the current destination waypoint. Each time the rover enters the circumference defined by the waypoint radius, the algorithm identifies the waypoint as reached and selects the next waypoint as the new target. This approach ensures that the rover follows the path accurately, constantly correcting its trajectory to stay within the target zone.

Updating the reference waypoint is a critical process that occurs each time the rover reaches the circle associated with the target waypoint. A counter, located outside the main function block as shown in figure 13, is updated by the algorithm's output count. This counter increments the value of the destination waypoint, providing the index of the new waypoint as input to the function block. The function block, in turn, recalculates the parameters necessary for the rover's movement towards the new target.

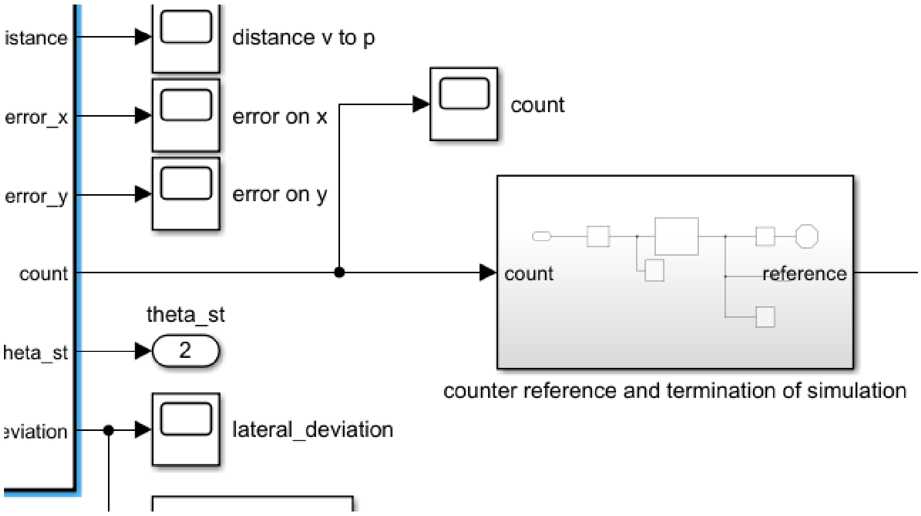


Figure 14: Counter block.

The waypoint radius must be set carefully to ensure optimal fit of waypoints along the route. A radius that is too large could reduce the precision of the path, while a radius that is too small could cause excessively abrupt movements. After several simulations, it was established that a waypoint radius of 0.35 meters offers a good compromise between accuracy and smoothness of movement.

Continuously updating the reference waypoint allows the rover to dynamically adapt to changes in the route, maintaining a precise and smooth trajectory. This approach ensures that the rover can effectively navigate along the predefined path, constantly correcting its direction in response to subsequent waypoints.

Calculating the angular error is critical to correcting the rover's direction and keeping it on the desired path. The angle θp is calculated via parametric calculations within the function block. This angle is obtained from the segment created from the center of mass point of the rover and the target point with respect to the coordinates. The angle θv, on the other hand, is created by the direction of the vehicle with respect to the coordinates, calculated by the real-time simulation model and provided as input to the algorithm.

To avoid excessive direction corrections caused by small angular errors, a threshold of 2.5 degrees was set. This threshold prevents the rover from making continuous small direction corrections, ensuring smoother and more stable movement. The algorithm then applies steering angle correction only when the angular error exceeds this threshold, improving the stability and precision of the rover's navigation.

The algorithm is designed to ensure that the rover follows the predefined path with high accuracy. Point-to-point logic, continuous waypoint updating and precise angular error calculation work in synergy to keep the rover on the desired trajectory, thus improving the efficiency and reliability of the navigation system.

The Kp parameter plays a crucial role in modulating the rover's motion correction. This parameter determines the responsiveness of the steering system: if Kp is too small, the steering angle response (θ_steering) will be too attenuated, making the vehicle less responsive and unable to follow the trajectory efficiently. On the contrary, a Kp value that is too high makes the steering angle response too nervous, causing oscillations and instability in the movement of the vehicle.

For the rover kinematic model, Kp can vary over a wide range of values, allowing some flexibility in the system response. However, for the dynamic model, it was determined that a value of Kp=5 offers balanced behaviour, ensuring a sufficiently fast response without causing instability. Furthermore, a minimum steering limit of 2.5 degrees was set to avoid too many direction corrections and improve the stability of the rover's movement.

The speed adjustment was calibrated based on the typical operating speeds of small agricultural rovers. The maximum speed was set at 7 km/h, while the minimum speed was set at 1.8 km/h. The speed regulation logic is designed to adapt to the conditions of the route: when the rover travels along a straight stretch it reaches the maximum speed, while when cornering the speed decreases proportionally to the steering angle applied. Over max steering angle, the rover travel at minimum velocity.

The formulas for speed regulation are the following:

$$max\ steering\ angle = \frac{\pi}{2} \tag{5}$$

$$Kr = 1 - min\left(\frac{\theta_{steering}}{max\ steering\ angle}, 1\right) \tag{6}$$

$$target\_speed = \min\ \_speed + Kr * (\max\ \_speed - \min\ \_speed) \tag{7}$$

This logic ensures that the rover's speed is modulated appropriately, maintaining a balance between speed and safety, especially during cornering maneuvers. Reducing speed at significant steering angles helps prevent the vehicle from overturning and improves steering accuracy.

To monitor path accuracy and provide visual feedback during the simulation, position errors along the x and y directions are calculated. These parameters allow to view and analyze the rover's deviation from target waypoints, giving you a clear understanding of how the simulation is progressing. By monitoring position errors, it is possible to identify any problems in the rover's behaviour and make the necessary corrections to improve the accuracy of the route.

The distance of the rover from the reference waypoint is also calculated. This calculation provides another useful visualization parameter during the simulation, allowing you to observe in real time how close the rover is to the target waypoint. The calculated distance provides an immediate indication of the rover's path-following performance, highlighting how effectively the rover can navigate between waypoints.

Various control and tuning techniques are employed to ensure that the rover can navigate precisely and stably along the predefined path. From steering control to speed regulation, through the calculation of position errors and distances, each component of the algorithm contributes to keeping the rover on the desired trajectory, minimizing errors and dynamically adapting to route conditions.

The outputs of the algorithm are essential for managing the movement of the rover and for monitoring its performance during the simulation. Below is a detailed description of each release.

Speed: The calculated speed is used to control torque vectoring, which adjusts the angular speeds of the rover's wheels. Proper speed management is critical to ensuring the rover can adapt to course conditions, especially during sharp turns or straight sections.

Distance: the current distance of the rover from the reference waypoint is measured. The distance helps evaluate how close the rover is to the target and determine whether the waypoint needs to be updated.

Error in X and Y: These errors indicate the rover's deviation from the target waypoint in the horizontal and vertical directions. They are visual feedback parameters during the simulation, useful for monitoring the precision of the rover in following the path.

Count: The count is a crucial result that determines whether the rover has reached the destination waypoint. When the rover enters the area of interest defined by the waypoint radius circle, the counter changes from 0 to 1, warning that the waypoint has been reached and needs to be updated to the next one.

θ_st: This parameter provides the steering angle needed to correct the rover's trajectory. The steering angle is calculated based on the angular error and is used by the torque vectoring system to guide the rover towards the target waypoint.

Lateral Deviation: This output is stored in the MATLAB workspace via the .out block and is used to create subsequent graphs and analyses. Lateral deviation offers a visual and numerical measurement of the quality of the path followed by the rover.

The algorithm results not only provide essential parameters for controlling the rover's motion, but also for detailed analysis of its performance. Speed and steering angle are used directly to control the vehicle, while distance, position errors and lateral deviation provide continuous feedback on the precision of the movement and the effectiveness of the algorithm. This data is critical to evaluating and optimizing the rover's autonomous navigation system.

3.2 Kinematic Model

The Simulink system with the kinematic model, shown in figure 15, was mainly used for the prototyping phase of the rover's autonomous driving algorithm. This phase is crucial as it permits to test and validate the algorithm in a controlled environment, reducing the complexities introduced by multi-body dynamics and allowing to focus on the fundamental aspects of control.

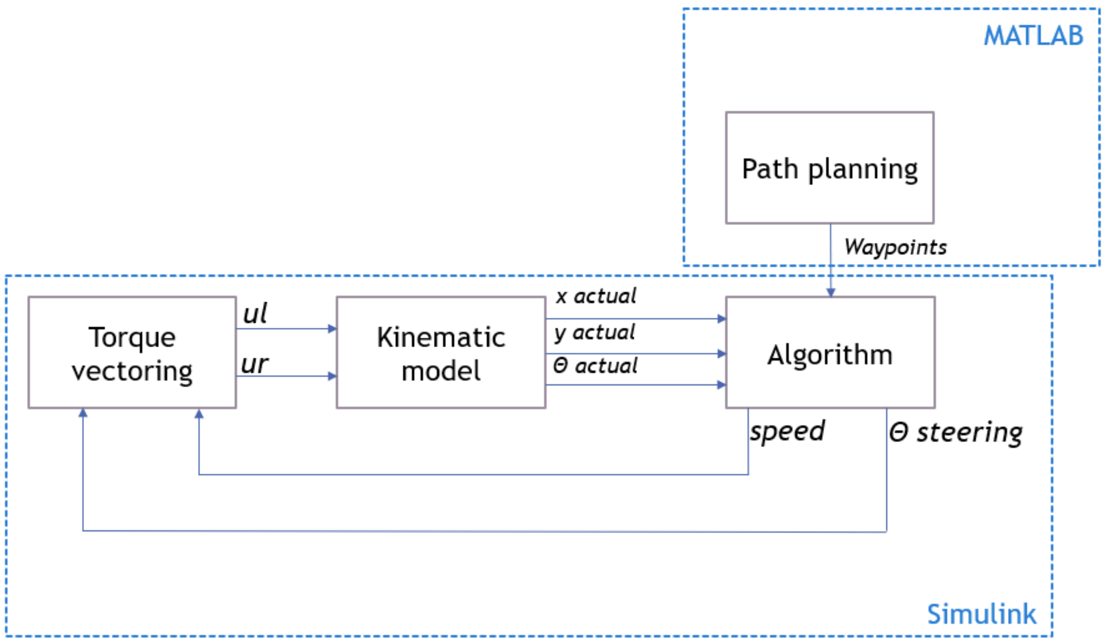


Figure 15: Kinematic model block scheme.

The main goal of using Simulink was to make the control algorithm work optimally. This approach allows to identify and fix problems in the algorithm itself before implementing it in a more complex model. By using a kinematic model, we were able to focus on the basic dynamics of the rover, simplifying many of the complexities introduced by multi-body dynamics.

The kinematic model used is based on past studies [1] and the formulas used are relatively simple. The compact-sized rover is schematically represented as a rectangle 1 m wide and 1.5 m long, with a wheel diameter of 0.4 m. Each wheel can be controlled independently of the others, allowing you to manage the speed of individual wheels when cornering. This type of vehicle does not require a physical steering system, since it can rotate on itself, having a minimum turning radius of zero but this results in increased lateral stress on the wheels when cornering.

$$\begin{cases} \dot{x} = \frac{r}{2}(u_r + u_l)\cos\theta = r u_v \cos\theta \\ \dot{y} = \frac{r}{2}(u_r + u_l)\sin\theta = r u_v \sin\theta \\ \dot{\theta} = \frac{r}{2l}(u_r - u_l)\sin\theta = \frac{r}{l} u_\omega \end{cases} \tag{8}$$

The values ur and ul indicate the right and left angular velocity, respectively, while θ indicates the orientation angle. The kinematic model was implemented in a 2D environment, ignoring vertical movements and considering only three degrees of freedom: two translational (longitudinal and lateral movements) and one rotational (yaw). Roll and pitch angles were ignored, as was lateral slip, since the rover's operating speed is relatively low. The wheel-ground contact is considered in pure rolling conditions.

The kinematic model aims to define the relationships between the operating conditions that the vehicle must satisfy (longitudinal and yaw speed) and the operating parameters necessary to achieve these conditions (angular speeds of each wheel). The rover is represented as a symmetrical four-wheel drive vehicle that is independent on each side. The mechanical simplicity of the rover, due to the absence of a physical steering system, is one of its main advantages.

To integrate the algorithm into the kinematic model, a block diagram was created in Simulink starting from the kinematic equations. The waypoints, generated separately, were provided to the system from workspace. The control algorithm receives as input the waypoints, the current position via the x and y coordinates of the kinematic model and the orientation θv. In output it provides the correction to be applied (error gain) to the orientation and speed of the vehicle.

The torque vectoring block, developed in Simulink, modulates the angular speeds of the wheels based on the speed and yaw correction calculated by the control algorithm.

$$\begin{cases} \omega 1 = \dfrac{(V - D * \dot{\theta}\_steering)}{R} \\ \omega 2 = \dfrac{(V - D * \dot{\theta}\_steering)}{R} \\ \omega 3 = \dfrac{(V + D * \dot{\theta}\_steering)}{R} \\ \omega 4 = \dfrac{(V + D * \dot{\theta}\_steering)}{R} \end{cases} \tag{9}$$

The angular speeds ω1 and ω2 refer to the left wheels and ω3 and ω4 to the right ones. The speed of the vehicle is indicated with V and the track with D. This block is fundamental for the correct execution of the rover's movements, as it guarantees that the angular speeds of the wheels are adequate for the necessary trajectory corrections. The implementation of this block required careful definition of kinematic relationships and continuous verification to ensure that the corrections applied were appropriate.

The generation of the waypoints was carried out separately and inserted into Simulink via a constant block containing the x and y coordinates of the points. These waypoints represent the target points that the rover must reach during its route. Choosing waypoints carefully is critical to ensuring the rover accurately follows the desired path.

The simulation was configured with a Kp gain of 5 and a waypoint radius of 0.35 meters. The solver used has a variable sampling time, allowing for efficient simulation. The results were compared continuously by viewing the model motion on the (*x,y*) map. This visualization allowed us to monitor the rover's behaviour in real time and make necessary changes to the control algorithm to improve the accuracy and stability of the system.

The results demonstrated that the system with the control algorithm is stable and robust under ideal simulation conditions. By varying the Kp value it was possible to observe how the vehicle's behaviour changed: a balanced Kp led to a better response when cornering. The main challenge was to develop a stable algorithm at a computational level, capable of maintaining the desired performance in different operating conditions. The simulation confirmed that the algorithm can effectively correct the rover's trajectory, keeping it on the desired path even in the presence of small initial errors in position or orientation.

The Simulink model allowed the control algorithm to be developed and tested in a controlled environment, providing a solid foundation for subsequent integration with the Adams multi-body model. The algorithm proved to be stable and robust, with a good ability to follow the generated waypoints and correct the rover's trajectory. The simulation highlighted the importance of a balanced Kp gain to obtain optimal behaviour of the rover in curves and in straight sections. This preliminary work in Simulink was crucial to developing a control algorithm that was reliable and ready to be tested in a more complex environment.

The implementation of the kinematic model in Simulink was based on some key assumptions to simplify the system:

- The vehicle is considered a rigid body, ignoring all effects related to deformability.
- The operating environment is 2D, so vertical movement is negligible.
- Wheel-ground contact occurs in pure rolling conditions, eliminating the need to consider lateral slips.

- Operating speed is slow enough to ignore complex dynamic effects.

The kinematic model defines the relationship between the operating conditions of the vehicle (longitudinal and yaw speed) and the operating parameters (angular speeds of the wheels). Using kinematic equations, the rover is represented as an independent, symmetric, four-wheel drive vehicle. This representation is ideal for simplifying vehicle control and testing the algorithm under ideal conditions. In the figure below there is a block scheme of kinematic model.

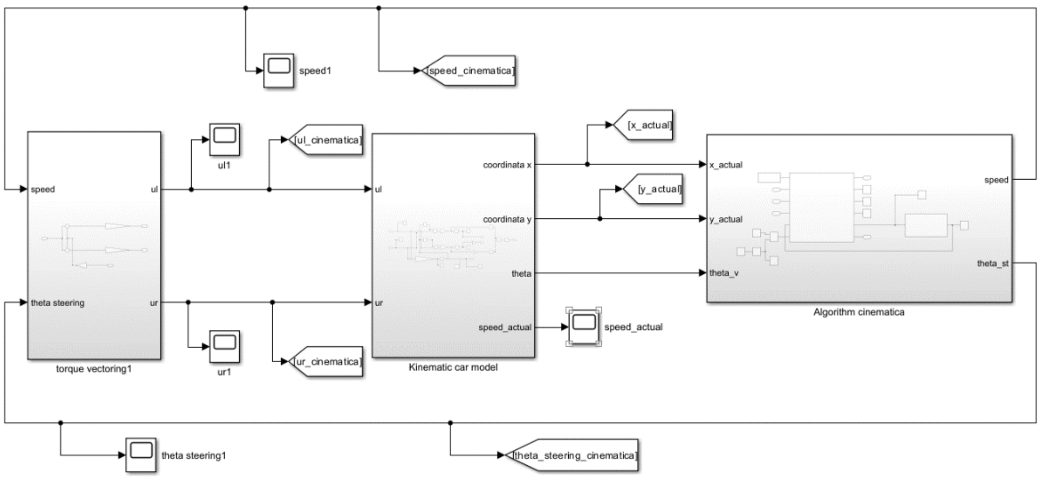


Figure 16: Kinematic model Simulink block scheme.

The torque vectoring block in Simulink modulates the angular speeds of the wheels based on the speed and yaw correction calculated by the control algorithm. This lock is essential to ensure that wheel speed is adequate for the necessary trajectory corrections. Implementing this block required careful definition of kinematic relationships and ongoing verification to ensure corrections were appropriate.

Waypoint generation was performed separately and inserted into Simulink via a constant block containing the x and y coordinates of the target points. The choice of waypoints is critical to ensuring that the rover accurately follows the desired path. Waypoints represent target points that the rover needs to reach and have been selected to form a route that the rover can easily follow. The implementation of waypoints in the Simulink model allowed the path-following algorithm to be tested under controlled conditions. The Simulink model receives the waypoints as input and calculates the rover's trajectory, displaying the model's motion on the *x-y* map.

The solver used has a variable or fixed sampling time, allowing for efficient simulation and the results were compared continuously by viewing the model motion on the *x-y* map.

This preliminary work in Simulink was crucial to developing a control algorithm that was reliable and ready to be tested in a more complex environment.

3.3 Dynamic Model:

Adams' multibody model was developed to improve the rover simulation, making it closer and suitable for real-world scenarios. Indeed, the kinematic model is based on ideal calculations that do not consider physical forces and dynamics, whereas Adams multibody model incorporates these physical variables, providing a more accurate representation of the rover's behaviour under realistic operating conditions. This fundamental difference between the two models allows to obtain a more detailed picture of the rover's performance, identifying potential problems and areas of improvement before the physical construction of the prototype. The advantages deriving from the use of a multibody model are evident: it allows you to perform virtual tests and optimizations, reducing the time and costs associated with physical prototyping.

3.3.1 Adams Model

Regarding the description of the multibody model, the body of the rover was represented as a rigid rectangle with uniform mass distribution, whereas the wheels were modeled considering their radius and thickness.

As part of the model implementation in Adams, the rover geometries were defined considering the typical dimensions of a small vehicle designed for precision farming and transport of small loads shown in figure 17. The rover has dimensions of 1.5 meters in length and 1 meter in width, measurements compatible with the narrow rows of an agricultural plantation. The rover's wheels, all identical, have a diameter of 0.4 meters and a width of 0.1 meters. The wheelbase, or the distance between the front and rear axles, is 1.1 metres, whereas the distance between the wheel-to-ground contact points of the wheels on the same axle is 0.9 metres.

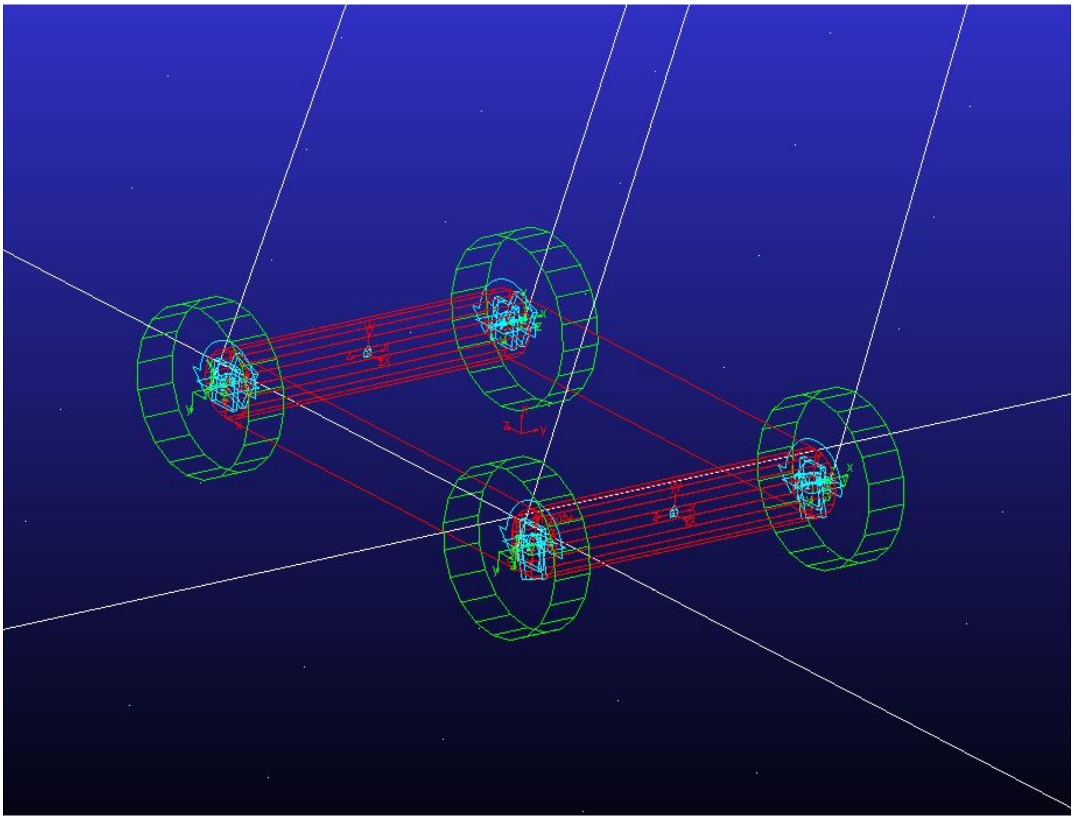


Figure 17: Multibody model of the rover.

The model is controlled in velocity and the rover receives in input the target angular speeds of each wheel. These angular speeds are determined by the control algorithm implemented in Simulink, which provides the commands necessary to guide the rover along the predefined path. Modeling of the wheels and rotation joints required particular attention to ensure that the forces and torques applied during the simulation were realistic and accurately reflected the rover's operating conditions.

The wheel-ground contact parameters as static and dynamic coefficient are values used in previous studies [24], to emulate the deformability of the ground.

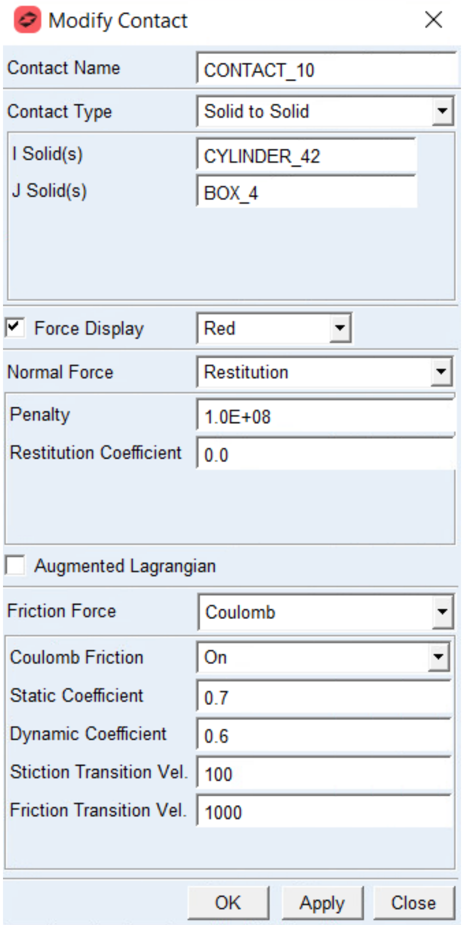


Figure 18: Tyre-ground parameters.

This approach allows to focus on the dynamics of the system without having to deal with the additional complexities related to the deformability of materials, while maintaining a high degree of accuracy in the simulation. The choice to consider the rover as a rigid body was made to reduce computational complexity, allowing results to be obtained more quickly without sacrificing the precision needed to evaluate the rover performance.

3.3.2 Co-simulation

Co-simulation represents an advanced methodology that allows the integration of different models and tools to make the most of the strengths of each software involved. This methodology is key to improving the quality of simulations, particularly when working on complex systems such as agricultural rovers. In the context of this thesis, the co-simulation between Simulink and Adams played a fundamental role in increasing the realism and reliability of the simulations, allowing to overcome the limitations of the kinematic model and include a dimension of realism that incorporates physical variables dynamics.

The main goal of this integration was to connect the multibody model developed in Adams with a control algorithm in Simulink, to improve the motion characteristics of the rover. The choice to use Simulink and Adams was motivated by the need to combine Simulink's ability to design and simulate control algorithms with the accuracy of Adams in multibody dynamic modeling. This combination made it possible to create a robust and realistic simulation environment, capable of predicting and solving potential problems already in the design phase, thus reducing the need to make further changes to the initial physical prototype and accelerating the rover development process.

The integration between Simulink and Adams was achieved through Simulink's native interface, which facilitates bidirectional communication between the two software. The process begins with configuring the desired inputs and outputs for each program. Through the Adams Plug-in function, an .slx file is then created in Simulink, containing a block representing the multibody model. This block is later integrated into the Simulink block system, allowing the connection of all inputs and outputs necessary for the simulation as represented below in figure 19.

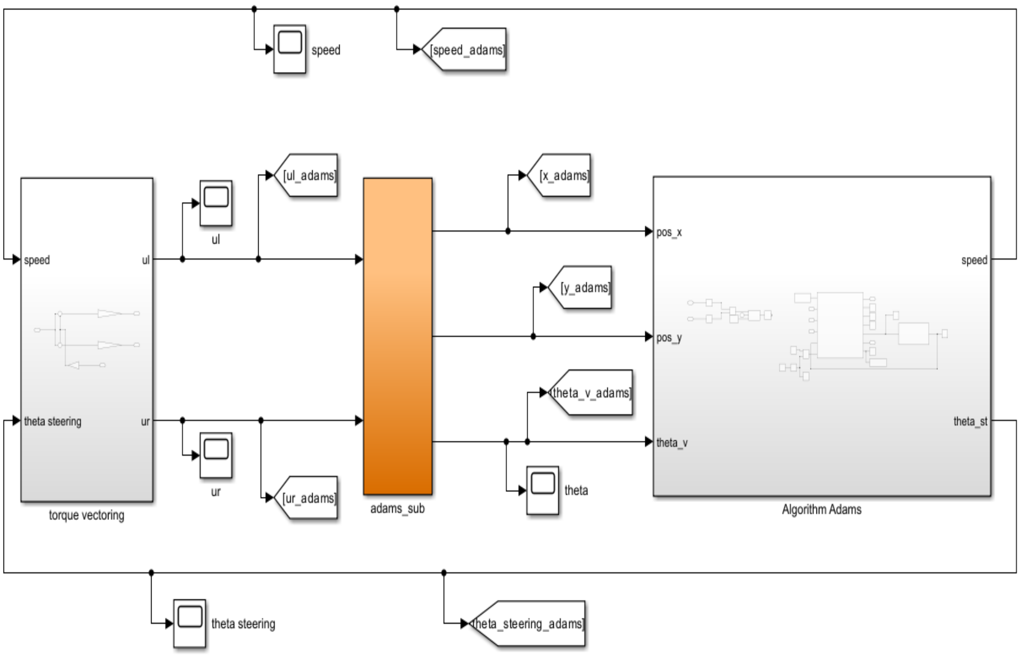


Figure 19: Dynamic model Simulink block scheme.

In Simulink the control algorithm was configured with specific parameters, including a waypoint_radius of 0.35 meters and a Kp of 5. Simulink uses a variable-time solver to provide flexibility during simulation, allowing the system to dynamically adapt to different operating conditions. In Adams, however, the multibody model of the rover was configured considering the physical dimensions and dynamic properties of the components. The rotational joints of the wheels receive angular velocities [rad/s] as input through the defined motion variables. The solver used is GSTIFF with formulation I3 and error 1.0E-03, chosen for its compatibility and stability during the simulations.

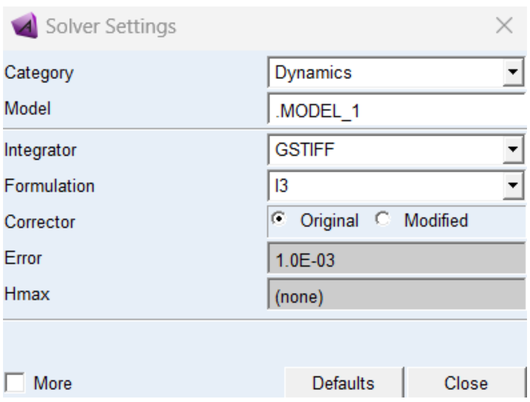


Figure 20: Solver settings.

Other solvers were tested but it was concluded that GSTIFF was the optimal choice.

Inputs are passed from Simulink to Adams. The angular velocities of the left and right wheels (ul and ur respectively in rad/s) are transmitted to Adams to control the rover's motion. These inputs allow you to realistically simulate the behaviour of the rover in response to control commands. The outputs are passed from Adams to Simulink. The x-y coordinates (in meters) and orientation (in radians) of the rover are returned to Simulink, providing crucial information for monitoring and correcting the rover's trajectory during the simulation.

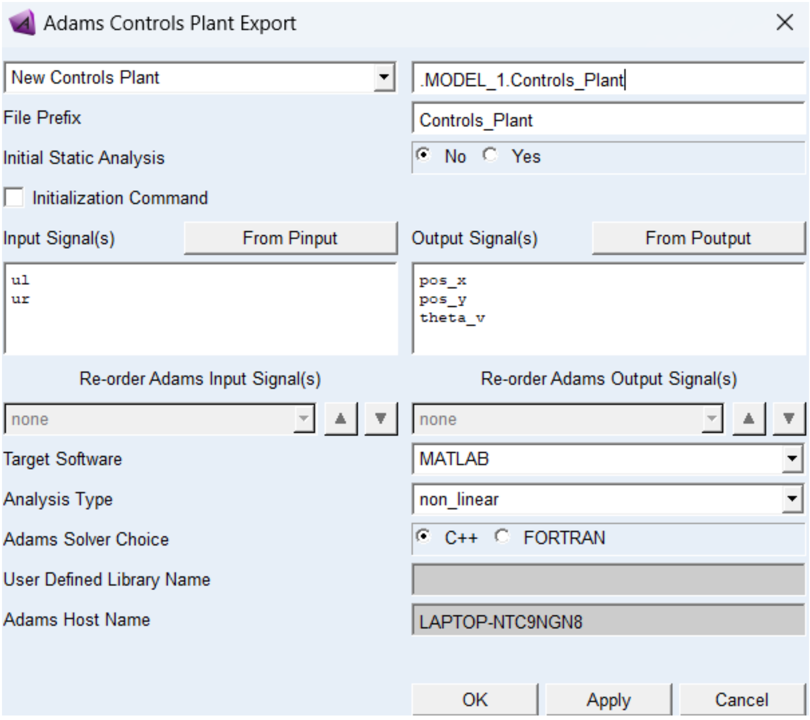


Figure 21: Plug-in configuration.

The co-simulation environment was entirely configured in Simulink, taking advantage of its integration and monitoring capabilities. The co-simulation methodology includes several phases, each crucial to ensuring the success of the simulation. Let's start from the initial configuration which involves defining all the necessary files using the Adams plug-in, making sure that both programs work in sync during the simulation. This step is essential to avoid discrepancies in the data exchanged between the two software. Waypoint generation must also be considered. Waypoints, which define the path the rover must follow, are generated in the initial part of route planning based on the desired route. These reference points are essential for guiding the rover along the predetermined path and for evaluating the effectiveness of the control algorithm. After the entire set-up part and definition of the fundamental parameters, the Co-Simulation is carried out.

With the initial setup, the system is ready to run the simulation, and during this phase Simulink and Adams work together to simulate the motion of the rover and the behaviour of the control system. During the simulation, the rover's motion is monitored in real time using visualization blocks such as the xy diagram for trajectory and the scope for orientation. The collected data is stored via sim.out blocks, which allow the graphs necessary for detailed analysis of the rover's behaviour to be subsequently drawn using MATLAB. This continuous monitoring is crucial to identify any problems during the simulation.

The control and simulation phase involves close interaction between Adams and Simulink. Control of the rover occurs entirely in Simulink, while Adams handles the data exchange with Simulink.

A step-by-step approach was taken to introduce co-simulation between Simulink and Adams, starting with an open loop configuration. This initial method for the Adams block was chosen to evaluate the rover's reaction to input data generated by the Simulink-integrated kinematic model. In the open-loop co-simulation system, the Adams block was given angular velocities to apply to the wheels of the multi-body model, allowing for immediate comparison between the theoretical model and the dynamic simulation.

The successive approach it was the integration of the Adams block in a closed loop system as shown schematically in figure 22. The angular velocities of the wheels, calculated by the control algorithm in Simulink, are provided as input to Adams. In return, Adams give the rover's x-y coordinates and the orientation (θv), which are used by Simulink to update motion control.

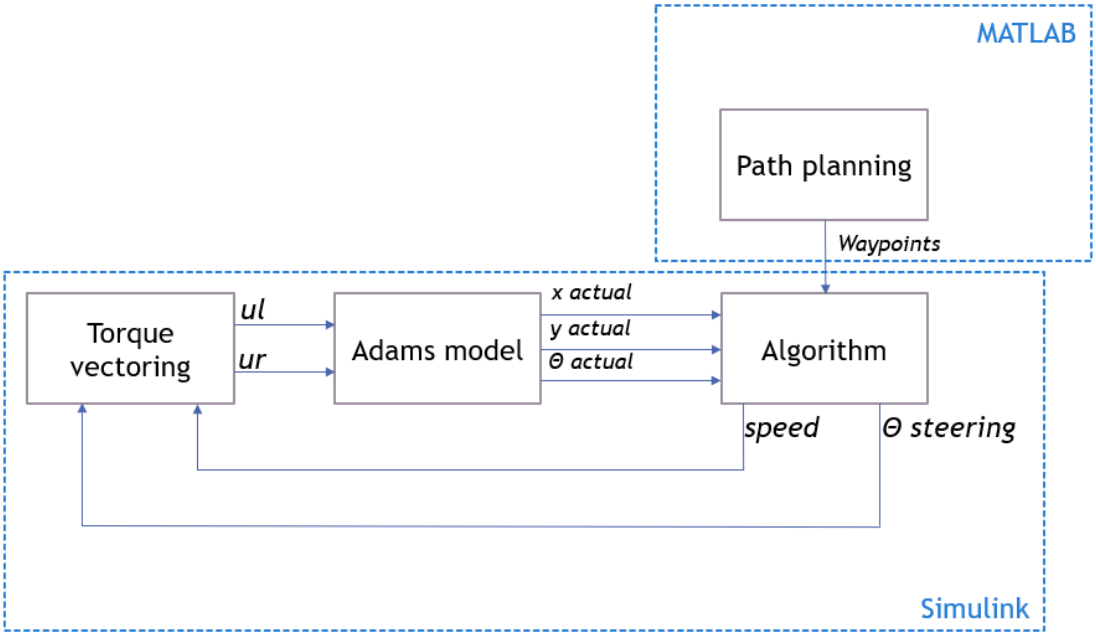


Figure 22: Dynamic model block scheme.

The control algorithm applied is proportional and applied with the differential criterion, used to correct the direction of the rover and ensure that it follows the path established by the waypoints. To verify the system, simulations were carried out on a map of 40x40 meters, which demonstrated the correct functioning of the rover in following the pre-set path.

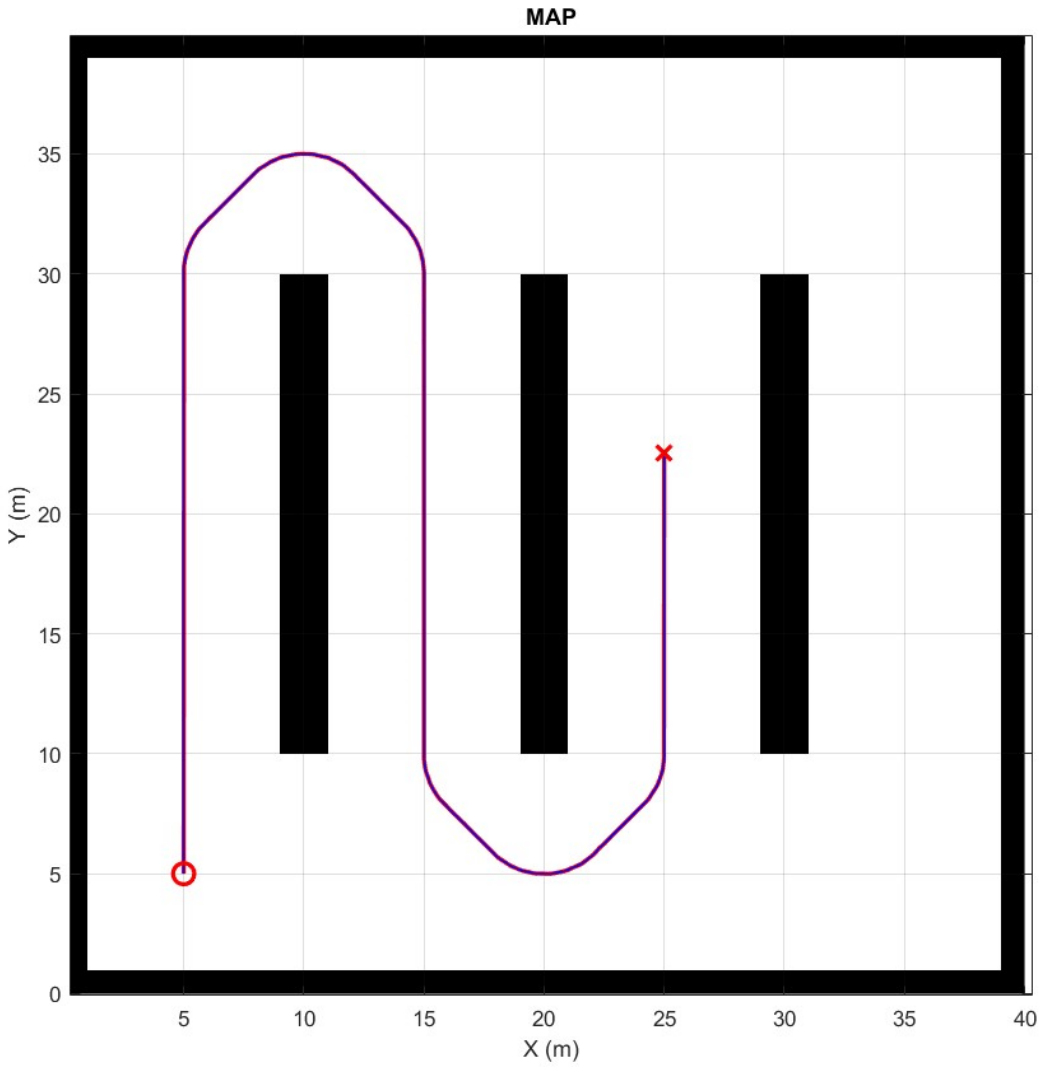


Figure 23: Simulation in Map 40x40.

This co-simulation process permits to test and verify the behaviour of the rover in a controlled environment, identifying any stability or performance issues before implementing the system in the real world.

Validation and simulation results confirmed that the rover can follow the path with an error of the order of a decimeter, indicating high accuracy of the system. The vehicle showed stability at set speeds of 7 km/h in straight paths and 1.8 km/h when cornering. The stability of the algorithm in sharp curves was mainly studied through the kinematic model, since correct operation in that context indicates that it will also work in the multibody model. This approach saved time given the long duration of co-simulation simulations between Adams and Simulink. The rover's ability to maintain stability and precision even at higher speeds was particularly significant, demonstrating that the system is robust and reliable.

The usefulness of this approach to simulate real-world conditions more accurately is highlighted. The results obtained are appropriate to the context, providing valuable data for possible future improvements of the system. To make the model even more realistic, it will be necessary to introduce additional physical parameters, allowing us to simulate more complex situations and obtain rover behaviour that reflects reality even more faithfully. The integration of the multibody model in Adams with Simulink represents a significant step forward in virtual prototyping, allowing potential problems to be anticipated and resolved before the physical construction of the rover.

This iterative simulation process allows system parameters to be optimized, improving the accuracy and reliability of the rover in real-world applications. With the introduction of further physical variables and the continuous optimization of the control algorithm, the system can become increasingly sophisticated and precise, leading to a highly performing rover capable of operating effectively in a wide range of environmental conditions.

Co-simulation offered several significant benefits, improving various aspects of the rover design and optimization process. The integration of the dynamic models significantly improved the accuracy compared to the simple kinematic model. By considering real physical variables and complex dynamics, co-simulation allowed us to obtain results that were much more realistic and representative of real operating conditions. Co-simulation simplified the design process, reducing the need for initial physical prototypes and accelerating system development. The ability to test and optimize the system in a virtual environment saved time and resources, while improving the quality of the final project. The combined use of Simulink and Adams allowed us to exploit the strengths of both software, offering a complete and versatile simulation environment. This flexibility allowed the system to be dynamically adapted to different operating conditions, further improving the effectiveness of the simulation.

The co-simulation results demonstrated high accuracy, with errors in the order of decimetres, and vehicle stability at different speeds. The simulation confirmed the effectiveness of the control algorithm even in tight curve conditions, demonstrating that a well-designed kinematic model can lead to good results even by integrating a multi-body model into the system. Simulation performance was evaluated in terms of performance accuracy, system responsiveness, and stability. The simulation demonstrated that the rover can precisely follow the predetermined path, even at different speeds and this demonstrates the effectiveness of the control algorithm and the validity of the integrated multibody model.

Co-simulation between Simulink and Adams has proven to be extremely useful for simulating real-world conditions and preparing for the prototype phase. The results obtained were adequate to the context, but for future improvements it will be necessary to introduce additional physical parameters into the Adams model and further develop the control algorithm in Simulink, including advanced features such as obstacle avoidance.

The main results obtained from the co-simulation confirm the good attitude of the integrated approach for the advanced simulation of complex systems such as agricultural rovers. The integration of Simulink and Adams allowed us to overcome the limitations of individual models, offering a complete and realistic solution for rover design and optimization.

Co-simulation represented a fundamental step for the success of the project, demonstrating the validity of the integrated approach for the advanced simulation of complex systems such as agricultural rovers. The combination of Simulink and Adams has made it possible to create a complete and realistic simulation environment, capable of significantly improving the design and optimization process, opening new possibilities for the future development of even more advanced and sophisticated systems.

4.RESULTS AND DISCUSSION

The goal was to develop a co-simulation system between Simulink and Adams that was stable and allowed the rover to follow a predefined track, made up of waypoints, with high accuracy. Co-simulation allows to integrate the control capabilities of Simulink with the dynamic physical realism of Adams, providing a complete picture of the rover's behaviour. A detailed analysis of the results is fundamental because it permits to identify areas of improvement and further optimize the system, improving future design and reducing development times and costs.

4.1 Analysis of simulation results

For the analysis of the rover's performance, specific simulation parameters were chosen and shown in the table below:

Kp	waypoint radious [m]	max speed [km/h]	min speed [km/h]
5	0,35	7	1,8

Table 1: Set parameters.

Furthermore, the contact parameters between the wheel and the ground in Adams' vision were set to maximize grip as can be seen in the following figure.

These parameters there was chosen to ensure a balance between precision and realism. The selection of waypoint radius and Kp consider the number of waypoints and the total distance to travel: a higher waypoint density requires smaller values to ensure that the rover can accurately follow the route. For this motivation it was selected different rates of distance between waypoints. The Velocities are based on the operational needs of an agricultural rover, while contact parameters are optimized to reflect realistic operational conditions in the field.

The calculation of lateral deviation is a critical element for evaluating the quality of the simulation and the performance of the rover. This parameter measures how much the rover deviates from the ideal trajectory defined by the waypoints. An accurate measurement of lateral deviation helps determine how effective the rover's control algorithm is at maintaining the planned trajectory.

The lateral deviation is calculated as the perpendicular distance of the rover to the imaginary line connecting the waypoints.

These coefficients represent the line between the current and next waypoints. The lateral deviation provides an index of how much the rover deviates from the line between two waypoints, and this value is fundamental for calculating further performance parameters such as the ATD (Average Trajectory Deviation), the OF (Oscillating Factor) and finally the RA (Relative precision). These complex parameters will serve to quantify the global performance of the navigation system.

The ATD represents the average value of trajectory deviations calculated for each position occupied by the rover. A low value of ATD indicates greater efficiency of the algorithm. The oscillation factor (OF) is used to evaluate whether the steering corrections made by the rover are adequate to keep the vehicle on the planned trajectory. Observing the deviation of the rover's trajectory during its mission, it can be noted that in case of large steering corrections, the deviation presents a wavy pattern. Conversely, a smoother progression indicates less aggressive steering corrections. The evaluation of the oscillating factor is carried out by counting the relevant peaks of the trajectory deviation curve. Also in this case, a lower number of peaks indicates a greater efficiency of the algorithm.

Based on the results obtained in this study, a final indicator is proposed to evaluate the overall quality with which the rover follows the planned trajectory. This indicator is called relative accuracy (RA) and can be determined using the following equation:

$$RA = \frac{1}{(ATD * OF)} \tag{10}$$

This indicator is defined as "relative" since it is not possible to evaluate the quality of the autonomous driving algorithm in absolute terms, given that its performance depends on various factors, such as the main characteristics of the vehicle and the operating environmental conditions.

The simulations were conducted on a 40x40 meter map with different configurations of distances between waypoints. The different configurations allows to evaluate how the rover behaves in situations of complex trajectories and how ATD and OF vary depending on the density of the waypoints.

Another set of simulations on a 40x40 meter map was performed with different position error values. This allows to understand the impact that position precision has on the performance of the algorithm, observing how ATD and OF vary in the presence of different position errors.

A simulation was conducted on a 40x40 meter map to compare the performance of the dynamic model versus the kinematic model using the optimal definition parameters. This comparison is essential to evaluate the reliability and accuracy of the two models under identical operating conditions, providing a detailed analysis of the differences in terms of ATD and OF between the two approaches.

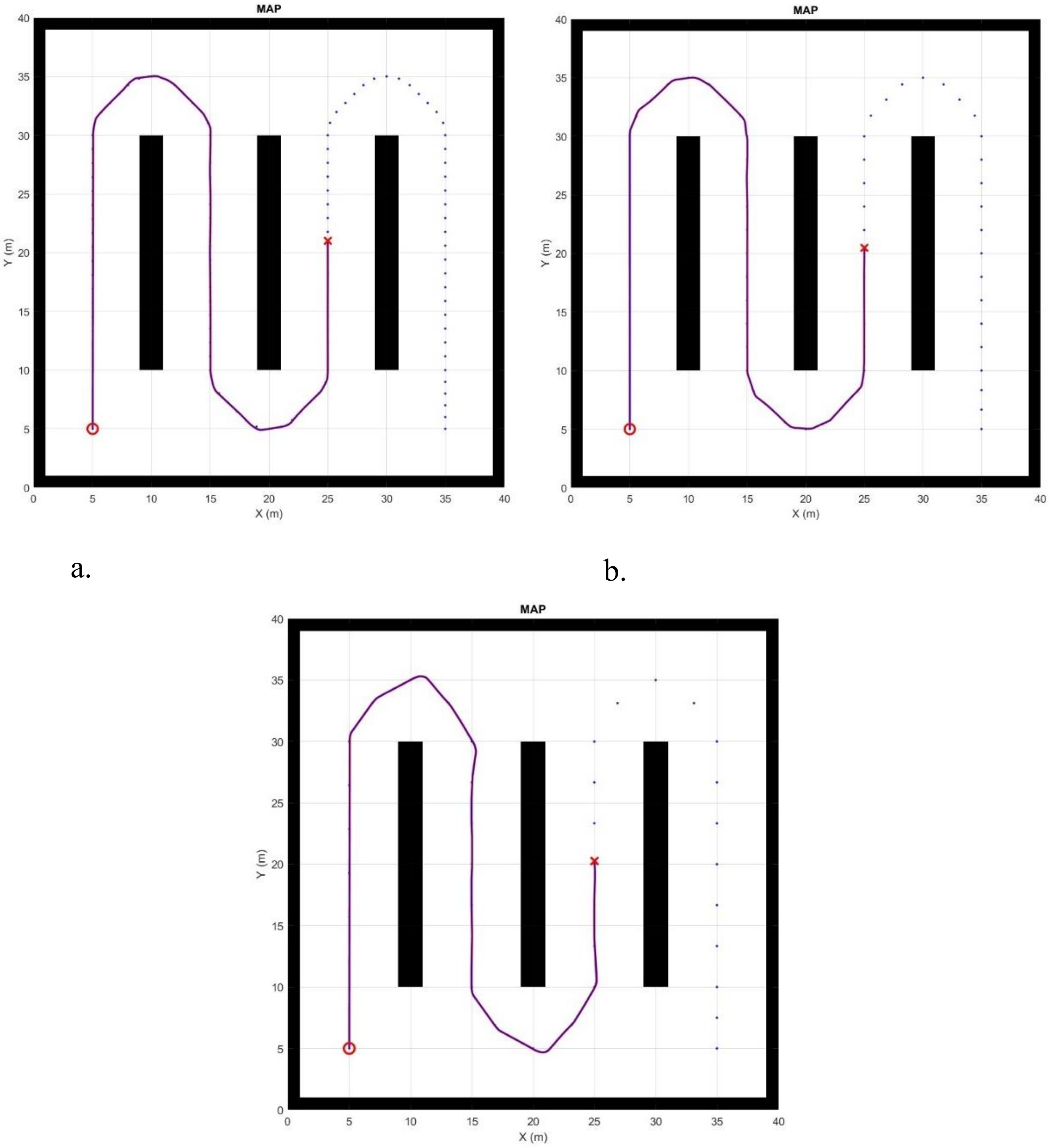
This detailed analysis of the simulations allows to better understand the performance of the algorithm under different operating conditions, providing a comprehensive evaluation of the effectiveness and efficiency of the implemented autonomous driving system.

Analysis of simulations with different waypoint distances on 40x40m map:

To evaluate the rover's performance, three simulations were performed on a 40x40 meter map with different configurations of distances between waypoints.

The figure illustrates the paths a b and c followed by the rover with waypoints positioned at distances of 1, 2 and 4 meters respectively. From observing these paths, we can see how the rover's accuracy in following the planned path decreases as the distance between waypoints increases:

- a: shows the rover's path with waypoints spaced 1 meter apart. In this configuration the rover manages to maintain high adherence to the planned route, making small direction corrections to maintain the correct trajectory.
- b: Displays the route with intermediate points spaced 2 meters apart. Here the rover's accuracy begins to decline, with more noticeable deviations from the ideal path. However, the rover still manages to follow the path with some efficiency.
- c: Represents the route with waypoints spaced 4 meters apart. In this case, the rover shows a significant loss of precision, with larger deviations and a decreased ability to follow the carefully planned path.



a. b.

c.

Figure 24: Path following: a=1m b=2m c=4m.

Trajectory deviation plots show how the rover's deviation from the ideal trajectory increases as the distance between waypoints increases. In particular, the graphs highlight a clear trend:

- distance of 1 meter, the trajectory deviation is minimal, indicating that the rover can follow the path with high precision.
- distance of 2 meters, the deviation from the trajectory increases, but remains within acceptable limits for practical applications.

- distance of 4 meters, the deviation is significantly greater, indicating a substantial reduction in the rover's ability to maintain the desired trajectory.

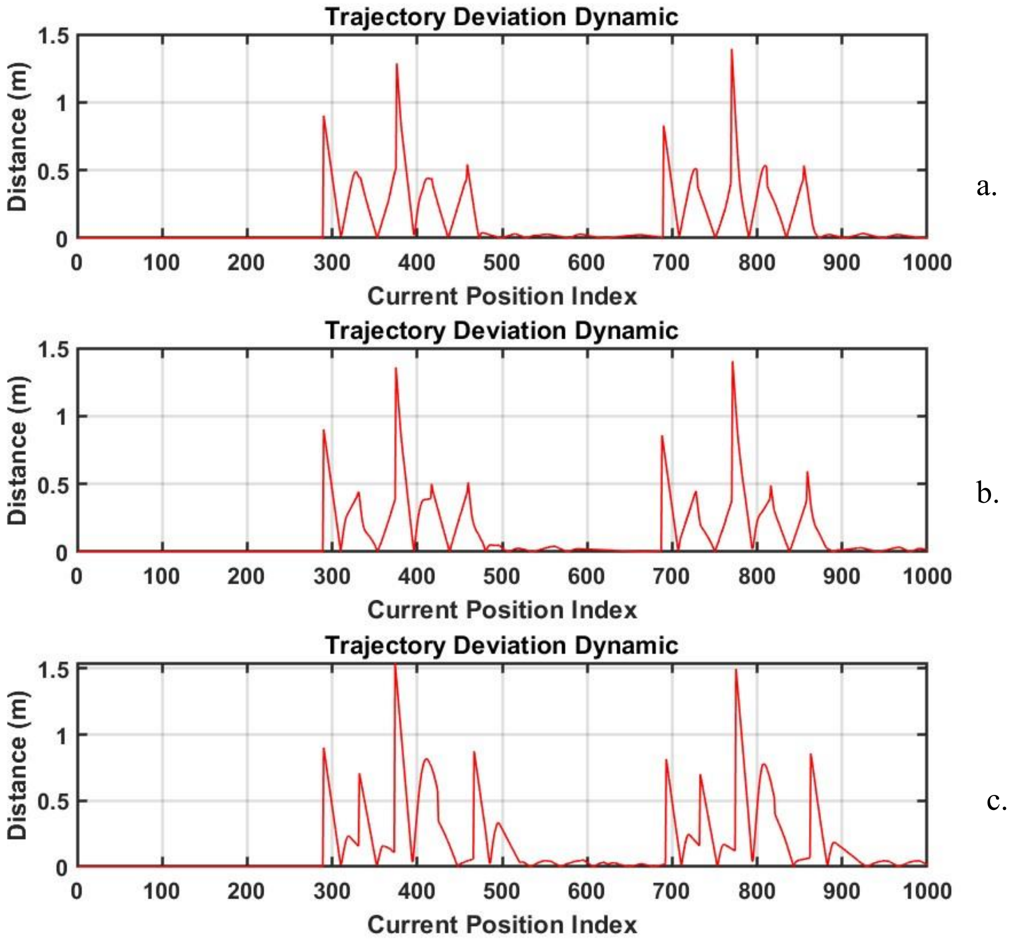


Figure 25: Trajectory deviation rate a=1m b=2m c=4m.

Table 2 shows the RA (Relative Accuracy), ATD (Average Trajectory Deviation), and OF (Oscillating Factor) values for each waypoint configuration. From the analysis of this data two fundamental aspects emerge. The RA factor decreases as the distance between waypoints increases. This was to be expected, as the greater the distance between waypoints, the less accurately the rover can follow its planned route. A lower RA indicates a decrease in the relative accuracy of the system. ATD increases as the distance between waypoints increases. A higher ATD means that the rover deviates more from the ideal trajectory, confirming the loss of accuracy observed in the trajectory deviation plots. The OF decreases as the distance between waypoints increases. A lower OF indicates that the system makes fewer heading corrections with waypoints further apart. This suggests that, with closer waypoints, the rover tends to make more micro-corrections to maintain the trajectory, resulting a nervous behaviour.

Rate	ATD [m]	OF	RA [1/m]	ΔRA% with respect to 1m rate
1 m	0,0211	48	0,9874	0%
2 m	0,0468	41	0,5212	-47,22%
4 m	0,1562	31	0,2065	-79,08%

Table 2: Performance parameters Rate waypoints.

The results in table 2 indicate that, as the distance between waypoints increases, a decrease in performance is observed, although this decrease is not particularly pronounced.

Using constant initial parameters such as r_waypoint = 0.35 meters and Kp = 5, it was noted that a waypoint step of 1 meter was the most accurate compared to steps of 2 and 4 meters. In particular, the 1-meter pitch configuration showed better trajectory control and lower average deviation than configurations with larger distances between waypoints. This suggests that a higher density of waypoints allows the rover to more precisely follow the planned route, thus reducing trajectory deviation and improving movement efficiency.

Analysis of simulations with different position error values on 40x40m map:

To analyze the impact of position error, a band-limited white noise block was used to simulate possible GPS error with randomly varying parameters. This error was added to the rover's x and y coordinates to introduce instability in the acquisition of position data. The error was imposed in an increasing and gradual manner to observe and evaluate the behaviour of the system under different position error conditions.

To analyze the impact of position error on the performance of the navigation system, the optimal parameters determined with a waypoint frequency of 1 meter were used. Figures a, b and c illustrate the paths followed by the rover with position errors of 1 cm, 2 cm and 4 cm respectively:

- a: shows the rover's path with a position error of 1 cm. In this configuration the rover manages to maintain good precision in following the planned path, with minimal deviations from the ideal trajectory.
- b: Displays the route with a position error of 2cm. Here the rover's accuracy begins to decrease, with more obvious deviations from the ideal path, but remains within acceptable limits.
- c: Represents the path with a position error of 4 cm. In this case, the rover shows a significant loss of precision, with large deviations and a reduced ability to accurately follow the planned path.

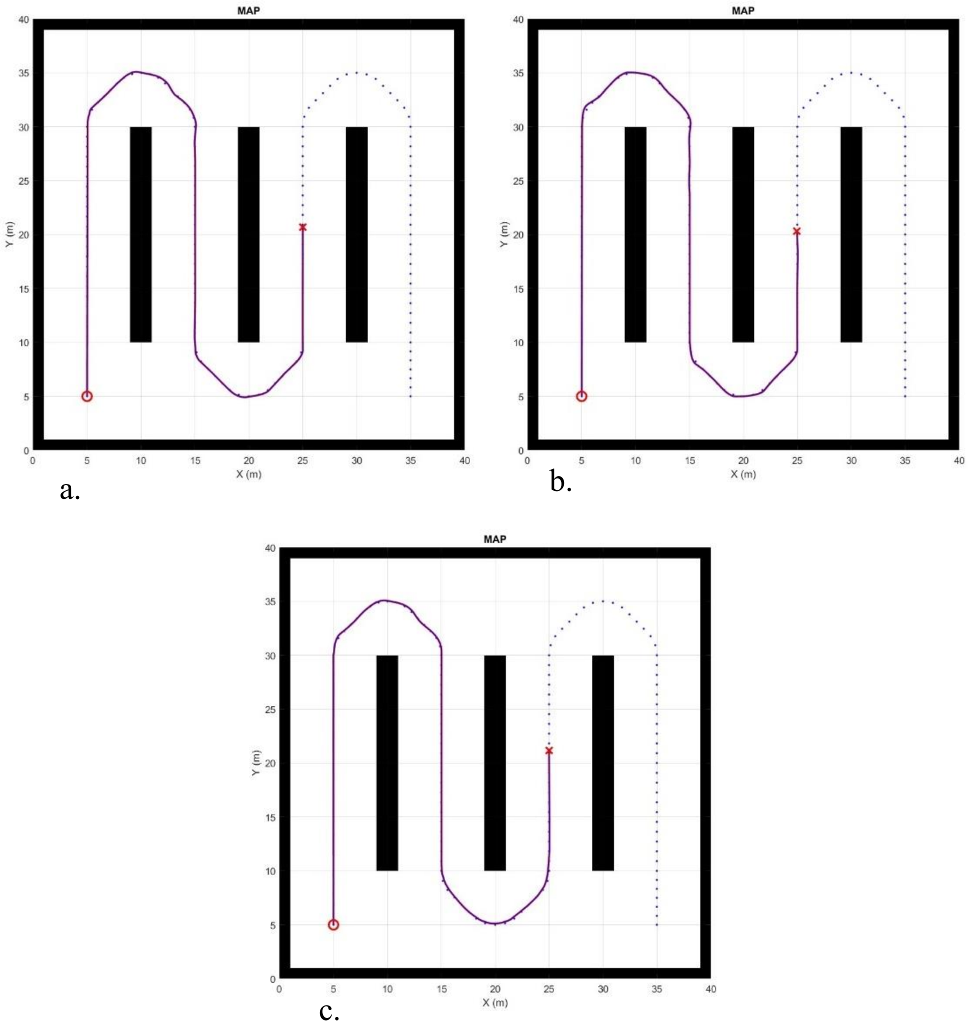
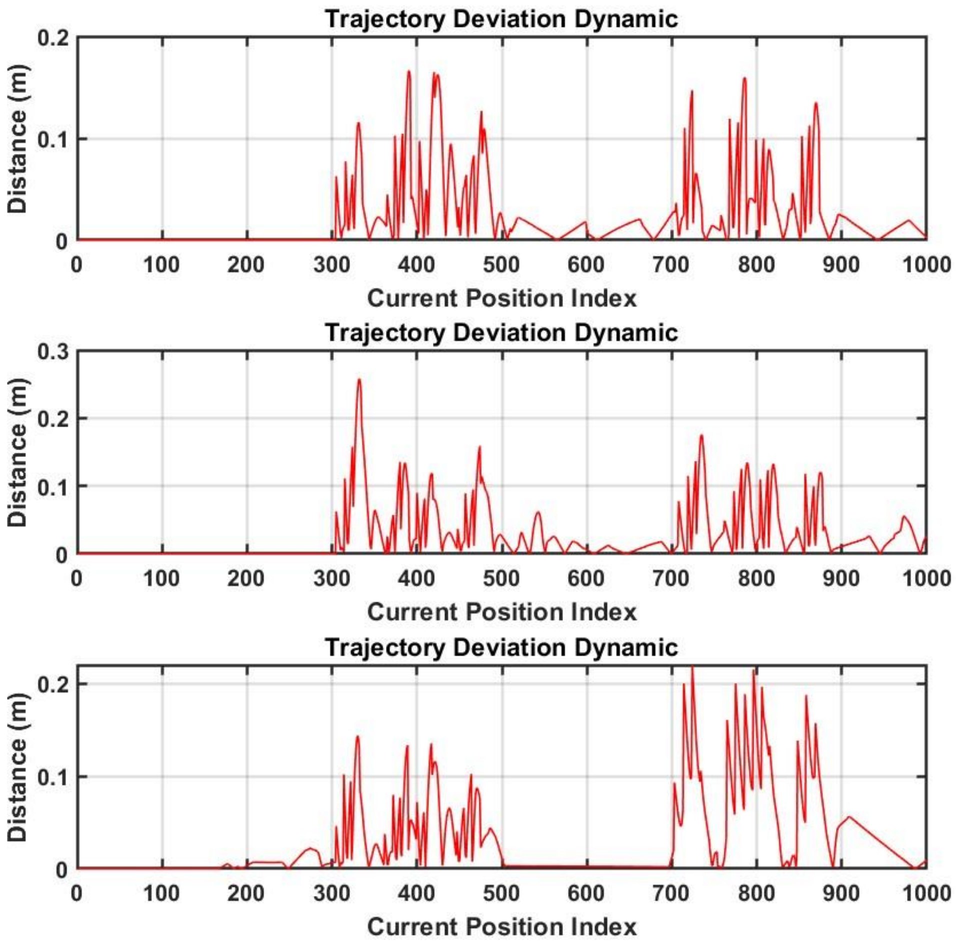


Figure 26: Path following a=±1cm b=±2cm c=±4cm.

Trajectory deviation plots provide a visual analysis of the rover's performance relative to the ideal trajectory. These graphs clearly show how increasing position error negatively affects the rover's accuracy:

- With a position error of 1 cm, the trajectory deviation is minimal, indicating that the rover can follow the path with high precision.
- With a position error of 2 cm, the trajectory deviation increases but remains within tolerable limits for practical applications.
- With a position error of 4 cm, the trajectory deviation is greater, but also follow the trajectory in a good way.



a.

b.

c.

Figure 27: Trajectory deviation error position a=±1 b=±2 c=±4.

Table 3 reports the values of RA (Relative Accuracy), ATD (Average Trajectory Deviation) and OF (Oscillating Factor) for each position error configuration. The following key points are from the data analysis. The RA factor decreases as the position error increases. This was to be expected, since a larger position error implies a lower accuracy in following the planned path. A lower RA reflects a decrease in the relative accuracy of the system. The ATD increases as the position error increases. A higher ATD means that the rover deviates more from the ideal trajectory, confirming the loss of accuracy observed in the trajectory deviation plots. OF increases slightly as position error increases. A very similar OF indicates that approximately equal direction corrections are occurring. This suggests that position error does not affect the OF parameter much.

error position	ATD [m]	OF	RA [1/m]	ΔRA% with respect no error
0 cm	0,0211	48	0,9874	0%
1 cm	0,0215	49	0,9492	-3,86%
2 cm	0,0260	52	0,7396	-25,09%
4 cm	0,0296	51	0,6624	-32,91%

Table 3: Performance parameters error position.

It is shown in table 3 that as the position error increases, the precision of the trajectory followed by the rover loses. However, despite this loss of precision, the system maintains sufficient stability to tackle the route reliably.

Even with higher position errors, the rover was able to complete its intended path without significant mission-compromising deviations. This demonstrates the robustness of the implemented autonomous driving algorithm, which manages to manage position errors while maintaining an acceptable trajectory and stable behaviour. The system's ability to adapt to position errors demonstrates its potential applicability in real-world scenarios, where sensor accuracy may vary due to different environmental factors.

Simulation of Kinematic and Dynamic models on 40x40m map:

Simulations were conducted on a 40x40 meter map to compare the rover's performance using kinematic and dynamic models.

The figure 28 illustrate the paths taken by the rover on a 40x40 meter map using both the kinematic and dynamic models. The differences between the two models are minimal, as both follow the planned trajectory exactly. This result highlights the robustness of the navigation algorithm, which maintains high performance regardless of the model used.

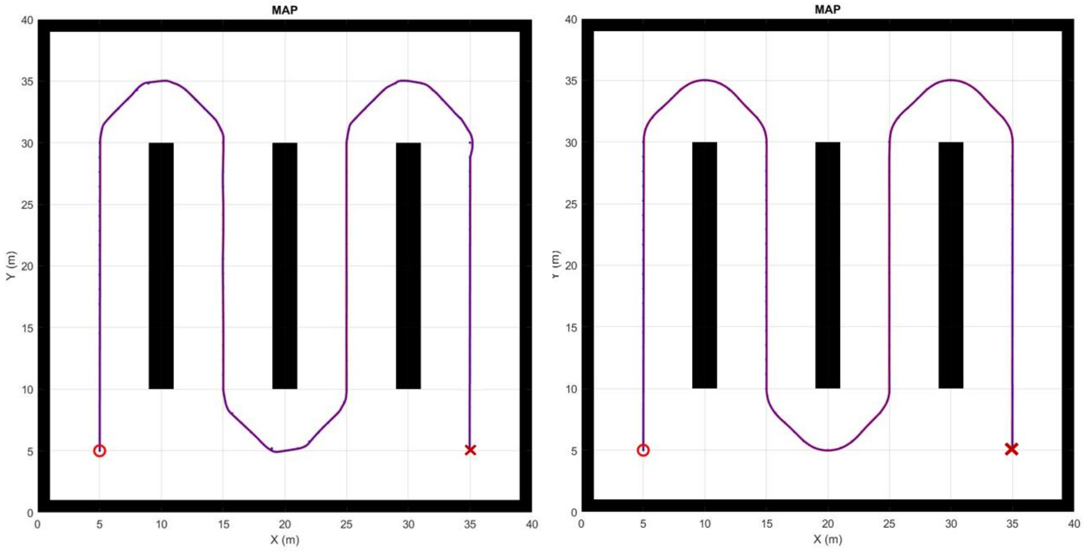


Figure 28: Dynamic and Kinematic map.

The analysis of the angular speeds of the left (ul) and right (ur) wheels show very similar behaviours between the two models. Angular velocities, crucial for direction control, have minimal differences, suggesting that the control algorithm is effective in both contexts.

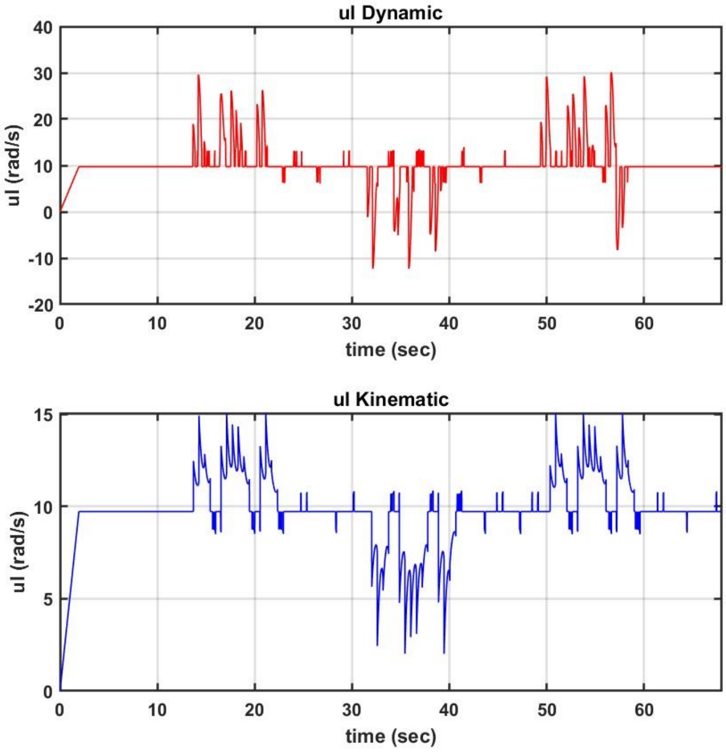


Figure 29: ul behaviour in Dynamic and Kinematic conditions.

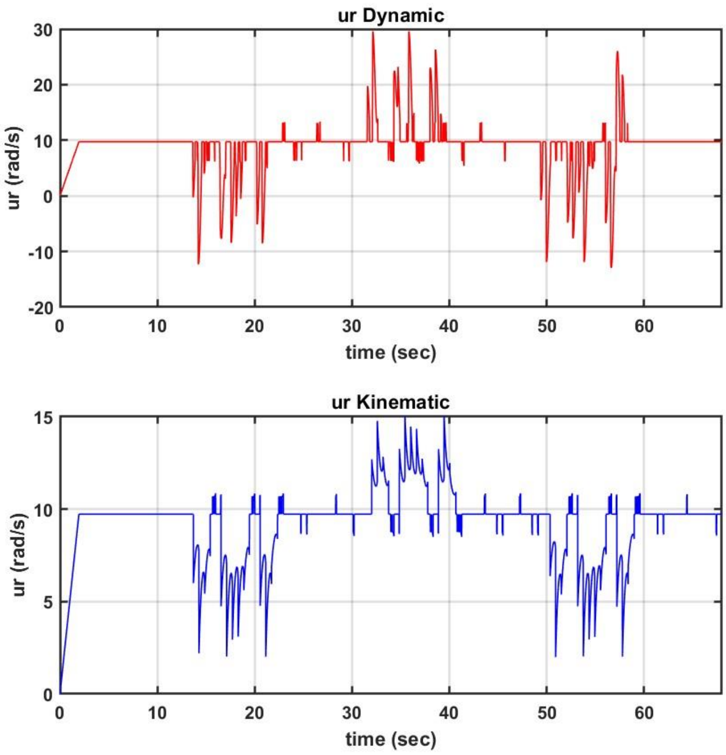


Figure 30: ur behaviour in Dynamic and Kinematic conditions.

The steering parameter theta, which indicates direction corrections, has some differences between models, with the dynamic model making slightly larger corrections when cornering. However, the overall trend remains similar, confirming the accuracy of the algorithm in keeping the rover on the desired trajectory.

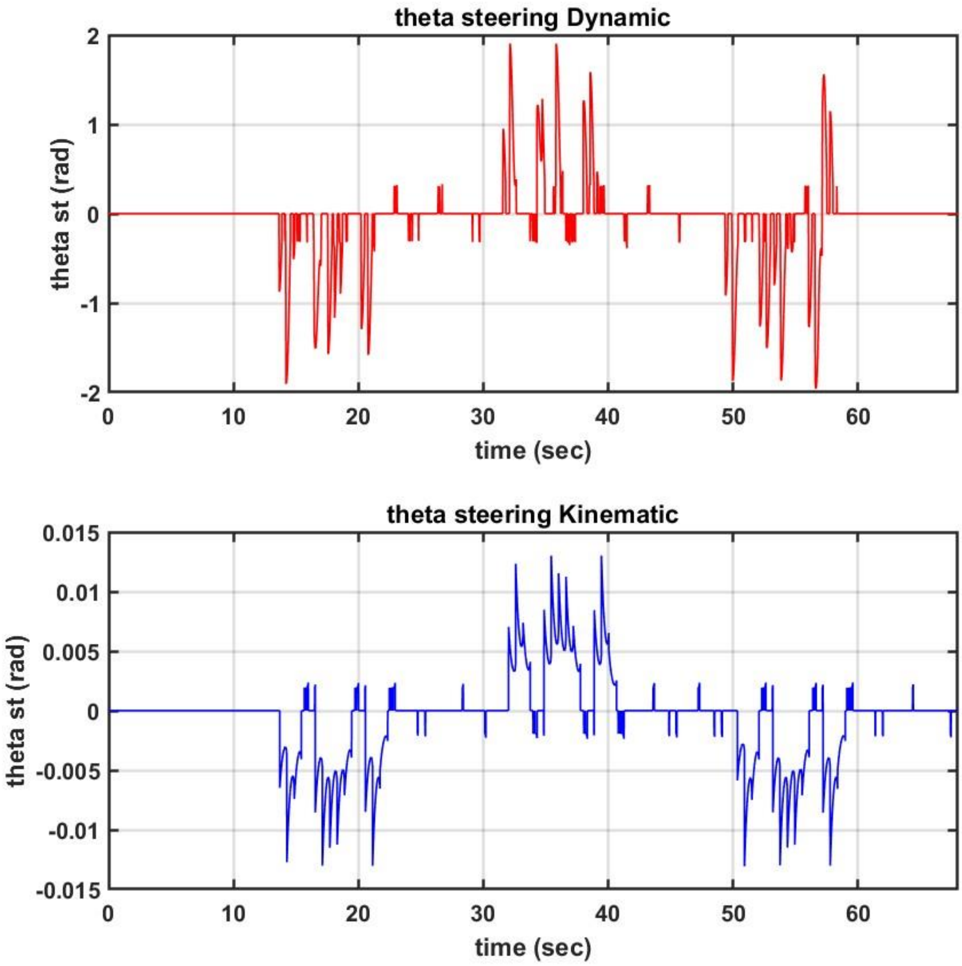


Figure 31: Theta steering behaviour in Dynamic and Kinematic conditions.

The rover's speed is also similar between models. In both cases the deceleration, a function of the steering angle, is adequate. This indicates that the speed control is effective, adapting to the needs of the route to ensure stability and precision.

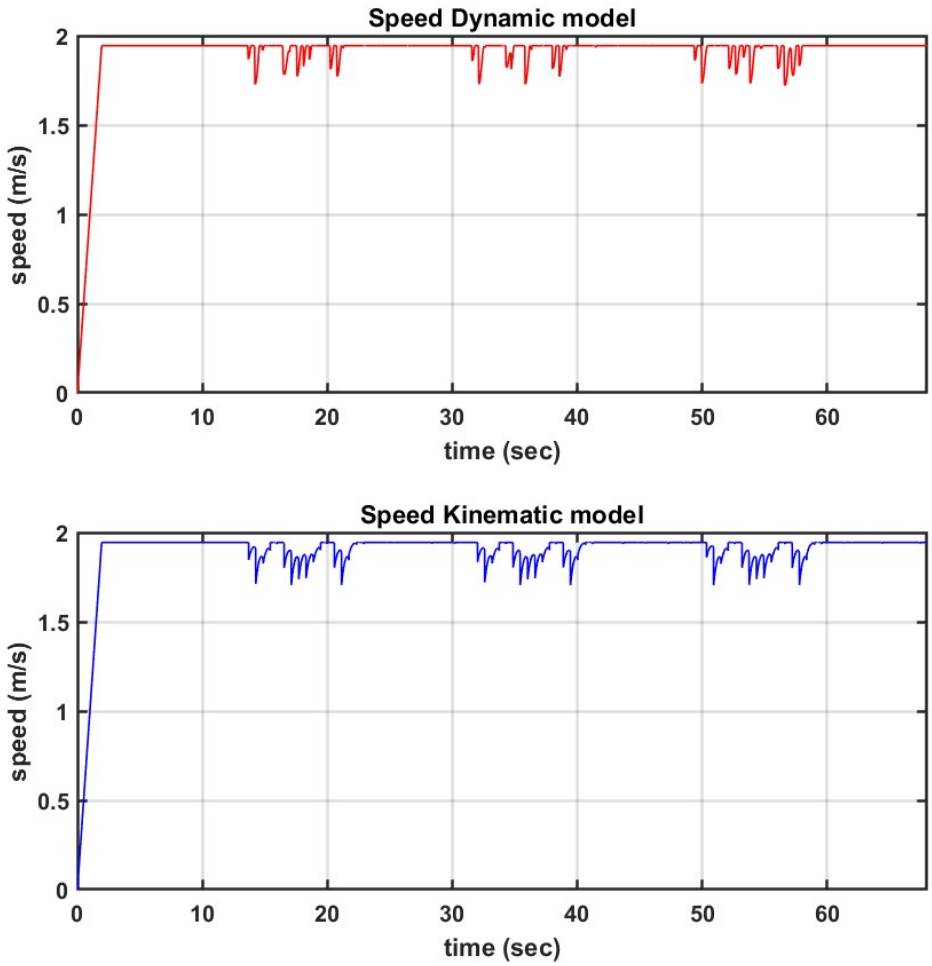


Figure 32: Speed behaviour in Dynamic and Kinematic conditions.

Analysis of trajectory deviation reveals interesting differences. The RA factor is significantly higher for the kinematic model than for the dynamic model, but the dynamic model still maintains a high value, indicating excellent performance. The ATD is very low in the kinematic model, signaling minimal deviation from the planned trajectory. In the dynamic model the ATD is higher but still acceptable. The OF values are similar, with the dynamics slightly lower, suggesting that the kinematic model makes more direction corrections.

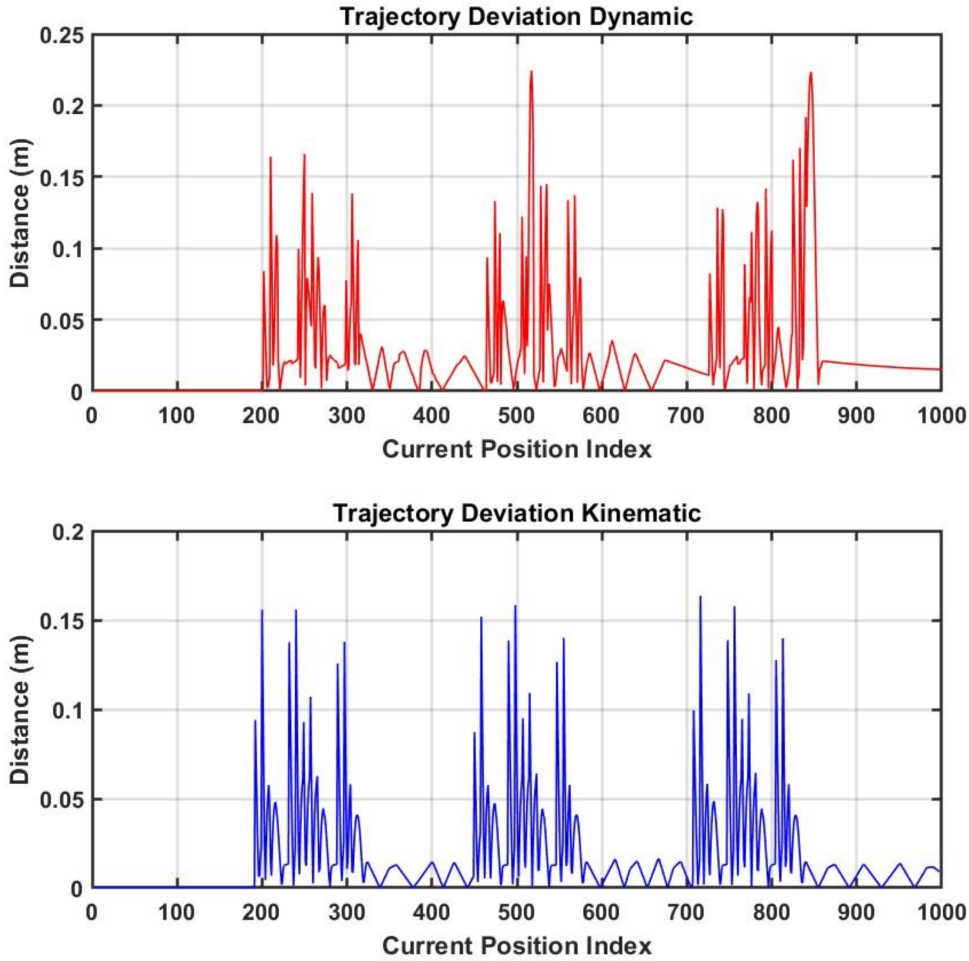


Figure 33: Trajectory deviation in Dynamic and Kinematic conditions.

	ATD [m]	OF	RA [1/m]	ΔRA% with respect to Kinematic
Kinematic	0,0177	65	0,8692	0%
Dynamic	0,0248	64	0,6300	-27,51%

Table 4: Performance parameters in Dynamic and Kinematic model.

These results in table 4 show that the kinematic model is useful for rapid performance evaluation of the navigation algorithm due to its simplicity and lower computational cost. The dynamic model, however, offers a more realistic representation of the rover's behaviour, including complex physical parameters that influence the actual motion of the vehicle. The similarity in results between the two models confirms the robustness and effectiveness of the navigation algorithm.

The approach using both models in the simulation allows for a complete and accurate evaluation of the navigation system performance. This method permits to optimize the algorithm in a simple context and validate it in a more realistic scenario, ensuring reliability and precision for practical applications on the rover.

Sensitivity Analysis:

The sensitivity analysis focuses on two aspects which are the variation of the Kp parameter and the variation of the velocity, comparing the results obtained between the dynamic and kinematic models.

In the case of Kp variation, it was observed that as this parameter increases, the dynamic model improves its performance. In particular, the mean trajectory deviation (ATD) decreases shown in figure 34, indicating a greater adherence to the planned path. The oscillating factor (OF), in figure 35, which represents the number of direction corrections, tends to increase slightly with higher Kp, with about 15 more corrections than at lower values. The relative accuracy (RA), shown in figure 36, tends to increase, indicating an overall improvement in performance. In contrast, the kinematic model shows a more constant behaviour as Kp increases. Only with lower Kp values is a reduction in RA observed, mainly due to a higher ATD, but overall, the system remains stable with respect to Kp variations.

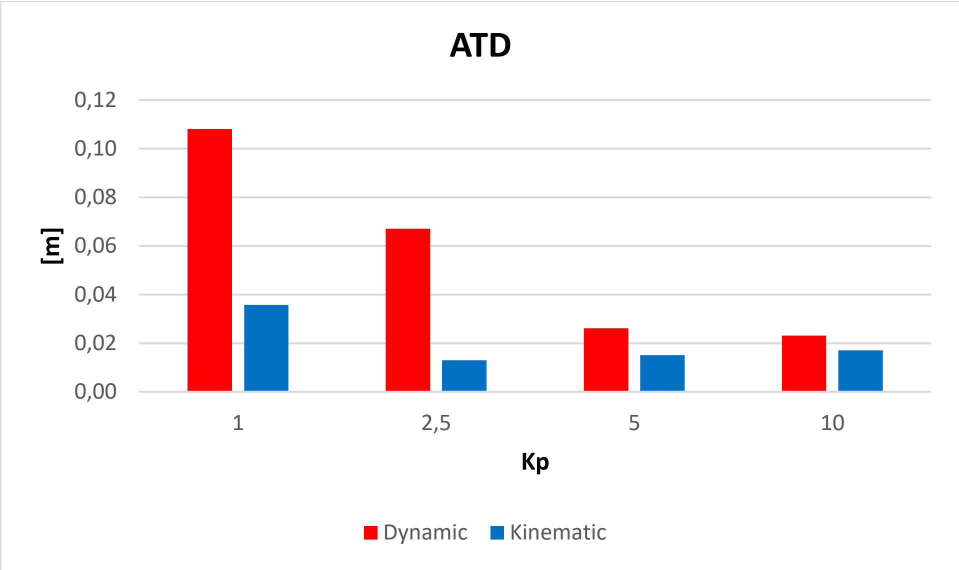


Figure 34: ATD with different Kp.

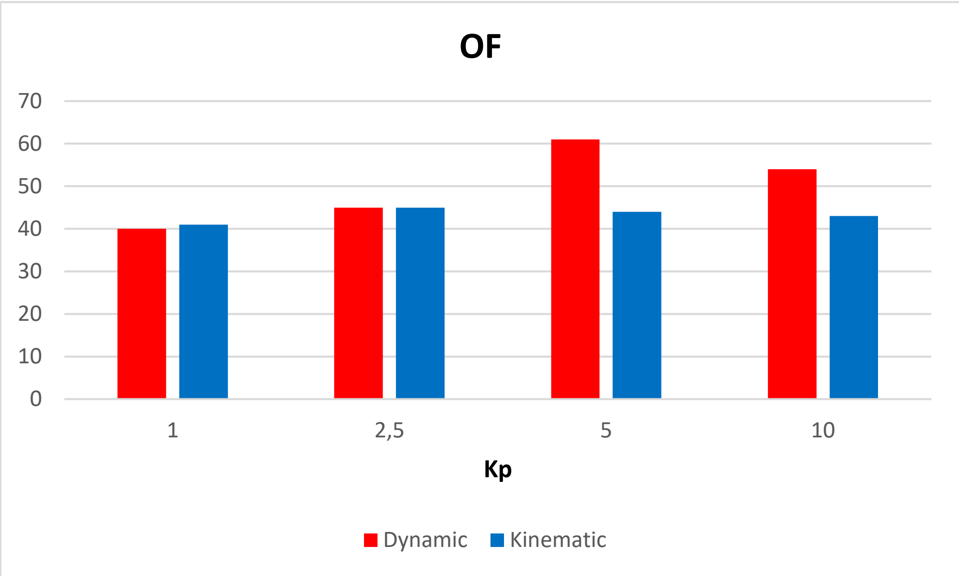


Figure 35: OF with different Kp.

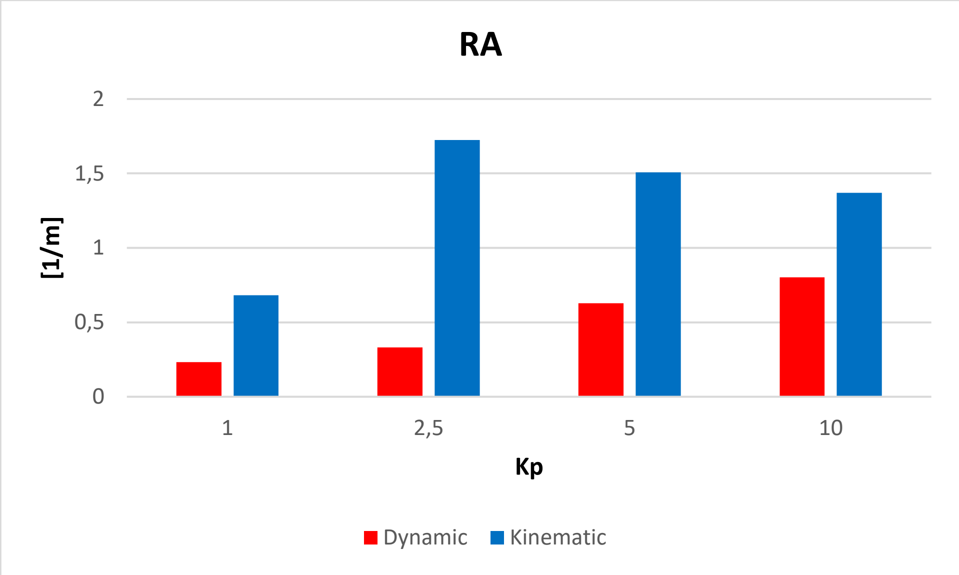


Figure 36: RA with different Kp.

Regarding the increase of the velocity, the results indicate that, in the dynamic model, the figure 37 shows that ATD increases with speed, suggesting that at high speeds the rover deviates more from the ideal path. The OF, in figure 38, remains almost constant, but with slight oscillations more marked than in the kinematic model. The RA parameter, in figure 39, follows a decreasing trend in the dynamic model, indicating a decrease in performance as the speed increases. Differently, in the kinematic model, the ATD and the RA remain practically unchanged as the speed increases. This behaviour highlights how the kinematic model, not being subject to real physical parameters, does not suffer significant distortions or reductions in performance with higher speeds.

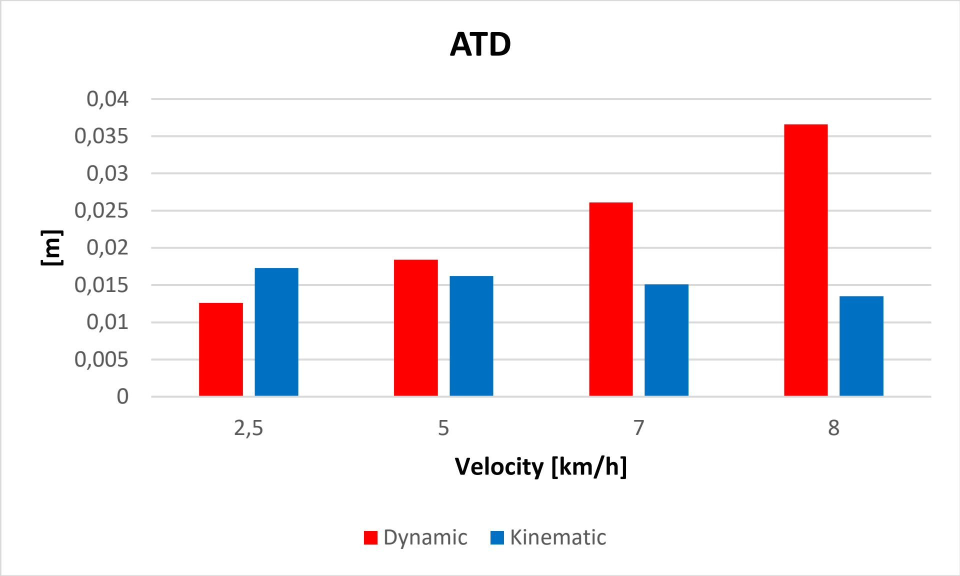


Figure 37: ATD with different speeds.

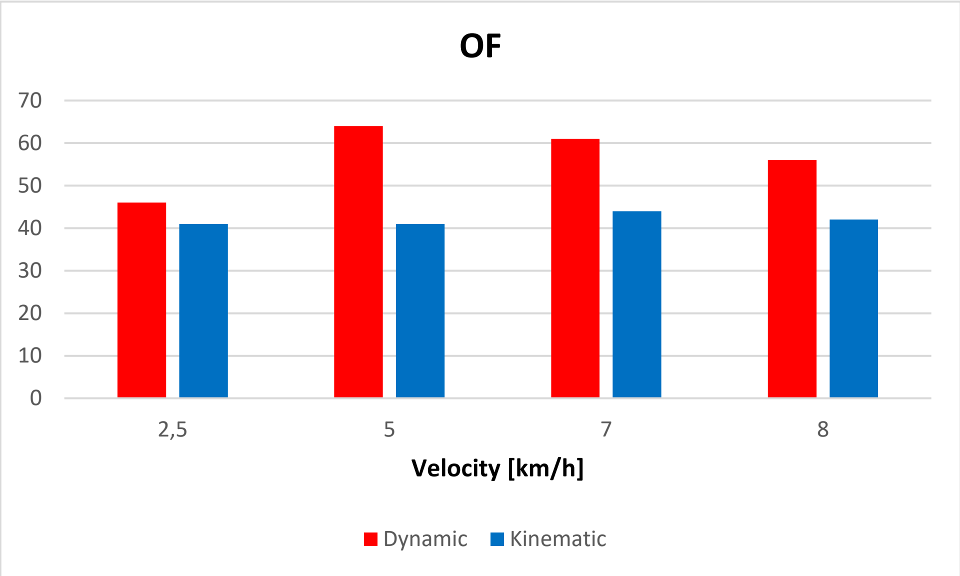


Figure 38: OF with different speeds.

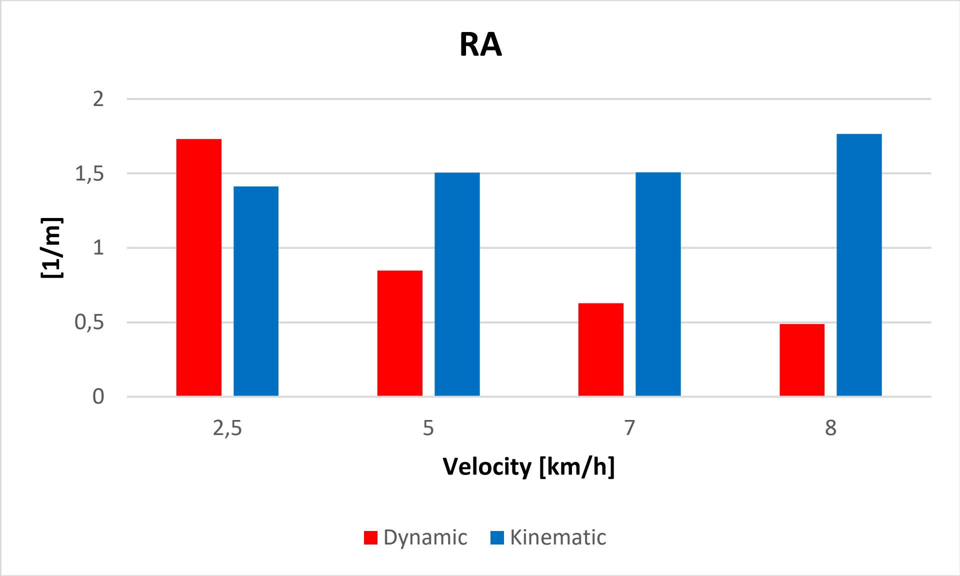


Figure 39: RA with different speeds.

The dynamic model is more sensitive to both Kp and velocity variations, showing improvements with higher Kp but worsening at high velocities. The kinematic model, on the other hand, is more stable and less affected by these variations, especially about velocity, where it maintains constant performance. To confirm this behaviour between Kp and the velocity is shown in figure 40 and figure 41 the 3D map of Kinematic model and Dynamic model respectively.

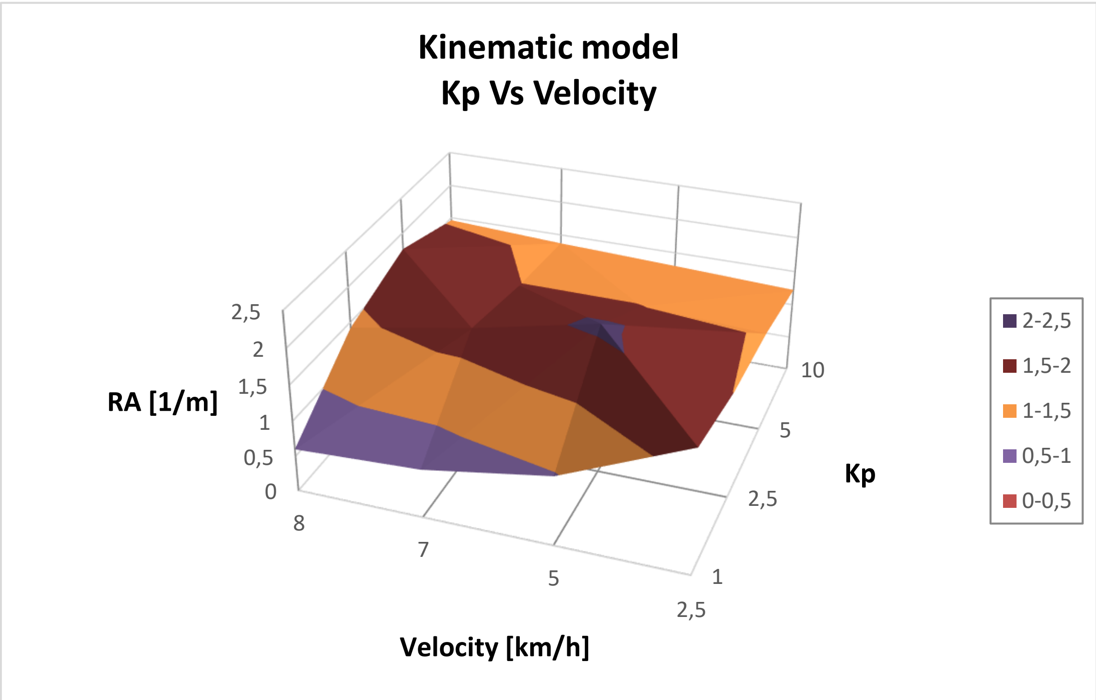


Figure 40: Kinematic model Kp vs Velocity.

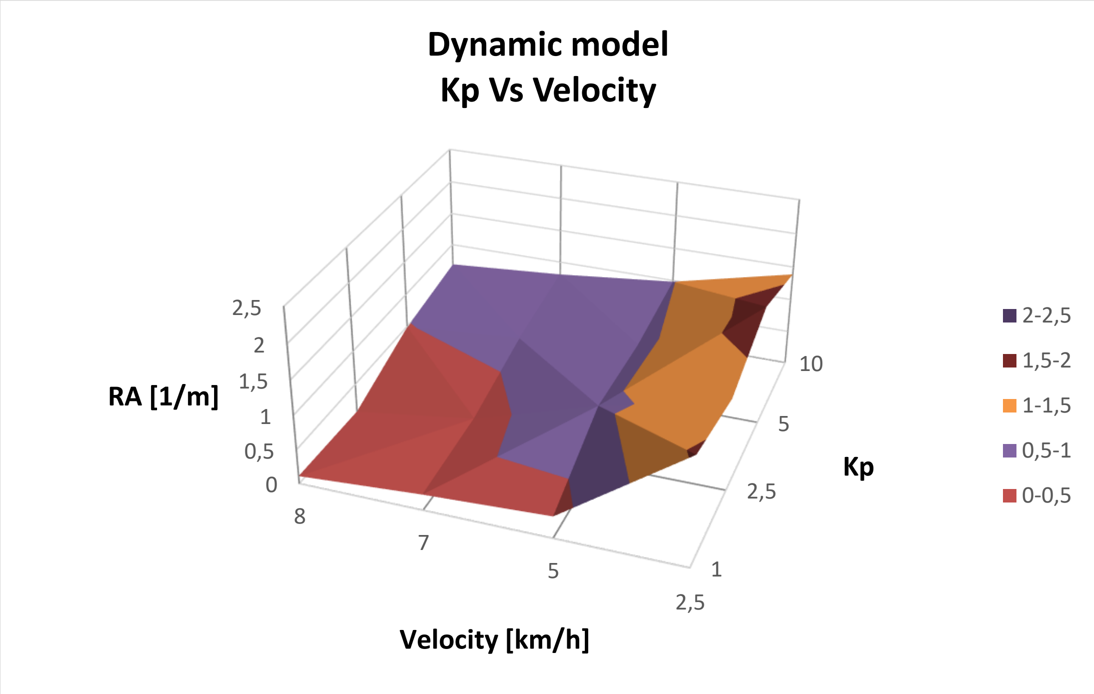


Figure 41: Dynamic model Kp Vs Velocity.

The importance of correct balancing of the set-up parameters to obtain an effective and accurate control system is highlighted:

- Kp, to modulate the corrective response for angular speeds
- waypoint_radius, essential for tracking
- distance between waypoints, essential parameter to define the route to follow
- torque vectoring, rover geometries that allow precise response

The system demonstrated high accuracy in results, with minimal error tolerance and stable behaviour. However, the system can be further elaborated by adding additional physical details to the multi-body model, such as forces and torques acting on the rover to make it more realistic. These improvements would make the simulation even more realistic, allowing for more precise and reliable rover design and optimization. Detailed analysis of the results provided valuable insights for future developments, highlighting key areas to focus on to further improve the rover's performance.

4.2 Discussion of the results

We want to discuss the results obtained where the main objective of this project was to develop a stable and realistic simulation system for a differential drive electric rover. The integration between Simulink and Adams was critical to achieving a simulation that was not only accurate but also reflected the real dynamics of the rover. This approach allowed the system to be continuously tested and improved, providing a solid foundation for future technological developments and practical implementations.

During the design phase, analysis of the results enabled constant incremental improvements. This iterative process demonstrated the importance of having adequate hardware resources for testing, considering the need for computing capacity, but this still led to the conclusion that the data collected was very satisfactory and often exceeded initial expectations. The rover managed to follow the predefined path with high precision, demonstrating the validity of the approach adopted. This precision was achieved thanks to a careful balancing of control parameters, such as the Kp value, which allowed optimizing the reactivity and stability of the system.

The key result of this work was to demonstrate that the integration between the two-software used for simulation and control of the rover produced a stable and highly versatile system. This stability and adaptability make the system suitable for a wide range of future applications, extending its potential use to other types of kinematic and dynamic models. The ability to maintain high and constant performance under different operating conditions and with various models represents an important goal in autonomous driving research. This demonstrates not only the effectiveness of the developed autonomous driving algorithm, but also the robustness of the overall simulation system.

The Relative Accuracy Factor (RA) proved to be a crucial tool for analyzing and understanding the rover's performance during simulations. This indicator combines two fundamental parameters: the Average Path Deviation (ATD) and the Oscillating

Factor (OF). Through RA it is possible to identify the movement characteristics of the rover and determine the changes necessary to optimize its performance.

A high ATD value indicates that the rover is not very accurate in following its planned route. This may suggest the need to increase steering gain (Kp) to improve trajectory accuracy. Instead, a high value of Oscillating Factor suggests that the rover is making too many micro-corrections in the direction of travel, resulting nervous and inefficient behaviour. To solve this problem, you can reduce the steering gain or, as implemented in this study, set a filter that ignores small radius steering up to 2.5 degrees.

The analysis of the simulations conducted on an area of 40x40 meters highlighted how the number of waypoints and the distance between them significantly influence the performance of the navigation system. Simulations have shown that a larger number of waypoints, with a smaller distance between them, improves system performance, allowing the rover to follow the route with greater precision. 1 meter between waypoints produced the best results in terms of trajectory accuracy.

Optimizing these parameters not only improves the RA factor, reducing the average trajectory deviation, but also reduces the number of course corrections required, improving the overall efficiency of the system.

The position error, simulated to emulate the inaccuracy of the GPS signal, was analyzed in detail. As the position error increased, a decrease in the rover's accuracy in following the planned path was observed, although the loss of accuracy was not excessive. This indicates that the system maintains a certain robustness even in the presence of significant positioning errors. The main factors contributing to position error include the quality of the GPS signal and environmental conditions.

The comparison between the kinematic and dynamic models, conducted on an area of 40x40 meters, highlighted distinct advantages and disadvantages for each approach. The kinematic model is advantageous due to its computational simplicity and its ability to provide rapid feedback on the performance of the algorithm under well-defined initial conditions. However, it does not consider all physical parameters that influence a multi-body system, limiting its accuracy in complex scenarios. On the other hand, the dynamic model offers a more realistic and complete view of the simulation, including the complex physical variables that influence the behaviour of the rover. This model, although more computationally demanding, provides more accurate and useful data to validate the system in real conditions. The gradual transition from the kinematic to the dynamic model was effective in obtaining a complete understanding of the rover's behaviour, allowing the algorithm to be refined before applying it in real-world contexts. This approach, applicable in various simulation scenarios, improves the accuracy and reliability of system performance predictions, making the proposed method versatile and robust for future applications.

In the literature, the Simulink-Adams co-simulation approach has been mainly applied to mechatronic systems, with fewer direct applications to vehicles. Many existing studies use open-loop systems, without considering motion feedback, or generate model inputs directly from Simulink source blocks. This project, in contrast, developed a system that autonomously follows a predefined path, without depending on the input source blocks in Simulink. This difference represents an innovation in the approach to vehicle co-simulation, adding a level of autonomy and realism uncommon in other studies. Furthermore, the ability to dynamically integrate the control algorithm with the multi-body model made it possible to obtain a simulation closer to real conditions.

Once validated and running, the simulation system developed in this project can be used to accelerate the prototyping process. This helps you get working prototypes faster and more effectively, reducing the risks and costs associated with development. In the specific context of this project, the system could be further refined to simulate the rover's behaviour in a real agricultural environment, such as a vineyard. This step would represent an important bridge between virtual simulation and practical application, allowing the rover to be tested in real conditions before its physical construction.

To further improve the system, several optimizations could be implemented:

- As for Simulink, integrating obstacle avoidance features and improving the control algorithm could significantly increase the rover's capabilities.
- As for Adams, improving the simulation environment to make it more realistic and complete would allow for an even more accurate representation of the rover's dynamics.

Adding additional physical details, such as the interaction between the wheels and the terrain, could significantly improve the accuracy of the simulation. Furthermore, implementing an energy management system could help evaluate the rover's efficiency in terms of energy consumption.

The results obtained were fundamental for the development and validation of the project. This work demonstrates the crucial importance of the design phase in the rover development process. The ability to test and improve the system in a virtual environment significantly reduces the risks and costs associated with physical prototyping. Furthermore, the approach adopted in this project provides a solid foundation for future developments and applications, helping to make the design and development process more efficient and reliable. This project demonstrated that with careful integration between simulation and control, it is possible to obtain a system that not only meets the initial requirements but can be continuously improved to address new challenges and applications.

5.CONCLUSIONS

The project was based on the integration of different software to achieve the set objectives. Matlab was used for the path planning part, allowing the rover's path to be defined, while Simulink provided the platform to develop the overall control system, while Adams allowed the creation of a detailed physical model for the simulation. This combination allowed us to obtain more positive results, highlighting the precision of the rover in following the defined path, the stability and reliability of the control algorithm, as well as the ease of comparison between the kinematic and multi-body theoretical models.

Analyzing the data it was seen that the multi-body model provides responses almost identical to those of the ideal model, indicating excellent efficiency. Regarding the simulation conditions, it was found that the setup needs to be adapted according to the type of simulation to facilitate the calculation. The behaviour of the rover was analyzed by varying the speeds, highlighting that too high speeds could lead to overturning. Sharp curves were initially tested in theoretical mode to validate the control algorithm, ensuring that the route can be followed even with unusual waypoint parameters. Subsequently, these curves were also simulated in co-simulation to observe whether the rover manages to follow trajectories different from the standard ones.

A key parameter of the project was the Kp value, which proved to be fundamental for the control of the rover's movement because an adequate value setting made it possible to obtain softer or more nervous movement responses, depending on the type of driving desired. Equally important was the radius of the passing points, as it automatically defines the vehicle's tolerance to correctly follow the route.

The developed system demonstrated high accuracy and reliability as the co-simulation between Simulink and Adams provided a realistic picture of the rover's behaviour, allowing the dynamics of the movement and the efficiency of the control algorithm to be analyzed in detail. These results confirm the validity of the approach adopted and the possibility of further future improvements.

This project demonstrates the crucial importance of simulation tools in the design of mobile vehicles, especially in the context of autonomous driving. The use of platforms such as Simulink and Adams have made it possible to develop a co-simulation system that not only guarantees an accurate representation of the rover's movement, but also significantly reduces prototyping times. The ability to simulate rover behaviour in a realistic virtual environment accelerated the design and optimization process, improving the overall effectiveness of the project.

An innovative aspect of this project, compared to existing literature, is the implementation of a fully automated closed-loop system. Unlike many previous studies that use open-loop models or depend on Simulink-generated inputs, the developed system follows a predefined path autonomously, improving the reliability and efficiency of motion control.

To validate and control the results, techniques such as ATD (Average Trajectory Deviation) and OF (Oscillating Factor), already used in previous studies, were used also to find the parameter RA (Relative Accuracy) necessary to obtain a comparison between simulations. These tools allowed the system's performance to be precisely valued, ensuring that the rover followed the path with minimal deviation and responded correctly to control inputs. The use of these techniques confirmed the robustness and reliability of the system, providing a clear picture of its capabilities.

The developed co-simulation system can be applied to a wide range of contexts involving differentially driven vehicles. Through an appropriate configuration of the initial parameters, it is possible to adapt the system to vehicles of different sizes and purposes, making it extremely versatile. This approach can also be extended to other areas of robotics, offering a valuable tool for testing and studying movement in various application scenarios. Co-simulation proves to be an effective and versatile method for the development and validation of autonomous vehicles, helping to reduce prototyping times and improve the precision of motion control. This project demonstrated that, with the use of advanced simulation tools, high levels of precision and reliability can be achieved, paving the way for further developments in the field of mobile robotics.

To further improve the simulation system, one of the most important technological developments to consider is the integration of an obstacle avoidance system. The implementation of an artificial vision system that allows the rover to detect and overcome obstacles along its path would significantly increase the operational autonomy of the vehicle. This improvement would allow the rover to navigate more complex and dynamic environments, adapting over simulation time, terrain variations and unexpected events.

Regarding the simulation environment, a further step forward would be to make the model in Adams more realistic. This includes a more detailed representation of the rover itself, with more accurate component models and a simulation environment that faithfully reproduces the characteristics of the terrain the rover must traverse. For example, adding variables such as wheel grip in different terrain conditions such as wet, dry or rocky, and simulating the effect of these conditions on the rover's performance would provide valuable data for design optimization. It would also be interesting to see how the rover performs in different trail situations, testing the system in a variety of environmental scenarios. For example, simulations that include varying slopes, natural obstacles such as rocks or roots, and different weather conditions such as rain or snow can provide a more complete understanding of the system's capabilities and limitations.

Future improvements should focus on integrating advanced technologies to avoid obstacles and create a more detailed and realistic simulation environment. These developments would not only increase the accuracy and reliability of the system, but also broaden the rover's potential applications in various operational contexts, making autonomous vehicle simulation more versatile and robust. The simulation system developed in this project was specifically designed for small agricultural vehicles, intended for use in vineyards and orchards. Its focus is terrain monitoring, data collection and transportation of small loads. However, the potential of this technology goes far beyond the agricultural sector.

Future developments could include the creation of virtual replicas of autonomous vehicles and their operating environments, which can be used to test and optimize performance in a virtual environment before implementing it in the real world. This approach can significantly reduce the costs and risks associated with physical testing, enabling faster, more iterative prototyping.

In the industrial sector, the adoption of these technologies can revolutionize production and logistics. Autonomous vehicles equipped with artificial intelligence and connected by IoT can optimize warehouse operations, improve supply chain management and increase transportation efficiency within industrial plants. Additionally, they can be used for predictive maintenance, identifying and resolving potential problems before failures occur.

In the research environment, advanced simulation technologies can accelerate the development of new control algorithms and navigation strategies. Researchers can use simulation platforms to test and validate new ideas in a safe and controlled way, reducing the time needed to bring innovations from the laboratory to the applied field.

Advanced simulation technologies, enriched by these innovations, can facilitate faster and more effective deployment of autonomous vehicles in a wide range of applications, improving the efficiency, safety and robustness of autonomous systems.

6. BIBLIOGRAPHY

1. *Rivella J.; Martelli, S.; Mocera, F.; Soma', Algoritmo di guida autonoma per rover elettrico in campo agricolo-industria 4.0.*
2. *Benterki, M. Boukhnifer, V. Judalet and C. Maaoui, "Artificial Intelligence for Vehicle Behavior Anticipation: Hybrid Approach Based on Maneuver Classification and Trajectory Prediction," in* IEEE Access*, vol. 8, pp. 56992-57002, 2020, doi: 10.1109/ACCESS.2020.2982170. keywords: {Trajectory;Hidden Markov models;Predictive models;Autonomous vehicles;Artificial neural networks;Artificial intelligence;autonomous vehicle;intention prediction;LSTM;maneuver classification;neural networks;trajectory prediction}.*
3. *JOSHI, Adit. Real-Time Implementation and Validation for Automated Path Following Lateral Control Using Hardware-in-the-Loop (HIL) Simulation. SAE Technical Paper, 2017.*
4. *KHAN, Imad M., et al. Multibody dynamics cosimulation for vehicle NVH response predictions. SAE International Journal of Vehicle Dynamics, Stability, and NVH, 2017, 1.2017-01-1054: 131-136.*
5. *FENG, Ying; ZHANG, Qiao. Multi-body dynamical modeling and co-simulation of active front steering vehicle. In: 2012 International Conference on Computer Application and System Modeling. Atlantis Press, 2012. p. 371-375.*
6. *LEVESLEY, M. C., et al. Multi-body co-simulation of semi-active suspension systems. Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics, 2007, 221.1: 99-115.*

7. *KHAN, Nawab, et al. Current progress and future prospects of agriculture technology: Gateway to sustainable agriculture. Sustainability, 2021, 13.9: 4883.*

8. *IGNATIOUS, Henry Alexander, et al. An overview of sensors in Autonomous Vehicles. Procedia Computer Science, 2022, 198: 736-741.*

9. *J. Heikkinen, E. V. Serykh and T. Minav, "A control strategy for an autonomous differential drive mobile robot,"* 2017 IEEE II International Conference on Control in Technical Systems (CTS)*, St. Petersburg, Russia, 2017, pp. 83-86, doi: 10.1109/CTSYS.2017.8109494. keywords: {Mobile robots;Trajectory;Wheels;Mathematical model;Simulation;DC motors;mobile robot;differential drive;controller;IMU},*

10. *Mohd Saifizi Saidonr, H. Desa and M. N. Rudzuan, "A differential steering control with proportional controller for an autonomous mobile robot,"* 2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, Penang, Malaysia, 2011, pp. 90-94, doi: 10.1109/CSPA.2011.5759849. keywords: {Wheels;Mobile robots;DC motors;Robot sensing systems;Proportional control;Tuning;mobile robot;differential steering control;straight line;PID},*

11. *J. Song, Z. Zuo, B. Zhu and X. Xia, "A New Planar Path-Following Control for Non-Holonomic Systems,"* 2019 IEEE 15th International Conference on Control and Automation (ICCA)*, Edinburgh, UK, 2019, pp. 254-259, doi: 10.1109/ICCA.2019.8899624. keywords: {Kinematics;Asymptotic stability;Mobile robots;Vehicle dynamics;Closed loop systems;Velocity control;Sliding mode control;planar path following;implicit path},*

12. *R. Singh, G. Singh and V. Kumar, "Control of closed-loop differential drive mobile robot using forward and reverse Kinematics,"* 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2020, pp. 430-433, doi: 10.1109/ICSSIT48917.2020.9214176. keywords: {Mobile robots;Robot kinematics;Wheels;Kinematics;Mathematical model;Service robots;Inverse kinematics;odometry;wheel encoders;differential drive mobile robot;state estimation;localization;proportional controller},*

13. *V. R. N and A. T. Mathew, "Design, simulation and implementation of cascaded path tracking controller for a differential drive mobile robot,"* 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India, 2015, pp. 1085-1090, doi: 10.1109/ICACCI.2015.7275754.*
keywords: {Mobile robots;Kinematics;DC motors;Angular velocity;Wheels;Robot kinematics;differential drive;cascaded control;kinematic and dynamic model;mobile robot},

14. *M. Ramírez-Neria* et al.*, "Distance and Angle Formation Scheme for Collision Avoidance of Differential-drive Mobile Robots,"* 2023 IEEE 6th Colombian Conference on Automatic Control (CCAC)*, Popayan, Colombia, 2023, pp. 1-6, doi: 10.1109/CCAC58200.2023.10333810.*
keywords: {Negative feedback;Robot kinematics;Robot sensing systems;Regulation;Mobile robots;Collision avoidance;Differential-drive mobile robots;Distance-based approach;Collision avoidance},

15. *X. Liu* et al.*, "Challenges and Opportunities for Autonomous Micro-UAVs in Precision Agriculture," in* IEEE Micro*, vol. 42, no. 1, pp. 61-68, 1 Jan.-Feb. 2022, doi: 10.1109/MM.2021.3134744. keywords: {Surveys;Software algorithms;Autonomous aerial vehicles;Software;Hardware;Robot localization;Sensors;Reliability;Task analysis;Payloads},*

16. *T. Brezina, Z. Hadas and J. Vetiska, "Using of Co-simulation ADAMS-SIMULINK for development of mechatronic systems,"* 14th International Conference Mechatronika*, Trencianske Teplice, Slovakia, 2011, pp. 59-64, doi: 10.1109/MECHATRON.2011.5961080.*
keywords: {Mechatronics;Mathematical model;Computational modeling;Solid modeling;Manipulators;DC motors;Control systems;Mechatronic;development;simulation;model;control;ADAMS;SIMULINK},
17. *FAN, Brian Su-Ming. Modeling and simulation of a hybrid electric vehicle using MATLAB/Simulink and ADAMS. 2007. Master's Thesis. University of Waterloo.*
18. *LI, Shengqin; HE, Le. Co-simulation Study of Vehicle ESP System Based on ADAMS and MATLAB. J. Softw., 2011, 6.5: 866-872.*
19. *LIU, S. L.; WANG, Y. Study on dynamic simulation of differential driving cargo transport vehicle. Journal of Intelligent Management Decision, 2023, 2.1: 38-45.*
20. *XU, Tao, et al. Differential drive based yaw stabilization using MPC for distributed-drive articulated heavy vehicle. IEEE Access, 2020, 8: 104052-104062.*
21. *YING, Feng; YIMING, Guo; HONGNI, Zhou. Control for vehicle handing stability based on ADAMS and Matlab. In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). IEEE, 2010. p. V8-546-V8-549.*
22. *Martelli, S.; Mocera, F.; Soma', A. Autonomous Driving Strategy for a Specialized Four-Wheel Differential-Drive Agricultural Rover. AgriEngineering 2024, 6, 1937–1958. https://doi.org/ 10.3390/agriengineering6030113*
23. *Costamagna M.; Mocera, F.; Soma' Studio comparativo di un modello di contatto ruota-terreno per una trattrice agricola in ambiente multibody.*