Politecnico di Torino
Universitat Politècnica De Catalunya

Master's Degree in Mechanical Engineering

Academic Year 2025/2026

Graduation Session 23/07

Design of a self-mobile robot for vineyards applications.

Rear suspension system and self-steering system design.

Relatori:

Massimo Sorli

Antoni Grau

Candidati:

Lucio Ramunno

Abstract

This work presents the design of the rear suspension and steering systems for a mobile robot intended for vineyard operations. To reduce dynamic loads on the vehicle and protect the grape crates from uneven terrain, a transverse torsion-bar suspension was designed. The system integrates seamlessly into the existing chassis, provides ±100 mm of vertical wheel travel. Analytical calculations and finite-element analysis (FEM) are used to verify stress calculations. Furthermore, a complete autonomous steering architecture was developed for reliable row navigation. This includes an outer-loop Pure Pursuit trajectory controller and an inner-loop PID. Validated with a dynamic model in Simulink, the integrated mechatronic system demonstrates trajectory tracking.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Context

In the last decade, field robotics in agriculture has progressed from an aspirational research topic to a practical tool deployed in production. This transition reflects persistent structural pressures on the sector: chronic shortages of seasonal labor, stricter environmental targets, and the need to raise output while reducing inputs through data-driven, precision operations.
Vineyards, in particular, present a demanding yet instructive setting for robotic development: inter-row spacing is tight, terrain can be uneven or sloped, the canopy occludes sensors, and the crop itself is delicate and damage-sensitive. These characteristics make vineyards a compelling proving ground for navigation and gentle load handling in real outdoor conditions.

Mobile platforms designed for viticulture must combine reliable mobility on variable soils with strict operational constraints: adherence to row geometry, limited soil disturbance, protection of the fruit, safe coexistence with human workers, and predictable behavior across the season. Within this envelope, autonomous vineyard *transport* emerges as a particularly tractable class of tasks. Shuttling loads along rows and between the field and collection points is repetitive and logistics-heavy, routes are structured and repeatable, and the perception requirements are lower than those of crop manipulation. As the review of the state of the art in Chapter 2 shows, however, most vineyard robots reported in the literature and on the market are *operation-oriented* (weeding, spraying,monitoring), while the dedicated transport of fragile loads has received comparatively little attention.

The present work addresses precisely this transport-oriented design problem. It builds on a preliminary design phase reported in [1], where the overall mechanical configuration of a low-cost vineyard transport robot was defined: a four-wheeled electric platform with front Ackermann steering, a deck sized for standard grape

crates, a rated payload of 350 kg, and a strict cost ceiling of 13,000. That study delivered a validated chassis concept but left two subsystems open, both of which become critical in vineyard operation. The first is the rear suspension: the legacy chassis provides no compliant element at the rear wheels, so terrain-induced accelerations would be transmitted directly to the fragile payload.
The second is the steering automation: the platform is intended to advance autonomously along the rows, which requires a control architecture that converts a reference path into reliable, well-damped steering action.

This thesis develops these two subsystems. On the mechanical side, a passive rear suspension based on a transverse torsion bar with trailing arms is selected, sized, and verified against static and fatigue criteria, and its integration into the existing chassis is detailed.
On the mechatronic side, a two-loop autonomous steering architecture is designed and validated in simulation: an outer Pure Pursuit trajectory-tracking loop generates the steering reference from the vehicle pose, and an inner proportional–integral (PI) servo loop, designed analytically by pole placement, drives the steering actuator to realize that reference. The two subsystems act as complementary layers of the same objective—protecting the load—at different points in the signal chain: the suspension attenuates the vertical disturbances coming from the terrain, while the steering controller shapes the lateral dynamics induced by path tracking.

1.2 Goal

The overarching goal of this work is to define and verify the key mechanical and mechatronic design choices that make the robot fit for vineyard transport operation, within the geometric, payload, and cost constraints inherited from the existing chassis.

The specific objectives are:

- **Rear suspension concept selection.** Compare the passive suspension layouts compatible with the legacy chassis—outboard coilovers, an inboard pushrod–rocker arrangement, and a torsion-bar arrangement—against packaging, cost, contamination-robustness, and wheel-travel criteria, and motivate the selection of the transverse torsion bar with trailing arms.

- **Torsion-bar design and verification.** Size the torsion bar(material, diameter, effective length, lever arm) so as to provide the prescribed ± 100 mm of vertical wheel travel under the fully loaded condition; verify the static strength through the Von Mises criterion and the infinite-life fatigue strength through the Goodman criterion; introduce an end-stroke bump stop to raise the fatigue safety factor to an acceptable level without sacrificing travel; define the complete chassis integration (splined interfaces, reaction blocks, bushings, base plates) and cross-check the analytical results with finite-element analysis.

- **Autonomous steering design and validation.** Assemble a simulation model of the steering system in Simulink, representing the vehicle with a three-degree-of-freedom single-track dynamic model and the not-yet-selected steering actuator with a first-order transfer function; design the inner PI controller analytically by pole placement and assess its stability (closed-loop poles, Routh–Hurwitz criterion, frequency-domain margins) and its robustness to the uncertainty on the actuator time constant; validate the complete two-loop system through a simulation campaign covering the necessity of the inner loop, the role of the anti-windup mechanism under steering saturation, the tuning of the Pure Pursuit look-ahead distance, and the rejection of a lateral disturbance representative of operation on sloped terrain.

Two boundaries of scope should be stated explicitly. First, the high-level autonomy layer—perception, mapping, and path planning, including the generation of the feasible end-of-row maneuver—is outside the scope of this thesis: the tracking controller operates on a given, precomputed reference path. Second, the specific hardware of the steering actuator is not selected here; the control design is deliberately formulated so that its conclusions do not depend on the precise value of the actuator time constant, and the methodology allows the gains to be updated once the actuator is identified.

1.3 Thesis structure

The thesis is organized to reflect a systems-engineering flow from requirements to verification:

- **Chapter 2** reviews the state of the art along three axes: autonomous platforms for vineyard operations, suspension architectures for agricultural ground robots, and path-tracking control strategies. It positions the present work—a low-cost, passively suspended, Ackermann-steered transport platform with an openly documented subsystem design—with respect to the existing research and commercial landscape.
- **Chapter 3** describes the operating concept of the collaborative transport mission, formalizes the system requirements and design specifications inherited from the customer constraints and from the previous chassis study, and locates the contribution of this work within the functional decomposition of the robot.
- **Chapter 4** develops the concept selection for the rear suspension: it recalls the role of the suspension in vineyard transport, compares the passive layouts

compatible with the legacy chassis, and motivates the choice of the transverse torsion bar as the primary candidate.

- **Chapter 5** details the design and analysis of the torsion bar: pre-load philosophy, governing relations and geometric constraints, material selection and iterative sizing, static and fatigue verification, the needed of a bump-stop, chassis integration, and FEM verification.
- **Chapter 6** introduces the foundations of the autonomous steering system: the two-loop control architecture, the trajectory-tracking problem and the Pure Pursuit law, the structure and actions of the PID family of controllers from which the inner servo is derived, and the Ackermann steering kinematics, from which the minimum turning radius and its operational consequences for the end-of-row maneuver are derived.
- **Chapter 7** presents the design of the autonomous steering actuation system: the vehicle dynamic model and its parameters, the construction of the Simulink model, the design of the inner PI loop by pole placement with the associated stability and robustness analysis, and the simulation campaign that validates the complete system.
- **Chapter 8** summarizes the results and outlines the future developments toward prototyping and field deployment.

Chapter 2

State of the Art

The design choices developed in this thesis–a passive torsion-bar rear suspension and a Pure Pursuit / PID autonomous steering system–do not arise in isolation, but within an active and rapidly growing body of research and commercial development on agricultural field robotics. This chapter frames the work along three axes. Section 2.1 reviews autonomous platforms for vineyard and precision-agriculture operations, distinguishing operation-oriented robots from the comparatively under-explored case of dedicated load transport. Section 2.2 turns to the two subsystems developed in this thesis: Section 2.2.1 reviews suspension architectures for off-road and agricultural unmanned ground vehicles (UGVs), while Section 2.2.2 examines path-tracking and steering-control strategies for ground vehicles, motivating the trade-off between geometric controllers such as Pure Pursuit and more accurate but more complex alternatives. Section 2.3 closes by positioning the present work with respect to the state of the art and, in particular, with respect to the chassis study [1] that this thesis extends.

2.1 Autonomous platforms for vineyard operations

Over the last decade, field robotics has shifted from an aspirational research topic to a practical tool deployed in commercial agriculture. This transition is driven less by novelty than by persistent structural pressures on the sector. Automation technologies—artificial intelligence, robotics, the Internet of Things, and remote sensing—are increasingly seen as a response to chronic labour shortages, climate-resilience challenges, and the need to optimize resource use [2]. In viticulture these pressures are particularly acute: vineyards are perennial, high-value crops worked along structured rows, where many operations are repetitive, labour-intensive, and increasingly difficult to staff, while tightening restrictions on agrochemicals push growers toward mechanical and data-driven alternatives.

These pressures have been met by a broad ecosystem of digital and automation technologies rather than by robots alone. Artificial intelligence and machine learning are used to interpret field data and drive decisions; networks of in-field IoT sensors and wireless sensor networks provide continuous monitoring of microclimate, soil moisture, and crop status; remote and proximal sensing, from multispectral and thermal drones to ground-based imaging, maps vineyard variability at scale; and autonomous robotic platforms carry out the physical tasks in the field [2]. These layers are increasingly integrated, with robots acting as mobile carriers for sensing, mapping, and treatment functions. Figure 2.1 situates the main families of technologies and the role of mobile robots within them [2].

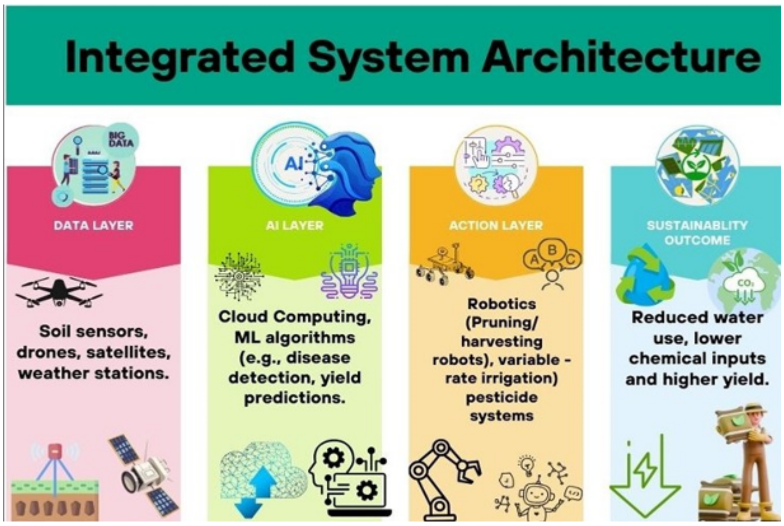


Figure 2.1: Integrated system architecture for viticulture. Source: Horizon ePublishing [2].

Within this landscape, the majority of vineyard robots reported in the literature and on the market are *operation-oriented*: they are conceived to perform a specific agronomic task. Autonomous robotic systems have been developed for selective harvesting, pruning, and pest management, improving operational efficiency while reducing manual labour [2], and complementary sensing platforms based on multispectral drones and ground sensors are used to monitor soil health, water stress, and canopy dynamics in support of precision-agriculture practices such as targeted irrigation and nutrient delivery [2]. Recent research platforms illustrate this trend concretely: fully autonomous robots have been built for high-frequency disease monitoring and AI-based leaf sampling in operational vineyards [3], and heterogeneous fleets combining unmanned ground and aerial vehicles have been validated over entire growing seasons for canopy and water-stress mapping [4].

A recurring observation in this work is that integrated, fully autonomous operation in real vineyards remains constrained by the difficulties of robust navigation, sensor fusion, and energy autonomy under field conditions.

The same operation-oriented emphasis characterizes commercially available systems. Weeding and inter-row maintenance dominate the market: straddle robots such as Naïo Technologies' *Ted* (Figure 2.2) are dedicated to mechanical weeding and canopy management in vineyard rows, guided by RTK-GNSS on previously mapped plots [5].

Figure 2.2: The Naïo Technologies Ted straddle robot for autonomous weeding and canopy management in vineyards. Source: Naïo Technologies [5].

These platforms confirm that autonomous mobility along vineyard rows is, by now, a mature capability when the payload is an agronomic tool rather than a delicate mobile load.

Dedicated *transport* of fragile loads, by contrast, represents a smaller share of reported vineyard robots. While several commercial transport platforms exist, the task has received comparatively less attention in the literature than crop-centered operations, and the engineering design of the subsystems it requires is rarely documented in open form. Yet shuttling grape crates along rows and between the field and collection points is a logistics-heavy, repeatable task that is well suited to automation: routes are structured, perception requirements are lower than for manipulation, and a transport platform provides a natural integration backbone for sensing and safety functions. The design priorities, however, differ from those of an operational robot. The vehicle must protect a fragile, shifting payload from terrain-induced accelerations, preserve ground clearance and stability on uneven inter-row soil, and coexist safely with human workers–requirements that place specific demands on the suspension and steering subsystems rather than on a task-specific end-effector. It is precisely this transport-oriented design problem, and the mechanical and mechatronic subsystems it implies, that the present thesis addresses.

2.2 Suspension and steering-control approaches for agricultural ground robots

The two subsystems developed in this thesis—the rear suspension and the autonomous steering controller—each sit within a distinct body of literature. This section reviews how agricultural and off-road ground robots address these two problems and indicates where the choices made in this work depart from the prevailing approaches. The detailed engineering motivation for each choice is deferred to the corresponding design chapters; the aim here is to position the work within the state of the art.

2.2.1 Suspension architectures for off-road and agricultural robots

Surveys of agricultural unmanned ground vehicles (UGVs) classify their running gear by the degree of compliance and control, ranging from rigid mountings through passive springing to semi-active, actively controlled, and hybrid systems [6]. The distinction is not one of mechanical layout but of how the suspension reacts to the terrain. A *passive* suspension relies on springs and dampers with fixed characteristics: it requires no external energy, sensing, or control, and simply reacts to terrain inputs. A *semi-active* suspension keeps passive springing but continuously adjusts its damping in real time. An *active* (or hybrid active–passive) suspension adds actuators that generate forces on demand, typically to keep the vehicle body level on slopes; this yields high terrain adaptability at the cost of actuators, sensing, control complexity, energy, and price.

Much of the recent UGV literature emphasizes active and hybrid suspensions, whose purpose is to preserve stability and keep the body level on the steep, irregular ground found in many hillside vineyards and orchards. Several studies report active, electronically controlled suspensions that level the chassis on side slopes [7, 8]. The benefit is quantifiable but comes at a clear cost in complexity: an LQR-controlled active suspension on a four-wheeled agricultural robot reduced roll, pitch, and vertical acceleration by roughly 40–70% relative to its passive counterpart [7], and an actively leveled three-wheeled chassis cut its leveling time from eight seconds (passive) to four (actively controlled) [8]—each improvement requiring additional actuators, sensors, and a dedicated control loop.

The present work deliberately occupies the opposite end of this spectrum. The robot is designed to operate along partially structured soils and level inter-row soils of a vineyard and it is bound by a strict cost target. Under these conditions the added mass, energy, and complexity of an active system are not justified: a simple, robust, low-cost *passive* suspension is sufficient to provide the compliance needed to protect a fragile payload from terrain-induced accelerations. A passive architecture is therefore adopted. Several passive layouts are nonetheless possible, among them outboard coilovers, pushrod-rocker linkages, and torsion-bar arrangements. The comparative assessment of these options, together with the detailed justification of the final choice against the packaging, cost, and wheel-travel constraints inherited from the existing chassis, is presented in Chapter 4.

2.2.2 Path-tracking and steering control

Path-tracking control is the component of autonomous navigation that converts a reference path into steering commands, and it has been studied extensively for unmanned agricultural ground vehicles (UAGVs). Recent systematic reviews classify the commonly adopted strategies into Proportional–Integral–Derivative (PID) control, Pure Pursuit, the Stanley Controller, Sliding-mode Control, Model Predictive Control (MPC), and learning-based methods [9]. A persistent difficulty, common to all of them, is achieving precise and robust tracking on unstructured terrain, where wheel slip and dynamic disturbances degrade performance [9].

These approaches trade simplicity against accuracy and modeling effort. Geometric controllers such as Pure Pursuit and Stanley compute a steering command directly from the geometric relationship between the vehicle and a look-ahead point on the path, without requiring an explicit dynamic model [10]; they were widely adopted precisely for their simplicity and effectiveness at low speeds in structured environments [11]. Model-based and optimization-based controllers (MPC, sliding-mode, PSO-tuned, fuzzy-hybrid schemes) generally achieve lower tracking error — in comparative studies„ the Stanley controller attains an average lateral error on the order of 0.19 m, better than Pure Pursuit but inferior to model-based or hybrid controllers [11] — at the cost of greater modeling, tuning, and computational burden.

For the present application–a wheeled platform moving at low speed along the structured rows of a vineyard–Pure Pursuit offers an attractive balance. Its geometric formulation is simple, computationally light, and robust at low speed, and it is well documented in real agricultural deployments: improved Pure-Pursuit schemes with adaptive look-ahead have reported straight-line tracking errors of only a few centimeters in field trials [12].

Pure Pursuit is therefore selected as the outer-loop path-tracking law, with a PID inner loop regulating the steering actuator to the commanded angle. The detailed controller formulation and tuning are developed in Chapter 7; more accurate model-based or adaptive schemes (Stanley, MPC, fuzzy/PSO-tuned controllers) are noted as directions for future work.

2.3 Positioning of this work

The preceding sections allow the contribution of this thesis to be located precisely. On the platform level (Section 2.1), most vineyard robots are operation-oriented and crop-centered, while dedicated transport of fragile loads is addressed by a smaller set of—largely commercial—platforms. On the subsystem level (Section 2.2), the prevailing trend in agricultural UGV suspensions is towards active and hybrid systems for slope leveling, and the most accurate path-tracking controllers are model-based or learning-based schemes of considerable complexity. This thesis deliberately departs from both trends, trading peak capability for simplicity, robustness, and low cost.

The closest existing systems are wheeled transport carriers for orchards and vineyards. The Ant Robotics *Adir Power* is a four-wheel electric carrier for transport and maintenance in orchards and vineyards, with a payload of up to 600 kg and slope capability to 40%, priced from €25,000 to €65,000 depending on configuration [13].

Figure 2.3: The Ant Robotics Adir Power, a four-wheel electric carrier for transport and maintenance in orchards and vineyards. Source: Ant Robotics (via Future Farming) [13].

The *Burro 8.2* is an AI-guided collaborative carrier for specialty crops that transports up to roughly 226 kg and is built mainly around a follow-the-worker operating mode [14].

Figure 2.4: The Burro 8.2 collaborative agricultural carrier. Source: Augean Robotics (via Agtecher) [14].

Against these references, the platform developed here occupies a distinct point in the design space. Its rated payload (350 kg) lies within the range of the closest carriers–above the Burro and below the Adir Power–but it targets a substantially lower cost: the project is bound by a cost ceiling of €13,000 [1], well below the entry price of both the Adir Power and the Burro. Beyond cost, two design choices set the platform apart, each representing a deliberate trade-off rather than an unqualified advantage.

The first concerns steering. Wheeled carriers of this class typically rely on skid or independent-wheel steering, which changes direction by deliberately skidding the wheels. This grants a very small turning radius–the ability to pivot almost on the spot, which is convenient for tight headland maneuvres–but at the price of lateral slip, increased tyre wear, disturbance of the inter-row soil, and jolts transmitted to the load during turns.

The present platform instead uses an Ackermann layout governed by a Pure Pursuit outer loop and a PID inner loop. Ackermann steering keeps the wheels rolling without lateral scrub, so the vehicle is more stable and predictable through a turn and gentler on both the fragile payload and the soil; the cost is a larger turning radius, which demands more space to change rows at the end of a corridor. This kinematic choice is also well matched to the chosen controller: the geometric Pure Pursuit law operates directly on the Ackermann steering angle and is simple, robust, and effective at the low speeds and along the structured rows in which the robot is intended to work.

The second concerns suspension and, more broadly, the level of engineering disclosure. The suspension arrangements of the commercial carriers are not published, so no direct comparison of compliance is possible. What distinguishes this work is therefore not a claimed superiority over undisclosed hardware, but the fact that its running gear is openly designed and verified: a passive suspension, specifically introduced to absorb terrain-induced accelerations and reduce the payload accelerations, and sized against static and fatigue criteria.

Table 2.1 summarizes this comparison. While payload, speed, and price are publicly specified for the commercial carriers, their steering and suspension designs are not disclosed–in contrast to the documented subsystem engineering that is the core contribution of this work.

Table 2.1: Comparison of the proposed platform with the closest existing wheeled transport carriers for orchards and vineyards.

	This work	**Adir Power** [13]	**Burro 8.2** [14]
Application	Orchard/vineyard transport/maint.	Orchard/vineyard transport/maint.	Orchard/vineyard transport/maint.
Drivetrain	Electric	4-wheel electric	4-wheel
Rated payload	350 kg	up to 600 kg	≈226 kg
Max. speed	low (1–4 m/s)	4 km/h	2.25 m/s
Slope capability	up to 15% (100kg)	up to 40%	—
Steering	Ackermann	skid steer	skid steer
Suspension	Passive	—	—
Indicative cost	≤ €13,000	€25,000–65,000	≈$24,000
Subsystem design	documented	not disclosed	not disclosed

In short, this thesis contributes a low-cost, wheeled transport platform for fragile loads in vineyards, and a documented design and analysis of its rear suspension and steering subsystems, extending the chassis study of [1] on which it builds.

Chapter 3

System Requirements and Project Decisions

3.1 Operating concept

Before formalizing requirements, it is worth describing concretely how the proposed robot is meant to operate in the field, since this operating concept is what drives the functional requirements that follow.

The robot is conceived as a collaborative transport platform that works *alongside* the human pickers rather than replacing them. During harvesting, the vehicle proceeds autonomously along the vineyard row, keeping a walking pace just ahead of the operators. As the workers move down the row picking grapes, they place the filled crates directly onto the robot's deck, so that the load is collected progressively without anyone having to carry crates back to a collection point. The robot maintains a prescribed lateral offset and heading within the row, holding a steady trajectory so that the deck stays within easy reach and the pickers can keep working without interruption.

When the robot reaches the end of a row, it executes an autonomous end-of-row maneuver: it leaves the current row, turns through the headland, and steers into the next row to resume the same collaborative advance. Throughout the mission the platform must move gently—limiting vertical accelerations transmitted to the fragile grape crates—and behave predictably and safely around the people working in close proximity. Once full, or at the end of the picking cycle, the loaded crates are transported to the collection point.

This concept—an autonomous, slow-moving, crate-carrying platform that follows the rows, is loaded on the move by the operators, and links rows through steering maneuvers—defines the tasks the robot must accomplish. The next section translates this concept into a set of functional requirements and, together with the customer constraints, into the design specifications that guide the rest of this work.

Figure 3.1 represents the conceptual scheme of the proposed operation: a top-view of the robot advancing autonomously along the inter-row lane, while the operator, walking behind, picks the grapes and progressively loads the filled crates onto the deck.

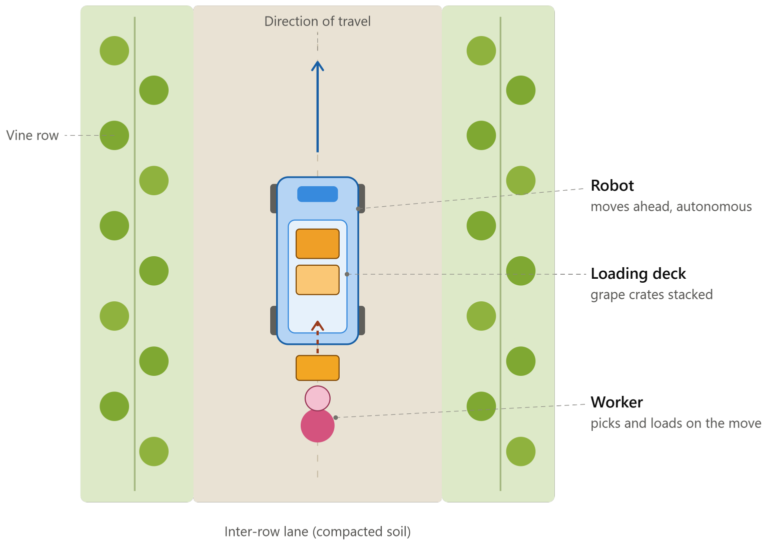


Figure 3.1: Conceptual scheme of the collaborative operation (top view): the robot advances autonomously along the inter-row lane while the operator, following behind, picks and loads the grape crates onto the deck.(Source: author's own elaboration)

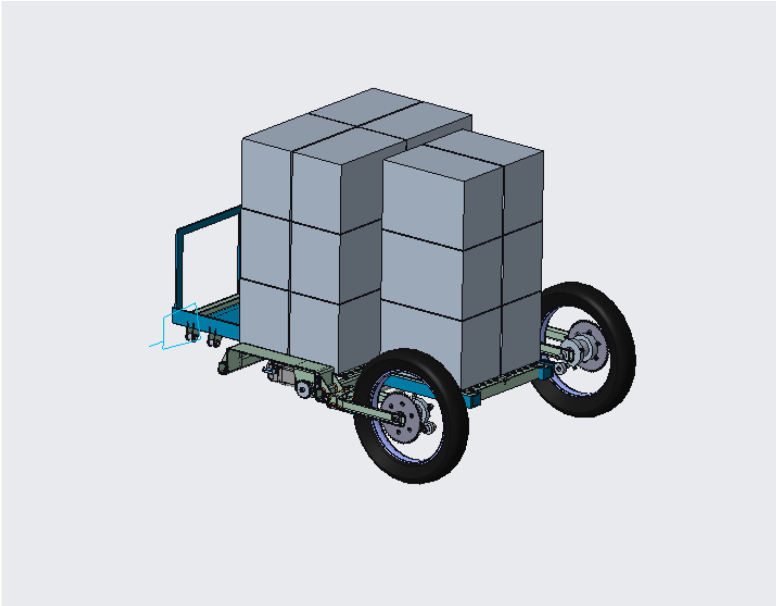


Figure 3.2: The chassis developed in [1] with the grape crates on the deck.

3.2 System requirements

Before any subsystem can be designed, the tasks the robot must accomplish and the constraints it must respect have to be made explicit. This section therefore translates the intended mission into a set of functional requirements, which are then consolidated, together with the customer constraints, into the design specifications that guide the rest of this work.

A typical mission comprises four main functions:

1. **Environment perception and trajectory following**: the robot shall perceive its surroundings and track the planned trajectory. It shall detect obstacles within the sensing range and either generate a collision-free path or command a controlled deceleration to a complete stop within the specified stopping distance.
2. **End-of-row manoeuvre**: the robot shall perform a safe turning manoeuvre to move from one row to the next.
3. **Load transport**: the robot shall carry grape crates on its deck, minimising the dynamic loads transmitted to the payload during traversal and manoeuvres.
4. **Operation on sloped terrain**: the robot shall remain stable and operable on the slopes encountered in vineyard environments.

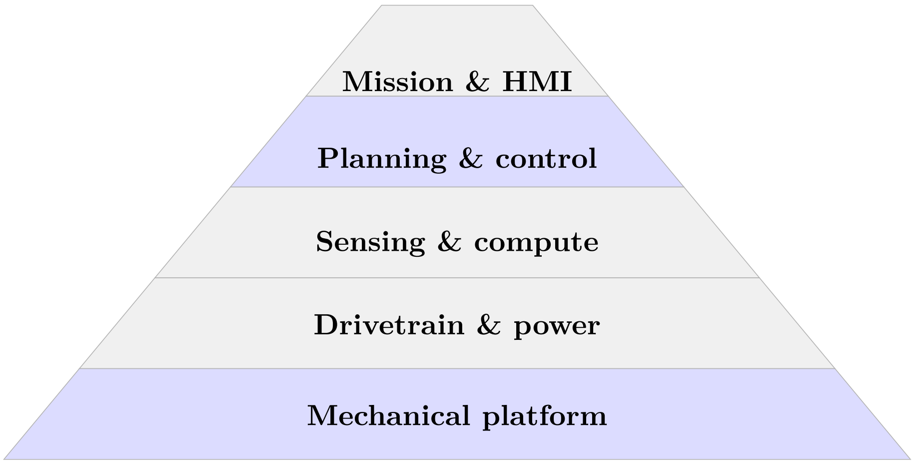
These functional requirements, combined with the customer constraints—such as the maximum overall dimensions, mass, and target cost—inform a broader set of design specifications. The most relevant to the scope of this thesis, together with the ones needed for a general, well-rounded understanding of the project, are summarized in Table 3.1, drawn from the previous work [1].

Item	**Specification / Target**
Operating environment	Compacted, walkable inter-row soil; uneven but not extreme; resistant to minor stone impacts.
Navigation	Collaborative autonomous navigation; status monitoring via mobile app.
Mission	Row perception & following; end-of-row maneuver; grape-crate transport.
Steering architecture	Front **Ackermann** steering
Nominal speed (level)	**1–4 m/s**; higher speeds excluded for worker safety and payload stability.
Speed on slopes	**1–2.5 m/s** recommended.
Obstacle stop distance	Full stop within $\leq$ **1 m** under nominal conditions.
Overall dimensions (max)	Width $\leq$ **1.6 m**; Height $\leq$ **2.0 m**; Length $\leq$ **2.5 m**.
Ground clearance	$\geq$ **0.30 m**.
Unladen mass	$\leq$ **300 kg**.
Payload capacity	**350 kg**.
Towing capacity	**100 kg** on slopes up to **15%**.
Pallet interface	EU pallet **1200** $\times$ **800** $\times$ **144 mm**.
Rear suspension travel	Rear-wheel vertical travel $\pm$100 **mm** about nominal (parallel-to-ground) position.
Power system	Easily swappable batteries; integrated charging port.
Sensors	Proximity sensors at multiple ranges.
Cost target	$\leq$ €13,000

Table 3.1: Project-wide specifications and design targets.

3.3 Functional decomposition.

The development of the robot is structured as a pyramid of work (Figure 3.3), organized into domains with clear prerequisite dependencies. The layers are described below, in order from the lowest (base) to the highest (apex). For each layer, the discussion that follows also states explicitly whether it is designed, extended, assumed, or left out of scope in this thesis, so that the contribution of this work can be located precisely within the overall stack.



Scope of this thesis:
- **Mechanical platform**—Design and sizing of the rear torsion-bar suspension;
- **Planning & control**—Design and tuning of a trajectory-tracking controller (fixed reference; no online re-planning).

Figure 3.3: Robot design pyramid.

Mechanical platform (base layer)

This layer comprises the physical hardware that bears loads and interfaces with the ground: chassis/frame, suspension elements, wheels/tires, hubs and forks. Its goal is to provide geometric compatibility (envelopes and clearances), structural stiffness and durability, and the ride compliance needed to protect the grape crates and maintain traction on the ground.

Powertrain & actuation

This layer comprises the components that generate and transmit motion: traction motors, gearboxes/reductions, mechanical transmissions (chain/belt/gear), motor drivers/inverters, batteries and power distribution, and the steering actuator (motor + gearbox) that delivers the torque and rate needed to command the front-wheel angle. It defines the plant dynamics, limits (saturation, rate), efficiencies, and energy budget.

Sensing & compute

This layer equips the robot with the measurements and computing required for autonomy and safety: LiDAR/cameras and proximity sensors for obstacle detection, GNSS/IMU/encoders for pose estimation, and the embedded compute and communications (MCUs/SBCs, CAN/Ethernet) that acquire, time-stamp, and fuse these signals.

Planning & control

Given the plant and the available measurements, this layer generates and regulates motion. It includes the **trajectory/row planner** and **trajectory-tracking controller** which minimizes lateral and heading errors. Supervisory logic enforces limits and safe behaviors (e.g., smooth deceleration to a stop upon obstacle detection).

Mission management & HMI (apex)

Denotes task-level and coordination—mode management (manual/assisted/autonomous), job sequencing (e.g., row selection, load/unload cycles), and safe-state handling on faults. Also design how the operator interacts with the robot: local controls and indicators, the mobile app for status/commands, teleoperation inputs, and diagnostic/alert feedback. This layer exposes a clear, safe interface to the user while orchestrating the lower layers.

Scope of the present work within the pyramid

This work extends a previous study [1] that developed the initial concept for the vineyard robot's chassis. That study, framed within a systems-engineering "pyramid" (from mechanical foundations upward), delivered a baseline mechanical platform—frame geometry, wheelbase and track width, deck layout for grape crates, and the main mounting interfaces for drivetrain and auxiliaries—mature enough to build a first prototype. Figure 3.4 documents the physical envelope and the packaging logic that the present thesis adopts as given.

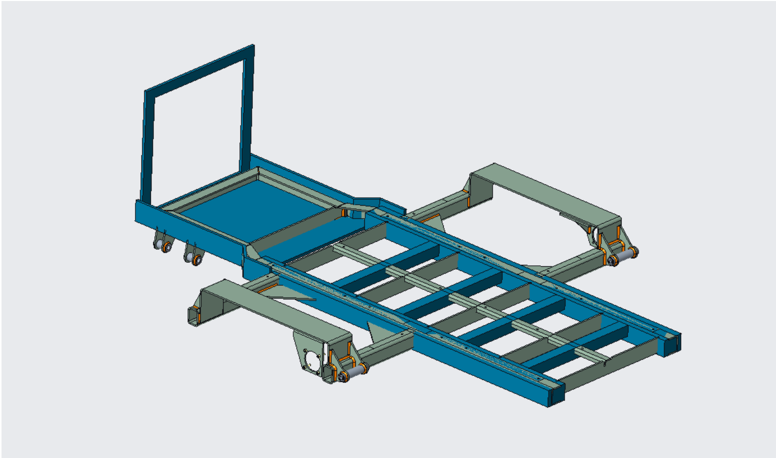


Figure 3.4: Initial version of the chassis.

Table 3.2: Real dimensions and dynamic-model parameters of the robot.

Parameter	Symbol	Value
Total mass (worst case)	m	650 kg
Wheelbase	L	1.6 m
Track width	w	1.3 m
CG-to-front-axle distance	a	0.6 m
CG-to-rear-axle distance	b	1.0 m

Building on this foundation and following the pyramid approach, the work reported here advances the subsequent layers. While the chassis established the overall dimensions and payload architecture, it also exposed a gap that becomes critical in vineyard operation: the absence of a dedicated suspension for the rear wheels. The intended environment is compacted soil between rows—walkable but uneven and often dusty. With fragile grape crates on the deck and people working behind the vehicle, the platform must limit vertical accelerations and absorb terrain inputs to do not compromise stability. For these reasons, the present work introduces a rear suspension system specifically tailored to the existing chassis.
The design brief is twofold: deliver a functional level of compliance and damping under realistic loads; and remain strictly compatible with the geometry and hardpoints defined by the prior prototype—no changes to overall geometry. In practical terms, this translates into allocating a vertical wheel travel of approximately ±100 mm about the nominal (parallel-to-ground) position, preserving the baseline ground clearance, and fitting the suspension hardware within the available space around the rear module.

Complementing the mechanical intervention, this thesis also develops the steering automation stack. A trajectory-tracking controller (outer loop) minimizes cross-track and heading errors with respect to a fixed, pre-defined reference trajectory, while an inner loop drives the steering actuator so that the actual front-wheel angle closely follows the commanded angle.
The control design is validated in simulation over representative vineyard scenarios across the nominal speed range; online re-planning and obstacle avoidance remain outside the scope of this work since in part of the trajectory planning system.

Chapter 4

Suspension Concepts and Selection

4.1 Role of suspensions, and their necessity in vineyard operations

A suspension is the mechanical system that connects the wheels to the chassis while allowing controlled relative motion. In practice it combines an *elastic element* that stores and releases energy with a *damping element* that dissipates energy to control oscillations. Together, these elements shape how vertical disturbances from the ground are transmitted to the structure and to the payload.

For a vineyard transport robot the suspension serves three primary purposes. First, **payload comfort**: Grape crates are fragile and sensitive to shocks and high-frequency vibrations; the suspension must limit vertical accelerations at the deck to reduce impact loads on the fruit, minimize the risk of crates tipping or falling off the robot, and attenuate load transients transmitted to the chassis—thereby lowering structural stresses and improving the durability of the frame and joints.
Second, **traction on uneven terrain**: compacted soil between rows is walkable but irregular; maintaining a more constant normal load at each wheel helps preserve grip and steering authority, especially at low speeds and on slopes.
Third, **stability and robustness**: the suspension contributes to ride height and attitude control, helping the platform maintain a consistent posture under load and during maneuvers.

These objectives translate into technical targets and constraints. The robot must accommodate a **wheel travel** of approximately $\pm 100\,\text{mm}$ at the rear to absorb surface irregularities and preserving the baseline ground clearance (about $0.30\,\text{m}$). The wheel rate (equivalent vertical stiffness) must be high enough to avoid excessive sag under the **350 kg** payload, yet compliant enough to filter terrain inputs.

At last: the elastic and guiding elements must fit within the existing chassis envelope and around the rear module, avoiding interference with the chassis and drivetrain.

A well-designed suspension therefore balances competing needs: isolation versus road holding, travel versus packaging. Before comparing specific layouts, it is worth recalling the level of the design space at which this choice is made. As discussed in Chapter 2, the broader classification of suspension systems—passive, semi-active, and active—was already examined, and the latter two were ruled out for this application: hybrid and active architectures, despite their slope-leveling and ride-control capabilities, were excluded on cost and complexity grounds, which are incompatible with the strict cost target of the project. At the opposite extreme, a rigid arrangement (i.e. no suspension at all) was likewise discarded, since the absence of any compliant element would fail to protect the fragile payload and would compromise traction and stability on the uneven inter-row soil—exactly the comfort and stability requirements set out above.

The design effort of this chapter is therefore restricted to the class of *passive* suspensions, which provide the compliance and damping required by the application with a simple, robust, and low-cost arrangement well suited to a strict cost target. Within this class, several layouts are possible; here we introduce and compare two concepts suitable for the rear axle of a compact vineyard robot—the torsion-bar with trailing arms and the coilover (spring–damper) unit—focusing on operating principles, integration implications, and trade-offs. Chapter 5 develops the detailed sizing and verification of the selected concept.

4.2 Passive suspension concepts

Coilover (spring–damper). The coilover combines a spring (elastic element) and a damper (dissipative element) acting in parallel. The essentials parts of a spring-dumper system are showed below, in Figure4.1.

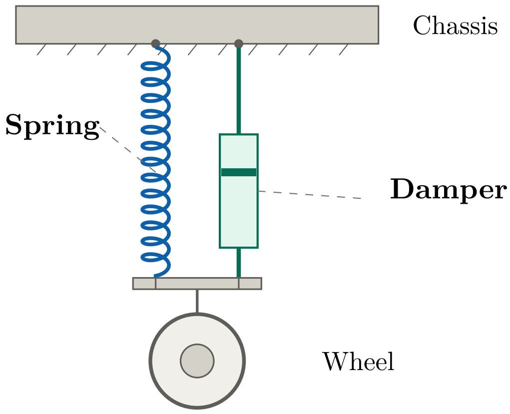


Figure 4.1: Schematic of a coilover unit: a coil spring and a hydraulic damper acting in parallel between the chassis and the wheel. Source: author's own elaboration.

The spring energy and viscous damping model are standard [15].
The **spring** stores elastic potential energy and provides the restoring force:

$$F_s = k_s x_s.$$

Where k_s is the elastic constant of the spring and x_s is the vertical travel distance from the rest position. It supports the static load and pulls the system back toward the equilibrium when displaced.
The **damper** does not store energy: it dissipates kinetic energy, with force proportional to velocity:

$$F_d = c \dot{x}_s.$$

Where again c is the dissipative constant of the dumper and $\dot{x}_s$ is the velocity at which the dumper is compressed or extended from the rest position.
In practice, the spring sets stiffness and ride height; the damper removes energy so that the return to equilibrium is controlled (no sustained oscillations or excessive overshoot).

In vineyard duty, coilovers are attractive for their built-in damping and on-field tunability (spring rate, preload), but they require vertical clearance, rigid mounts to avoid side-loading the rod, and sealing/protection for rod and seals against dust and mud.

Below are three practical coilover layouts used on small off-road machines and mobile robots. They differ mainly in packaging, contamination exposure, kinematic tuning freedom (motion ratio), and cost/complexity.

- **Vertical coilover**:
  Mounted close to the wheel carrier or fork, nearly vertical, acting directly between arm tip and chassis.
  Pros: simplest brackets, low cost, near-constant motion ratio.
  Cons: tall package; rod/seals highly exposed to mud/dust.

- **Inclined coilover**:
  The coilover connects the trailing arm (at an intermediate point) to the chassis at an angle to reduce height.
  Pros: lower vertical package; some tuning via lever arm/angle (motion ratio shaping).
  Cons: motion ratio varies with travel, still exposed to contamination, still needed a vertical clearance.

- **Pushrod–rocker coilover**:
  A short pushrod drives a bell-crank (rocker) that compresses an inboard coilover mounted inside the frame. (Figure 4.2)
  Pros: more compact and fits better with the deck.
  Cons: highest complexity, weight, and cost (extra joints/rocker/axles); tighter tolerances and more maintenance points.

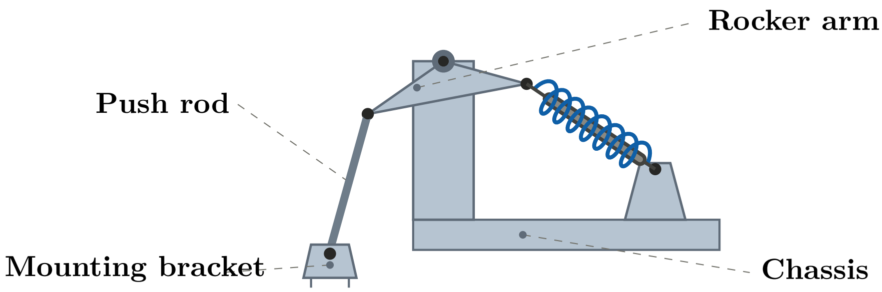


Figure 4.2: Schematic of a pushrod-rocker unit. Source: author's own elaboration.

Torsion-bar with trailing arms. A torsion bar provides springing by working in shear: vertical wheel motion rotates a trailing arm that applies torque to a transverse bar, which twists elastically and generates a restoring torque proportional to the twist angle. For small deflections we can assume:

$$\theta = \frac{TL}{GJ}, \qquad \tau = \frac{Tr}{J}, \tag{4.1}$$

where T [N $\cdot$ m] is the applied torque G [Pa = N/m^2] is the shear modulus, J [m^4] the polar second moment (solid or tubular bar), L [m] the effective bar length, and r [m] the radius of the bar. Damping is not intrinsic and is provided by tires/bushings, bump-stops, or a small auxiliary damper if needed.
Pros: very compact transverse packaging (frees deck volume), robust against contamination, low routine maintenance, good fit with tight rear envelopes and the allocated ±100 mm wheel travel.
Cons: stiffness retuning requires changing bar geometry/lever arm; careful fatigue design and sealing of supports are needed. Well suited when deck clearance and packaging around the rear module are the primary constraints.

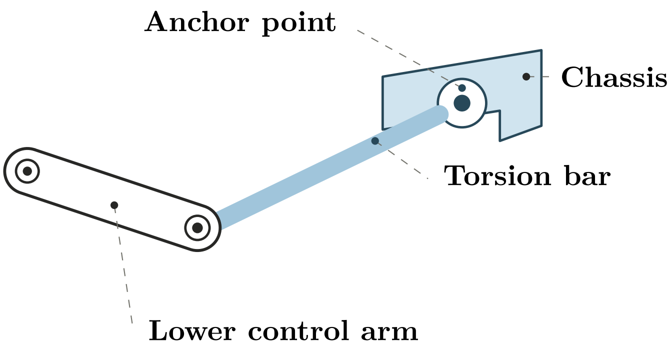


Figure 4.3: Schematic of a torsion bar unit. Source: author's own elaboration.

4.3 Integration, packaging and maintenance

In Chapter 3 we have seen the client specifications. In particular we know that the vertical rear wheel travel must be ±0.1 m, and the ground clearance ≥ 0.30 m. Figure 3.4 shows that, given the existing chassis and rear-fork geometry (tight rear envelope and legacy hard-points), **vertical or inclined outboard coilovers can not be installed without fabricating new towers/brackets on the frame**. Although such modifications to the chassis geometry are technically feasible, solutions that are more compatible with the existing structure are preferred: provided they meet the functional requirements, they avoid unnecessary redesign of a chassis that has already been validated, and they keep added weight and packaging disruption to a minimum. Given these considerations, the most convenient coilover arrangement is the pushrod-rocker layout, which enables a viable implementation without major changes to the existing chassis design.

An inboard pushrod–rocker would, in principle, relocate the spring–damper unit inside the frame and improve packaging around the deck; however, the rocker cannot be oriented perpendicularly to the longitudinal direction because its sweep would intrude into the volume reserved for grape crates. To preserve the deck envelope, the rocker must be reoriented and split into two symmetric, transversely mounted units—one per side—driven by lateral pushrods and arranged to keep the central corridor clear, as illustrated in Figure 4.4.
This transverse layout resolves the interference with the payload but introduces additional linkage volume and part count (two rockers, two pushrods, extra pivots), with corresponding impacts on mass, complexity, and maintenance.

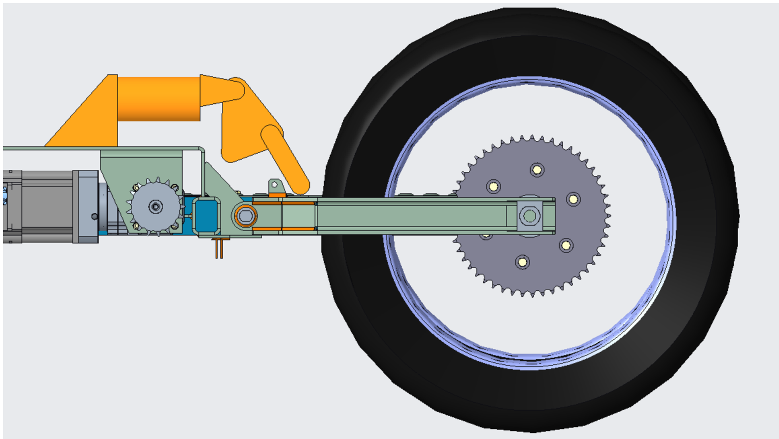


Figure 4.4: Pushrod-rocker mounting design idea.

The last option compatible with the already existed chassis is the torsion bar system.
A torsion bar can be routed either transversely or longitudinally, with distinct packaging consequences. A transverse bar passed through the frame minimizes the vertical stack around the wheel module and keeps the deck volume clear; typically only the arm pivots and the bar supports occupy space near the rear module. By contrast, a longitudinal bar tends to increase the local vertical stack at the wheel module and along the frame rails (detailed explanation in 5) , which can tighten clearances around the deck and service interfaces.

Torsion-bar suspensions have few parts and generally low scheduled maintenance, but fatigue-critical details (e.g., splines and shoulders) must be designed and inspected carefully. Because torsion bars provide stiffness but not intrinsic damping, a small auxiliary damper or end-stroke bump stops may be necessary to achieve the desired dynamic response.

4.4 Comparative assessment and selection

Under these constraints, only two concepts remain truly mountable on the current chassis:
(i) a *transverse torsion bar* with trailing arms
(ii) a *pushrod–rocker* layout driving a coilover.
The torsion-bar solution is compact and frame-friendly, but it requires two bars (one per wheel) and careful routing to avoid compromising the 0.30 m ground clearance. Moreover, with a fully loaded robot mass of about 650 kg (300 kg unladen + 350 kg payload), the bar must be sized from high-strength spring steel with adequate fatigue performance; commonplace low-alloy steels are not sufficient at the required torsional stiffness–stress combination.
The pushrod–rocker alternative offers kinematic freedom, but in our chassis the rocker/coilover assembly would still sit outboard of the frame rails, leaving the damper more exposed to dust and mud, and the linkage adds parts, cost, and alignment effort.
A hybrid layout—torsion bar combined with a pushrod–rocker—would split the elastic and damping duties but is not warranted in this application. The objective is not full chassis isolation or maximum ride comfort for the payload; rather, it is to prevent crate ejection and to keep structural stresses within acceptable limits. In this light, added kinematic complexity and cost are difficult to justify. The primary question, therefore, is whether *torsion bars* can meet the requirements, as it is the simplest and more economical solution (avoiding the pushrod–rocker altogether).

On this basis, the concept selection proceeds with the **torsion-bar suspension** as the primary candidate.

Chapter 5

Torsion-bar design and analysis

5.1 Design intent and pre-load philosophy

The rear suspension must guide each wheel in both directions—upward over bumps and downward into ruts—while attenuating the robot's vertical oscillations.

A torsion bar mounted with the trailing arm exactly parallel to the ground at the no-load position would provide restoring torque for upward motion, but essentially no "pull-down" capability into depressions. To ensure symmetric compliance, the bar must be pre-indexed by a downward angle θ at assembly. In this downward pre-indexed angle θ, the torsion bar is in its rest (stress-free, zero-torque) position. Under the maximum loaded condition (total mass $\sim 650\,\mathrm{kg}$: $300\,\mathrm{kg}$ unloaded + $350\,\mathrm{kg}$ payload), the wheel must settle to the nominal parallel-to-ground geometry after a vertical deflection of exactly $h = 100\,\mathrm{mm}$. This choice is conservative: at lighter loads the deck sits slightly higher and the effective wheel rate increases.

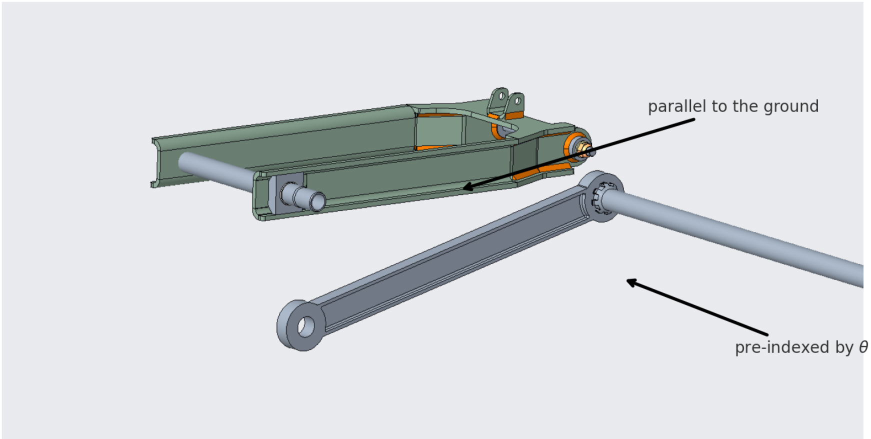


Figure 5.1: Difference between the zero position of the fork (parallel to the ground) and the zero position of the torsion-bar (pre-index).

The geometric link between vertical wheel travel and twist is

$$\sin\theta = \frac{h}{B}, \qquad \text{with} \quad h = 100\,\text{mm}, \tag{5.1}$$

where B is the effective lever arm from the arm pivot to the wheel contact line.

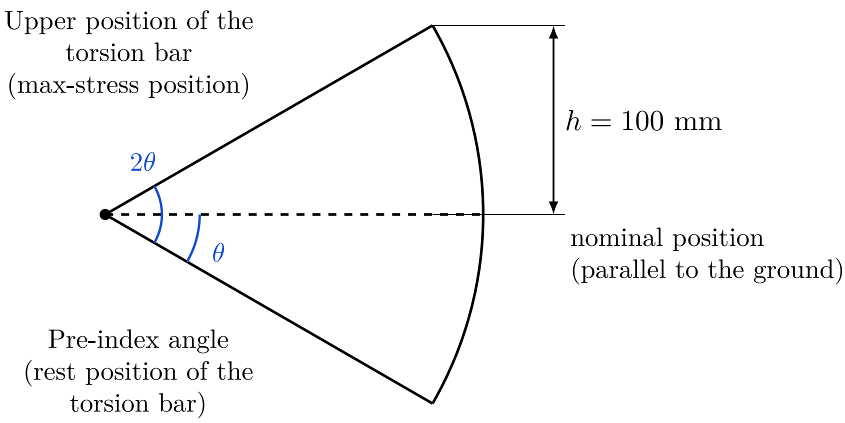


Figure 5.2: Angular sweep of the torsion-bar arm: rest (pre-index) position, nominal position parallel to the ground, and upper (maximum-stress) position, with vertical travel $h = 100$ mm.
Source: author's own elaboration.

5.2 Governing relations and geometric constraints

The torsion bar is sized from the torsion relations in [16], where J is the polar moment of inertia. For a bar of free length L with a shear module G and k_θ the elastic constant.

$$\theta = \frac{T\,L}{G\,J} \qquad [\text{rad}] \tag{5.2}$$

$$k_\theta = \frac{T}{\theta} = \frac{GJ}{L} \qquad [\,\text{N m rad}^{-1}\,] \tag{5.3}$$

$$\tau = \frac{T\,d}{2J} \qquad [\text{MPa}] \tag{5.4}$$

For a solid circular section of diameter d

$$J = \frac{\pi d^4}{32} \qquad [\text{mm}^4] \tag{5.5}$$

The torque is generated by the vertical wheel force acting through the arm:

$$T = F\,B \qquad [\text{Nm}] \tag{5.6}$$

where B is the lever arm and F is

$$F = \frac{mg}{4} \qquad [\text{N}] \qquad \text{(weight evenly distributed).} \tag{5.7}$$

The design objective is to find the bar length L, the diameter d (thus $J = J(d)$) and the lever arm B so that, under the maximum robot mass—assumed evenly distributed—the torsion bar twists by θ_0 at equilibrium. This angle must match with θ_b which is the angle that ensure a vertical travel of $100mm$ with the same lever arm used for θ_0.
This set of requirements can be expressed mathematically as the following chain of equalities:

$$\theta_b = \arcsin\left(\frac{h}{B}\right) = \theta_0 = \frac{T\,L}{G\,J(d)} = \frac{F\,B\,L}{G\,J(d)} \tag{5.8}$$

5.2.1 Geometric constraints from the legacy chassis

Before sizing the bar, it is necessary to fix the geometric envelope imposed by the existing chassis, since the available space directly bounds the two main geometric variables of the problem—the effective bar length L and the lever arm B—and therefore constrains the whole sizing.

The rear module of the legacy chassis leaves a limited room for the installation of the torsion bars. A dedicated study of the rear-module geometry, clearances, and mounting points established that the maximum usable bar length is

$$L_{\max} = 1.16 \text{ m}, \tag{5.9}$$

where L denotes the *effective* (torsionally active) length of the bar, i.e. the portion that actually twists, excluding the splined end that engages the trailing arm and the section clamped to the chassis. These end features do not contribute to the elastic compliance and are therefore not counted in L.

This figure refers to a *transverse* installation, with the bar mounted perpendicular to the robot's longitudinal axis.
Among the possible arrangements, **the transverse orientation is the only one that simultaneously accommodates the required lever arm and the bar length** within the rear envelope, while allowing simple and symmetric fixtures on the frame—one bar per wheel, each with its own independent elastic element. A longitudinal arrangement would require less vertical space and would easily meet the ground-clearance requirement; however, it is precluded by the rear envelope and, more fundamentally, by the kinematic coupling between lever arm and bar length. In fact, for small angles:

$$\frac{h}{B} \approx \frac{F\,B\,L}{GJ} \quad \Rightarrow \quad L \approx \frac{h\,GJ}{F\,B^2},$$

The required bar length scales with $1/B^2$, so the reduced lever arm imposed by a longitudinal routing would demand a bar far longer than the available space. Moreover, a longitudinal mounting would entail a more complex implementation and more intricate kinematic linkages, further discouraging this arrangement.

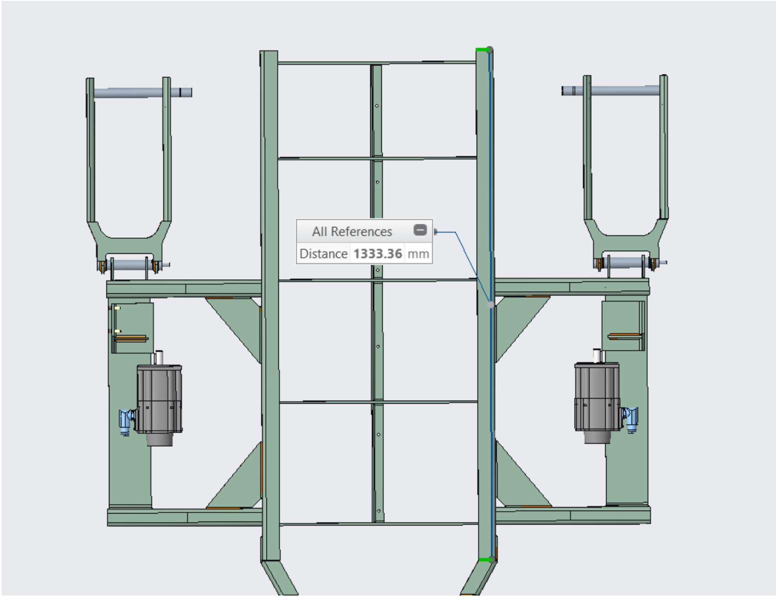


Figure 5.3: Bottom view of the chassis, with the available longitudinal space.

Because L is fixed at its maximum available value, it cannot be used as a free parameter to relieve the bar stresses: any increase in length that would lower the torsional stress is simply not available. The sizing is therefore carried out by iterating the bar diameter d and the lever arm B within the narrow ranges permitted by the chassis, with $L = L_{\max}$ held at its upper bound throughout.This decision also took into account the locking mechanisms required to constrain the two bars and the available locations to secure them to the chassis.

5.3 Material choice and iterative sizing

Because several variables interact (d, B, material properties...), the sizing started from the material choice.
The bar operates as a purely elastic torsional element cycled a very large number of times over its service life; the material must therefore combine a high elastic limit—to keep the working stresses safely within the elastic regime under the required twist—with good fatigue strength, while remaining inexpensive and readily available, consistent with the strict cost target of the project. These requirements point directly to the family of *spring steels*: low-alloy steels specifically developed for elastic components such as springs and torsion bars, designed to be quenched and tempered to a high yield-to-tensile ratio and good fatigue performance. The search was therefore oriented towards this class, looking for a grade that offers the necessary strength once heat-treated while keeping material and processing

costs low. Within this family, a quenched-and-tempered spring steel, **51CrV4**, was selected. From datasheet values in [17]

- E=210 [GPa];
- G=80 [GPa];

The values of $R_{p0.2}$ and R_m are taken from [17]. The table of tempering provides the material properties obtained on a ⌀10 mm round after quenching at 850 °C, oil-quenched, and tempered for 2 hours; the diagram reports R_m, $R_{p0.2}$, elongation, reduction of area, and hardness as a function of the tempering temperature.

Table of tempering values obtained at room temperature on round of Ø 10 mm after quenching at 850 °C in oil

HB		615	595	577	577	550	525	504	455	421	390	371	344	297	253
HRC		58	57	56	56	54.5	53	51.5	48	45	42	40	37	31.5	25
R	N/mm²				2170	2050	1960	1840	1650	1490	1340	1250	1140	990	850
Rp 0.2	N/mm²	1500	1590	1700	1750	1750	1720	1650	1530	1400	1270	1170	1130	900	700
A	%				6.8	7.6	7.8	8.0	8.5	9.8	11.2	12.5	14.6	19.0	22.5
Kv	J		8	10	16	16	15	16	26	28	31	38	46	94	135
Tempering at	**°C**	**50**	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**	**550**	**600**	**650**	**700**

Figure 5.4: Tempering diagram for 51CrV4. Source: [17]

The choice of the tempering temperature deserves a specific justification, since it directly governs the strength values used in the fatigue assessment. The values of R_m and $R_{p0.2}$ adopted in this work are read from the tempering table of the material [17]. A tempering temperature of 450 °C is selected, corresponding to $R_m = 1490$ N/mm^2 and $R_{p0.2} = 1400$ N/mm^2.

This choice is a deliberate compromise rather than an attempt to maximize tensile strength. Lowering the tempering temperature would indeed raise R_m further (up to roughly 2050 N/mm^2 at 200 °C), but this would not benefit the fatigue behaviour: for steels the unnotched endurance limit saturates above $R_m \approx 1400$ N/mm^2, stabilizing at $S'_f \approx 700$ N/mm^2 [**Fenollosa**], so no gain in fatigue strength is obtained by pushing R_m beyond this threshold. On the contrary, a lower tempering temperature increases hardness and reduces ductility and toughness, raising the fatigue stress-concentration factor K_f: since fatigue cracks initiate precisely at the splined ends and fillets of the bar, a more notch-sensitive material would worsen, not improve, the fatigue margin. A harder, more brittle bar would also be more vulnerable to surface defects and would propagate cracks with little warning—an undesirable trait for a load-bearing component operating close to people.

Tempering at 450 °C therefore provides a high elastic limit—keeping the bar comfortably within the elastic regime and preserving the pre-load—while retaining sufficient ductility to limit notch sensitivity. It places the material in the high-strength range where the endurance limit is already saturated, without entering the brittle regime where fatigue performance no longer improves and crack sensitivity grows.

- $R_{0.2}$=1400 [MPa];
- R_m=1490 [MPa].

With L fixed at its maximum available value $L_{\max} = 1.16$ m (Section 5.2.1) and the material properties set by the chosen steel, the two remaining variables—the bar diameter d and the lever arm B—are still coupled through the consistency condition of Eq. (5.8), which requires the bar to reach the prescribed twist θ_0 under maximum load while satisfying $h = 100$ mm. Among the feasible combinations, the configuration adopted uses a torsion-bar lever equal to the fork lever ($B_{\text{bar}} = B_{\text{fork}}$). This 1:1 leverage simplifies the mechanical linkage, minimizes parasitic side loads and alignment errors, and reduces backlash and tolerance stack-ups.

For a fixed value of B, a feasible and packaging-compliant solution was found at:

$$\boxed{d = 0.026\,\text{m}} \qquad \boxed{B = 0.45\,\text{m}} \qquad \boxed{L = 1.16\,\text{m}}$$

The consistency of the selected diameter is verified by computing, for $d = 26$ mm, the twist angle produced by the nominal torque and the corresponding vertical travel. The polar second moment of area is

$$J = \frac{\pi d^4}{32} = \frac{\pi (0.026)^4}{32} = 4.486 \times 10^{-8} \text{ m}^4. \tag{5.10}$$

The vertical force per wheel and the nominal torque are

$$F = \frac{mg}{4} = \frac{650 \cdot 9.81}{4} = 1594 \text{ N}, \qquad T = FB = 1594 \cdot 0.45 = 717 \text{ N m}. \tag{5.11}$$

From the torsion relation, the twist angle under this torque is

$$\theta_0 = \frac{TL}{GJ} = \frac{FBL}{GJ} = \frac{1594 \cdot 0.45 \cdot 1.16}{80 \times 10^9 \cdot 4.486 \times 10^{-8}} = 0.2319 \text{ rad} = 13.28^\circ. \tag{5.12}$$

The corresponding vertical wheel travel follows from the geometric link $h = B \sin \theta_0$:

$$h = B \sin \theta_0 = 0.45 \cdot \sin(13.28^\circ) = 0.1034 \text{ m} \approx 103 \text{ mm}. \tag{5.13}$$

The resulting travel $h \approx 103$ mm satisfies the target $h = 100$ mm: the 26 mm bar is the smallest standard diameter that still reaches the required stroke, providing a slightly higher travel (103 mm) at maximum load. The dimensions of B and L reflect the available rear-module envelope for a *transverse* bar orientation (perpendicular to the chassis/fork direction).
A larger diameter would make the bar too stiff to achieve the prescribed 100 mm.

5.4 Static strength check (Von Mises criterion)

To bound the elastic stresses, the worst-case torsional demand is taken at the "top" position, corresponding (by construction) to a total twist of $2\theta_0$ from the pre-indexed assembly position. For the value of the torque (T) at that angle position the same 5.2 is used, with a value of $2\theta_0$.
With the sized geometry ($d = 26\,\text{mm}$) the peak torque is T_{max}=**1435** [Nm].

The maximum shear stress therefore:

$$\tau_{\text{max}} = \frac{T_{\text{max}}\, d}{2J} = 416 \qquad \text{[MPa]} \tag{5.14}$$

For pure shear, the Von Mises equivalent stress [16] reduces to

$$\sigma_{\text{VM}} \ = \ \sqrt{3}\,\tau, \tag{5.15}$$

so, using the material yield from the datasheet[**Material_diagram**] the allowable shear stress is:

$$\boxed{\tau_{\text{allow}} \ = \ \frac{R_{0.2}}{\sqrt{3}} = 808 \qquad \text{[MPa]}}. \tag{5.16}$$

We reference the (yield) strength $R_{0.2}$, not the ultimate tensile strength R_m, because the torsion bar must operate strictly in the elastic regime. Any entry into plasticity would produce permanent twist and loss of preload, alter the effective stiffness, and compromise the bar's primary function as an elastic energy store; therefore the static check is bounded by yield not by R_m.

The corresponding static safety factor is therefore

$$\text{SF} \ = \ \frac{\tau_{\text{allow}}}{\tau_{\text{max}}} \ \approx \ \frac{808}{416} \ \approx \ 1.944, \tag{5.17}$$

which satisfies the elastic strength requirement for the intended duty.

5.5 Fatigue strength assessment

5.5.1 Fatigue fundamentals

Unlike static failure—governed by a single peak stress exceeding the material yield/ultimate, strength—fatigue is driven by stress (or strain) cycles that nucleate and grow cracks even when nominal stresses remain well below the static limit. The relevant design quantity is the endurance limit of the component, i.e., the maximum fully reversed stress amplitude it can sustain for a very large number of cycles without failure.

Fatigue failure occurs under cyclic loading: even when peak stresses are below the static strength, repeated cycles can initiate and grow cracks until rupture.

A generic stress cycle is characterized by the maximum and minimum stress [18], $\sigma_{\max}$ and $\sigma_{\min}$, from which the alternating and mean components are defined as

$$\sigma_a = \frac{\sigma_{\max} - \sigma_{\min}}{2}, \qquad \sigma_m = \frac{\sigma_{\max} + \sigma_{\min}}{2}, \qquad R = \frac{\sigma_{\min}}{\sigma_{\max}}.$$

Common cycle types are: fully reversed ($R = -1$, $\sigma_m = 0$), pulsating ($R = 0$, $\sigma_m \neq 0$). The worst cycle is the fully reversed. The fatigue behaviour of the

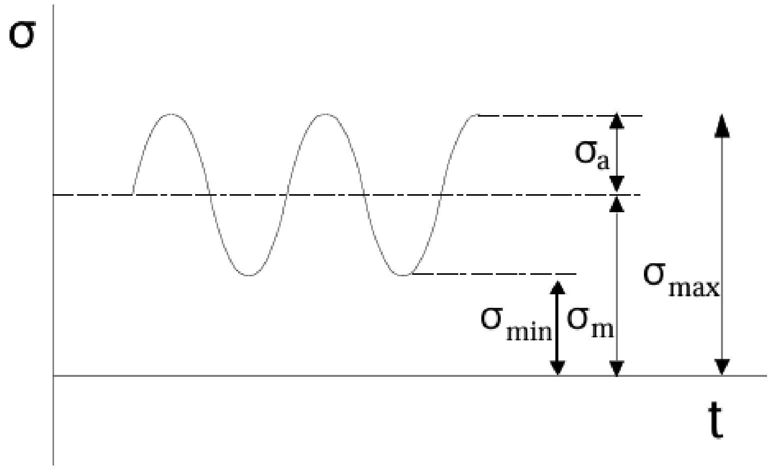


Figure 5.5: Type of stresses. Source:[18]

material is summarized by the S–N (Wöhler) diagram, which plots the stress amplitude against the number of cycles to failure N on a logarithmic scale. For steels and ferrous alloys, the curve flattens into a horizontal asymptote beyond roughly 10^6–10^7 cycles; the corresponding stress is the endurance limit of the material, S'_f. Below S'_f, under fully reversed loading, infinite life is expected. In the simplified S–N construction adopted here, the knee is conventionally placed at 10^6 cycles [18].

A common design check is the Goodman relation or Goodman graph. The Goodman diagram represents the infinite-life criterion in the (σ_m, σ_a) plane. All infinite-life operating points (σ_m, σ_a) must lie on or below this line. When the mean stress is not zero, a tensile σ_m reduces the admissible alternating stress, and the infinite-life condition is no longer expressed by a single horizontal line. A common design criterion is the Goodman relation, which represents the infinite-life boundary in the (σ_m, σ_a) plane: all operating points must lie on or below the line

$$\frac{\sigma_a}{S_f} + \frac{\sigma_m}{R_m} \leq 1, \tag{5.18}$$

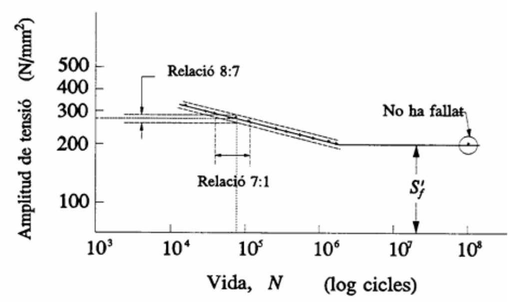


Figure 5.6: Example of a generic Wohler graph. Source:[18]

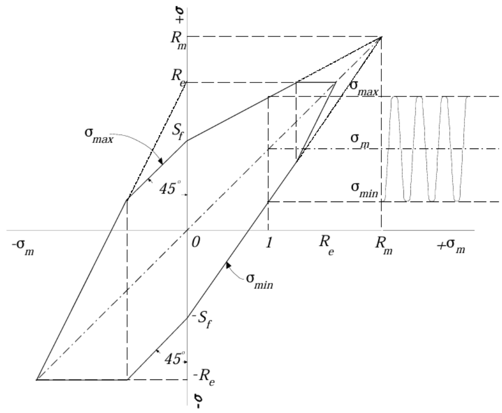


Figure 5.7: Example of a generic Goodman graph. Source:[18]

Figure 5.7 clearly shows that, increasing the tensile mean stress $\sigma_m > 0$ the point shifts to the right and reduces the admissible alternating component σ_a. The same construction applies in shear by replacing σ with τ and using $S_{f\tau}$.

5.5.2 Method

Following the guideline in [18], the component endurance limit is evaluated starting from the unnotched endurance limit S'_f of the material and applying the empirical modifying factors.

$$\boxed{S_f \;=\; k_l\, k_d\, k_s\, \frac{1}{k_f}\, S'_f} \tag{5.19}$$

where:

- S'_f is the baseline endurance limit of the material. From Table 3 (Figure5.8) in [18]

Taula 3

Acer forjat	$S'_f = 0{,}5\ R_m$ ($R_m \leq$ 1.400 N/mm^2) $S'_f =$ 700 N/mm^2 ($R_m >$ 1.400 N/mm^2)
Acer fos	$S'_f = 0{,}4\ R_m$
Fosa	$S'_f = 0{,}4\ R_m$
Alumini Magnesi Aliatges	$S'_f = (0{,}3 \div 0{,}4) \cdot R_m$ [$N = 5 \cdot 10^8$]

Figure 5.8: Table 3 in [18] illustrates different values of unnotched endurance limit for different steels.

For forged steels with $R_m > 1400\,\mathrm{MPa}$ we adopt $S'_f = 700\,\mathrm{MPa}$.

$$S'_f = 700 \qquad [\mathrm{MPa}].$$

- k_l (*load factor*) accounts for the type of load. For torsional loading, the load factor is set to

$$k_l = 0.58$$

- k_d (*size factor*) reduces the limit for larger diameters/sections due to the greater highly stressed volume and defect probability. According to Table 6 in [18], for diameters $7.6 < d \leq 50\,\mathrm{mm}$, the size factor is

$$k_d = 0.85$$

- k_s (*surface finish factor*) penalizes roughness, machining marks and surface treatments that act as micro-notches. According to Figure 3.5 in [18], the surface finish factor is

$$k_s = 0.9$$

- k_f (*fatigue stress concentration factor*) amplifies the local alternating stress due to geometric notches (steps, keyways, splines). It derives from the elastic stress concentration k_t and notch sensitivity q as

$$k_f \;=\; 1 + q\,(K_t - 1), \qquad 0 \le q \le 1,$$

  Stress raisers occur at the bar ends where the connection to the trailing arm and the retention features are machined. The torque-transfer interface is a splined stub; according to the selected spline standard (showed in paragraph 5.6), the nominal major diameter is 32 mm. Given the high stress level and the fact that notches are prime sites for fatigue initiation, a sharp shoulder between the 26 mm bar and the 32 mm spline was avoided. Instead, a gradual diameter transition was adopted to reduce the elastic stress concentration. Increasing the fillet radius r relative to the shaft diameter d raises r/d and lowers the geometric stress–concentration factor $k_t \to 1$; for ductile steels the notch sensitivity $q \to 1$ since r is higher, but the overall result is a fatigue stress concentration factor further reduced with a good final estimation of

$$k_f = 1.15$$

  The detailed diagrams for the estimation of q and k_t values used for the calculation of k_f are illustrated in Figure 3.14 and 3.23 in [18].

The (5.19) relation is valid for fully reversed cycles about the origin, where the mean stress is zero ($\sigma_m = 0$). In the present case, the bar is subjected to one-sided torsion: the outer fibres are always sheared in the same sense, with a minimum stress essentially equal to zero (wheel at -100 mm vertical travel). The loading is therefore pulsating, with $\sigma_{\min} = 0$ and $\sigma_m \neq 0$.

A nonzero mean stress moves the operating point on the Goodman diagram and reduces the admissible alternating stress if infinite life is required.

In this different case, the residual-stress (local-stress) approach is adopted.

For ductile materials: the fatigue stress concentration factor is applied directly to the nominal stresses $\sigma_a = k_f\,\sigma_{a,\text{nom}}$ and $\sigma_m = k_f\,\sigma_{m,\text{nom}}$.
Consequently, the component endurance limit no longer carries the $\frac{1}{k_f}$ term; it is taken as

$$S_f = k_l\,k_d\,k_s\,S'_f.$$

The objective is to compute the component endurance limit S_f using case-specific modifying factors; then determine the operating point and assess life fatigue either analytically via the Goodman inequality or graphically on the Goodman diagram. For a fluctuating (nonzero-mean) stress cycle—the pulsating case—the operating point on the Goodman diagram can be recast by introducing an equivalent fully reversed amplitude.

$$\sigma_{a,\mathrm{eq}} \;=\; \frac{R_m\,\sigma_a}{R_m-\sigma_m}$$

So the 5.18 can be re-written as:

$$\sigma_{a,\mathrm{eq}} \;\leq\; S_f.$$

Same formulas and considerations can be applied for shear stress. In particular, in our case we have to verify:

$$\boxed{\tau_{a,\mathrm{eq}} \;=\; \frac{R_t\,\tau_a}{R_t-\tau_m} \;\leq\; S_f.} \tag{5.20}$$

Where $R_t = 0.8R_m = 1192$ [MPa].

5.5.3 Fatigue results and solution

By applying the calculated coefficients:

$$S_f = 311 \ \ \text{[MPa]}.$$

With the previously established $\tau_{\max}$ at the critical section, the local (notch-root) alternating and mean shear stresses are obtained as

$$\tau_a \;=\; \frac{k_f\,(\tau_{\max}-\tau_{\min})}{2}, \qquad \tau_m \;=\; \frac{k_f\,(\tau_{\max}+\tau_{\min})}{2}.$$

For a pulsating cycle ($R = \tau_{\min}/\tau_{\max} = 0$) one has $\tau_{\min} = 0$, hence

$$\tau_a = \tau_m = \tfrac{1}{2}\,k_f\,\tau_{\max} = 239 \ \ \text{[MPa]}$$

.

Our

$$\tau_{a,\mathrm{eq}} = 299 \text{ [MPa]}.$$

It follows that

$$\boxed{\tau_{a,\mathrm{eq}} = 299 \text{ [MPa]} < S_{e\tau} = 311 \text{ [MPa]}}$$

Thus, with a 26 mm bar in 51CrV4, both requirements are met simultaneously: the full ±100 mm travel is preserved *and* the infinite-life condition is satisfied, since

$\tau_{a,\mathrm{eq}} \approx 301\,\mathrm{MPa} < S_f \approx 311\,\mathrm{MPa}$.
However, the resulting fatigue safety factor is only

$$SF = \frac{S_f}{\tau_{a,\mathrm{eq}}} \approx 1.03.$$

The resulting fatigue safety factor of about 1.03, although above the Goodman infinite-life line, is too small to be acceptable for a load-bearing component operating close to people; a value in the range 1.3–1.5 is desirable. Two design levers are available to raise it: reducing the usable travel (which lowers the peak twist, and hence the peak shear) or increasing the bar diameter. Neither can be pushed arbitrarily, because both interact with the prescribed $\pm 100\,\mathrm{mm}$ travel; the following two studies quantify the trade-off and motivate the final choice. Raising the material strength is not an option, since the endurance limit of steel saturates above $R_m \approx 1400\,\mathrm{MPa}$ and no further fatigue gain is obtained beyond that threshold.

Study 1 — maximum admissible travel for a target safety factor ($d = 26\,\mathrm{mm}$). The first study keeps the sized 26 mm bar and asks the inverse question: what is the largest stroke compatible with a prescribed fatigue safety factor? Imposing a target $SF = 1.3$ on the Goodman check, the admissible equivalent alternating stress is

$$\tau^*_{a,\mathrm{eq}} = \frac{S_f}{SF} = \frac{311}{1.3} = 239.2 \text{ [MPa]}.$$

Inverting the equivalent-amplitude relation (5.20) for the pulsating cycle ($\tau_a = \tau_m = k_f\,\tau_{\max}/2$) gives the corresponding peak shear at the notch,

$$\tau^*_{a,\mathrm{eq}} = \frac{R_t\,\tau_a}{R_t - \tau_m} \quad\Longrightarrow\quad \tau_{\max,\mathrm{cap}} = 346.5 \text{ [MPa]},$$

to be compared with the unclipped $\tau_{\max} = 416\,\mathrm{MPa}$ reached at full stroke. The associated torque follows from (5.4) for $d = 26\,\mathrm{mm}$,

$$T = \frac{2\,J\,\tau_{\max,\mathrm{cap}}}{d} = 1196 \text{ [Nm]},$$

the twist from the torsion relation (5.2) with $L = 1.16\,\mathrm{m}$,

$$\theta = \frac{TL}{GJ} = 0.387 \text{ [rad]} \approx 22^\circ,$$

and the corresponding vertical stroke from the geometric link (5.1) with $B = 450\,\mathrm{mm}$,

$$h = B\sin\theta \approx 168 \text{ [mm]}.$$

This is the total admissible stroke of the bar from its stress-free (pre-index) position. Keeping the stress within the $SF = 1.3$ bound therefore requires limiting the travel itself. Centering this admissible stroke on the nominal (parallel-to-ground) position yields a symmetric travel of

$$\pm\tfrac{1}{2}\,h = \pm 84\ [\mathrm{mm}].$$

In other words, with the 26 mm bar, the full ±100 mm target is not compatible with a fatigue safety factor of 1.3: guaranteeing that margin over the whole stroke costs about 15 mm of travel per side.

Study 2 — increasing the bar diameter to 27 mm**.** The second study takes the opposite lever and increases the bar diameter to $d = 27\,\mathrm{mm}$. By changing the diameter, the inertia changed into $J = \pi d^4/32 = 5.22 \times 10^{-8}\,\mathrm{m}^4$ and the equilibrium twist at maximum load becomes

$$\theta_0 = \frac{T_0 L}{GJ} = 0.199\ [\mathrm{rad}] = 11.42^\circ,$$

so that the vertical settling under the fully loaded robot is

$$h_0 = B \sin\theta_0 = 89.1\ [\mathrm{mm}].$$

The stiffer bar therefore no longer settles by the full 100 mm at maximum load, but by about 90 mm. Evaluating the worst-case shear at the top position ($2\theta_0$),

$$\tau_{\max} = 371\ [\mathrm{MPa}],$$

and, for the pulsating cycle, the notch-root alternating and mean components $\tau_a = \tau_m = k_f\,\tau_{\max}/2 = 213\,\mathrm{MPa}$ give the equivalent fully reversed amplitude

$$\tau_{a,\mathrm{eq}} = \frac{R_t\,\tau_a}{R_t - \tau_m} = 260\ [\mathrm{MPa}],$$

whence the fatigue safety factor

$$SF = \frac{S_f}{\tau_{a,\mathrm{eq}}} = \frac{311}{260} = 1.20.$$

By symmetry about the nominal position, this configuration corresponds to a bar built for a travel of about ±90 mm rather than ±100 mm, with a fatigue safety factor of 1.20. An end stroke is required at the top position of the bar.

Design choice. The two studies expose the same underlying trade-off from opposite directions: because the bar is rigidly coupled to the fork, the peak stress is set by the travel, and margin can only be bought back by giving up stroke — either by clipping the travel at fixed diameter (Study 1) or by stiffening the bar, which reduces the load-induced settling (Study 2). Neither lever reaches the full ± 100 mm at an ample margin: the choice is ultimately between prioritizing vertical travel or fatigue safety factor.
The 27 mm diameter of Study 2 was retained, since it offers a still generous ± 90 mm of travel, adequate to absorb the terrain irregularities of the inter-row soil, together with a fatigue safety factor of 1.20. A further, less obvious advantage of this choice is that under the fully loaded robot the bar settles by $h_0 = 89$ mm, essentially half of its admissible stroke, so that the equilibrium position is symmetric by construction. The 26 mm bar does not share this property: its admissible travel is ± 85 mm, yet the payload alone twists it by 103 mm, beyond the compression limit, and using it would require re-indexing the bar against its natural equilibrium and adding a mechanical end stop on the compression side to keep the stroke symmetric.

5.6 Chassis integration and mounting scheme

5.6.1 Arrangement

A first packaging option considered was a **fore–aft arrangement**, with the two bars placed one behind the other across the rear module. This layout would inevitably give the left and right arms different effective lever lengths, producing unequal torques for the same wheel travel. The consequence would be non-uniform wheel rates unless two different bars were designed and stocked, and the linkage geometry needed to route torque from each wheel to its respective bar would become unnecessarily complex. In short, the fore–aft option trades symmetry and commonality for added parts, alignment effort, and mismatched suspension behavior.

An alternative is a vertical **"under/over" arrangement**, stacking the two transverse bars within the rear envelope. This configuration preserves identical lever arms and a single bar specification on both sides, optimizes the production process (same specifications for both sides), simplifies the arm interfaces (same splined hubs and indexing), and keeps the torque path short and rigid. The critical point, however, is ground clearance: the lower bar and its arm sweep must not infringe the minimum 0.30 m clearance set in Chapter 3. The integration therefore centers on placing the bars through bushings, using splined or keyed anchor blocks for angular indexing, providing axial retention shoulders. This solution maintains commonality of parts and symmetric kinematics while respecting the chassis hard points and the clearance requirement.

5.6.2 Spline design

After fixing the transverse installation, the next design choice is how to transmit torque into the bar, so how to couple the bar to the trailing arm, and how to restrain the opposite end against rotation while providing axial location. The solution must be stiff, precise and repeatable in assembly.

A splined end was selected for the arm–bar interface so that torque is carried without clamp-slip and the arm angle can be indexed discretely at assembly. Among available spline families, a straight-sided serration per DIN 5463 [19] (closely aligned with the ISO 14 family of straight-sided splines) offers a good balance of manufacturability and strength: teeth are prismatic and can be produced economically by broaching; backlash is small and the fit is robust in dirt. As an alternative, the involute spline per DIN 5480 uses a gear-like involute tooth form rather than the rectangular profile of DIN 5463, providing superior load distribution and centering and supporting higher torques with tighter tolerance control—at the expense of more complex tooling.

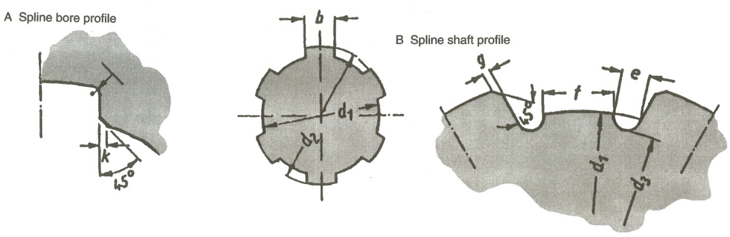


Figure 5.9: DIN 5463 profile

In practice, DIN 5463 is attractive when simplicity and cost dominate and a concentric pilot or shoulder can be provided for centering; DIN 5480 is preferred when maximum torque capacity, repeatable centering, and interchangeability across suppliers are primary drivers.

For ease of 3D modeling and meshing in the FEM environment, the arm–bar interface was represented as a straight-sided spline according to **DIN 5463**.

Nevertheless, the preferred choice remains an involute spline per DIN 5480, which offers more uniform tooth load distribution (greater strength), improved centering, and reduced backlash compared with straight-sided serrations.

The original intent was to seat both the trailing-arm hub and the opposite-end reaction block against positive shoulders machined on the torsion bar, so as to provide inherent axial location. On the reaction side, a reduced-diameter straight-sided spline (DIN 5463) was envisaged as a pilot serration, complemented by a circlip seated in a groove to complete the axial retention.

This configuration, however, proved incompatible with the fatigue requirement. A positive shoulder introduces an abrupt change of section, and the resulting fillet must be kept small to seat the mating part; the circlip groove adds a second sharp notch. Both are severe stress raisers. The FEM analysis of this arrangement confirmed it: high stress concentrations developed at the shoulder–spline transition and at the circlip groove, with local peaks well above the allowable limit, indicating a high likelihood of early fatigue failure (Figure 5.10).

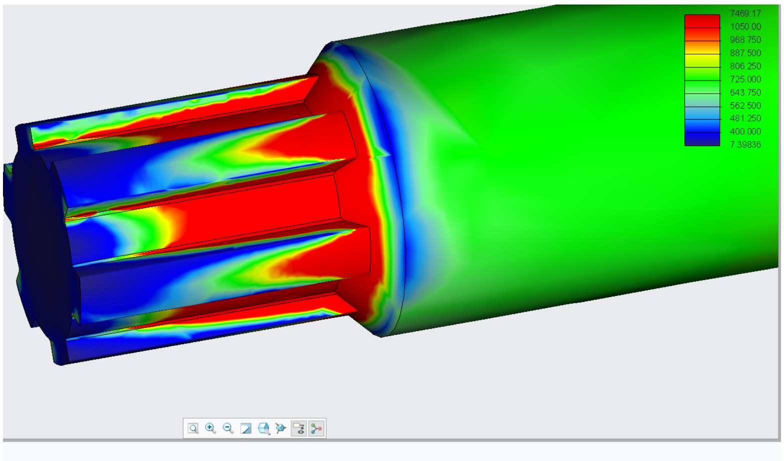


Figure 5.10: FEM result of the configuration with positive shoulders on the torsion bar: stress concentrations at the section changes.

Crucially, this outcome is inconsistent with the value assumed in the fatigue assessment. The whole infinite-life verification was carried out with a fatigue stress-concentration factor $K_f = 1.15$, i.e. a deliberately low value that can only be attained with smooth, gradual diameter transitions and generous fillet radii. A positive shoulder, with its abrupt step and necessarily small fillet, yields a far higher stress-concentration factor and would make $K_f = 1.15$ unrealistic, invalidating the fatigue margin established earlier. The shoulder-based retention was therefore rejected: the geometry required to keep K_f low at the bar ends is simply not compatible with seating a part against a positive shoulder.

A second arrangement was adopted instead, which removes the shoulders altogether and relies on a single straight-sided spline of larger diameter, machined with smooth transitions, and on two circlip grooves for the axial retention of the bar. With no abrupt shoulder, the diameter changes can be blended with comparatively large fillet radii, keeping the local stress-concentration factor close to the assumed

$K_f = 1.15$ and thus preserving the fatigue verification.

Under the selected DIN 5463 specification, the outside (major) diameter is set to $D = 32\,\text{mm}$. For the chosen size series, the standard further specifies six straight-sided teeth ($z = 6$), a nominal tooth width $b = 6\,\text{mm}$, a root fillet radius $r = 0.3\,\text{mm}$, a chamfer $k = 0.4\,\text{mm}$ and an axial tooth length $l = 30\,\text{mm}$. Both spline ends were designed with circlip grooves, so that snap rings provide positive axial retention of the bar at both the trailing-arm hub and the reaction block. Adhering to the standard geometry and fit class, the resulting arrangement is shown in Figure 5.11.

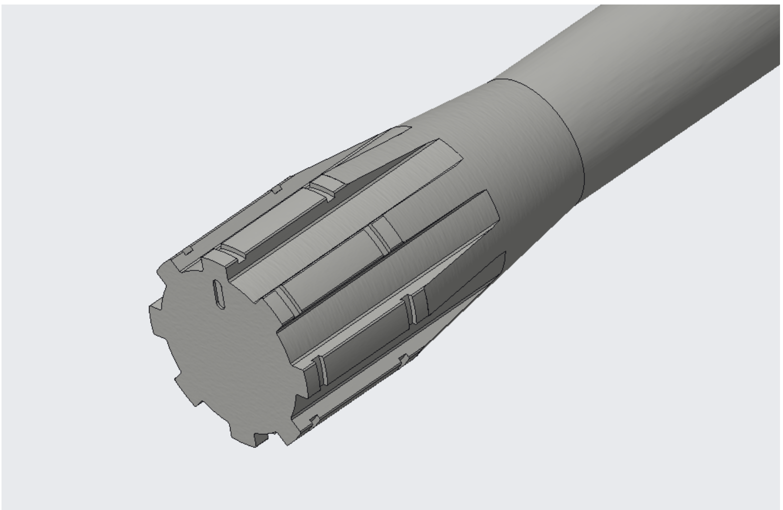


Figure 5.11: Final configuration of DIN 5463 on the torsional bar.

5.6.3 Implementation on the chassis

In Figure 5.11 the groove is visible on the spline, providing positive axial retention in both directions and thus fully blocking the axial motion of the bar. Between the shaft diameter and the $D = 32\,\text{mm}$ spline major diameter a generous fillet is introduced to lower the elastic stress concentration and improve fatigue life, consistently with the value $K_f = 1.15$ assumed in paragraph 5.6.2.

To guarantee correct angular installation, an orientation notch is machined on both end faces of the bar. During assembly the notch is used to clock the bar to the prescribed downward pre-index angle θ, which defines the zero reference of the torsion-bar stroke (coincident with the fork droop position at $-100\,\text{mm}$). The trailing-arm hub and the opposite reaction block carry matching orientation features, so that the bar can be installed only with all notches aligned. This simple, robust arrangement prevents misassembly, ensures a repeatable angular datum on both sides, and allows quick visual verification of the correct indexing during maintenance.

Torque is transmitted from the bar to the fork by a short link rod (dog-bone) pinned between the torsion-bar arm and a clevis on the fork (Figure 5.12). The arms and the reaction blocks share the same spline specification, which provides positive torque transmission together with discrete angular indexing; by contrast, the link rods and the reaction blocks are the only components that must be tailored in geometry and dimensions to the left/right packaging.

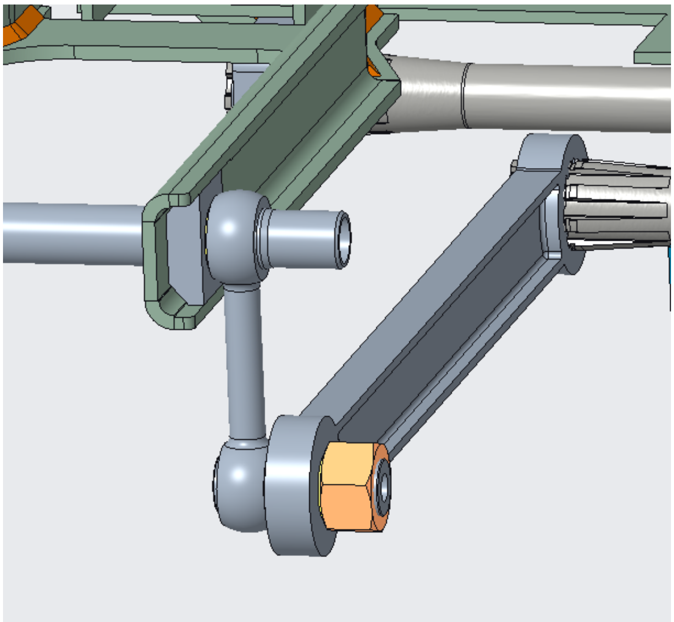


Figure 5.12: Link rod pinned between the torsion-bar arm and a clevis on the fork.

The torsion bars are mounted so that the torque reaction closes through the main frame rails, i.e. into the stiffest portions of the chassis. A pair of base plates, profiled to match the rail spacing, was modelled and is to be fillet-welded to the rails; these plates provide a stiff, permanent interface and include precision locating features (pilot bores) so that the reaction blocks can be positioned uniquely and repeatably. Mounting holes are also provided for the torsion-bar supports.

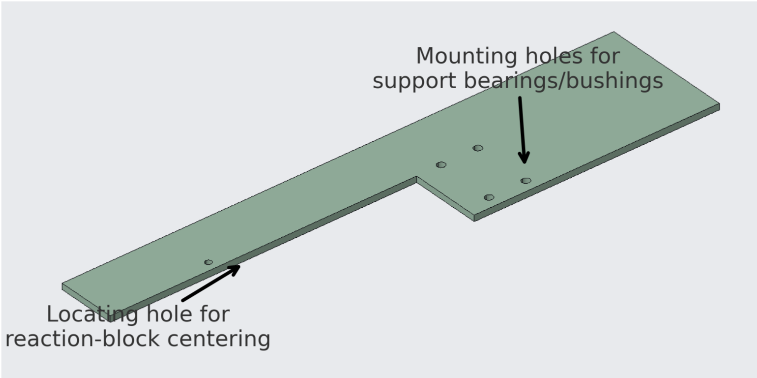


Figure 5.13: Base plates welded to the frame rails.

To avoid parasitic side loads on the bar and on its supports, the two reaction blocks are aligned so that the bar axis lies in the *same vertical plane* as the fork rotation axis, with a small intentional vertical offset below the pivot height: 60 mm for the upper reaction block and 105.5 mm for the lower one (Figure 5.14). This alignment keeps the lever line nearly straight, simplifies the linkage geometry, and reduces bending of the bar caused by misalignment. Once welded, the base plates provide a stable datum; shims at the block–plate interface allow fine adjustment of parallelism and height.

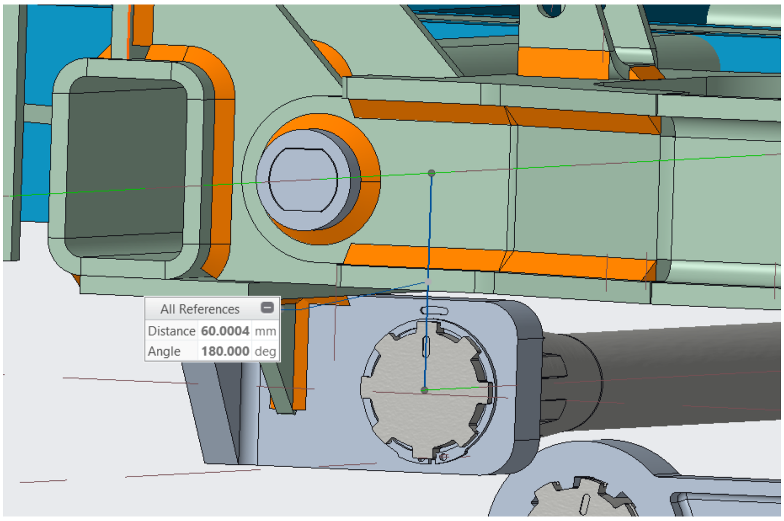


Figure 5.14: Vertical distance and alignment between the pivot height and the upper reaction block.

Because of the stepped geometry of the torsion bar—with larger end diameters at the spline/hub interfaces—the support envelope does not accommodate standard rolling bearings of adequate capacity. Rolling bearings were therefore excluded and bushings were selected instead. Bushings provide a compact housing, tolerate contamination and slight misalignment, and are significantly more economical; for the slow, small-angle oscillatory motion required here they offer more than sufficient service life (Figure 5.15).

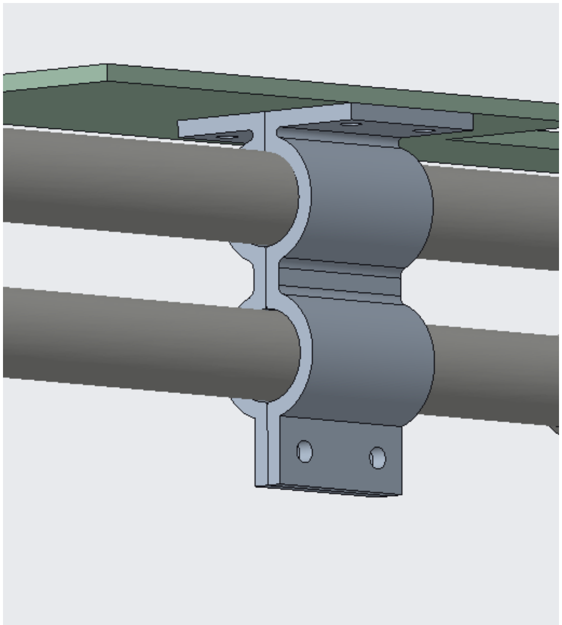


Figure 5.15: Bushing design.

A delicate aspect of this arrangement is the design of the reaction blocks, which must react the maximum bar torque while working essentially as cantilevers: the torque is transferred at the base of the vertical block into the base plate, which is loaded by the resulting overturning moment. Given the available space and the possible fixing methods—bolting or welding—several solutions were attempted, converging on a welded attachment complemented by a single bolt for accurate positioning of the block during assembly.

Successive iterations of the block geometry were carried out with the aim of maximizing the available weld length, so as to distribute the stress over a larger seam rather than concentrating it: the geometry was progressively reshaped. The optimized block geometry is shown in Figure 5.16.

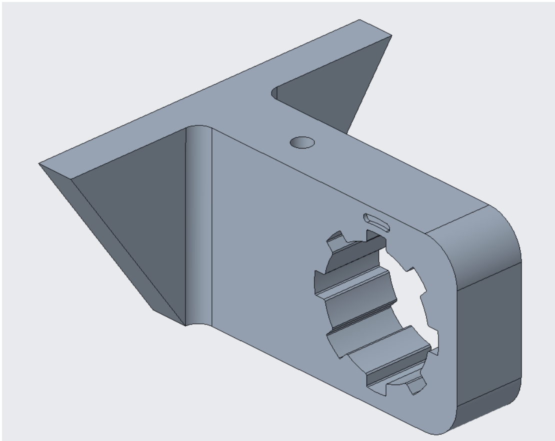


Figure 5.16: Reaction block: the T-shaped design increases the available weld length between the block and the base plate.

All parts in the splash zone are provided with mud guards and corrosion protection, and drain/inspection holes are added to the base plates. The resulting complete assembly is shown in Figure 5.17.

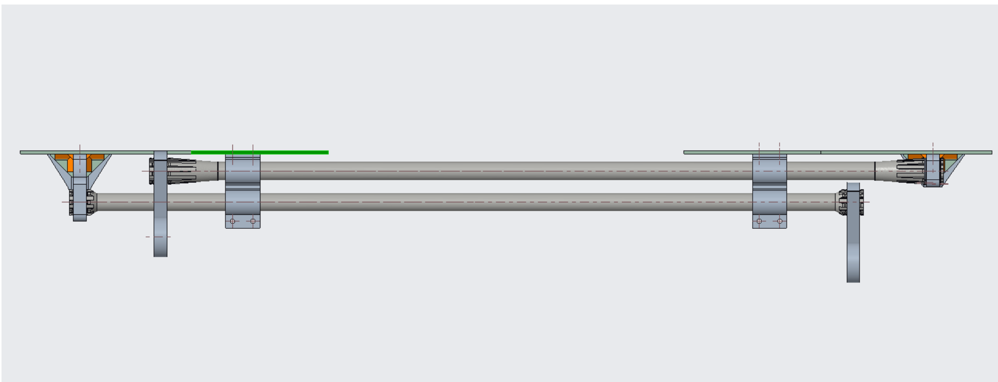


Figure 5.17: Final assembly.

5.7 FEM model

For the numerical verification; after giving the material properties to the torsional bar; boundary conditions were defined to reproduce the service load path through the splined interfaces. The driving torque from the trailing arm was applied to the torsion bar with the correct sense of rotation and distributed over the engaged spline flanks, acting only on the load-carrying faces rather than uniformly around the circumference. A dedicated local mesh refinement was introduced on the tooth flanks, at the tooth root, and at the shoulder fillet to capture the stress gradients with fidelity.

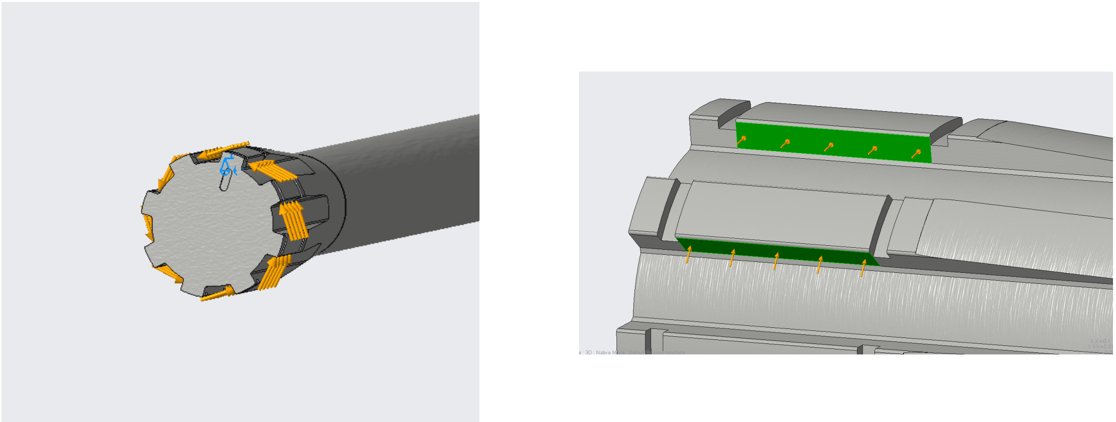


Figure 5.18: Torque application only on the interested faces.

At the opposite end, the bar was inserted into the reaction block, and the boundary conditions were applied to the base plate and to the interface between the base plate and the reaction block, rather than directly on the bar. This setup was chosen to verify not only the torsional behavior of the bar, but also the contact between the spline of the bar and the reaction block, allowing inspection of the internal stress state—both in the spline teeth and in the bar—via section views across the interface. Figure 5.19 shows the spline engagement between the torsion bar and the reaction block. The purple arrows, visible on the base plate, indicate the boundary conditions applied to restrain the assembly and react the bar's rotation.

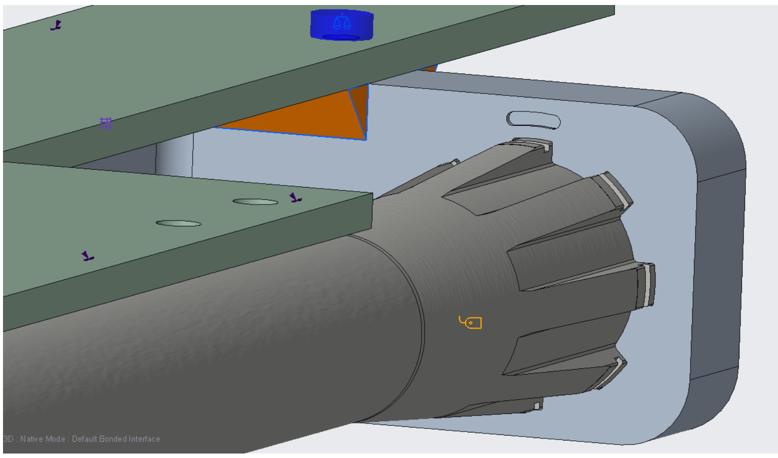


Figure 5.19: Bar constrained by the reaction block to simulate contact conditions.

The FEM results are consistent with the analytical calculations presented earlier. In the static sizing, the outer-fibre shear was $\tau_{\max} = 371\,\mathrm{MPa}$. Since the solver reports Von Mises stress, for pure shear one expects

$$\sigma_{\mathrm{VM}} = \sqrt{3}\,\tau_{\max} \;\approx\; \sqrt{3} \times 371\,\mathrm{MPa} \;\approx\; 643\,\mathrm{MPa},$$

which matches the FEM field showed in Figure5.20. Moreover, the very gradual transition (large shoulder fillet) between the spline's major diameter and the bar shank markedly reduces the local notch-driven stresses, bringing the peak values at the shoulder close to those in the uniform portion of the bar.

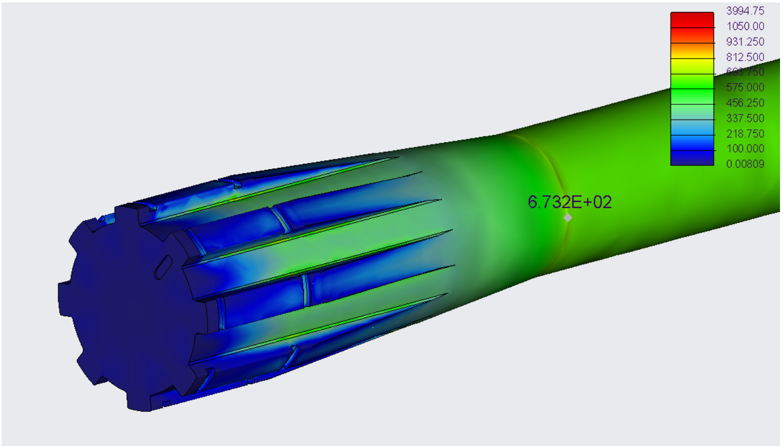


Figure 5.20: FEM result:free-side.

Fig. 5.21 is a sectional view of the bar and report the radial stress distribution.

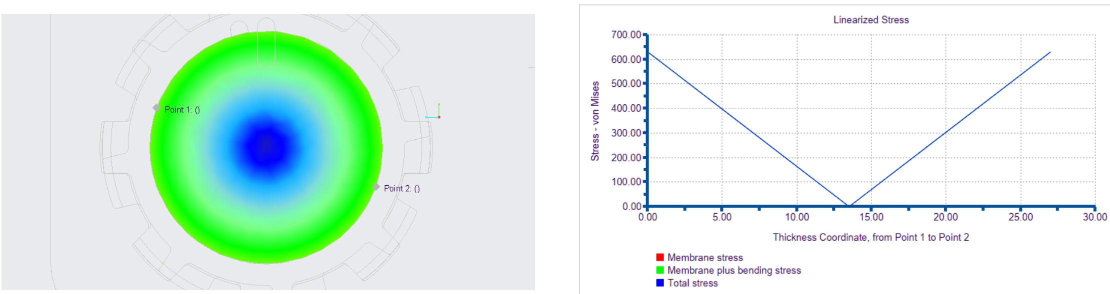


Figure 5.21: FEM stress result in section and linearized behaviour.

In the contact analysis of the engaged splines, very high local stresses appear at geometric discontinuities (sharp edges, tooth corners). These are classic linear–elastic contact singularities: an ideal sharp corner in contact can produce a theoretical stress field that increases without bound as the evaluation point approaches the edge. In reality, small fillets/chamfers, surface roughness, and local yielding limit the stress; the FE model, however, reports large peaks at the idealized corner.

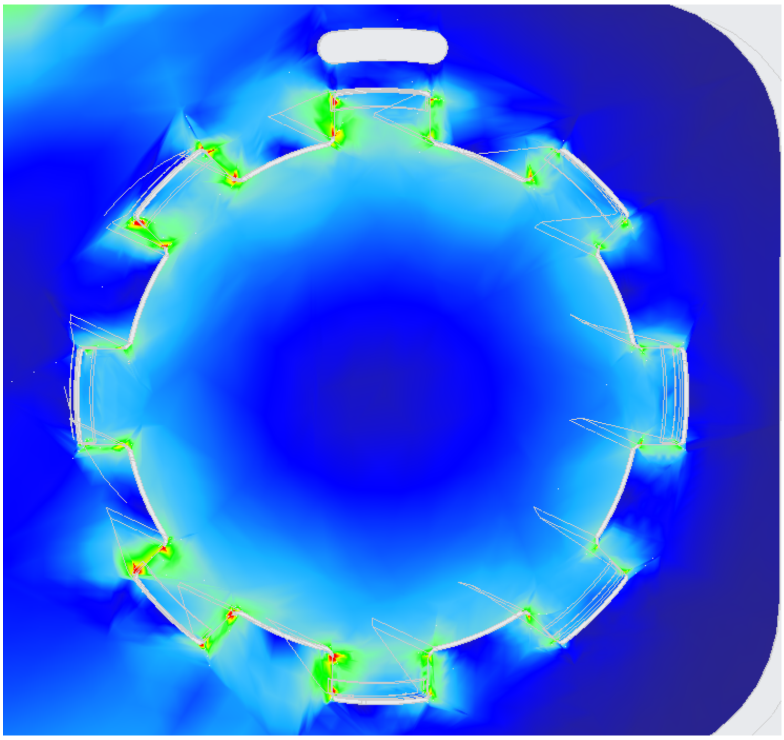


Figure 5.22: FEM result: reaction block-side.

By cutting the model on a slightly offset inspection plane—i.e., moving a small distance away from the edge/corner where the peaks appear—the stress field quickly returns below the allowable limit. This confirms that the extreme values are confined to idealized singularities (sharp edges, contact corners) rather than reflecting the stresses in the load-bearing region. Consequently, design checks should rely on converged, area-averaged stresses away from the singular point and on geometry with the actual fillets/chamfers included.

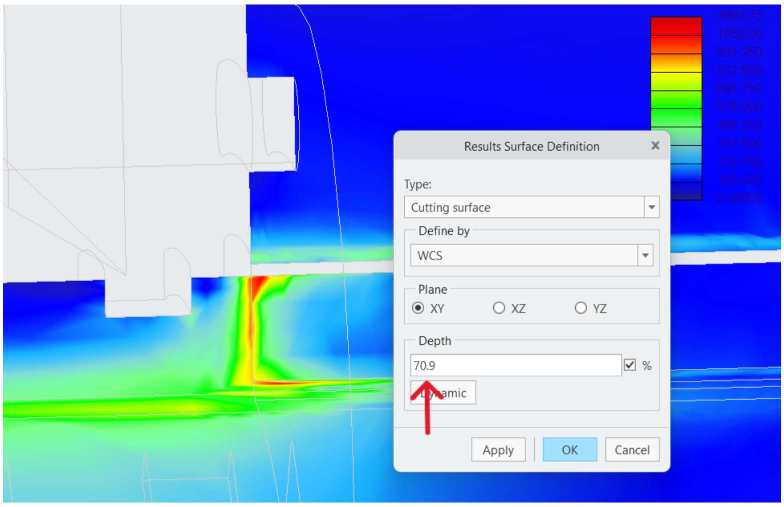


Figure 5.23: Local stress of the spline.

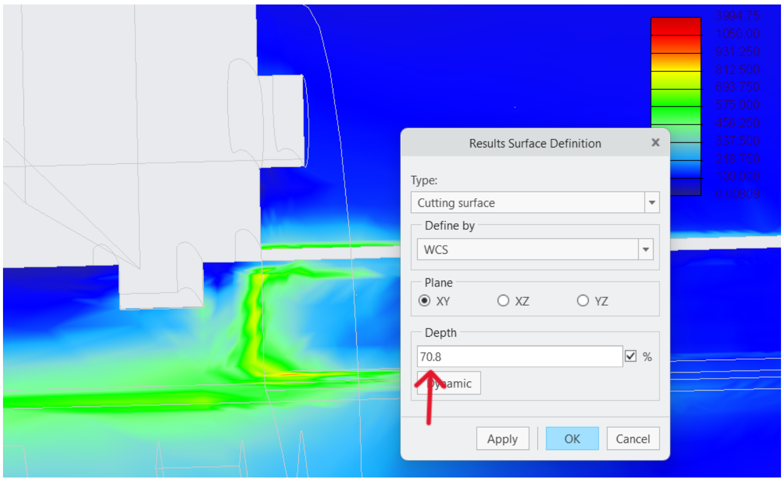


Figure 5.24: Local stress of the spline.

5.8 Summary and remarks

This chapter developed the rear suspension for the existing vineyard-robot chassis, starting from the geometric constraints imposed by the legacy platform and arriving at a verified design. Within the available envelope—an effective bar length fixed at $L_{\text{max}} = 1.16\,\text{m}$ and a lever arm $B = 0.45\,\text{m}$—a transverse torsion-bar layout was selected, with one bar per rear wheel. A quenched-and-tempered 51CrV4 spring steel, tempered at 450 °C, was adopted; the tempering temperature was chosen as a compromise between the elastic limit and notch sensitivity.

The bar was sized starting from these constraints, together with the pre-index strategy introduced to obtain a symmetric compliance about the nominal (parallel-to-ground) position. A first solution with a diameter $d = 26$ mm met the prescribed ± 100 mm vertical travel and satisfied the infinite-life condition, but only with a fatigue safety factor of about 1.03—above the Goodman line, yet too low for a load-bearing component operating close to people. Two options were therefore examined: keeping the 26 mm bar and clipping the admissible travel,or increasing the bar diameter and accepting a slightly reduced load-induced stroke.

The second option was retained, adopting a diameter $d = 27$ mm. This configuration yields a fatigue safety factor of 1.20 and a static safety factor of 2.18, with a still generous ± 90 mm of vertical travel—adequate to absorb the terrain irregularities of the inter-row soil. Beyond the larger travel, the 27 mm bar has a further advantage: under the fully loaded robot it settles by 89 mm, essentially half of its admissible stroke, so that the equilibrium position is symmetric by construction. The 26 mm bar does not share this property—the payload alone would twist it beyond its compression limit—so its use would require re-indexing the bar against its natural equilibrium and adding a mechanical end stop on the compression side to keep the stroke symmetric.

The analytical results were cross-checked by FEM, which confirmed the stress levels predicted by the hand calculations.

The integration with the chassis was then defined: bar retention and angular indexing through a DIN 5463 spline, torque transmission via a link rod to the fork, and reaction blocks welded to base plates on the main frame rails. The reaction blocks were designed as a compromise between bolting and welding: given the tight spaces and the geometry of the existing chassis, a bolted interface proved impractical, and a welded connection was adopted instead. The T-shaped design of the block was conceived precisely to increase the available weld length and thus distribute the stress over a larger joint. A few remarks should finally be made on the dynamic behavior and on the overall viability of the solution.

The design deliberately relies on the torsion bar as the sole elastic element, with no dedicated damping. At the low operating speeds of the robot this may be acceptable: the residual damping provided by the tires,the compliant soil, and the material hysteresis could be sufficient to attenuate most of the terrain inputs. A dedicated study of the system's response to the terrain-induced accelerations at the operating speeds should nonetheless be carried out, to rule out persistent oscillations or resonance that would compromise both the operation of the platform and the safety of the grape crates.

A second point concerns the reaction blocks, which work essentially as cantilevers, the bar torque being reacted at the base of the vertical block through its weld to the base plate: the strength of this welded joint under cyclic loading should be verified as part of the detailed design, since welded joints exhibit a lower fatigue strength than the base material. Both aspects are taken up among the future developments in Chapter 8.

More generally, this study shows that adopting a torsion bar as the rear suspension of the robot is technically feasible: the selected layout meets the travel and infinite-life requirements within the constraints of the existing chassis.

With the mechanical platform now defined, the work moves up the design pyramid to the planning-and-control layer. The next chapter addresses the autonomous steering of the robot, developing the trajectory-tracking controller and the inner steering loop that allow the platform to follow the rows.

Chapter 6

Autonomous Steering: Architecture and Control Foundations

6.1 Autonomous steering and its control requirements

As established in Chapter 2, the use of an autonomous collaborative platform in vineyard operations is motivated by the structural pressures of the sector and by the repetitive, structured nature of in-row tasks.
Building on that motivation, this section does not revisit *why* autonomy is desirable, but rather examines *what autonomous steering specifically requires* of the control system, and introduces the components that the remainder of this chapter develops.

Autonomous steering is, at its core, a closed-loop regulation problem: the vehicle must measure its own state, compare it against a desired trajectory, and continuously adjust its steering input so as to drive the tracking error toward zero. Translating this principle into a vineyard transport platform imposes a set of concrete requirements on the control architecture.

The first is *tracking repeatability.* The vehicle must hold a prescribed lateral offset and heading along the row with centimetric consistency, so as to avoid inadvertent contact with vines and trellis hardware.

The second is *disturbance rejection.* On the uneven, partially structured soil of a vineyard, the vehicle is subject to small heading perturbations induced by ruts, soil camber, and local slope. The steering controller must attenuate these disturbances rather than propagate them, keeping the vehicle on its reference despite terrain-induced deviations.

The third concerns the *actuation limits*. The physical steering system cannot move instantaneously nor beyond a finite angular range; the control commands must therefore respect curvature and rate constraints. At the collaborative operating speeds considered in this work (approximately 1–4 m/s), this also contributes to safety, since steering and speed commands can be shaped to remain within limits that are safe for human workers operating nearby.

Finally, the steering behavior is coupled to *payload protection*. The lateral acceleration experienced by the deck is governed by the time evolution of the steering angle: an abrupt change in steering angle produces an abrupt change in path curvature and hence a sharp lateral acceleration. By regulating the steering angle with a well-damped response—free of overshoot and oscillation, as enforced by the inner-loop design—the controller yields a smooth steering profile and thus limits the lateral accelerations transmitted to the fragile grape crates. The control system and the passive suspension sized in the previous chapters therefore act as two layers addressing the same objective—protecting the load—at different points in the signal chain: the suspension attenuates vertical disturbances from the terrain, while the steering controller shapes the lateral dynamics induced by path tracking.

These requirements are met through a feedback architecture organized in two nested loops.
This thesis adopts a **two-loop structure**: an outer trajectory-tracking loop, which generates a steering reference from the geometric tracking errors, and an inner PID steering servo, which drives the physical steering angle to follow that reference.

The remainder of this chapter provides the foundational background on the components of this architecture and serves as an introduction to Chapter 7, where the actual design and tuning are carried out. Specifically, the following sections outline:
(i) the trajectory controller (outer loop) that produces a steering reference from lateral and heading errors;
(ii) the PID steering servo (inner loop) that drives the physical steering angle to track that reference;
(iii) the Ackermann steering kinematics adopted for this robot, which also determine the minimum turning radius and, with it, the end-of-row maneuvering constraints discussed at the close of the chapter.

6.2 Architecture overview and block diagram

The distinction between the two control loops introduced above mirrors a broader separation between two layers of the autonomous system. At the top, a **high-level autonomy** layer performs perception and mapping, fuses sensors, and plans or updates a reference path in response to the environment—rows, headlands, and obstacles—producing a nominal trajectory. This layer, essential for a fully autonomous robot, lies outside the scope of this thesis.

The present work focuses on the layer below: trajectory **tracking**. Given a precomputed path, the tracking layer continuously compares the robot's measured state against the reference and produces the steering commands that keep the vehicle on track. Figure 6.1 shows the signal flow adopted throughout this work and the cascade of components that make up the low-level system.

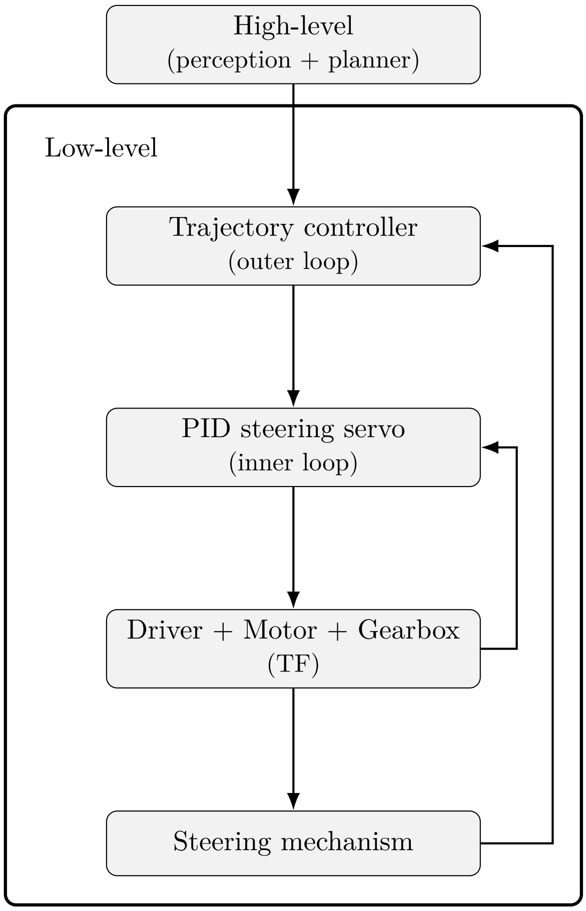


Figure 6.1: Two-layer architecture of the autonomous steering system. Block diagram of the low-level:trajectory-tracking detailed.

Following the signal from the reference path to the wheels, the **trajectory controller** (outer loop) computes a steering reference from the lateral and heading errors; the **PID steering servo** (inner loop) drives the steering actuator so that the physical steering angle follows that reference, with a filtered derivative term and anti-windup; the **driver, motor, and gearbox** convert the control command into actuator motion, subject to current and voltage limits; and the **Ackermann mechanism** maps the actuator motion into the individual wheel steering angles. The remainder of this chapter examines each of these components in turn.

6.3 Trajectory tracking and the Pure Pursuit controller

6.3.1 The trajectory-tracking problem

Trajectory tracking is the outer-loop control task that drives the vehicle to follow a planned path. The path is expressed as a curve $\mathcal{P}(s) = [x_d(s),\, y_d(s),\, \psi_d(s)]$, parameterized by the arc length s, where x_d, y_d define the desired position and ψ_d the desired heading at each station along the path; a reference speed $v_d(s,t)$ may optionally be associated with it. As established, the path itself is assumed to be provided by the high-level planning layer; the tracking controller treats it as a given input and is responsible only for following it.

At each control cycle, the controller compares the vehicle's current pose with the path. The current position is first *projected* onto the path $\mathcal{P}$, i.e. the closest point on the path is identified, located at the station $s^\star$. Relative to that point, two geometric errors describe how badly the vehicle is tracking the path:

- the **lateral error** (or cross-track error) e_y: how far the vehicle is displaced sideways from the path;
- the **heading error** $e_\psi = \psi - \psi_d(s^\star)$: how much the vehicle is pointing away from the local path direction.

Intuitively, e_y tells the vehicle *how far off* the path it is, while e_ψ tells it *how badly it is aimed.* Reducing both is what allows the vehicle to rejoin the path smoothly: a controller that corrected only the lateral displacement, ignoring how the vehicle is oriented, would tend to overshoot the path and oscillate around it.

The goal of the trajectory controller is therefore to produce, at each instant, a steering reference δ_{ref} that drives both errors toward zero.

6.3.2 The Pure Pursuit controller

Among the trajectory-tracking methods reviewed in Chapter 2, this work adopts **Pure Pursuit**, a geometric tracking law well suited to the low-speed, structured-row operation considered here.

The path is provided as an ordered sequence of 2D waypoints.
The core idea of Pure Pursuit is intuitive: at each control step, the controller looks ahead along the path and picks a single *goal point* G lying on the reference trajectory at a fixed distance l_d from the vehicle—the *look-ahead distance*. The controller then steers the vehicle along the unique circular arc that starts at the vehicle's current position, is tangent to its current heading, and passes through G (Fig. 6.2).

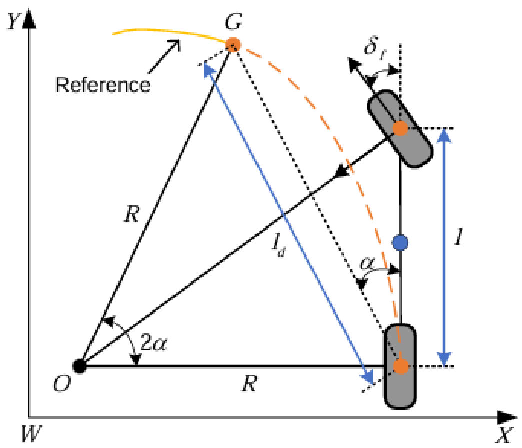


Figure 6.2: Pure Pursuit geometry. Source: [9]

This circular arc defines the turning radius R that the vehicle must follow to reach the goal point. The geometry is governed by a single angle, α, the *heading error*: the angle between the vehicle's current heading and the line-of-sight to the goal point G. From the geometry of the arc, the turning radius is

$$R = \frac{l_d}{2 \sin \alpha}. \tag{6.1}$$

The radius is then converted into the actual command for the steering mechanism. Using the bicycle (single-track) relation between turning radius and steering angle, which will be derived in Section 6.5, the required front-wheel steering angle δ_f follows as

$$\delta_f = \arctan\left(\frac{2\,l\,\sin\alpha}{l_d}\right),$$

where l is the vehicle wheelbase. These two relations are the heart of the method: the heading error α and the look-ahead distance l_d together determine, at each instant, the curvature of the arc and hence the steering angle that the inner loop must realize.

6.3.3 Role of the look-ahead distance

The look-ahead distance l_d is the single, central design parameter of Pure Pursuit, and the steering law above shows that it directly shapes the commanded arc. Its choice reflects a trade-off between responsiveness and smoothness.

A *short* l_d makes the vehicle aim at a nearby point, producing tight arcs and a highly responsive behavior that corrects tracking errors quickly; however, on high-curvature segments this can induce oscillations about the path. A *long* l_d, conversely, makes the vehicle aim further ahead, yielding smoother trajectories at the cost of reduced accuracy, since the vehicle tends to cut corners and converge to the path more slowly—a particular concern when negotiating sharp turns or operating in the confined space of a vineyard row. The look-ahead distance is therefore tuned to balance accuracy and smoothness for the low-speed operation considered in this work. It can also, in principle, be adjusted online as a function of speed or path curvature.

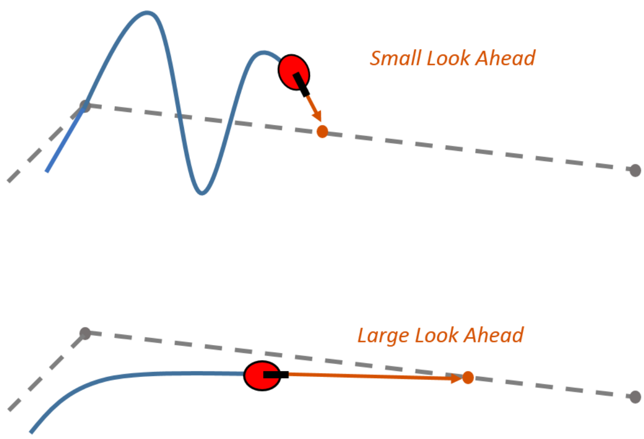


Figure 6.3: Effect of the look-ahead distance on path tracking: a short l_d causes oscillations about the path, while a longer l_d produces smoother but wider tracking. Source: [20].

6.4 The PID steering servo

While the trajectory controller decides *where* the wheels should point, it does not by itself guarantee that the steering actuator actually reaches that angle.
The steering reference δ_{ref} produced by Pure Pursuit must be physically realized by the steering hardware, whose dynamics—motor and gearbox inertia, friction, transmission compliance-make the actual wheel angle lag behind the command.

This is the role of the inner loop: a feedback controller that continuously drives the actual steering angle δ toward the reference δ_{ref}.

This two-loop arrangement is a cascade: the outer loop (Pure Pursuit) generates the steering reference, and the inner loop (the PID servo described here) tracks it. For the cascade to work properly, the inner loop must be significantly faster than the outer one, so that, from the viewpoint of the trajectory controller, the steering actuator can be regarded as reaching its commanded angle almost instantaneously. This separation of time scales is a standard requirement of cascade control and will guide the choice of the controller bandwidth in Chapter 7.

6.4.1 Feedback structure and the transfer function

The inner loop is organized as a classical feedback control system (Fig. 6.4). Working in the Laplace domain, every signal is represented by its transform: the reference $\delta_{\text{ref}}(s)$ is the desired steering angle; the error $E(s) = \delta_{\text{ref}}(s) - \delta(s)$ is the difference between the desired and the measured steering angle; the controller $G_c(s)$ processes this error and produces the command $U(s)$ sent to the actuator; the plant $G_p(s)$ models the actuator dynamics (motor and gearbox) and converts the command into the actual steering angle $\delta(s)$; finally, the sensor $H(s)$ measures δ and feeds it back to the summing junction, closing the loop.

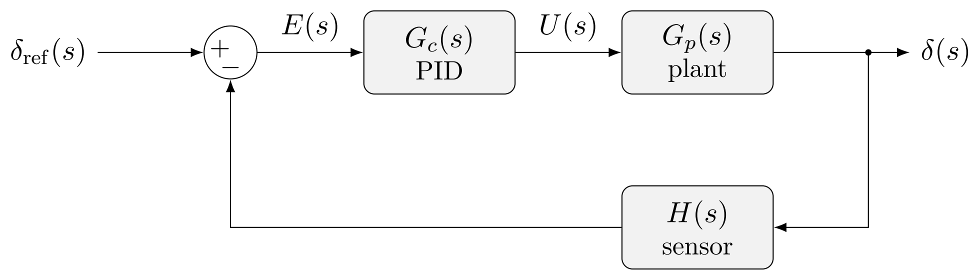


Figure 6.4: Feedback structure of the inner steering loop: the error between the reference steering angle δ_{ref} and the measured angle δ drives the PID controller $G_c(s)$, whose command actuates the plant $G_p(s)$; the sensor $H(s)$ closes the loop. Source: author's own elaboration.

A *transfer function* is the ratio, in the Laplace domain, between the transform of a system's output and that of its input, assuming zero initial conditions. It fully describes how a linear system responds to any input, and it is the basic tool used throughout this work to analyze the loop. The plant transfer function, for instance, is defined as $G_p(s) = \delta(s)/U(s)$: the relation between the command applied to the actuator and the resulting steering angle.

Combining the blocks, the *closed-loop* transfer function from reference to output is

$$\frac{\delta(s)}{\delta_{\text{ref}}(s)} = \frac{G_c(s)\,G_p(s)}{1 + G_c(s)\,G_p(s)\,H(s)},$$

which, under the common assumption of an ideal unity-gain sensor ($H(s) = 1$), reduces to

$$\frac{\delta(s)}{\delta_{\text{ref}}(s)} = \frac{G_c(s)\,G_p(s)}{1 + G_c(s)\,G_p(s)}.$$

The denominator of this expression, set to zero, gives the **characteristic equation** of the closed loop, whose roots—the closed-loop *poles*—determine the stability and the dynamic behavior of the system. This is the starting point of the stability analysis carried out in Chapter 7.

6.4.2 The step response and its performance indices

The behavior of the closed loop is conventionally characterized through its *step response*: the output obtained when the reference is suddenly changed (for instance from 0 to 1). This response is described by a set of standard performance indices—*rise time*, *overshoot*, *settling time*, and *steady-state error* e_∞—which quantify how quickly, how accurately, and how smoothly the actual steering angle reaches the commanded value.
The purpose of the controller is to shape this response so thatδ tracks δ_{ref} as fast and as smoothly as the application requires.

6.4.3 The three control actions

The PID controller acts on the error through three terms—proportional, integral, and derivative (Figure6.5).

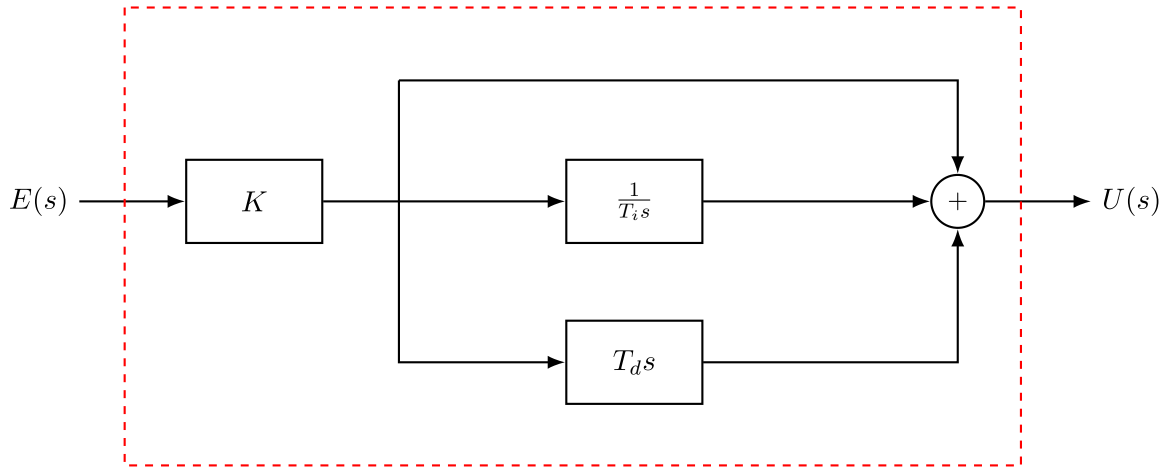


Figure 6.5: Internal structure of the PID controller: the error $E(s)$ is processed in parallel by the proportional (K), integral ($1/T_i s$), and derivative ($T_d s$) actions, whose sum forms the control command $U(s)$. Source: author's own elaboration.

Its transfer function can be written in two equivalent forms:

$$G_c(s) = K\left(1 + \frac{1}{T_i s} + T_d s\right) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s},$$

with $K_p = K$, $K_i = K_p/T_i$, and $K_d = K_p T_d$.
The two formulations are equivalent and related by $K_p = K$, $K_i = K_p/T_i$, and $K_d = K_p T_d$. The gains K_p, K_i, K_d express each action as a direct gain, while the equivalent parameters T_i and T_d—the integral time and the derivative time, both measured in seconds—express the integral and derivative actions in terms of characteristic times: a smaller T_i corresponds to a stronger integral action (since K_i and T_i are inversely proportional), while a larger T_d corresponds to a stronger derivative action. In this work the gain form (K_p, K_i, K_d) is adopted, as it maps directly onto the stability analysis of Chapter 7.
Each term has a distinct and complementary effect on the response.

Proportional action (K_p). The proportional term produces a command proportional to the instantaneous error. Increasing K_p speeds up the response and reduces the steady-state error, but it cannot eliminate it entirely, and excessive values reduce stability and increase overshoot.

Integral action (K_i). The integral term accumulates the error over time, and it is this accumulation that drives the steady-state error to zero and rejects constant disturbances. Its benefit comes at a cost: increasing K_i tends to reduce stability and increase overshoot. A common tuning strategy is to first set the transient response with K_p, then add integral action to remove the residual steady-state error.

Derivative action (K_d). The derivative term responds to the rate of change of the error, adding damping that reduces overshoot and can improve stability. However, because differentiating amplifies measurement noise, the derivative term is often problematic in practice; it also has no effect on the steady-state error.

6.5 The Ackermann steering mechanism and the minimum turning radius

The two steering references discussed so far—the curvature produced by Pure Pursuit and the angle tracked by the PID—must ultimately be realized by the physical steering mechanism of the vehicle.
This robot employs **Ackermann steering**, the arrangement adopted since the beginning of this work; this section describes its geometry, derives the relation between steering angle and turning radius, and uses it to determine the vehicle's minimum turning radius.

6.5.1 Geometric principle

Ackermann steering is the geometric condition that, during a steady turn with pure rolling and no lateral slip, the axes of all wheels intersect at a single point, the *instantaneous center of rotation* (ICR). Because the inner front wheel follows a tighter arc than the outer one, the two front wheels cannot share the same steering angle: the inner wheel must steer more. An ideal Ackermann linkage enforces this purely kinematic constraint, minimizing tire scrub at low speed (Fig. 6.6).

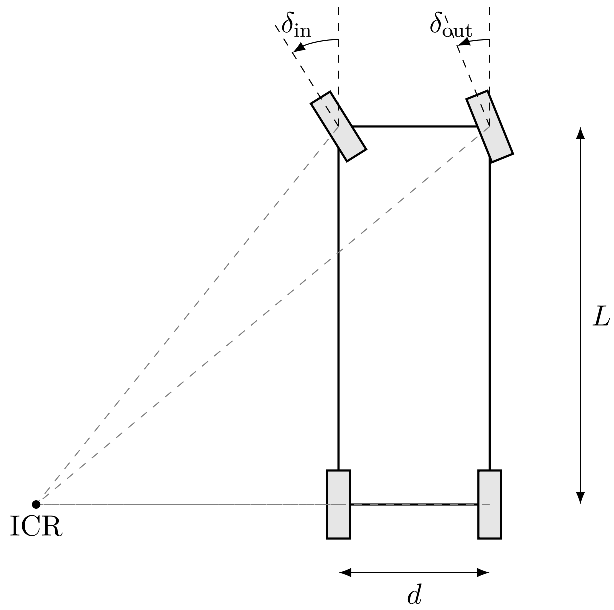


Figure 6.6: Ackermann steering geometry: in a turn, the inner front wheel steers by a larger angle (δ_{in}) than the outer one (δ_{out}) relative to the straight-ahead direction (vertical dashed lines), so that the perpendiculars to all four wheels meet at a common instantaneous center of rotation (ICR). Here L is the wheelbase and d the front track width. Source: author's own elaboration.

6.5.2 The bicycle (single-track) model

A convenient abstraction is the *bicycle*, or single-track, model, in which the two front wheels are replaced by a single virtual wheel at the front-axle midpoint, steered by an equivalent angle δ. This model rests on a set of assumptions that hold well in the present application:

- the vehicle moves at **low speed**, so that lateral accelerations are small;
- for the purpose of this geometric relation, the wheels are assumed to roll without lateral slip, so that each wheel moves along its own plane and the instantaneous center of rotation is well defined;
- the motion is **planar**: the vehicle is described in the horizontal x–y plane, without roll, pitch, or vertical motion.

Under these conditions the vehicle rotates about a single ICR (Figure6.7).

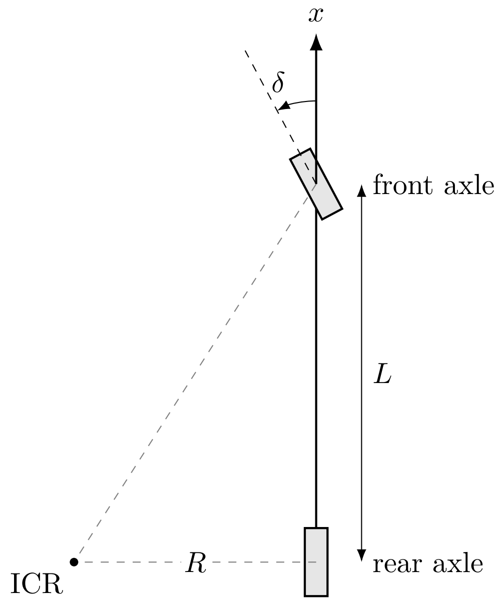


Figure 6.7: Single-track (bicycle) model: the two front wheels are replaced by a single virtual wheel steered by the angle δ.

Denoting by R the distance from the rear-axle midpoint to the ICR, the steering geometry gives

$$\tan\delta = \frac{L}{R}, \qquad \kappa = \frac{1}{R} = \frac{\tan\delta}{L},$$

where κ is the path curvature, and therefore

$$\boxed{\delta = \arctan(L\,\kappa) = \arctan\left(\frac{L}{R}\right)} \tag{6.2}$$

6.5.3 From bicycle steer to individual wheel angles

Figure 6.6 clarifies why the two physical front wheels cannot share the same angle: following arcs of different radii about the common ICR, their planes intersect it at different angles. Let d be the front track width and $\delta_{\text{in}}, \delta_{\text{out}}$ the inner and outer front-wheel angles. Given the equivalent bicycle steer δ, with $R = L/\tan\delta$, the ideal Ackermann relations are

$$\delta_{\text{in}} = \arctan\left(\frac{L}{R - \frac{d}{2}}\right), \qquad \delta_{\text{out}} = \arctan\left(\frac{L}{R + \frac{d}{2}}\right).$$

Since $R - \frac{d}{2} < R + \frac{d}{2}$, it follows that $\delta_{\text{in}} > \delta_{\text{out}}$ for any finite R, confirming that the inner wheel steers more.

6.5.4 Connection to the Pure Pursuit law

The bicycle relation (6.2) is precisely what links the Pure Pursuit geometry to the steering command.

Recalling that Pure Pursuit produces a commanded curvature $\kappa = 1/R$ from the look-ahead geometry, with the turning radius derived in (6.1), substitution into (6.2) gives the steering law

$$\boxed{\delta = \arctan\left(\frac{2L\,\sin\alpha}{l_d}\right)} \tag{6.3}$$

which maps the instantaneous heading error α and the look-ahead distance l_d into the equivalent bicycle steering angle. This is the reference that the inner PID loop is tasked to track.

6.5.5 Minimum turning radius and the need for an end-of-row maneuver

The choice of Ackermann steering for this robot is motivated by its behavior on the soft, structured ground of a vineyard. Unlike skid (differential) steering, which turns the vehicle by driving the wheels on the two sides at different speeds and therefore relies on the wheels slipping sideways against the ground, Ackermann steering turns through pure rolling: each wheel follows its own arc without scrubbing.
This yields more accurate low-speed tracking, a well-defined yaw behavior, and—of particular importance between vine rows—minimal disturbance of the soil, since the wheels are not dragged laterally across it.

This advantage, however, comes at a cost. Because Ackermann steering turns through rolling rather than by pivoting in place, it cannot achieve arbitrarily tight turns: the vehicle has a finite minimum turning radius, larger than that of a skid-steered platform, and consequently requires more space to change rows at the end of a corridor.

The bicycle relation quantifies this limit. The steering angle is mechanically limited to a maximum value $\delta_{\max}$; substituting it into $R = L/\tan\delta$ gives the **minimum turning radius**

$$R_{\min} = \frac{L}{\tan\delta_{\max}}.$$

For the present robot, with $L \approx 1.6\,\mathrm{m}$ and $\delta_{\max} \approx 45°$, this evaluates to $R_{\min} \approx 1.6\,\mathrm{m}$.

This value has a direct operational consequence.

In the vineyard considered here, the rows are spaced approximately 1.8 m apart. Turning from one row into the immediately adjacent one with a single half-circle (U-turn) would require a turning radius of about 0.9 m, well below the achievable $R_{\min} = 1.6\,\mathrm{m}$: a direct U-turn into the neighboring row is therefore geometrically infeasible for this vehicle, and cannot be completed in a single forward arc.

The most plausible operating strategy is therefore to work on *alternate rows*. Skipping one row places the target lane at a lateral distance of about 3.6 m, so that the U-turn connecting the two requires a radius of about 1.8 m—above $R_{\min}$, and thus executable in a single forward maneuver. Should consecutive-row operation be required instead, the adjacent-row transfer can be completed with a **fishtail** (switchback) maneuver, which introduces a reverse segment to realign with the next row within a much smaller footprint. Such a maneuver, however, concerns the generation of a feasible reference trajectory—which arcs, which reverse segments, in which sequence—and therefore belongs to the high-level path-planning layer that, as stated in Chapter 6, lies outside the scope of this thesis.

There, the controller is exercised on a reference path that simulates the route the robot would follow under the alternate-row strategy: the U-turns it contains are set at the corresponding turning radius and are used as *demanding test inputs* for the steering controller, imposing a large, sustained steering demand that drives the inner loop towards its saturation limits. The path then includes a chicane representative of a typical high-level trajectory re-planning to avoid an obstacle, which imposes rapid reversals of steering direction. Together, these segments stress the inner loop and reveal its tracking accuracy, its response speed, and its behavior near the steering saturation limits.

6.5.6 Remarks on validity

The bicycle-to-Ackermann conversion above is purely *kinematic*, and is accurate at the low speeds considered here, where slip angles are small and the ICR construction is meaningful.
As speed and lateral acceleration increase, tire slip and dynamic effects—yaw rate, lateral velocity, load transfer—alter the relationship between curvature and steering, and a dynamic model with tire cornering stiffness becomes necessary. Such a dynamic model is precisely what is adopted for the simulation environment in Chapter 7; the kinematic formulation established here provides the geometric mapping from the look-ahead geometry to the steering angle, while the dynamic response of the vehicle is captured by the three-degree-of-freedom model described there.

Chapter 7

Design of the Autonomous Steering Actuation System

7.1 Introduction

The previous chapter established the theoretical foundations of the autonomous steering system: the trajectory-tracking law (Pure Pursuit), the working principle of the PID controller, and the Ackermann kinematics relating steering angle and path curvature.
This chapter brings those elements together into a complete simulation model, designs the inner steering controller, analyzes its stability, and validates the overall system through a campaign of simulations.

The chapter is organized as follows.
Section 7.2 describes the dynamic model of the vehicle adopted in simulation—the three-degree-of-freedom single-track model—together with its governing equations, parameters, and modeling assumptions.
Section 7.3 details the construction of the Simulink model: the generation of the reference path, the setup of the Pure Pursuit outer loop, the conversion from target direction to steering reference, and the closure of the outer guidance loop.
Section 7.4 addresses the inner steering loop in full: the modeling of the actuator through its transfer function, the design of the PID controller by pole placement, the choice of the controller structure, and the stability analysis of the resulting closed loop, supported by its step response.
Finally, Section 7.5 presents the simulation campaign, which validates the tuning of the guidance and control parameters and examines the rejection of external disturbances representative of operation on sloped terrain.

7.2 The vehicle dynamic model

The motion of the robot in simulation is reproduced by the `Vehicle Body 3DOF` block (single-track configuration) of the Simulink Vehicle Dynamics Blockset. As anticipated in Chapter 6, this block is the dynamic counterpart of the single-track abstraction introduced there: it shares the same geometric simplification—the two wheels of each axle are lumped into a single equivalent wheel lying on the vehicle centerline—but, unlike the purely kinematic model used to derive the steering law, it solves the planar rigid-body equations of motion, computing the lateral tyre forces that arise during cornering and the resulting yaw motion.
The equations of motion implemented by the block are reported in the MathWorks documentation [21].

7.2.1 Physical principle of the model

Before writing the equations, it is useful to describe how the model reasons.
The vehicle is treated as a rigid body moving in the horizontal plane, with three degrees of freedom: the longitudinal motion along the vehicle-fixed x-axis, the lateral motion along the y-axis, and the yaw rotation ψ about the vertical axis. The block adopts the SAE vehicle coordinate system, fixed to the body with the x-axis pointing forward, and with the origin at the center of gravity.

The lateral motion originates from the tyres. When a tyre rolls while its plane is not exactly aligned with its direction of travel, it develops a small angle between the two, the slip angle; in response, the tyre generates a lateral force that, within the normal operating range, is proportional to that slip angle. The constant of proportionality is the *cornering stiffness* of the tyre.
These lateral forces, acting at the front and rear axles, produce both the lateral acceleration of the vehicle and, because they act at a distance from the center of gravity, the yaw moment that turns the vehicle. The equations below simply formalize this chain: from the vehicle state to the slip angles, from the slip angles to the tyre forces, and from the forces to the accelerations.

7.2.2 Equations of motion

The block is used in the *External longitudinal velocity* configuration, in which the forward speed is prescribed externally and the tyre lateral forces are computed from the slip angles through a linear cornering-stiffness model. Under this setting the longitudinal dynamics are treated as quasi-steady, so that the longitudinal acceleration is taken as zero.

Slip angles. Denoting by $\dot{x}$ and $\dot{y}$ the longitudinal and lateral velocities in the vehicle frame, by $r = \dot{\psi}$ the yaw rate, by a and b the distances from the center of gravity to the front and rear axles, and by δ_f, δ_r the front and rear steering angles, the front and rear slip angles are

$$\alpha_f = \arctan\left(\frac{\dot{y} + a\,r}{\dot{x}}\right) - \delta_f, \qquad \alpha_r = \arctan\left(\frac{\dot{y} - b\,r}{\dot{x}}\right) - \delta_r.$$

Each slip angle is the difference between the direction in which the axle is actually travelling (the arctangent term, which combines the lateral velocity with the contribution of the yaw rate) and the direction in which its wheel is pointing (the steering angle). For this rear-unsteered robot, $\delta_r = 0$.

Tyre lateral forces. The lateral force at each axle is proportional to its slip angle through the cornering stiffness C_{yf}, C_{yr}, scaled by the friction coefficient and by the ratio of the instantaneous vertical load to the nominal load:

$$F_{yf} = -\,C_{yf}\,\alpha_f\,\mu_f\,\frac{F_{zf}}{F_{z,\mathrm{nom}}}, \qquad F_{yr} = -\,C_{yr}\,\alpha_r\,\mu_r\,\frac{F_{zr}}{F_{z,\mathrm{nom}}}.$$

The negative sign expresses that the lateral force opposes the slip, tending to realign the tyre with its direction of travel.

Rigid-body dynamics. The lateral and yaw motions then follow from Newton's and Euler's equations for the planar rigid body:

$$\ddot{y} = -\,\dot{x}\,r + \frac{F_{yf} + F_{yr} + F_{y,\mathrm{ext}}}{m}, \qquad \dot{r} = \frac{a\,F_{yf} - b\,F_{yr} + M_{z,\mathrm{ext}}}{I_{zz}}, \qquad \ddot{x} = 0,$$

where m is the vehicle mass, I_{zz} the yaw moment of inertia, and $F_{y,\mathrm{ext}}, M_{z,\mathrm{ext}}$ any external lateral force and yaw moment applied to the body. The term $-\dot{x}\,r$ is the centripetal contribution due to describing the motion in the rotating vehicle frame, and $\ddot{x} = 0$ reflects the imposed constant forward speed. Finally, the pose in the earth-fixed frame is obtained by integrating

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi, \qquad \dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi, \qquad \dot{\psi} = r.$$

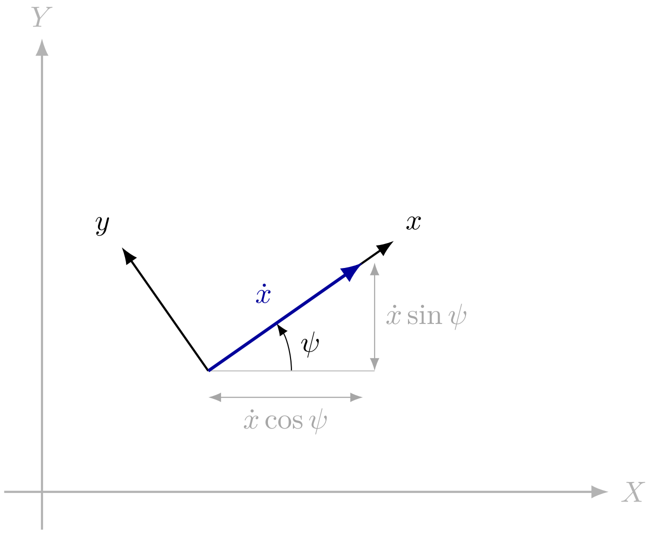


Figure 7.1: Transformation from the vehicle-fixed frame (x, y) to the earth-fixed frame (X, Y). The vehicle is rotated by the yaw angle ψ; the longitudinal and lateral velocities are projected onto the fixed axes. Source: author's own elaboration.

These parameters, once integrated, will return the exact coordinates (so the exact position) of the robot step by step. Those information will be used by the Pure Pursuit as the feedback of the outer loop.

7.2.3 Configuration and assumptions

The model is configured consistently with the operating conditions of the robot, under the following assumptions:

- **Planar motion.** Only longitudinal, lateral, and yaw motions are modeled; pitch, roll, and vertical dynamics are neglected. This is appropriate for the low-speed operation considered here and is the reason the single-track 3DOF model is preferred over higher-fidelity multi-body models.
- **Imposed longitudinal velocity.** The forward speed is prescribed and held constant at $v = 2\,\mathrm{m/s}$, representative of the cruise speed within the rows; accordingly the longitudinal acceleration is zero. The model computes the lateral and yaw response produced by the steering input. This is consistent with the Pure Pursuit controller, which likewise assumes a constant linear velocity.
- **Lateral forces from slip.** The tyre lateral forces are computed from the slip angles through a linear cornering-stiffness model, so that tyre slip is explicitly represented in the vehicle response.

- **Steering input.** The front wheel angle $\delta(t)$, produced by the inner steering loop, is the control input that drives the lateral and yaw motion; the rear axle is unsteered.

7.2.4 Model parameters

The parameters supplied to the block are summarized in Table 7.1. The mass is taken at its worst-case (fully loaded) value. The center of gravity is assumed approximately central, with equal distances from the front and rear axles. The yaw moment of inertia, the center-of-gravity height, and the tyre cornering stiffness—not directly available at this preliminary design stage—are estimated as described below.

Parameter	**Symbol**	**Value**
Vehicle mass (worst case)	m	650 kg
CG-to-front-axle distance	a	0.6 m
CG-to-rear-axle distance	b	1 m
Yaw moment of inertia (estimated)	I_{zz}	≈ 230.2 kg m^2
CG height (estimated)	h	≈ 0.9 m
Front/rear cornering stiffness (est.)	C_{yf}, C_{yr}	15 000 N/rad
Forward speed	v	2 m/s

Table 7.1: Parameters of the Vehicle Body 3DOF single-track model.

Yaw moment of inertia. Since a detailed mass distribution of the prototype is not yet available, the yaw moment of inertia is estimated by approximating the vehicle body as a homogeneous rectangular parallelepiped of mass m, length L_{veh}, and width W_{veh}, rotating about its vertical axis through the center of gravity:

$$I_{zz} = \frac{m}{12}\left(L_{veh}^2 + W_{veh}^2\right).$$

With the overall dimensions $L_{veh} = 1.6\,\text{m}$ and $W_{veh} = 1.3\,\text{m}$,

$$I_{zz} = \frac{650}{12}\left(1.6^2 + 1.3^2\right) \approx 230.2 \text{ kg m}^2.$$

This first-order estimate is adequate for the present study; a more precise value would require the detailed CAD mass model of the final vehicle.

Center-of-gravity height. The center-of-gravity height is estimated at roughly 45% of the overall vehicle height ($H_{veh} = 2.0\,\text{m}$), reflecting the concentration of mass in the lower part of the vehicle, giving $h \approx 0.9\,\text{m}$. Under the constant-speed operation assumed here, the precise value of h has a negligible influence on the results, since it intervenes mainly in the longitudinal load transfer associated with acceleration and braking, which are absent in these simulations.

Cornering stiffness. As the tyres are not yet selected, the cornering stiffness is assumed equal for both axles (identical front and rear wheels) and set to a value representative of narrow off-road tyres. Reported cornering-stiffness values for narrow tyres are of the order of several thousand newtons per radian per tyre; accounting for two tyres per axle, an axle value of $C_{yf} = C_{yr} = 15\,000$ N/rad is adopted. This value is to be identified experimentally once the tyres and the operating surface are defined.

7.2.5 Representation of terrain slope

The single-track 3DOF model is planar and does not represent a three-dimensional tilt of the vehicle. The effect of a lateral ground slope is therefore introduced not as a geometric inclination, but as an equivalent external lateral force applied to the center of gravity through the block external-force input $F_{y,\text{ext}}$. For a lateral slope of angle φ, this force is

$$F_{y,\text{ext}} = m\,g\,\sin\varphi,$$

the component of gravity along the vehicle lateral axis. This representation captures the dominant effect of the slope—the lateral force that pushes the vehicle off its path and must be rejected by the control loops—while neglecting secondary out-of-plane effects such as lateral load transfer, consistently with the planar formulation. This disturbance is used in the dedicated test of Section 7.5.

7.3 Construction of the Simulink model

This section describes how the complete steering system is assembled in Simulink. The model reproduces the two-loop architecture introduced in Chapter 6.
The description proceeds in the natural order of the signal flow.
The reference path is first defined and sampled into a set of waypoints, which are supplied to the Pure Pursuit block; from the current vehicle pose, Pure Pursuit produces a target direction, converted by a dedicated subsystem into the reference steering angle $\delta_0(t)$.This reference is compared with the actual steering angle to form the error that drives the PID controller, whose command is applied to the actuator.
The actuator output then passes through a `saturation` block that enforces the mechanical steering limit of ±45°—the maximum angle the wheels can physically reach—yielding the actual steering angle $\delta(t)$.

This saturated angle enters the steering input of the Vehicle Body 3DOF block and, at the same time, is fed back to close the inner loop. Among the vehicle block outputs, the pose signals (X, Y, ψ) are selected, which serve two purposes: they are fed back to the Pure Pursuit block—closing the outer guidance loop—and they are sent to the visualization block that displays the robot's motion along the vineyard rows.

7.3.1 Scenario and reference path

As discussed in Chapter 6, the path used here reproduces the route the robot would follow under the *alternate-row* strategy: the robot travels along a row, performs a 180° U-turn into the next-but-one row, traverses it in the opposite direction, performs a second U-turn, and finally executes an obstacle-avoidance chicane at the end of the last row. The U-turns are set at the alternate-row turning radius (about 1.8 m), above $R_{\min}$ and thus individually executable in a single forward pass.

Although geometrically feasible, these U-turns are deliberately demanding: at that radius they call for a steering angle close to the vehicle's maximum, driving the command into saturation—precisely the condition of interest, since it lets the controller be examined at and beyond the actuator limits. The chicane, representative of a typical high-level trajectory re-planning to avoid an obstacle, complements them by imposing rapid reversals of steering direction.

Saturation also exposes a well-known issue of integral action, *integral windup*: while the command is saturated the tracking error does not vanish, yet the integral term keeps accumulating beyond what the actuator can deliver, and on leaving saturation this excess produces large overshoots and sustained oscillations. Its severity, and the improvement obtained with the anti-windup mechanism, are examined in Section 7.5.

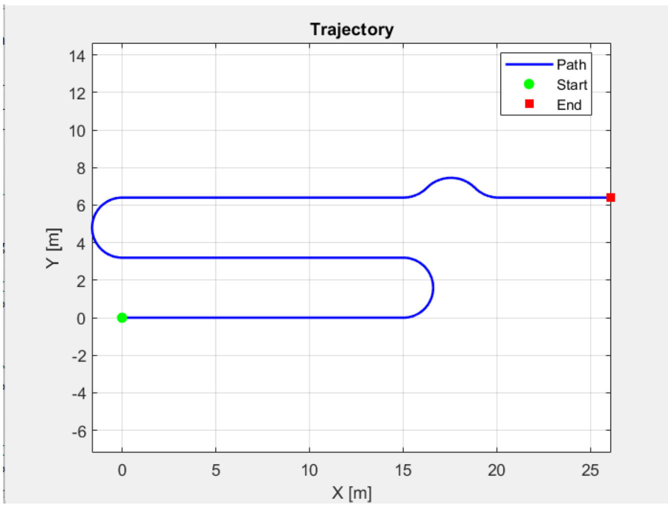


Figure 7.2: Reference path generated in MATLAB: serpentine motion across three rows followed by an obstacle-avoidance chicane. All turns respect the minimum turning radius.

The generation script produces the arrays of waypoint coordinates together with the initial pose of the robot,

$$\{x_i\}_{i=1}^{N}, \qquad \{y_i\}_{i=1}^{N}, \qquad (X_0,\, Y_0,\, \psi_0),$$

where $\{x_i\}, \{y_i\}$ are the waypoint coordinates, sampled along the path, and (X_0, Y_0, ψ_0) is the initial position and orientation of the vehicle. These arrays are passed directly to the Simulink model as the path input for the Pure Pursuit controller.

Initial orientation. The initial yaw angle ψ_0 is deliberately set with a 30° misalignment with respect to the local path direction at the starting point. This choice makes the steering transient more severe and easier to interpret: if the robot were initialized already aligned with the path, the performance of the controller would become visible only later, at the first curve, whereas the imposed misalignment produces a clear alignment-to-path transient from the very beginning of the simulation.

7.3.2 Outer loop: Pure Pursuit

In Simulink, the waypoint vectors $\{x_i\}$ and $\{y_i\}$ are imported from the workspace and concatenated into a single $N \times 2$ matrix, which is supplied to the `Pure Pursuit` block as its path input. The block is configured with a look-ahead distance L_d; its influence on the behavior of the guidance loop is analyzed separately in Section 7.5, where its value is tuned.

At each step, the Pure Pursuit block outputs the *target direction*, denoted $\alpha(t)$, that is, the heading from the current robot pose to the look-ahead point on the path (Chapter 6).
This angle expresses where the target point lies relative to the vehicle heading, and is not yet a steering command.

7.3.3 Conversion from target direction to steering reference

The target direction $\alpha(t)$ is routed into a dedicated subsystem that implements the algebraic conversion from target direction to the desired steering angle $\delta_0(t)$, according to the Pure Pursuit steering law derived in Chapter 6(Eq. (6.3)). The output of this subsystem is the *reference* steering angle $\delta_0(t)$, the wheel angle the vehicle should realize in order to follow the path.

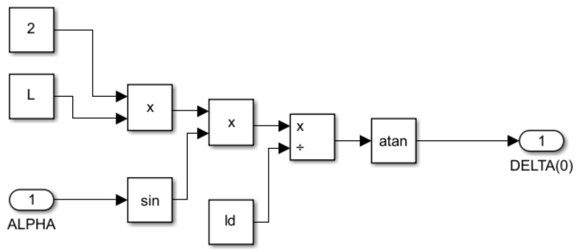


Figure 7.3: Block system for the conversion of $\alpha(t)$ into $\delta_0(t)$.

7.3.4 Inner loop: error, controller, and actuator

The reference steering angle $\delta_0(t)$ is compared with the actual steering angle $\delta(t)$ by a `Sum` block configured as $\delta_0 - \delta$, producing the steering error

$$e_\delta(t) = \delta_0(t) - \delta(t).$$

This error drives the `PID Controller` block, whose output is the command sent to the steering actuator. The actuator is represented by its transfer function $G_p(s)$—a first-order model whose derivation and parameters are the subject of

Section 7.4—implemented in Simulink through a `Transfer Fcn` block.
The output of the actuator is the actual steering angle $\delta(t)$.

The inner loop is closed by feeding $\delta(t)$ back to the `Sum` block. In this way the inner loop continuously drives the actual steering angle toward the reference produced by the guidance layer.

Finally, a `Saturation` block is placed on the steering signal before it enters the vehicle block, limiting the steering angle to the mechanical maximum of $\pm 45°$. This reproduces the physical limit of the wheels' maximum rotation and prevents the simulation from commanding steering angles—and hence path curvatures—that the real robot could not physically achieve.

7.3.5 Vehicle dynamics and closure of the outer loop

Once limited by the `Saturation` block, the actual steering angle $\delta(t)$, is fed to the `Vehicle Body 3DOF` block. The block, once filled with all the parameters described in Section 7.2, is initialized with the pose (X_0, Y_0, ψ_0) and the constant forward speed $v = 2\,\mathrm{m/s}$, and integrates the equations of motion to produce the time histories of the vehicle state, in particular its position and orientation $(X(t), Y(t), \psi(t))$.

The vehicle state is collected in an output bus, from which a `Bus Selector` extracts the pose components required by the Pure Pursuit block to compute the next look-ahead point. This closes the *outer loop*: the updated pose feeds the Pure Pursuit geometry, which updates the target direction $\alpha(t)$; the conversion angle subsystem produces the new reference $\delta_0(t)$; and the inner loop drives the actuator to realize $\delta(t)$, which in turn moves the vehicle.
The two loops thus operate together at every integration step, the inner one realizing the steering command and the outer one regenerating it from the resulting motion.

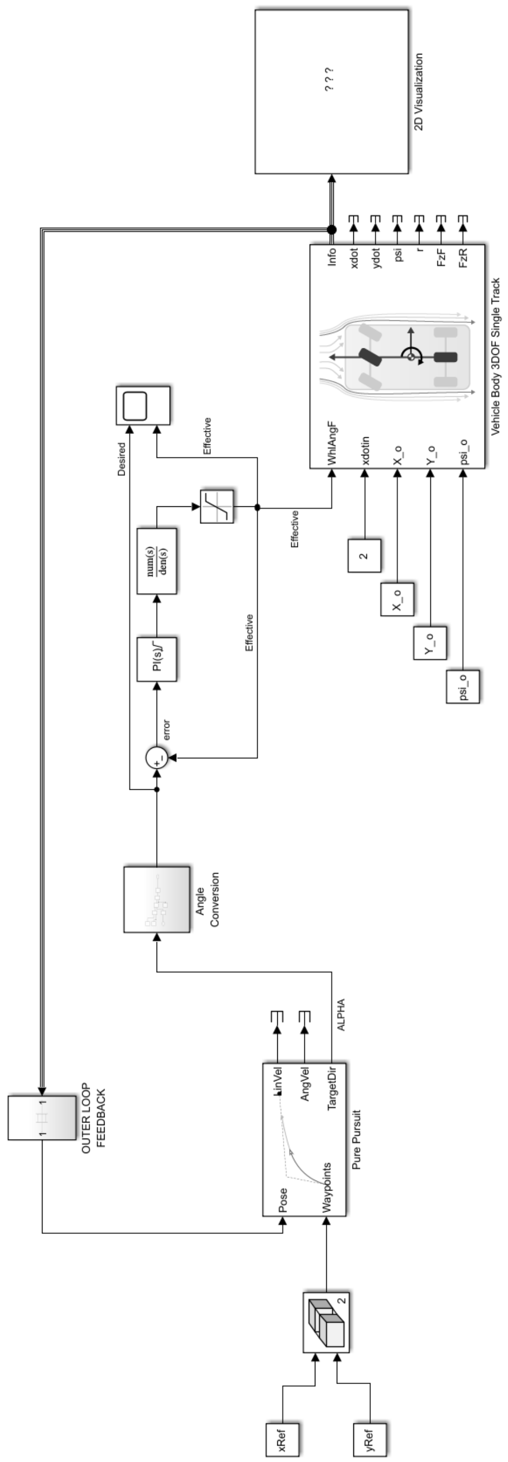


Figure 7.4: Complete Simulink model of the autonomous steering system.

7.3.6 Visualization

For qualitative validation, the vehicle pose and the scenario entities are also connected to a visualization block (scenario viewer), which displays the robot's motion relative to the vine rows and the headland turns during the simulation. This graphical feedback was used to verify that the exported waypoints reproduce the intended field path and that the 30° initial misalignment produces the expected alignment transient. In addition, dedicated scope record the reference and actual steering angles and the vehicle trajectory over time, providing the quantitative signals used in the simulation campaign of Section 7.5.

7.4 The inner steering loop: modeling, design, and stability

This section develops the inner control loop in full. It first models the steering actuator as a transfer function, then designs the controller by pole placement, motivates the choice of a PI structure, and finally analyzes the stability and the dynamic response of the resulting closed loop.

7.4.1 Choice of a PI structure

The controller is deliberately chosen as a PI rather than a full PID; that is, the derivative gain is set to zero.

The derivative action differentiates the error signal, which amplifies high-frequency measurement noise—a significant drawback on a robot operating over rough terrain, where the steering feedback is affected by vibration and would require additional filtering.
The integral term, by contrast, is essential: it guarantees zero steady-state error in the tracking of the steering reference and rejects the constant load disturbances acting on the steering channel—such as the torque induced by a lateral ground slope.
A PI controller thus provides all the required properties while avoiding the drawbacks of derivative action.

7.4.2 Transfer function of the steering actuator

The inner loop must drive the physical steering angle $\delta(t)$ to follow the reference $\delta_0(t)$ generated by the guidance layer.
The steering actuator—an electric motor with its gearbox and driver—converts the control command into the motion of the steering mechanism. A detailed

electromechanical model of this actuator would require the parameters of a specific motor; however, as stated in the earlier chapters, the motor has not yet been selected. At this design stage the actuator is therefore represented by its dominant input–output dynamics rather than by a detailed model of its internal components.

Any current-controlled electromechanical actuator, regulated internally by its own current and velocity loops, behaves as a system that brings the steering angle to its commanded value with a single dominant time constant. This behavior is captured by a first-order transfer function

$$G_p(s) = \frac{\delta(s)}{U(s)} = \frac{1}{\tau s + 1}, \tag{7.1}$$

where $U(s)$ is the control command, $\delta(s)$ the actual steering angle, and τ the actuator time constant.
The transfer function (7.1) aggregates the internal current and velocity loops of the real drive into a single first-order lag: it does not neglect them, but represents their combined effect at the level of abstraction appropriate here. The steady-state gain is unity, meaning that, at rest, the actuator reaches exactly the commanded angle.

The time constant is set to a nominal value $\tau = 0.25\,\mathrm{s}$, representative of a small steering actuator of this class. Since the motor is not yet selected, τ is an estimate; its precise value will be identified experimentally, through a step test on the actual actuator, once the motor is chosen. For this reason, the robustness of the control design with respect to variations of τ is explicitly verified at the end of this section.

7.4.3 Second-order model and canonical form

With the PI controller acting on the first-order actuator, the closed inner loop is a second-order system. It is therefore useful to recall the standard form of a second-order system, which provides a direct link between the closed-loop poles and the transient behavior:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0, \tag{7.2}$$

where ω_n is the natural frequency and ζ the damping ratio.
These two parameters govern the response: ω_n sets the speed of the response (how fast the system reacts), while ζ sets the degree of damping (how oscillatory the response is). For $0 < \zeta < 1$ the poles are complex conjugate,

$$s_{1,2} = -\zeta\omega_n \pm j\,\omega_n\sqrt{1-\zeta^2},$$

with negative real part $-\zeta\omega_n$: the real part determines the decay rate of the transient, and the imaginary part the frequency of the residual oscillation. A larger ζ reduces the overshoot; the value $\zeta = 1$ gives a critically damped, non-oscillatory

response.
Designing the controller then amounts to placing the closed-loop poles at a chosen (ζ, ω_n)—the *pole placement* method.

7.4.4 Controller design by pole placement

The controller acts on the steering error $e_\delta = \delta_0 - \delta$.
Using a proportional–integral (PI) controller,

$$G_c(s) = K_p + \frac{K_i}{s},$$

the open-loop transfer function is $G_c(s)\,G_p(s)$, and the closed-loop characteristic equation $1 + G_c(s)G_p(s) = 0$ becomes, with $G_p(s) = 1/(\tau s + 1)$,

$$1 + \left(K_p + \frac{K_i}{s}\right)\frac{1}{\tau s + 1} = 0 \quad \Longrightarrow \quad \tau s^2 + (1 + K_p)\,s + K_i = 0.$$

Dividing by τ,

$$s^2 + \frac{1 + K_p}{\tau}\,s + \frac{K_i}{\tau} = 0. \tag{7.3}$$

Comparing (7.3) term by term with the canonical form (7.2) gives the two design equations

$$\boxed{K_p = 2\zeta\omega_n\tau - 1, \qquad K_i = \omega_n^2\,\tau} \tag{7.4}$$

which express the controller gains directly in terms of the desired damping ratio and natural frequency. Once ζ and ω_n are chosen, the gains follow analytically—no trial-and-error tuning is required.

Choice of the damping ratio. A damping ratio $\zeta = 0.9$ is selected. This value is deliberately close to critical damping: it yields a fast response with negligible overshoot (well below 1%). The reason for avoiding overshoot is specific to this application: the inner loop output is the steering angle, and an overshoot in the steering angle would mean the wheels momentarily turning *more* than commanded, injecting an unwanted disturbance into the outer guidance loop and, ultimately, into the vehicle trajectory. A well-damped, non-oscillatory steering response is therefore preferable to a faster but oscillatory one.

Choice of the natural frequency. A natural frequency $\omega_n = 5\,\mathrm{rad/s}$ is chosen. This value sets the bandwidth—and hence the response speed—of the inner loop. It must be high enough for the steering servo to react quickly relative to the guidance loop, but not so high as to demand unrealistically aggressive actuator commands or to amplify measurement noise. The resulting inner-loop settling time is of the order of $t_s \approx 4/(\zeta\omega_n) \approx 0.9\,\mathrm{s}$, where the factor 4 corresponds to the 2% settling criterion: the transient envelope decays as $e^{-\zeta\omega_n t}$, and the time for it to fall below 2% of the final value is $t_s = -\ln(0.02)/(\zeta\omega_n) \approx 4/(\zeta\omega_n)$.
This is substantially faster than the dynamics of the outer guidance loop, which, at the low operating speed of $2\,\mathrm{m/s}$, evolves over several seconds. This separation of time scales is the condition required for the cascade of the two loops to operate correctly: the inner loop realizes the steering command fast enough that, from the viewpoint of the guidance loop, the actuator can be regarded as nearly instantaneous.

Resulting gains. Substituting $\tau = 0.25\,\mathrm{s}$, $\zeta = 0.9$, and $\omega_n = 5\,\mathrm{rad/s}$ into (7.4) gives

$$K_p = 2(0.9)(5)(0.25) - 1 = 1.25, \qquad K_i = (5)^2(0.25) = 6.25.$$

7.4.5 Stability analysis

The stability of the designed inner loop is now verified analytically and in the frequency domain.

Routh–Hurwitz criterion. As introduced in Chapter 6, the stability of a closed-loop system is determined by the denominator of its closed-loop transfer function. Setting this denominator to zero yields the *characteristic equation* of the system, whose roots are the closed-loop poles.
The system is asymptotically stable if and only if all these poles have negative real part, so that every transient term decays in time rather than growing. For the inner loop, the closed-loop transfer function derived in Chapter 6 has denominator $1+G_c(s)G_p(s)$; substituting $G_c(s) = K_p + K_i/s$ and $G_p(s) = 1/(\tau s+1)$ and setting it to zero gives exactly the characteristic equation

$$1 + \left(K_p + \frac{K_i}{s}\right)\frac{1}{\tau s + 1} = 0 \quad \Longrightarrow \quad \tau s^2 + (1 + K_p)s + K_i = 0,$$

whose roots are the closed-loop poles.

The corresponding closed-loop poles lie at

$$s_{1,2} = -\zeta\omega_n \pm j\,\omega_n\sqrt{1-\zeta^2} = -4.5 \pm j\,2.18,$$

Stability for any actuator time constant. The pole locations above confirm that the designed system is stable. It is worth going one step further and asking whether this stability depends on the specific value of the actuator time constant τ—which, as noted, is only an estimate at this stage. This question is answered by applying the Routh–Hurwitz criterion to the characteristic equation in symbolic form, keeping τ, K_p, and K_i as parameters.

The characteristic polynomial $\tau s^2 + (1 + K_p)s + K_i$ has coefficients $a_2 = \tau$, $a_1 = 1 + K_p$, $a_0 = K_i$, giving the Routh array

$$\begin{array}{c|cc} s^2 & \tau & K_i \\ s^1 & 1 + K_p & 0 \\ s^0 & K_i & \end{array}$$

For a second-order system the criterion reduces to requiring that all the first-column entries be positive:

$$\tau > 0, \qquad 1 + K_p > 0, \qquad K_i > 0.$$

The first condition is satisfied by *any* physical actuator, since a time constant is always positive; the other two hold for any positive gains. The inner loop is therefore stable for *every* $\tau > 0$, independently of its actual value. This is a significant robustness property: even if the selected motor turned out to be slower than assumed (a larger τ), the closed loop would remain stable—only its speed of response would change, not its stability. The stability of the design does not rely on the accuracy of the estimate $\tau = 0.25\,\mathrm{s}$.

The stability margins were evaluated from the Bode plot of the open-loop transfer function $L(s) = G_c(s)\,G_p(s)$, shown in Figure 7.5. The analysis returns an infinite gain margin and a phase margin of 83.75° at the gain-crossover frequency of 5.47 rad/s. The gain margin is infinite because the open-loop phase, starting at $-90°$ due to the integral action and approaching $-180°$ only asymptotically, never actually reaches $-180°$ at any finite frequency: there is no phase-crossover frequency, and the gain can therefore be increased without limit without destabilizing the loop. The large phase margin of about 84° confirms that the closed loop is not only stable but also well damped, in agreement with the design value $\zeta = 0.9$ and with the closed-loop poles found earlier. The gain-crossover frequency of 5.47 rad/s is consistent with the chosen natural frequency $\omega_n = 5\,\mathrm{rad/s}$, confirming the intended bandwidth of the inner loop.

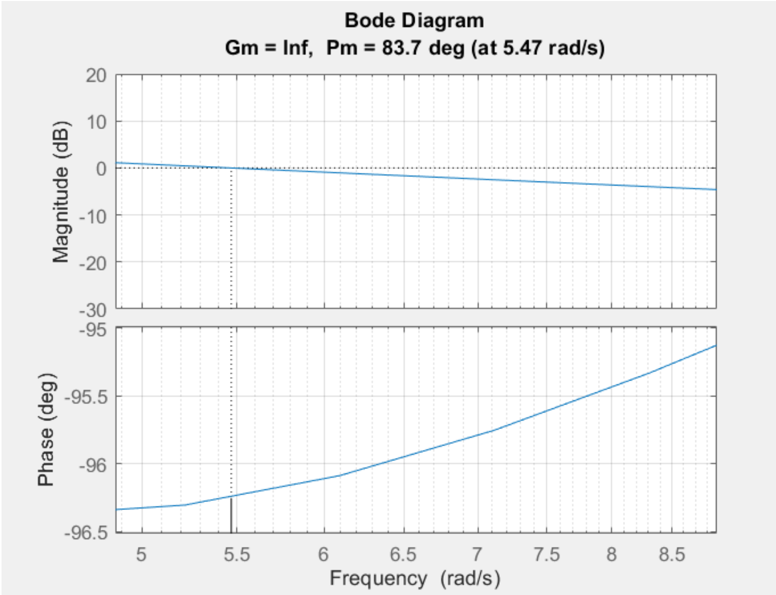


Figure 7.5: Bode plot of the open-loop transfer function of the inner steering loop, showing the phase margin and the absence of gain crossover above $-180°$.

7.4.6 Step response and robustness to actuator uncertainty

The closed-loop step response of the inner loop is shown in Figure 7.6. With the designed gains, the actual steering angle reaches the commanded value with the expected behavior: a fast rise, negligible overshoot, and a settling time of about $0.9\,\mathrm{s}$, matching the (ζ, ω_n) targets of the design.

Because the actuator time constant τ is an estimate, the same response is evaluated for three values, $\tau = 0.1$, 0.25, and $0.4\,\mathrm{s}$, spanning the plausible range of the not-yet-selected motor. As shown in Figure 7.6, the response remains stable and well damped across the whole range: the settling time varies moderately, but no oscillatory or unstable behavior appears. This confirms that the control design is robust to the uncertainty on the actuator dynamics, and that the conclusions do not depend on the precise value of τ.

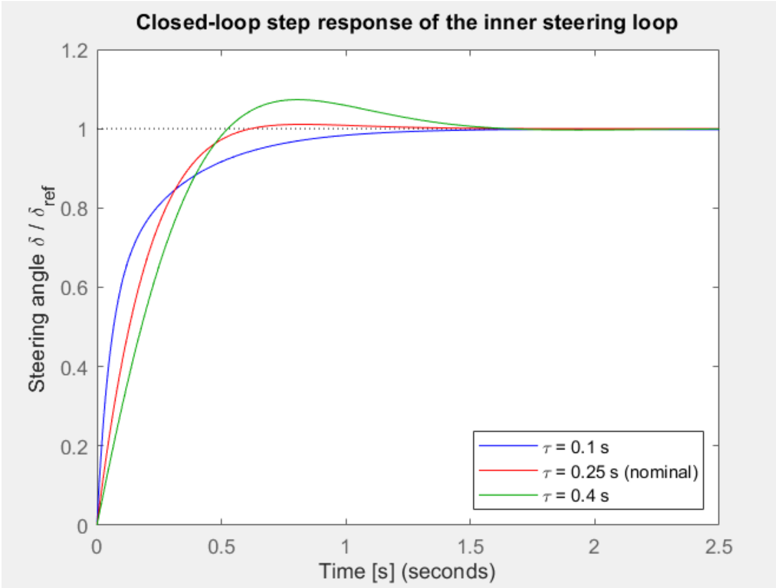


Figure 7.6: Closed-loop step response of the inner steering loop for three values of the actuator time constant τ, showing stable and well-damped behavior across the plausible range.

7.5 Simulation campaign

Having designed the inner loop and verified its stability, this section validates the complete system in simulation.
The campaign is organized as a progression: it first demonstrates that the PI play an important role into the system (Section 7.5.1), then examines the effect of the anti-windup mechanism during saturated maneuvers (Section 7.5.2), tunes the look-ahead distance of the guidance loop (Section 7.5.3), and finally assesses the rejection of an external disturbance representative of operation on sloped terrain (Section 7.5.4).
All runs use the reference path described in Section 7.3—the serpentine trajectory with U-turns and an obstacle-avoidance chicane—at a constant forward speed of 2 m/s, with the vehicle initialized with a 30° heading misalignment to expose the alignment transient from the start.

7.5.1 Verification of the control design

The first test verifies the role of the inner control loop by comparing two configurations that differ only in whether the controller is active. The plant, the reference path, and all other settings are kept identical; only the controller gains are changed.

Controller disabled ($K_p = K_i = 0$). In the first run the controller gains are set to zero. The outer guidance loop remains fully active—the Pure Pursuit controller keeps computing a valid steering reference $\delta_0(t)$ from the vehicle pose—but the inner loop produces no actuator command, so the reference is never turned into an actual steering angle.
As a result, the wheels do not steer and the vehicle proceeds essentially straight, ignoring the path entirely. This run makes explicit the division of roles between the two loops: the guidance layer decides where the wheels should point, but without the inner loop that command is never realized, and trajectory tracking fails completely.

Controller active (designed PI). In the second run the inner loop uses the PI controller designed by pole placement in Section 7.4 ($K_p = 1.25$, $K_i = 6.25$). With the controller active, the actual steering angle follows the reference produced by Pure Pursuit: the vehicle resolves the initial 30° heading misalignment, aligns onto the first row, and then negotiates the straight rows, the U-turns, and the obstacle-avoidance chicane while staying on the path.

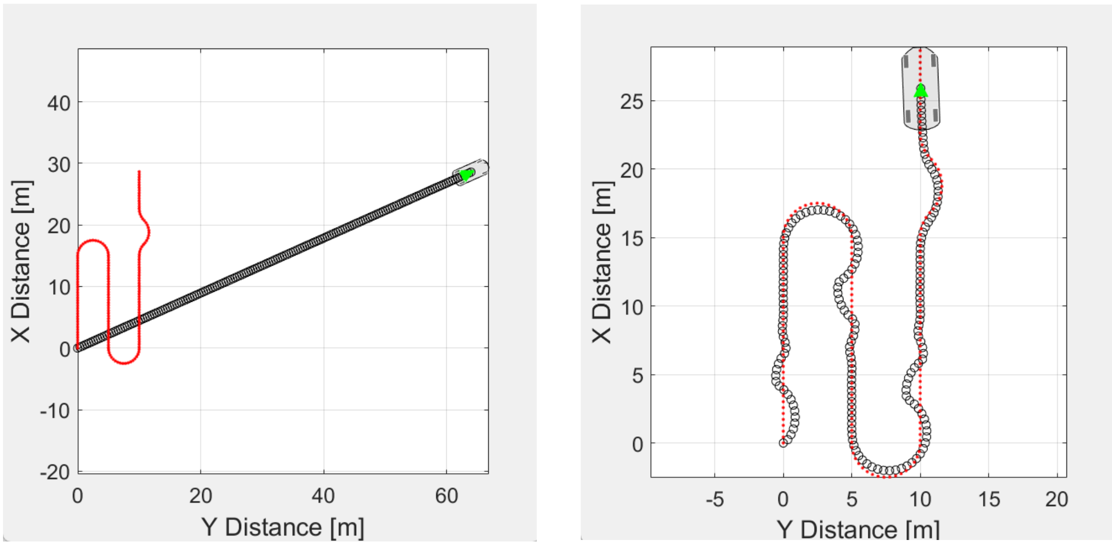


Figure 7.7: On the left the result of the simulation with the gains $K_p = K_i = 0$. On the right the result of the simulation with the gains $K_p = 1.25$ $K_i = 6.25$.

The comparison showed in Figure7.7 demonstrates the necessity of the inner control loop: the guidance layer alone is not sufficient to move the vehicle along the path, since its steering reference must be physically realized by the actuator through the inner loop. With the designed PI controller this is achieved, and the resulting configuration is adopted as the baseline for all subsequent tests.

7.5.2 The role of the anti-windup

With the designed PI controller the vehicle tracks the path, but reveals a residual issue linked to the steering saturation. When the steering angle demanded by the guidance law exceeds the physical limit of $\pm 45°$, the command saturates. This section examines what happens in the saturated segments, and why an anti-windup mechanism is required.

Origin of the problem: integral windup. While the steering command is saturated, the actuator delivers its maximum angle and the tracking error does not vanish; nevertheless, the integral term of the controller keeps integrating this persistent error, accumulating—"winding up"—to a value well beyond what the actuator can realize. Two effects follow, and they compound each other.

First, on the steering signal itself (Figure 7.8): without anti-windup the effective angle remains saturated over markedly longer intervals, and when it finally leaves the limit it does so abruptly—an almost instantaneous change of angle that is, in itself, physically unrealizable. This is the over-charged integrator discharging: the command stays pinned at the limit until the accumulated integral is finally unwound.

Second, and more importantly, this delay couples back into the guidance loop and sets up an oscillation of the whole system.
Because the steering is held turned for longer than required, the vehicle does reach the path, but it arrives while still steering, so it overshoots and crosses to the other side. The Pure Pursuit controller, which computes its reference from the current vehicle pose, now sees the vehicle off the path on the opposite side and reacts by demanding a sharp counter-steer to bring it back; this large demand again exceeds the steering limit and saturates once more, feeding a new windup cycle
This feedback between the two loops explains why, without anti-windup, even the desired steering angle oscillates and grows in amplitude: each overshoot forces the guidance law to request a corrective angle, which could saturate again.
On the trajectory this manifests as the vehicle weaving about the row after each maneuver before it settles.

Enabling the anti-windup breaks this cycle at its source. With the integral term prevented from accumulating during saturation, the steering leaves the limit promptly and the vehicle approaches the path gently, tangentially, rather than overshooting it. The guidance loop is then never forced into the sharp counter-steering that sustained the oscillation, so both the delay on the steering signal and the trajectory oscillations at the exit of the turns are strongly reduced, as shown in Figures 7.9 and 7.10.

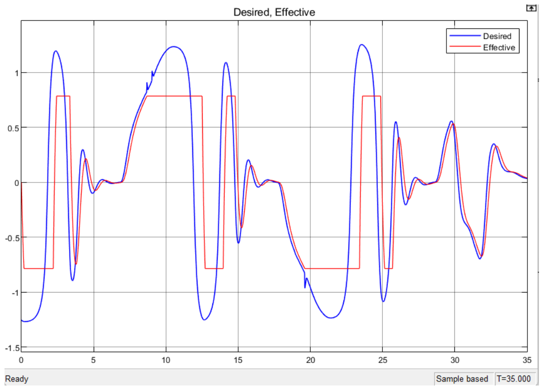


Figure 7.8: Steering behavior without anti-windup.

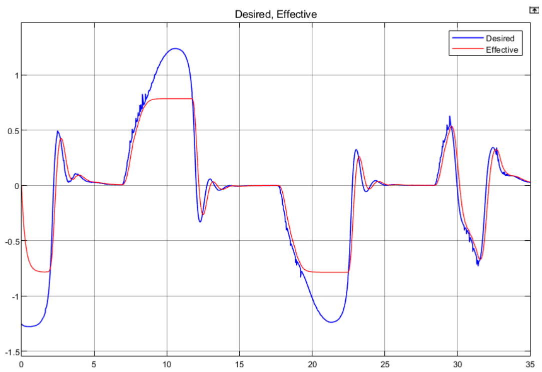


Figure 7.9: Steering behavior with anti-windup. Oscillations strongly reduced.

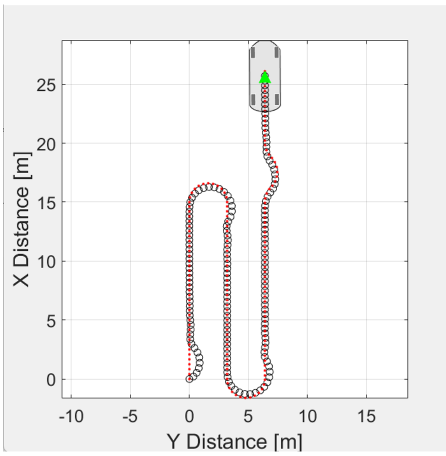
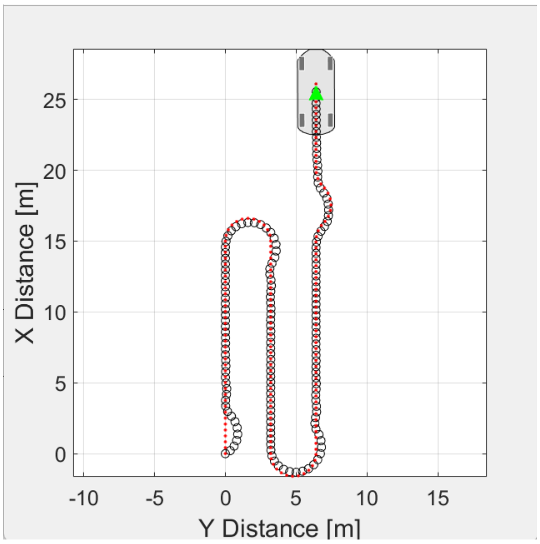


Figure 7.10: Result of the simulation with the anti-windup on.

7.5.3 Tuning of the look-ahead distance

All the simulations presented so far used a look-ahead distance $L_d = 1.5\,\mathrm{m}$.
This parameter governs how far ahead along the path the Pure Pursuit controller aims, and it sets a fundamental trade-off in the behavior of the guidance loop as anticipated in Chapter6. To make this trade-off explicit, two further runs are compared against the baseline: an aggressive setting $L_d = 0.8\,\mathrm{m}$ and a smoother setting $L_d = 2.2\,\mathrm{m}$. The comparison is carried out on both the vehicle trajectory and the steering signal.

Short look-ahead ($L_d = 0.8\,\mathrm{m}$): accurate but nervous. With a short look-ahead the controller aims at a point close to the vehicle, and therefore corrects its heading strongly to reach it. As a result the vehicle follows the path more accurately—it cuts the curves less and stays closer to the reference through the turns.
This accuracy, however, comes at a price that is visible on the steering signal: the response is noticeably more nervous than the baseline, with larger and more frequent steering oscillations, as the controller reacts sharply to every small deviation from the path. The aggressive look-ahead thus improves tracking but degrades the smoothness of the steering action (Figure7.11).



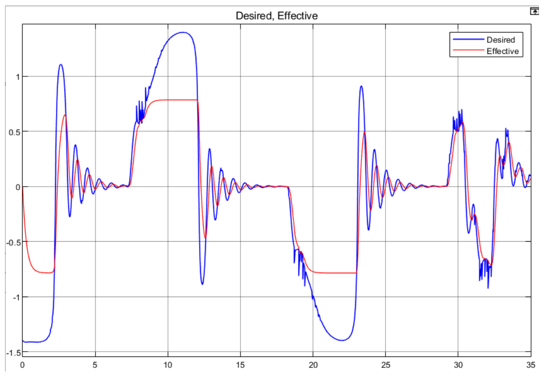


Figure 7.11: Path following and steering response with a $L_d = 0.8$ m.

Long look-ahead ($L_d = 2.2\,\mathrm{m}$): smooth but wide. With a long look-ahead the opposite behavior is obtained. Aiming at a point farther ahead, the controller demands gentler steering corrections: the steering signal is smoother and the oscillations are reduced with respect to the baseline, because the requested curvature is milder. This gentler action, however, causes the vehicle to cut the curves more markedly—entering them early and exiting wide. In a vineyard, the vehicle would risk leaving its lane and encroaching on the vine rows, ending up where it should not. The smooth response is therefore obtained at the expense of a tracking error (Figure7.12).

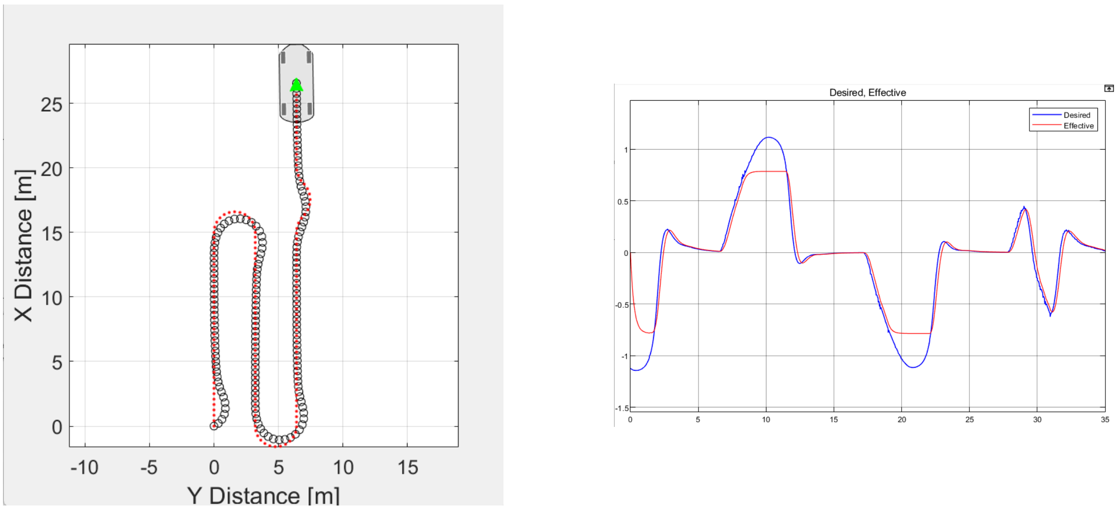


Figure 7.12: Path following and steering response with a $L_d = 2.2$ m.

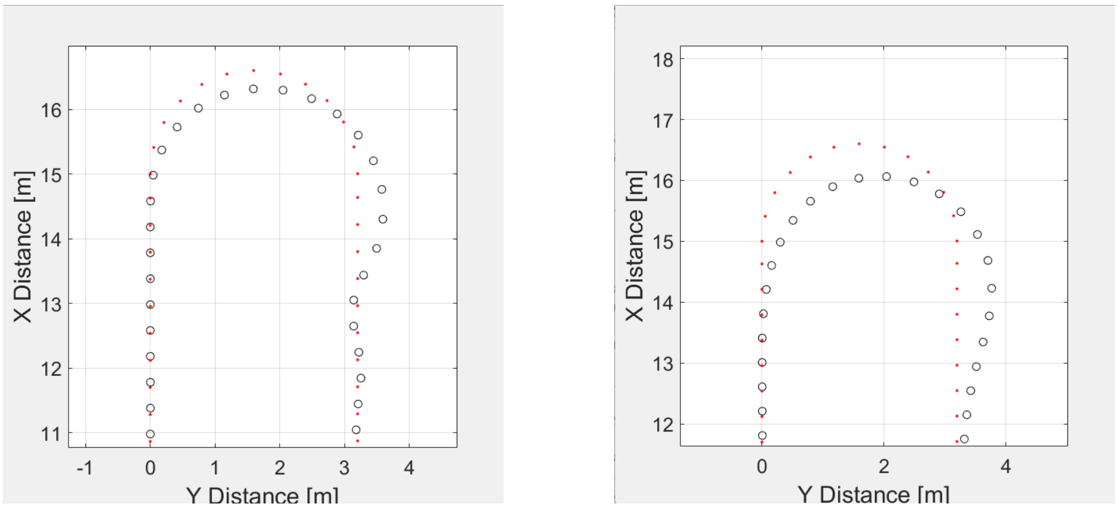


Figure 7.13: Detail of the U-turn trajectory for two look-ahead settings, with the ideal reference path shown in red. Short look-ahead ($L_d = 0.8\,\mathrm{m}$) on the left and long look-ahead ($L_d = 2.2\,\mathrm{m}$) on the right.

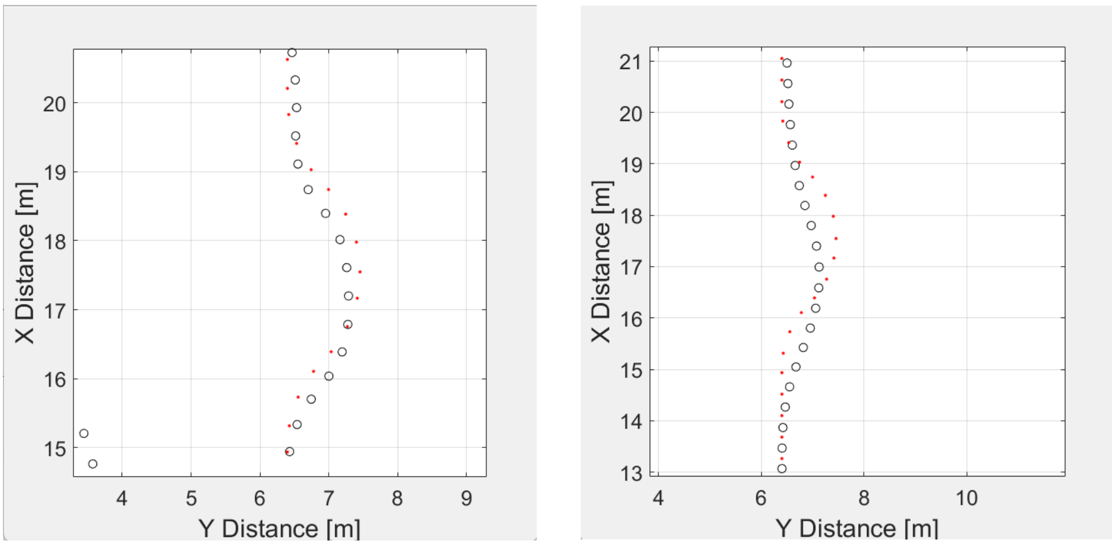


Figure 7.14: Detail of the chicane trajectory for two look-ahead settings, with the ideal reference path shown in red. Short look-ahead ($L_d = 0.8\,\text{m}$) on the left and long look-ahead ($L_d = 2.2\,\text{m}$) on the right.

Chosen value. The two settings illustrate the two opposing failure modes: too small a look-ahead gives an accurate but oscillatory, nervous behavior, while too large a look-ahead gives a smooth but imprecise one that cuts the path. The baseline value $L_d = 1.5\,\text{m}$ is chosen as a compromise between the two: it keeps the tracking error within acceptable bounds—without the vehicle leaving its lane—while maintaining a steering action that is smooth enough to avoid the nervous oscillations of the aggressive setting.
This value is retained for the remaining tests.

7.5.4 Disturbance rejection: lateral ground slope

The final test assesses the robustness of the closed-loop system against an external disturbance representative of real field operation: a lateral ground slope.
When the robot travels along a laterally inclined surface, the component of gravity along the vehicle's lateral axis pushes it sideways, tending to make it drift off the intended path. A good steering system must reject this disturbance and keep the vehicle on the row.

Modeling the slope as a lateral force. As discussed in Section 7.2, the single-track model is planar and does not represent a three-dimensional tilt of the vehicle. The effect of a lateral slope of angle φ is therefore introduced through the external-force input of the Vehicle Body 3DOF block, as a lateral force applied at the center of gravity,

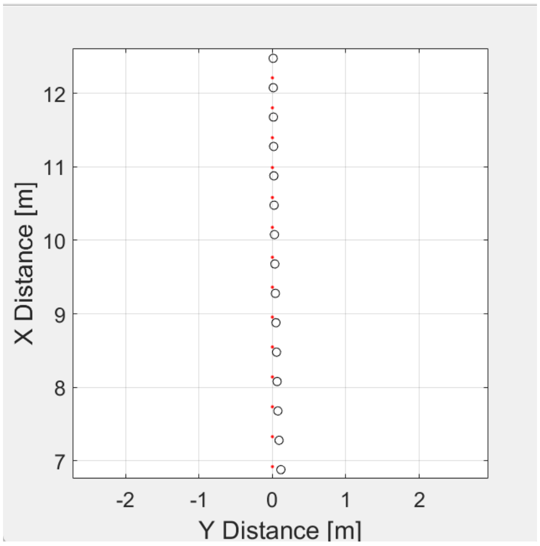
$$F_{y,\mathrm{ext}} = m\,g\,\sin\varphi.$$

This is the component of gravity acting along the vehicle's lateral axis. The external-force input is set to $[\,0,\ F_{y,\mathrm{ext}},\ 0\,]$: the longitudinal force and the yaw moment are left at zero, since a uniform lateral slope produces a pure lateral force at the center of gravity and no yawing moment. The disturbance thus captures the dominant effect of the slope—the lateral push that must be rejected by the control loops—consistently with the planar formulation.

For a lateral slope of 10°, the external lateral force is

$$F_{y,\mathrm{ext}} = 1107\,\mathrm{N}.$$

Results. With the lateral force applied, the vehicle is pushed toward the downhill side and initially deviates from the reference path. The guidance loop detects the resulting lateral error and commands a corrective steering angle that counteracts the drift, while the inner loop realizes it.
As a result, the vehicle recovers the path and continues to track it with a small residual deviation of $\approx 0.1\,\mathrm{m}$. Figure 7.15 shows the trajectory under the slope disturbance compared with the undisturbed case.



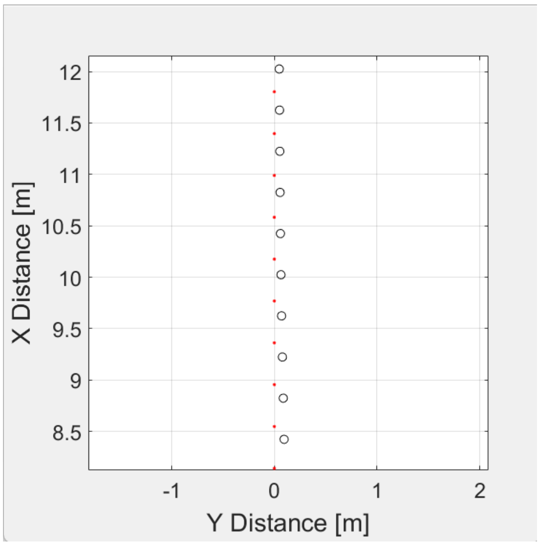


Figure 7.15: Trajectory under the slope disturbance compared with the undisturbed case

The rejection of the disturbance is also visible on the steering signal itself. On the straight rows, in the undisturbed case (image on the left in Figure 7.16) the steering reference settles close to zero, as the vehicle proceeds straight without need for correction; under the slope disturbance (image on the right in Figure 7.16), instead, the reference exhibits small but persistent adjustments, as the controller commands a slight continuous steering angle to oppose the lateral component of gravity that would otherwise push the vehicle downhill. These small alignment inputs, absent in the undisturbed case, are the signature—on the steering channel—of the disturbance-rejection action that keeps the vehicle on the row. However, the system will take more time to reach and keep the target position.

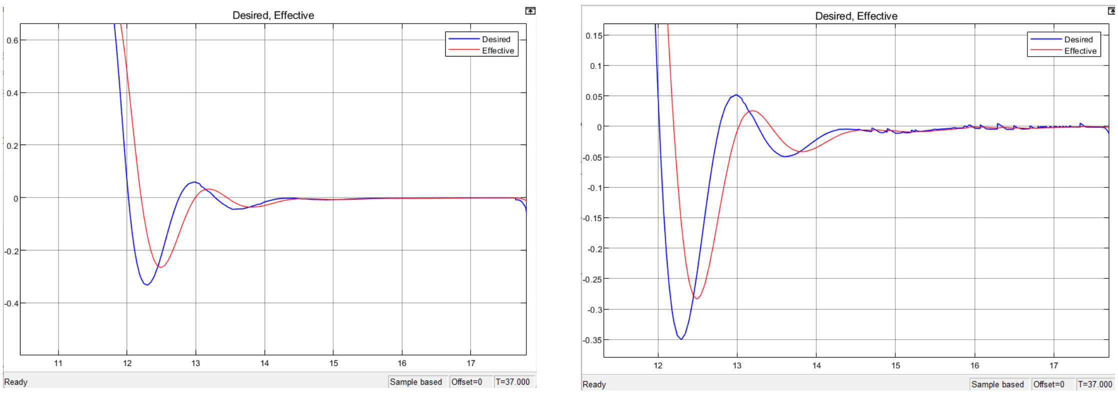


Figure 7.16: Steering reference and effective angle on the straight row following a U-turn, without (lesft side) and with (right side) the lateral slope disturbance.

7.6 Conclusions

This chapter brought together the elements introduced in Chapter 6 into a complete simulation model of the autonomous steering system, designed its inner controller, analyzed its stability, and validated the overall behavior through a campaign of simulations.

The vehicle was represented by the three-degree-of-freedom single-track model, in its dynamic form, which computes the lateral tyre forces from the slip angles and the cornering stiffness. The choice of this model, together with the assumptions of planar motion and constant forward speed, was shown to be appropriate for the study of steering control at the low operating speeds of the robot: it captures the dominant lateral and yaw dynamics while avoiding the complexity—and the additional, currently unavailable parameters.

The steering actuator was modeled as a first-order transfer function, aggregating the internal current and velocity loops of the drive into a single dominant time constant. On this basis, the inner steering controller was designed analytically by pole placement: expressing the closed-loop characteristic equation in the canonical second-order form allowed the proportional and integral gains to be derived directly

from a chosen damping ratio and natural frequency, rather than by trial and error. A PI structure was adopted, omitting the derivative action to avoid amplifying measurement noise on a vehicle operating over rough terrain, while retaining the integral term for its disturbance-rejection properties. The stability of the resulting loop was confirmed analytically—through the closed-loop poles.
The step response confirmed the intended fast, well-damped behavior and its robustness to the uncertainty on the actuator time constant.

The simulation campaign validated the design on a serpentine test path with U-turns and an obstacle-avoidance chicane.

The main findings are the following:
The comparison between the active and disabled controller confirmed the necessity of the inner loop, without which the steering reference computed by the guidance layer is never realized. The interaction between the steering saturation and the integral action was found to cause windup during the tightest maneuvers, coupling back into the guidance loop and producing an oscillation of the trajectory; enabling a clamping anti-windup mechanism removed this effect, confirming that the saturation block and the anti-windup are necessary and complementary.
The tuning of the look-ahead distance illustrated the fundamental trade-off of the Pure Pursuit law—between tracking accuracy and smoothness of the steering action—and motivated the chosen value as a compromise between the two opposing failure modes. Finally, the rejection of a lateral disturbance representative of operation on sloped terrain showed that the system keeps the vehicle on the path under a persistent lateral force, with a bounded residual offset.

Taken together, these results show that the proposed two-loop architecture, with an inner steering controller designed by pole placement, provides stable and effective trajectory tracking for the vineyard robot under the conditions considered.

It should be emphasized that the contribution of this work lies in the control architecture, the design method, and the simulation model, rather than in the specific numerical values of the parameters, which are preliminary estimates: the methodology developed here allows the design to be updated, once the motor, the tires, and the operating conditions are defined, without structural changes.

Chapter 8

Conclusions and Future Work

This thesis developed two subsystems of a mobile robot for vineyard transport, building on the chassis defined in the preliminary study [1]: a rear suspension and an autonomous steering system. The two parts belong to different layers of the design pyramid—the mechanical platform and the planning-and-control layer—and each was treated, sized, and verified on its own terms in Chapters 5 and 7, where the respective results and limitations are discussed in detail. This final chapter does not repeat those discussions; it places the two contributions in a single perspective at the project level and gathers the work that remains before the designs can be brought to hardware.

8.1 Overall contribution to the project

Although the two subsystems are technically unrelated, they serve the same overarching goal that motivated the project: to move a fragile payload gently and reliably along vineyard rows, on uneven, partially structured terrain. They act at different points of the same problem.

The rear suspension works on the *vertical* dynamics, absorbing the terrain-induced accelerations transmitted to the deck; the steering controller works on the *lateral* dynamics, shaping the steering profile so that path tracking does not inject abrupt lateral accelerations into the load. In this sense they are two complementary layers of a single load-protection strategy.

At the project level, the work advances the vineyard robot from the bare chassis of the previous study toward a platform with a defined rear running gear and a validated steering-control architecture. The rear suspension contributes a compact torsion-bar concept, sized and verified against static and fatigue criteria and integrated within the existing chassis; the steering system contributes a complete two-loop architecture, with an inner controller designed by pole placement and validated in simulation.

8.2 Future work

Rear suspension

- Study the dynamic response of the suspension, which relies on the torsion bar as its sole elastic element, with no dedicated damper. This entails identifying the natural frequency of the sprung system, characterizing its behavior when excited by the terrain inputs at the operating speeds, and verifying whether the working conditions stay clear of resonance—assessing, in particular, whether the residual damping provided by the tires, the compliant soil, and the material hysteresis is sufficient to attenuate the accelerations transmitted to the grape crates.

- Verify the fatigue strength of the welded joint between the reaction blocks and the base plates. The blocks work essentially as cantilevers, the bar torque being reacted at the base of the vertical block through its weld to the base plate; since welded joints exhibit a lower fatigue strength than the base material.

Autonomous steering

- Select the actuator and identify the estimated parameters experimentally—the actuator time constant through step tests, the tyre cornering stiffness and yaw inertia through dedicated tests—then update the control gains through the same pole-placement procedure.

- Introduce an explicit limit on the steering rate, representing the maximum speed of the real actuator, and verify it against the demands of the maneuvers once the motor is chosen.

- Extend the guidance loop with an adaptive look-ahead—longer on the straights, shorter through the turns—and replace the constant-speed assumption with a curvature-aware speed profile that slows the vehicle in the turns, reducing both the payload accelerations and the steering demand.

- Develop the high-level autonomy layer—perception, mapping, and trajectory planning, including a safety supervisor for emergency stop and obstacles avoiding.

8.3 Closing remarks

The two parts of this thesis deliver a verified rear-suspension concept and a complete, validated steering-control workflow for the vineyard transport robot, each advancing a distinct layer of the platform. Both are honest preliminary designs—aware of their assumptions, supported by analysis and simulation, and structured so that the move to hardware refines parameters within an established method rather than restarting the work. The future work identified above traces the path from these designs to a robot capable of safe, gentle, and reliable operation among the vines.

Bibliography

[1] M. Torrentó. «Conceptualization and Design of an All-Terrain Mobile Robot for Viniculture». Master's thesis. Barcelona, Spain: Universitat Politècnica de Catalunya (UPC), 2025 (cit. on pp. 1, 5, 11, 12, 15, 16, 18, 98).

[2] B. Moganapathi, P. S. Kavitha, I. Muthuvel, A. Senthil, D. Balachandar, and P. P. Mahendran. «The AI-viticulture nexus: Robotics and precision technologies for sustainable vineyards». In: *Plant Science Today* 12.sp1 (2025). DOI: `10.14719/pst.9713` (cit. on pp. 5, 6).

[3] «Autonomous Robotic Platform for Precision Viticulture: Integrated Mobility, Multimodal Sensing, and AI-Based Leaf Sampling». In: *Actuators* 15.2 (2026), p. 91. DOI: `10.3390/act15020091` (cit. on p. 6).

[4] «Experimental Assessment of Autonomous Fleet Operations for Precision Viticulture Under Real Vineyard Conditions». In: *Proceedings (MDPI)*. 2026. DOI: `10.3390/proceedings2026134047` (cit. on p. 6).

[5] Naïo Technologies. *Ted — Autonomous straddle robot for vineyards*. `https://www.naio-technologies.com/en/ted/`. Accessed: 2026-06-24 (cit. on p. 7).

[6] «Agricultural unmanned ground vehicles: A review from the stability point of view». In: *Revista Ciência Agronômica* (2021). Verify volume/pages/DOI from the SciELO page (cit. on p. 8).

[7] «Linear Quadratic Regulator (LQR) Control for the Active Suspension System of a Four-Wheeled Agricultural Robot». In: (2019). Verify journal, volume, pages, and DOI before use (cit. on p. 8).

[8] «Study on Chassis Leveling Control of a Three-Wheeled Agricultural Robot». In: *Agronomy* 14.8 (2024), p. 1765. DOI: `10.3390/agronomy14081765` (cit. on p. 8).

[9] Hui Liu Yafei Zhang et al. «A Systematic Review of Modeling and Control Approaches for Path Tracking in Unmanned Agricultural Ground Vehicles». In: *Agronomy* 15.10 (2025). DOI: `10.3390/agronomy15102274` (cit. on pp. 9, 61).

[10] «Application of Navigation Path Planning and Trajectory Tracking Control Methods for Agricultural Robots». In: *Agriculture* 16.1 (2025), p. 64. DOI: 10.3390/agriculture16010064 (cit. on p. 9).

[11] «Trajectory tracking control for autonomous vehicles: A systematic PRISMA review of models and strategies». In: *Annual Reviews in Control* (2026). ScienceDirect S1367578826000039 (cit. on p. 9).

[12] «Path Tracking Control in Autonomous Agricultural Vehicles: A Systematic Survey of Models, Methods, and Challenges». In: *Agriculture* 15.23 (2025), p. 2522. DOI: 10.3390/agriculture15232522 (cit. on p. 9).

[13] Future Farming. *New Adir Power platform for transport and applications in orchards and vineyards.* https://www.futurefarming.com/tech-in-focus/autonomous-semiauto-steering/autonomous-vehicles/ant-robotics-launches-adir-power-platform-for-transport-and-applications-in-orchards-and-vineyards/. Accessed: 2026-06-26. 2026 (cit. on pp. 10, 12).

[14] Augean Robotics. *Burro Generation 8.2 collaborative agricultural robot.* https://agtecher.com/en/robotics/burro-generation-8-2-cutting-edge-collaborative-robo/. Accessed: 2026-06-26. 2026 (cit. on pp. 11, 12).

[15] Daniel J. Inman. *Engineering Vibration.* 4th. Boston, MA: Pearson, 2014 (cit. on p. 23).

[16] Antonio Strozzi. *Fondamenti di costruzioni di macchine.* Esculapio, 2023 (cit. on pp. 30, 35).

[17] Lucefin Spa. *51CrV4 Quality.* URL: https://www.lucefin.com/wp-content/files_mf/152933444851CrV4.pdf (cit. on p. 33).

[18] Jacint B. Peiró. *La fatiga dels elements mecánics: 3.* UPC S.L, 2010 (cit. on pp. 36–39).

[19] DIN Deutsches Institut für Normung. *DIN 5463:1955-09 — Spline Shaft Connections with Straight Flanks; Medium Series.* Accessed: 2025-09-23. 1955 (cit. on p. 44).

[20] The MathWorks, Inc. *Pure Pursuit Controller.* MATLAB Documentation, Navigation Toolbox. Available at: https://it.mathworks.com/help/nav/ug/pure-pursuit-controller.html, accessed June 2026. 2026 (cit. on p. 62).

[21] The MathWorks, Inc. *Vehicle Body 3DOF — 3DOF Rigid Vehicle Body to Calculate Longitudinal, Lateral, and Yaw Motion.* MATLAB Documentation, Vehicle Dynamics Blockset. Available at: https://it.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html, accessed June 2026. 2026 (cit. on p. 72).