



UNIVERSITÀ DI TRENTO

Department of Information Engineering and Computer Science

Bachelor's Degree in
Computer, Communication and Electronic Engineering

FINAL DISSERTATION

Mechatronics Design of the ALPINE Climbing Robot

Supervisor
Michele Focchi

Student
Luca Hardonk

Academic year 2024/2025

Acknowledgments

I would like to thank my family for giving me the opportunity to pursue this academic path and for their constant support throughout these years. I am especially grateful to Professor Luigi Palopoli and Professor Michele Focchi for their guidance, for making the realization of this project possible and for the valuable knowledge and insights I gained under their supervision. Finally, I wish to thank all the colleagues and people I met along the way, whose collaboration and presence contributed to making this journey both constructive and rewarding.

Table of Contents

1 Introduction.....	8
1.1 Project Motivation.....	8
1.2 State of the Art.....	9
1.3 Requirements and Objectives of the Project.....	9
1.4 ALPINE Robot Overview.....	10
2 Mechanical Implementation of the Winch System.....	12
2.1 Reference Jumps.....	12
2.2 CAD Design.....	12
2.2.1 Mechanical Components of the Winch.....	13
2.3 Sizing of the Electrical Motor.....	15
2.4 Variable Transmission Calculations.....	16
2.5 Rope Design and Testing.....	18
2.5.1 Load Testing.....	19
2.5.2 Creep Testing.....	19
3 Electrical Implementation of the Winch System.....	20
3.1 Electrical Schematic.....	20
3.2 Wiring of the Winch.....	21
4 Software Architecture.....	22
4.1 Communication Network.....	22
4.2 Embedded Firmware of the Winch.....	22
4.3 Winch Telemetry Node.....	23
5 Friction Estimation.....	24
5.1 Nonlinear Friction Model.....	24
5.2 Parameters Identification Methodology.....	24
5.3 Experimental Results and Model Validation.....	26
6 Mechanical Design of the ALPINE Body.....	27
6.1 CAD Design.....	27
6.2 ALPINE Body Components.....	28
7 Electronic Design of the ALPINE Body.....	31
7.1 Electrical Schematic.....	31
8 Leg System of the ALPINE Climbing Robot.....	33
8.1 Piston Performance Calculations.....	33
8.2 Piston Stroke, Volume, Acceleration, and Timing.....	33
8.3 Required Air Flow and Tube Sizing.....	33
8.4 Pneumatic Diagram.....	34
9 Software Architecture of the ALPINE Body.....	35
9.1 Onboard Firmware for Real-Time Operation.....	35
9.2 ESP-NOW Dongle: Firmware and ROS2 Integration.....	36
10 Conclusions.....	37
11 Bibliography.....	38

Abstract

This thesis presents the development of a mechatronic winch system and A cLimb ing robot for oPerations in Mountain Environments (ALPINE). The project stems from the need for a compact, lightweight, and efficient jumping platform capable of operating on inclined surfaces, where precise and real-time motion control are essential for safe and reliable operation. The study begins with an analysis of the state of the art and the definition of system requirements, followed by a simulation phase to estimate key parameters and guide design choices. The mechanical implementation is divided into two main components: the winch system and the ALPINE body. The first step focused on the CAD based design of the winch, including motor sizing, a variable transmission mechanism, and the integration of a custom rope solution optimized for strength and electrical conductivity. In parallel, a lightweight CAD designed frame was developed for the jumping robot, capable of performing jumps according to the defined requirements. Particular attention was given to balancing structural compactness with load-bearing capability, validated through extensive 3D-printed component testing to ensure robustness under dynamic loads while remaining lightweight and practical for integration. On the software side, the work focused on low-level programming for the embedded controllers, including the development of real-time drivers, actuator interfaces and a ROS2 based communication layer. This software framework integrates advanced odometry and a friction estimation module, enabling accurate winch control and improving overall system interoperability. The integrated hardware-software prototype was validated through targeted experiments that demonstrated the effectiveness of the propulsion and orientation-control strategies and confirmed the potential of the system for real-world climbing operations. The thesis concludes with a discussion of the limitations identified during testing and outlines possible directions for future improvements, particularly in terms of robustness, so that the system can evolve beyond a prototype into a reliable climbing platform.

1 Introduction

1.1 Project Motivation

Mountain environments represent some of the most challenging and hazardous operational scenarios where humans are often required to perform maintenance tasks such as removing unstable rocks, trimming dangerous vegetation, or installing protective netting to mitigate landslides. These activities are not only physically demanding but also inherently risky due to factors such as steep slopes, unstable terrain, unpredictable weather conditions and limited accessibility. The growing awareness of climate change and its effects such as increased frequency of landslides, rockfalls and erosion further intensifies the need for continuous and safe monitoring of these environments. Traditionally, such interventions are performed by specialized teams using ropes, scaffolds, or helicopters, exposing workers to significant danger and incurring high logistical costs. These limitations have prompted increasing interest in robotic solutions that can automate or assist in these operations. Among the various robotic strategies investigated in recent years, tethered climbing robots have emerged as a particularly promising solution; as demonstrated in recent work such as the ALPINE platform [7]. These systems exploit rope tension to achieve enhanced stability and controlled movement, allowing navigation on steep or vertical surfaces with greater safety and energy efficiency. Compared to adhesive or legged robots which often struggle with terrain irregularities, limited grip, and low payload capacity tethered systems offer higher force capabilities and the ability to remain stationary for long periods of time with minimal energy consumption. Their structure also allows integration of heavier tools or sensors, making them well-suited for operations such as inspection, anchoring, and maintenance of complex mountain environments.

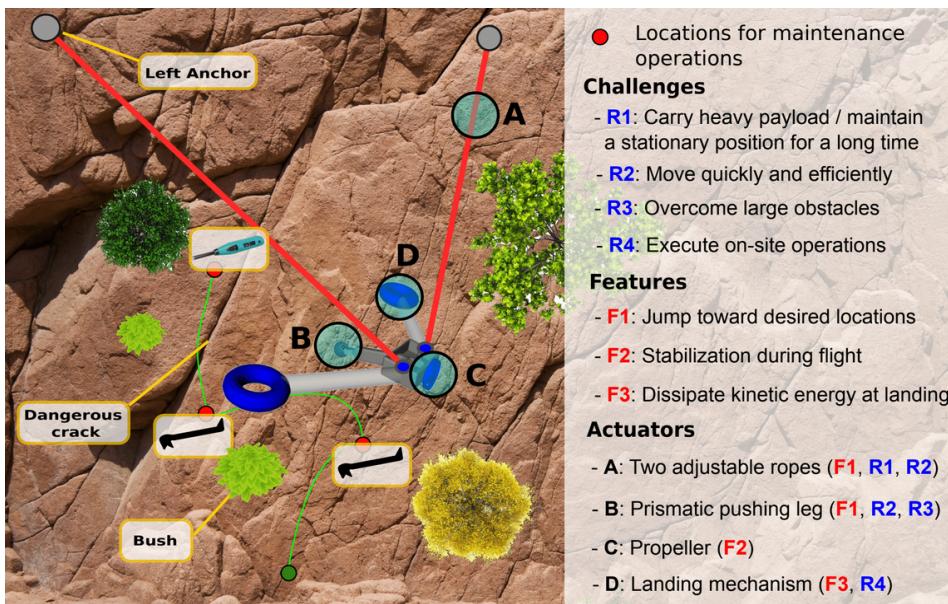


Figure 1: ALPINE climbing robot concept

and software design of a modular winch unit that provides reliable traction, precise control, and **ROS2** based telemetry (Robot Operating System). The goal is to contribute to the development of field ready climbing platforms that can operate semi autonomously in complex and dangerous mountain terrains, enhancing both the safety and efficiency of intervention strategies.

This thesis is motivated by the need to design a practical, lightweight, and reliable mechatronic winch-based climbing system capable of proving the feasibility of the ALPINE Climbing Robot deployment.

Drawing inspiration from the recent ALPINE robot project [7], the proposed system focuses on the mechanical, electrical,

1.2 State of the Art

In recent decades, the development of climbing robots has followed multiple design philosophies, each with distinct advantages and limitations depending on the target environment and application. A first family comprises bio-inspired walking climbing robots, such as Stickybot [14], which use adhesive structures to grip vertical surfaces. These robots draw inspiration from animals like geckos and wasps, offering mobility over smooth surfaces. However, they suffer from low payload capacity, are highly sensitive to terrain conditions, and lack the ability to overcome large obstacles, making them unsuitable for irregular or wet environments where grip reliability is critical. A second family consists of hybrid flying-climbing robots, which combine the agility of aerial drones with limited surface interaction capabilities. Examples include SCAMP [37], which uses propellers to adhere to walls, and CAROS (Climbing Aerial RObot System) [13], a quad-rotor equipped with wheels for wall transitions. More advanced variants employ propeller-wheeled locomotion to maintain proximity to walls [38], [2] and to perform inspection tasks such as crack detection [41] and hammer-based structural testing [26]. Despite their versatility, these robots are limited by payload and energy capacity, restricting their suitability for prolonged heavy-duty tasks. A third and increasingly promising category includes tethered or rope-based climbing robots, developed for use cases ranging from sports broadcasting [42] to planetary exploration [33]. These robots navigate by controlling the tension and length of one or more cables, allowing efficient traversal of steep and vertical surfaces. In most cases, the cable anchor points are fixed [15], and motion control is achieved through optimization-based strategies that address the system's nonlinear and constrained dynamics. Some rope-based systems, such as Aerones [1], have reached high technology readiness levels and are used in wind turbine maintenance, although only on smooth and predictable surfaces. Other notable examples include the Axel/DuAxel [24], a planetary rover that rappels down cliffs, and aerial tethered systems [35] that navigate glacial terrains by swinging between anchor points. Recent advances, such as the CLIO platform [8] and bio-inspired dragline robots [40], have introduced high-speed navigation through active winching mechanisms, offering advantages over conventional adhesive or gait-based robots [7]–[8]. However, these systems still face challenges in autonomy, particularly when executing complex tasks in unstructured environments.

1.3 Requirements and Objectives of the Project

The project aims to design a climbing system capable of meeting operational needs in alpine environments, which current solutions struggle to address. The system is based on a winch driven, rope assisted climbing robot and is designed to fulfill the following key requirements:

R1: Support heavy payloads and maintain a stable position on vertical surfaces for long periods without consuming energy.

R2: Move quickly and efficiently using combined jumping and lateral/vertical movement strategies.

R3: Overcome significant obstacles along the

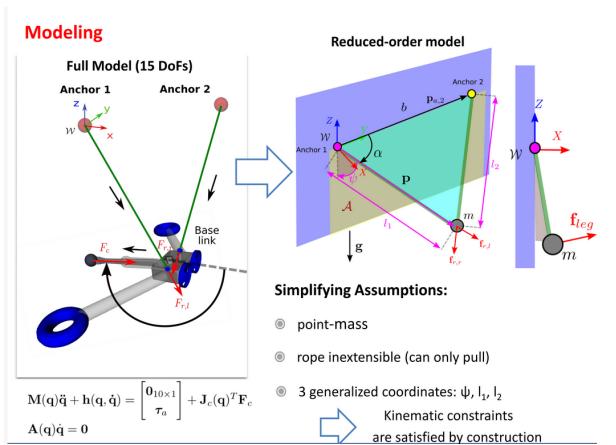


Figure 2: Comparison between the fully detailed dynamic model (15 DoFs) used in simulation and the reduced-order model adopted for control, planning, and odometry.

path, such as bushes and rock formations, by remaining detached from the wall during motion.

R4: Perform complex maintenance operations autonomously or semi-autonomously, using a controlled landing mechanism that dissipates kinetic energy and ensures stable contact.

R5: As a first prototype, the robot must weigh at most 5 kg to ensure safe handling and testing.

This thesis focuses on the mechatronic design of the ALPINE climbing robot, beginning with the estimation of core physical quantities derived from simulated omnidirectional key jumps. The worst case scenario was then identified to determine the critical parameters for construction. The work also addresses the control of all actuation systems and the integration of ROS2 to ensure ease of operation.

1.4 ALPINE Robot Overview

The conceptual design to meet the required movements of the proposed ALPINE robot, illustrated above in **Figure 2** through a schematic representation of the main force interactions, led us to devise a system architecture organized into two primary subsystems:

Two Symmetrical Winch Assemblies:

Two symmetrical winches are employed to generate the tensile forces required for rope-based locomotion during the robot's flight phase. Each winch comprises:

- A **brushless motor and winch**: the main winch motor connected to the winch via a **timing belt transmission**. The drum winds and unwinds the rope, which not only supports the full load of the robot but also serves as a conduit for power delivery to downstream systems (see Chapter 2.5).
- A **synchronous roller**, tasked with accurately monitoring rope displacement and enabling dynamic computation of the motor's gear ratio to solve the variable gear transmission problem (see Chapter 2.4).
- A **linear actuator** operating a **mechanical brake**, ensuring zero energy consumption during stationary phases and providing a fail-safe mechanism for secure operations.

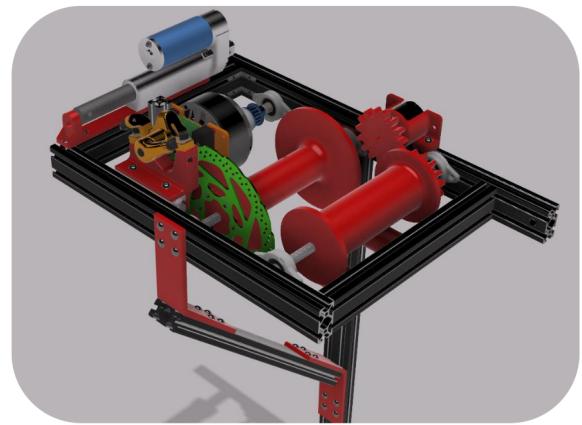


Figure 3: CAD drawing of the winch assembly

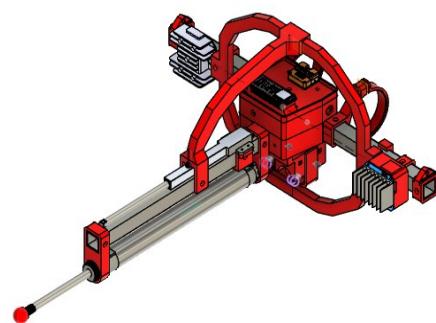


Figure 4: CAD drawing of the ALPINE body

ALPINE body Robotic Platform

The central robotic body is designed to accommodate future payloads and integrate actuation systems for jumping and stabilization. It includes:

- A **propulsion system**: based on a prismatic actuator, or "leg" capable of delivering the impulsive force needed to detach the robot from the wall. In addition to providing thrust, this actuator also functions as a compliant element,

dissipating impact energy mitigating shocks during landing.

- A **attitude control system** incorporates a **pitch-adjustment linear actuator**, which

regulates the orientation of the leg to ensure that it remains perpendicular to the wall surface during landing. This improves stability and reduces impact stresses on the structure. To support state estimation, **two inertial measurement units** (IMUs) are employed: one dedicated to determining the overall orientation of the robot, and the other to track the spatial orientation of the rope. Furthermore, the architecture is designed to **optionally** include **additional orientation control devices**, such as propellers, reaction wheels, gyroscopes, or cold gas thrusters, which can be used to enhance trajectory stability during flight.

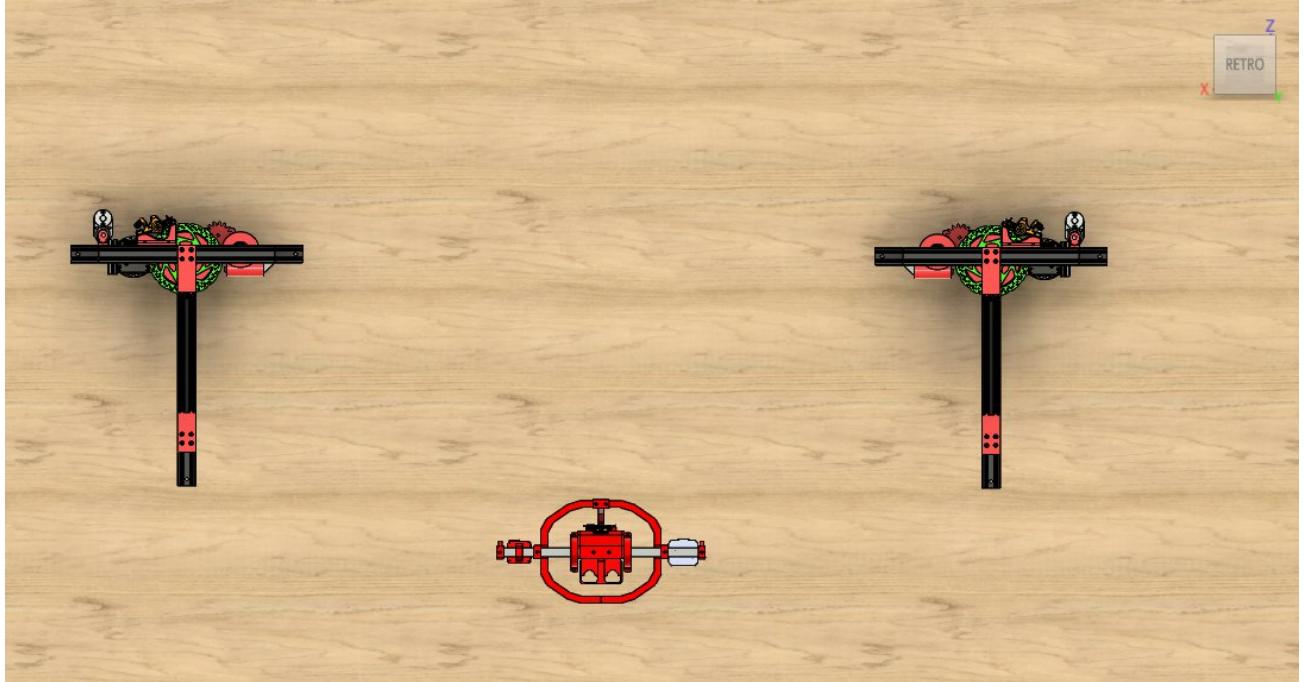


Figure 5: Full assembly mounted on the wall, cables not rendered

A detailed description of each subsystem, along with the design rationale and performance considerations, will be provided in the subsequent chapters of this work.

2 Mechanical Implementation of the Winch System

2.1 Reference Jumps

To initiate the design process, the first step is the definition of key parameters. For the ALPINE robot, this involves estimating the force and torque necessary to achieve significant jumps with a **5 kg prototype**, chosen as the initial design target. To ensure robustness and provide margin for future enhancements, such as the integration of advanced sensors or drilling apparatus, a **worst-case scenario** approach was adopted, considering a **total robot mass of 10 kg**. The required parameters were obtained using the simulator developed for this robot by Professor Focchi in MATLAB [9].

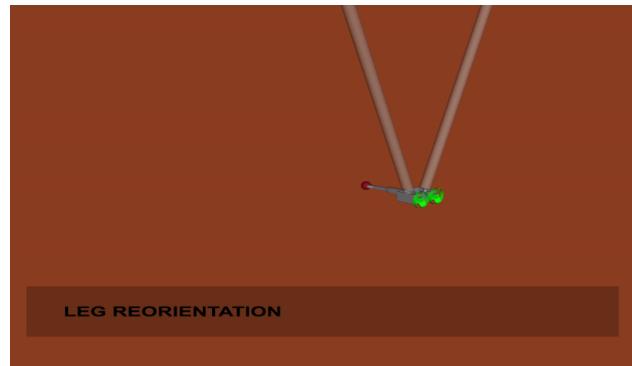


Figure 6: Simulation Results of jumps in multiple directions
considerations, the trajectory corresponding to the third line in **Table 1** was selected.

Lift-off and landing points are expressed as vectors (x,y,z) in meters, defined with respect to a reference frame whose origin is located at the top-left winch (Section 6.2, Diagram 1).

1	jumps	Start point	end point	5kg				
				consumed energy [J]	time [s]	Average system power [w]	max power right [w]	Max power left [w]
4	10.5, 2.5, -6	0.5, 4, -4		250	1.21	206.611570247934	453	181
5	20.5, 2.5, -10	0.5, 2.5, -2		503	1.67	301.197604790419	314	290
6	30.5, 1, -10	0.5, 4, -2		545	1.43	381.118881118881	437	230
7	40.5, 1, -5	0.5, 4, -5		212	1.3	163.076923076923	338	171
8	50.5, 2.5, -2	0.5, 2.5, -10		208	1.73	120.231213872832	179	160
10								
11	jumps	Start point	end point	10 kg				
				consumed energy [J]	Time [s]	Average system power [w]	max power right [w]	Max power left [w]
13	10.5, 2.5, -6	0.5, 4, -4		214	1.65	129.69696969697	254	65
14	20.5, 2.5, -10	0.5, 2.5, -2		963	1.65	583.636363636364	608	591
15	30.5, 1, -10	0.5, 4, -2		989	1.65	599.393939393939	696	431
16	40.5, 1, -5	0.5, 4, -5		132	1.51	87.4172185430464	132	39
17	50.5, 2.5, -2	0.5, 2.5, -10		472	1.75	269.714285714286	342	344

Table 1: multiple jumps results from simulation

2.2 CAD Design

The mechanical structure of the winch system was designed using *Autodesk Fusion 360*, with the dual objective of achieving structural strength and portability. The frame is constructed from standardized 20×40 mm extruded aluminum profiles (commonly referred to as **2040** profiles), as illustrated in Figure 7 (black contour in side view). Aluminum was chosen for its high strength to weight ratio, which reduces the overall mass of the system and facilitates safe handling and installation at elevated positions. The 2040 profiles were selected over 20×20 mm

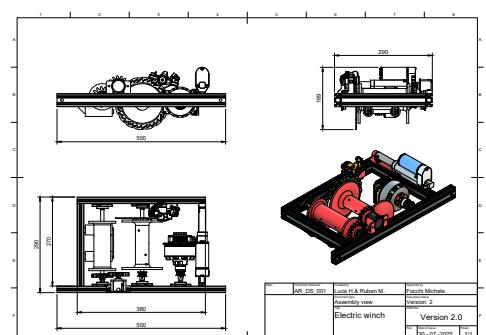


Figure 7: orthographic projections of the winch:
front, top, and side views

alternatives because their greater width provides two longitudinal slots on each side. This configuration enables the attachment of **90° L-shaped brackets** at every corner, securing the frame with eight independent connections across the rectangular geometry. The result is a redundant and highly rigid structure, ensuring both mechanical robustness and safety. Furthermore, the compact rectangular arrangement of the frame optimizes the system for transport and facilitates installation in elevated constrained environments.

2.2.1 Mechanical Components of the Winch

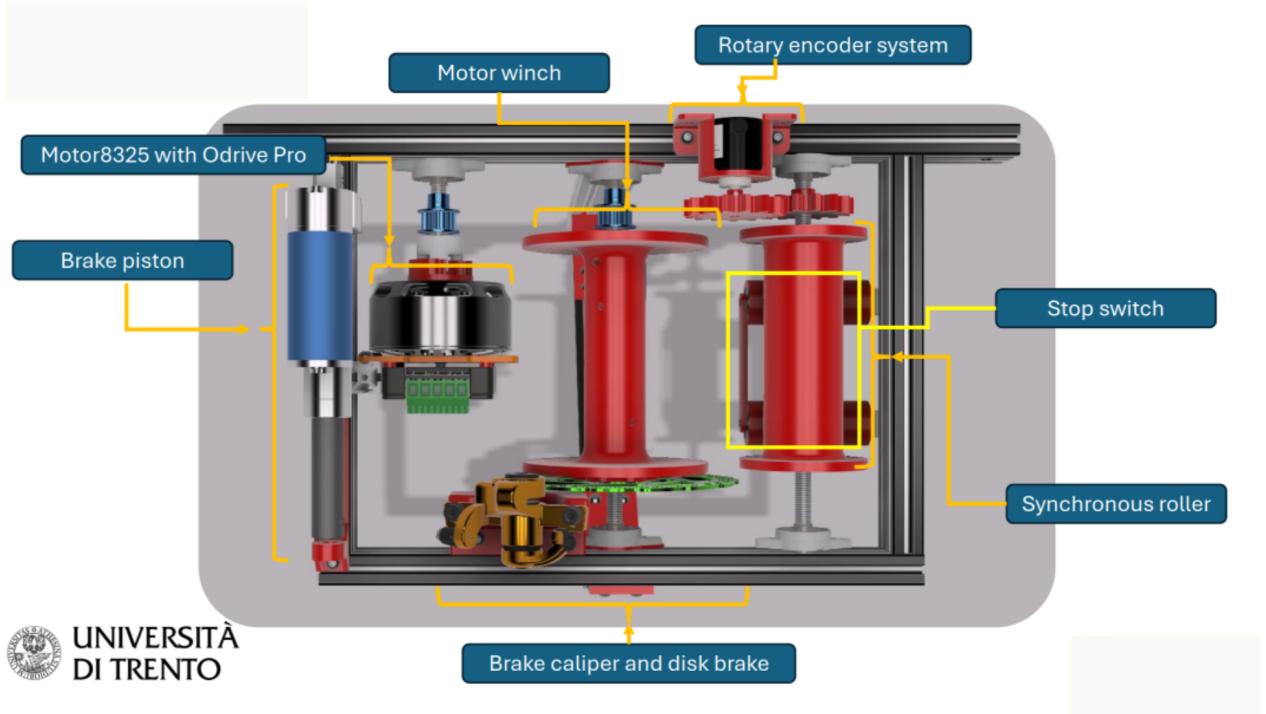


Figure 8: winch components

The winch system is primarily composed of the following active elements:

- **BLDC Motor with Absolute Encoder:** The winch actuation is provided by a brushless DC (**BLDC**) motor (**ODrive** model **M8325S, 100 KV**, datasheet [27]), paired with an **ODrive Pro** [28] motor driver for closed-loop control. The motor is equipped with its **high-resolution onboard absolute encoder** datasheet [29], enabling precise torque control and position feedback. To manage regenerative braking, excess energy is dissipated through a dedicated **Regeneron Clamp** module [30] (see Figure 15). Torque is transmitted from the motor to the main motor winch via a standard **HTD-5M timing belt drive**. This solution offers flexibility in motor placement, facilitating a compact overall layout and allows for adjustment of the gear ratio even if belt tensioning is required. For the present design, a **1:1 gear ratio** was selected, as the winch must be capable of operating at high rotational speeds while simultaneously delivering substantial torque (see Section 2.3). A direct-drive configuration therefore represents the optimal compromise between speed capability and torque delivery. Compared to gear or chain transmissions, a timing belt ensures efficient and quiet power transfer while reducing wear and maintenance requirements, making it particularly suitable for this application.
- **Braking System:** A critical safety feature of the winch is the integrated braking system. A

disk brake is rigidly attached to the main motor winch (Fig 8), enabling complete immobilization of the rope, even in the event of a belt or transmission failure. The brake is actuated by a mechanical caliper driven by a **linear actuator** called **brake piston or just brake** (Fig. 8 left) connected through a tensioned wire. Notably, the actuator is **non-back-drivable** and capable of exerting a holding force of 1500 N. This ensures that the system remains locked even under heavy loads and in the event of a power failure; an essential property that satisfies the system requirement of maintaining the robot in a stationary position without continuous power consumption, substantially decreasing constant power expenditure during long-lasting maintenance operations (see Fig. 8, bottom-center and left).

- **Synchronous Roller for Physical Rope Measurement:** Mounted at the top of the winch structure, the synchronous roller fulfills both mechanical and sensing functions. Structurally, it supports part of the vertical rope load by redirecting tension through its shaft and bearings attached to the frame. This reduces the load transmitted directly to the motor winch thereby lowering the torsional demand on the motor shaft and thus on the 3D printed roller. From a sensing perspective, the roller is instrumented with a 600 pulses per revolution (PPR) **rotary incremental encoder** (Fig 8), which directly measures angular displacement of the shaft. Rope displacement is then obtained by multiplying this angular displacement by the roller circumference while rope velocity is derived as the time derivative of displacement (see Section 2.4). The encoder is coupled to the roller shaft through a gear set, making it possible to increase or decrease the effective resolution of the measurement. For example, by selecting a gear ratio greater than **1:2** (roller:encoder), a single turn of the roller produces double the turns of the encoder, thereby increasing the effective resolution of the measurement. In the present design however, a **1:1** gear ratio was selected for simplicity and reliability. (Figure 8, top-right) This configuration ensures measurement accuracy that is independent of the rope winding pattern on the main winch drum. Multi-layer winding on a drum changes the effective drum radius and introduces non-linearity in displacement estimation. By employing a dedicated measurement roller, these effects are eliminated, removing the need for a complex spooling or guiding mechanism and thereby simplifying both the design and the control strategy as discussed in Section 2.4.
- **Stop switch.** The winch system is equipped with a mechanical end stop (Fig. 8, right), designed to prevent over-retraction of the rope. This stop consists of a buoy-like structure (Fig. 11) that becomes physically blocked between the two horizontal rollers and the synchronous roller when the rope is fully retracted. The mechanism serves a dual purpose: it enables calibration, since the buoy corresponds to a known reference position, and it provides a safety function by mechanically preventing the rope from winding further. In this way in case of an uncontrolled behavior or malfunction, it avoids potential damage to the payload or to the winch system itself.
- **Torque Control and Variable Radius Compensation:** During jump maneuvers, the robot operates in torque-control mode. To accurately translate the desired rope tension into the corresponding motor torque, it is necessary to account for the variable effective radius of the winch drum as the rope wraps and unwraps. This compensation is implemented through real-time calculations (see Section 2.4) based on a geometric model of the drum, enabling consistent control performance even in the presence of non-uniform winding.

2.3 Sizing of the Electrical Motor

Based on the simulation parameters reported in **Table 1** (class: 10 kg, row 3), we can estimate the motor velocity and torque requirements. At this stage, friction is not included in the raw power calculation; it will be modeled later in **Chapter 5** to optimize trajectory tracking. It is worth noting that the robot mass has already been conservatively over sized and the analysis considers the worst-case operating scenario. Under these conditions, the influence of friction on the raw power estimate can be considered negligible.

Data:

Robot mass: $m = 10 \text{ kg}$

Jump reference (m): $p_0 = [0.5, 1, -10] \rightarrow p_f = [0.5, 4, -2]$

Displacement: $\Delta p = [0, 3, 8] \text{ m}$

Energy consumed: $E = 1000 \text{ J}$

Time: $t = 1.65 \text{ s}$

Average power: $P_{\text{avg}} = 600 \text{ W}$

Peak power: $P_{\text{peak}} = 700 \text{ W}$

Max rope velocity: $v_{\text{rope,max}} = 7.8 \text{ m/s} \approx 8 \text{ m/s}$

Max rope acceleration: $a_{\text{rope,max}} = 16 \text{ m/s}^2$

Winch drum diameter: $d = 0.05 \text{ m}$

Drum radius: $r = 0.025 \text{ m}$

Maximum angular speed of the drum:

$$\omega = \frac{v_{\text{rope,max}}}{r} = \frac{8}{0.025} = 320 \text{ rad/s} \quad (2.1)$$

Converted to revolutions per minute:

$$n = \omega \cdot \frac{60}{2\pi} = 320 \cdot \frac{60}{2\pi} \approx 3056 \text{ RPM} \approx 3000 \text{ RPM} \quad (2.2)$$

Force and torque on the drum (dead lifting):

$$F = m(g + a_{\text{rope,max}}) = 10(9.81 + 16) = 258.1 \text{ N} \quad (2.3)$$

$$\tau = Fr = 258.1 \times 0.025 = 6.45 \text{ Nm} \approx 6.5 \text{ Nm} \quad (2.4)$$

Based on the results of the jump simulations, we determined that the winch system requires a motor capable of delivering at least **6.5 Nm** of torque, reaching speeds up to **3000 RPM**, and handling at least **700 W** of power. After evaluating several options, we selected the **M8325S 100 KV** motor (datasheet[27]) in combination with the **ODrive Pro** motor driver (datasheet[28]) as part of the **ODrive** motor and controller ecosystem [31], which best satisfied both the performance requirements and the integration constraints of the system.

This configuration provides several key advantages:

- **High continuous torque output**, ensuring reliable performance under load with an adequate safety margin.
- **Precise control resolution and fast response**, supported by advanced frequency modulation capabilities of the ODrive platform.
- **Regenerative braking support**, with the ability to connect an external regeneration clamp and shunt resistor (see schematic in Chapter 3) it ensures that excess energy is safely dissipated during deceleration or downward jumps. In these conditions, gravity performs negative work on the system, causing the motor to act as a generator. The

resulting energy is directed into the shunt resistor and converted into heat via the **Joule effect**, thereby preventing over voltages on the DC bus and avoiding motor overheating.

In addition, specific control strategies were implemented, such as the use of a mechanical caliper brake for safety and to provide a reference for resetting encoder readings, as well as ensuring

that peak torque demands can be met during dynamic maneuvers. This approach guarantees robust and precise control during critical phases such as jumping and landing while also improving thermal and energy management.

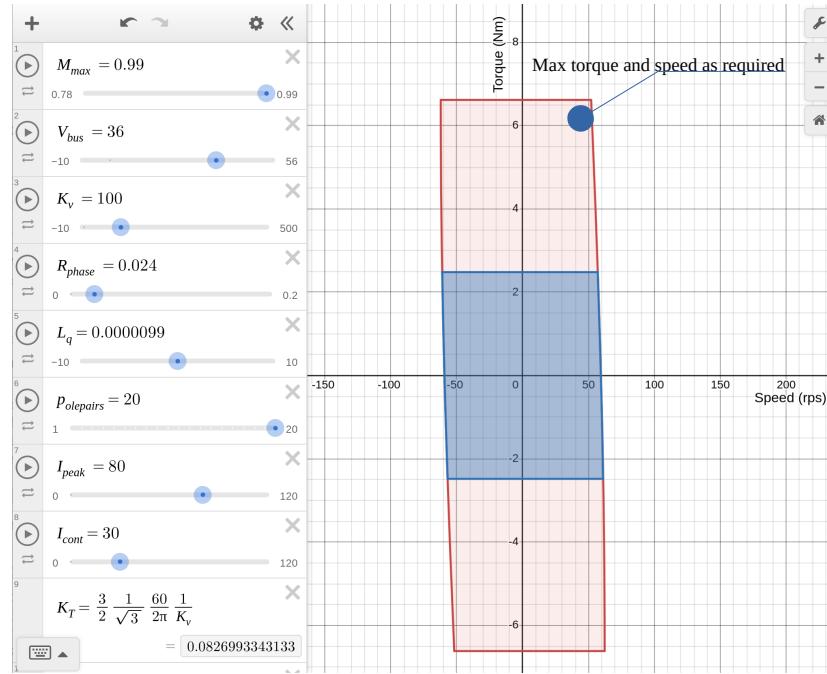


Figure 9: speed/torque curve

Finally, using the official ODrive motor in conjunction with its official online Demos motor parameter calculator [3] we verified that the selected motor-controller pair meets the required torque, speed, and power specifications for our intended application (Fig. 9).

2.4 Variable Transmission Calculations

In order to eliminate the need for mechanically complex spooling systems, such as the Lebus grooved drum method [16], rope displacement is measured through a dedicated synchronous roller. This approach avoids reliance on the winding pattern of the main motor winch drum, which would otherwise introduce a time-varying effective radius. This approach ensures that measurement accuracy remains unaffected by the layering of the rope on the drum. Multi-layer winding can introduce significant errors and **non-linearities** in displacement estimation due to variations in the effective drum radius. By delegating the measurement task to a dedicated roller, these effects are removed, thereby simplifying both the mechanical design and the control strategy.

The calculation of the effective transmission ratio must account for the winding geometry of the motor-driven winch drum. The synchronous roller is directly coupled to an incremental encoder through a **1:1 gear ratio**, while the rope is wound around the motor drum with a nominal radius R_3 (see Fig 10). As successive layers accumulate, an additional thickness x , dependent on the current winding state effectively increases the drum radius and modifies the real transmission ratio. The motor winch drum itself is connected to the motor shaft via a **1:1 belt drive** (see Chapter 2.2.1), thus called motor winch, so any change in the effective drum radius directly influences the relationship between motor angular velocity and rope linear velocity. In industrial winch applications, this challenge is typically addressed through complex spooling mechanisms or direct measurement systems, such as capacitive or optical thickness sensors, which increase both cost and design complexity. In contrast, the use of a synchronous roller provides a robust and efficient means of indirectly estimating the rope layer thickness without resorting to

additional sensing hardware, highlighting one of the methodological contributions of the present work. A schematic figure of the transmission is shown in Figure 10.

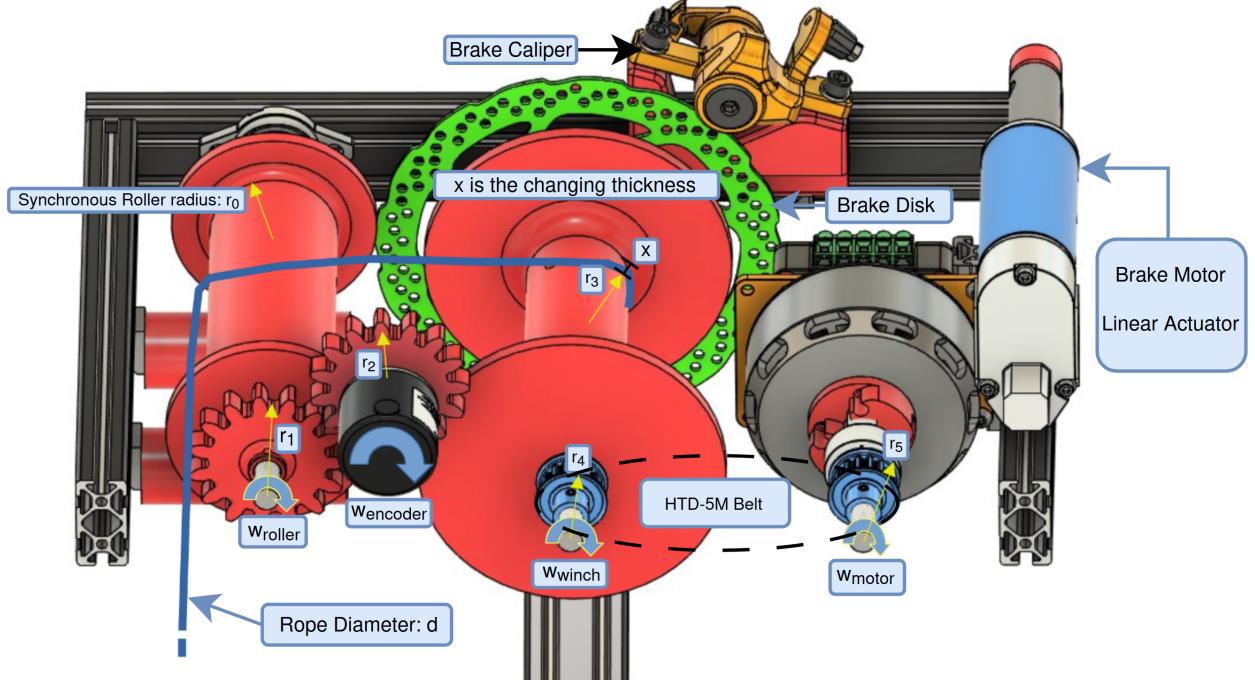


Figure 10: Schematic of the transmission system in the winch assembly

To see the full derivation of x accounting for every gear ratio, refer to [21]

The quantities of interest are the rope's linear velocity v_{rope} , and the rope tension F_{rope} , both aligned with the rope center line. The roller angular velocity is obtained from the encoder as:

$$\omega_{\text{roller}} = \omega_{\text{encoder}} \cdot \frac{r_2}{r_1} \quad (2.5)$$

Thus, the rope velocity is:

$$v_{\text{rope}} = \omega_{\text{roller}} \cdot \left(r_0 + \frac{d}{2} \right) = \omega_{\text{encoder}} \cdot \frac{r_2}{r_1} \cdot \left(r_0 + \frac{d}{2} \right) \quad (2.6)$$

To compute rope force, we use the motor torque τ_{motor} , estimated from the ODrive Pro driver. The transmission ratio between motor and roller is defined as:

$$G = \frac{\omega_{\text{roller}}}{\omega_{\text{motor}}} = \frac{r_5}{r_0} \quad (2.7)$$

Therefore, the force acting on the rope is:

$$F_{\text{rope}} = \frac{\tau_{\text{motor}}}{r_5} \quad (2.8)$$

Since $r_5 = G \cdot r_0$ by Eq.(2.7), substituting into Eq.(2.8) gives:

$$F_{\text{rope}} = \frac{\tau_{\text{motor}}}{G \cdot r_0} \quad (2.9)$$

2.5 Rope Design and Testing

One of the fundamental design choices for the ALPINE robot is the decision to have a centralized design of power supply, where the power of the robot directly travels through the support ropes rather than carrying batteries onboard. This approach significantly reduces the robot's inertia, improving agility and stability during jumping maneuvers while also increasing payload capacity for sensors and tools. To achieve this, the support ropes must not only withstand high tensile loads but also conduct substantial electrical current from the winch power supply (see chapter 3) to the robot. This requirement led to the need for conductive ropes. Initially, a **zinc-coated** steel wire was selected due to its excellent tensile strength and good electrical conductivity. However, further evaluation revealed that this material was unsuitable for high-current applications: zinc-



Figure 11: Custom hybrid cable. The outer PET braided sleeve is shown in black, while the inner silicone-insulated copper conductor wire is red colored. The top red buoy-like component serves as a mechanical end stop.

coated steel is approximately **42 times less conductive than copper**, resulting in excessive voltage drop and heat dissipation. Considering that the robot's future capabilities will include active stabilization, propeller-assisted jumps, and other energy-intensive subsystems, copper emerged as the only viable conductor material. Nevertheless, pure copper cables present a mechanical challenge in this context. Although copper offers outstanding conductivity, standard copper wires lack the required tensile strength and wear resistance to endure dynamic shear loads and continuous bending cycles during winch operation. To resolve this, a custom hybrid cable was developed and later patented. This cable integrates a flexible **22 AWG** (American Wire Gauge) silicone-insulated copper conductor (rated for continuous currents up to **30 A**) inside a **PET** (polyethylene terephthalate) expandable braided sleeve. In this configuration:

- The copper core ensures high electrical conductivity with minimal stiffness.
- The PET braided sleeve provides the primary load-bearing capacity, offering excellent abrasion resistance, tensile strength, and flexibility.

When installed, the hybrid cable is wound onto the motor winch drum, with the copper conductor terminating in the **aluminum rotating shaft** designed to deliver electrical power from the winch system to the robot while keeping it in place. The PET braided sleeve is wound on the winch drum too so that all tensile forces are supported by the outer braid, leaving the inner copper conductor stress-free and suspended. Illustrated in Figure 12, the aluminum transfer shaft is mechanically supported by the frame via precision **ball bearings**, while being **electrically insulated** with a custom 3D-printed dielectric shield. This insulation prevents parasitic currents from propagating through the robot's structural frame, thereby safeguarding sensitive electronics and ensuring safe power transfer. By separating the mechanical and electrical load paths, the design reduces conductor fatigue, extends cable service



Figure 12: electrically insulated winch shaft

life and ensures reliable power delivery in harsh alpine environments. Notably, the introduction of an electrically insulated rotating shaft as part of the power transmission system and the custom made conductive rope constitutes a specific design solution developed in this work by reducing the weight, thus the inertia, of the ALPINE body by eliminating the need of onboard heavy batteries thereby enabling agile jumps.

2.5.1 Load Testing

To validate the load-bearing capacity of the custom cable, incremental weights were progressively applied until reaching the design target of **500 N** of force. The cable successfully supported the full load without exhibiting any measurable electrical degradation, indicating no internal damage or conductor strain. This confirms the mechanical robustness and electrical reliability of the hybrid construction under static tensile loads.

2.5.2 Creep Testing

Creep resistance was tested by suspending a 1-meter section of the cable under a constant **10 kg** load over various durations: **6 h, 12 h, 24 h, and 36 h**. Across all intervals, the total deformation did not exceed **1 mm**, which is within the resolution limit of the measuring instrument. Only a minor initial settling of the PET sleeve around the inner conductor was observed, confirming excellent long-term dimensional stability and minimal plastic deformation over time.

3 Electrical Implementation of the Winch System

3.1 Electrical Schematic

The electrical system of the winch is organized with a clear separation between high-power and low-power domains, ensuring safe operation under all conditions, including complete power failure. As illustrated in Figure 13, each winch is equipped with its own electrical panel, which connects to the 36 V high-power bus. This voltage was chosen as the nominal operating voltage. From this panel, power is distributed via dedicated wiring to the winch, where the motor and motor driver is located with the brake piston. Within the electrical panel, the voltage is stepped down to supply low-voltage applications, such as the micro-controller, the brake system, and the various feedback components. A more in depth explanation of:

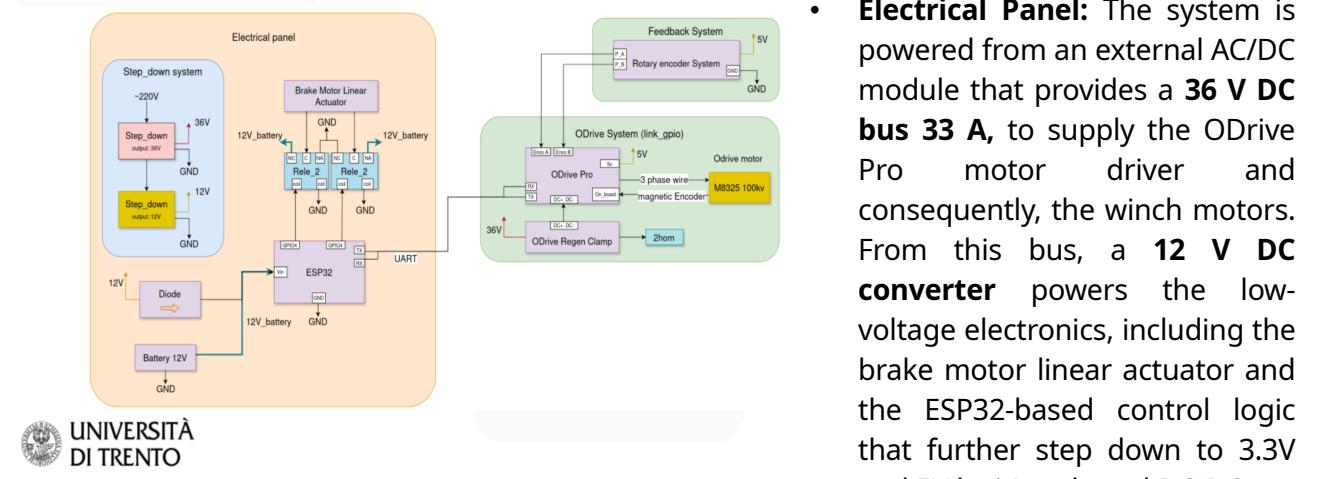
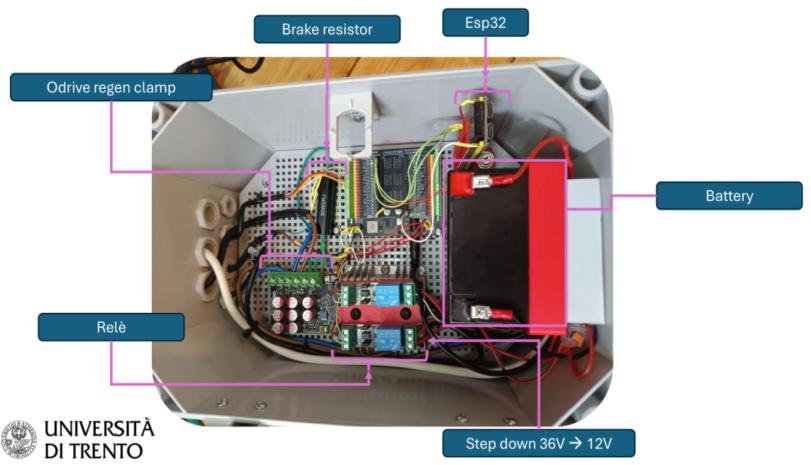


Figure 13: logical electrical diagram

a **12 V 5 Ah backup lead acid battery**, dedicated exclusively to emergency brake operation. Importantly, a **diode** is placed between the DC-DC converter and the battery (Fig 15). This prevents reverse current from flowing into the converter when the system is powered off, thereby avoiding slow overnight discharge of the emergency battery. Critically, this architecture guarantees that the brake can still be activated even in the event of a complete power failure as the backup battery directly powers the motor brake linear actuator whenever the primary supply is unavailable.

- Feedback System:** The low-power domain is managed by an **ESP32 micro-controller** located in the electrical panel. The ESP32 provides regulated **5 V and 3.3 V rails** for peripheral devices and serves as the logic bridge between the ODrive Pro and the external computer through a **ROS2 interface** (see [Figure 14: Electrical panel components](#))

The same 12 V line also charges



Chapter 4). Communication between the ESP32 and the ODrive Pro occurs via **UART at 2 Mbaud**, a choice made to minimize latency and guarantee fast response in real-time control. The ODrive Pro receives the **36 V DC bus** and drives the BLDC motor, while processing feedback from two independent encoders: the **driver's onboard absolute magnetic encoder** and the **synchronous roller encoder**, which together provide precise monitoring of both motor shaft dynamics and rope displacement. To safely handle regenerative energy during deceleration and rope retraction, the high-voltage bus passes through an **ODrive Regen Clamp** [30] equipped with a shunt resistor of **2 Ω**, which dissipates excess energy and prevents over voltage on the DC bus.

- **Brake Motor:** The brake is implemented as a **12 V DC linear actuator** with integrated limit switches, mechanically engaging a stationary brake on the motor winch. Electrically, the actuator is controlled by a **relay-based H-bridge**, constructed from two relays wired in parallel for synchronous switching. For simplicity and robustness, the relay pin pairs are driven directly by the ESP32 through **pin 4**, which ensures both directions of actuation are controlled simultaneously. In the absence of control signals or during a fault, the relays revert to their default state, causing the brake to engage automatically. Combined with the backup battery, this ensures that the brake will always engage, even in the case of a total power outage, thereby preventing uncontrolled rope release or descent.

3.2 Wiring of the Winch

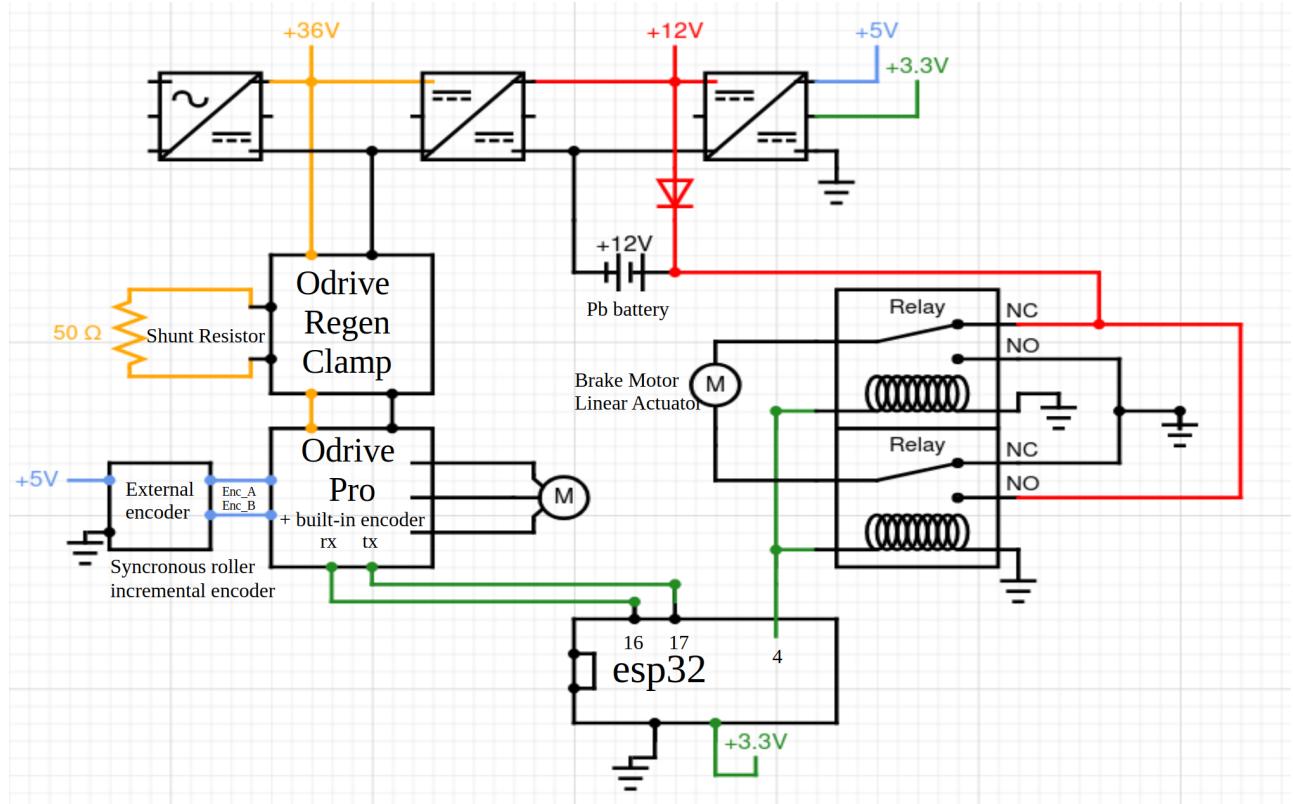


Figure 15: Electrical connections of the winch system

4 Software Architecture

4.1 Communication Network

The communication network of the ALPINE Climbing Robot is organized around a central host computer, which serves as the main interface for all ROS2 nodes and executes both the offline optimization for calculating optimal jump trajectories and optimal control algorithm both developed by my colleague Ruben Malacarne [17] specifically for this project. **Three USB** connections link the host to the embedded subsystems: the left winch, the right winch, and the **ESP32 “Dongle.”** The winch controllers, based on **ESP32** micro controllers, receive high-level commands from the host and relay them via **UART** to the **ODrive Pro** motor drivers. In parallel, they continuously stream telemetry at frequencies of **200 Hz**, ensuring real-time monitoring of motor and brake status. The dongle, by contrast, acts purely as a communication bridge. Connected to the host via **USB**, it forwards commands over the **ESP-NOW** wireless protocol to the ALPINE Body, and likewise relays feedback from the robot back to the computer. This effectively makes it a dedicated **USB-to-radio** interface. From a bandwidth perspective, the three **USB** connections operate without contention, and even at their maximum speed of **2 Mbit/s** they remain well within the capabilities of the **USB 1.1** standard. The true limitation of the communication bandwidth lies in the wireless link; the **ESP-NOW** bridge constrains bidirectional communication with the jumping robot to approximately **100 Hz**. This aspect, along with its implications for control performance, is analyzed in detail in Chapter 9.

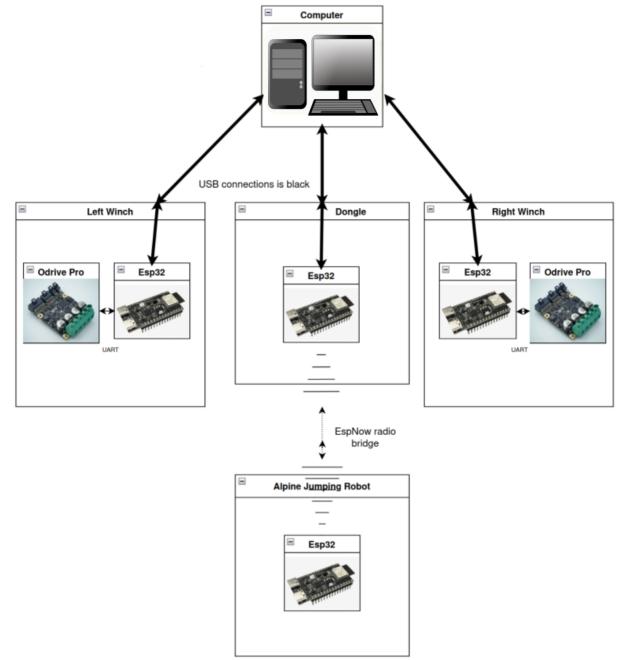


Figure 16: Communication network diagram. Images courtesy of Espressif [5], ODrive Robotics [28], and Pixabay [34].

4.2 Embedded Firmware of the Winch

The embedded firmware running on the **ESP32** micro-controller, has been chosen for its high clock speed (240 MHz) and its capability to expand connectivity through Wi-Fi or ESP-NOW [see Chapter 7]. The micro-controller serves as the real-time interface between the **ODrive Pro** motor driver and the **host computer**. Communication with the **ODrive** is established via **UART** at **2 Mbit/s** to minimize latency, while the **ESP32** also manages custom brake control logic, accessible through simple serial commands to engage, release, and monitor its state. The firmware operates in two complementary modes: a deterministic **200 Hz** telemetry loop and an on-demand commands which include motor commands and brake control logic. In the telemetry loop, a pre-compiled set of ODrive variables is queried and transmitted as a compact **CSV** record over **USB** at **1 Mbit/s**, enabling efficient, low-jitter data transfer compared to sequential ASCII requests. To maintain the **200 Hz** rate, all records must be processed within the **5 ms loop interval**; otherwise, real-time performance is degraded. As inputs, the micro-controller accepts individual textual commands from the host, such as brake control or direct **ODrive** read/write operations, which are slower but provide flexibility for diagnostics and asynchronous

configuration without the need to reprogram the firmware. This architecture ensures that the **ESP32** reliably manages time-critical telemetry and brake control functions, while remaining extensible for non-time-critical operations. The complete implementation, along with detailed documentation, is available in the dedicated project repository [20]. Notably the real-time firmware is the same for both winches left and right and they are differentiated by the serial **ID** of the micro-controller, set as parameter in the ROS interface as explained in the next section, 4.3.

4.3 Winch Telemetry Node

The ROS2 telemetry node of the winches constitutes the software interface between the low-level string-based serial communication of the ESP32 firmware and the higher-level control framework running on the host computer. Its primary role is to abstract the ASCII protocol used by the ODrive Pro and ESP32, exposing all functionalities through standardized ROS2 topics and services, thereby enabling seamless integration of the winch subsystems into the overall control architecture. The node establishes a USB serial link to each winch unit (default baud rate **1 Mbit/s**), with configurable parameters such as the serial port, polling rate, and logical identifier `<id>` (e.g., “sx” for the left and “dx” for the right winch). This identifier is used to dynamically generate unique namespaces (`.../winch/<id>/...`), allowing a single code base to operate multiple winches without duplication. At the low level, the ESP32 firmware streams compact CSV telemetry lines at 200 Hz, which the node republishes directly to `/winch/<id>/telemetry/csv` for logging, while also parsing them into a structured custom ROS2 message (`winchTelemetry`). In parallel, the node translates control targets into textual commands understood by the ESP32, exposing torque, velocity, and position references via the topics `/winch/<id>/target_torque`, `/winch/<id>/target_velocity`, and `/winch/<id>/target_position` but these commands directly drive the motor so they are rarely accessed by high-level control from these topics, instead, they are used internally by this node. This node also encapsulates all mode switching and brake actuation as asynchronous ROS2 services, such as `/set_idle`, `/set_closed_loop`, `/set_torque_mode`, `/set_velocity_mode`, `/set_position_mode`, and `/set_brake`; These asynchronous services set the driver mode and the brake position and depend on pre-compiled serial strings and provide confirmation feedback of the service calls through the CSV telemetry stream, ensuring deterministic and safe operation. Beyond raw motor parameters (turns, rad/s, Nm), the node implements real-time rope-centric abstraction, converting telemetry into physically meaningful variables required by higher-level readings. This enhancements includes: (1) the estimation of the **variable gear ratio** from motor data and synchronous roller velocities to capture the effect of changing drum effective radius during rope winding and unwinding (Chapter 2.4). (2) The **rope force computation** that compensates for both gear ratio variation and frictional effects, thereby enabling direct control of rope tension through the topic `/winch/<id>/cmd_rope_force` (see Section 5 for friction modeling). (3) Lastly **Rope kinematics**, namely rope length [m] and velocity [m/s], referenced to a configurable zero point set by a dedicated service `/set_initial_rope_position` for initialization purposes. By consolidating raw telemetry data acquisition, motor-to-rope conversion, and service-based control within a single ROS2 node, the system ensures modularity, eliminates duplication of kinematic transformations across nodes, and provides a consistent interface to upstream nodes. Consequently, advanced algorithms such as optimal control developed by Ruben Malacarne [17] can operate entirely in rope-referenced quantities (force, length, velocity), without the need to handle implementation details. In essence, the telemetry node acts as an abstraction layer, bridging low-level communication with control-ready signals for future researches and extensions. Implementation of the code at [19].

5 Friction Estimation

5.1 Nonlinear Friction Model

The starting point for the friction estimation process is the dynamic equation of the motor load system:

$$\tau_m = \tau_{fr}(\dot{\theta}) + I\ddot{\theta} + \tau_{gp} \quad (5.1)$$

Where:

- τ_m : measured motor torque (from telemetry)
- $\tau_{fr}(\dot{\theta})$: true friction torque (function of velocity)
- $I\ddot{\theta}$: inertial torque (acceleration component)
- τ_{gp} : gravity or passive torque (e.g., cables, bias)

For constant velocity trajectories in the horizontal plane, the inertial and gravitational contributions can be neglected, yielding:

$$\tau_m \approx \tau_{fr}(\dot{\theta}) \quad (5.2)$$

Friction Model:

To estimate the friction torque, we rely on the identification and compensation method proposed by Susanto [39], originally developed for robotic manipulators. This approach provides a structured way to capture the nonlinear effects of friction, including Coulomb and viscous components, and to integrate them into the control framework.

$$\tau_{fr}(\dot{\theta}) = \text{sgn}(\dot{\theta}) \left[\tau_c + (\tau_s - \tau_c) e^{-\frac{|\dot{\theta}|}{\dot{\theta}_{th}}} \right] + k_v \dot{\theta} \quad (5.3)$$

Which can be rewritten in parameter form:

$$\tau_{fr}(\dot{\theta}) = \text{sgn}(\dot{\theta}) \left[P_1 + P_2 e^{-\frac{|\dot{\theta}|}{P_3}} \right] + P_4 \dot{\theta} \quad (5.4)$$

Where:

- $P_1 = \tau_c$ = Coulomb friction
- $P_2 = \tau_s - \tau_c$ = stiction contribution
- $P_3 = \dot{\theta}_{th}$ = transition velocity for exponential decay
- $P_4 = k_v$ = viscous coefficient

This friction model captures:

- **Static friction:** (stiction) dominates at almost zero speeds.
- **Coulomb friction:** dominates at moderate speeds.
- **Viscous friction:** dominates at high speeds

5.2 Parameters Identification Methodology

The goal is to estimate parameters P_1, P_2, P_3, P_4 , from experimental telemetry. The process involves four main stages:

Step 1 Velocity Ramp Design:

To excite the friction dynamics, the motor is driven through sharp ramps in angular velocity $\dot{\theta}$, transitioning from $[-\omega_i, +\omega_i]$. The ramps should:

Include constant velocity segments to minimize inertial effects ($I\ddot{\theta} \approx 0$).

Sample different velocities to be able to characterize the viscous friction part of the model.

Step 2 Data Acquisition:

From the Telemetry node, record:

$\dot{\theta}$ = motor velocity (rad/s)

τ_m = motor torque (Nm)

By rearranging Eq. (5.1), the friction torque can be isolated as:

$$\hat{\tau}_{fr} = \tau_m - I\ddot{\theta} - \tau_{gp} \quad (5.5)$$

In practice, during constant-velocity segments the inertial term tends to zero, while the gravitational and passive terms are pre-calculated and treated as constant. Their overall effect is that equation 5.5 simplifies just to the motor torque.

Step 3 Regression Matrix Construction (Detailed Explanation)

The Stribeck friction model (Eq. 5.4) is nonlinear with respect to parameter P_3 , since it appears in the denominator inside the exponential term.

This nonlinearity makes direct regression on (P_1, P_2, P_3, P_4) impractical, as it would require a fully nonlinear optimization (e.g., Levenberg–Marquardt), which is computationally expensive and prone to convergence to local minima when noisy experimental data are used.

To simplify the problem, we temporarily fix $P_3 = \dot{\theta}_{th,i}$ to a trial value. This reduces the model to a linear form in the remaining parameters P_1, P_2, P_4 :

$$E_i = e^{-\frac{|\dot{\theta}_i|}{P_3}} \quad (5.6) \quad \hat{\tau}_{fr}(\dot{\theta}_i) = \text{sgn}(\dot{\theta}_i) (P_1 + P_2 E_i) + P_4 \dot{\theta}_i \quad (5.7)$$

Since E_i is now a known constant for each data point, the equation can be written in matrix form:

$$A_i = [\text{sgn}(\dot{\theta}_i), \text{sgn}(\dot{\theta}_i)E_i, \dot{\theta}_i] \quad (5.8)$$

$$y = \hat{\tau}_{fr}, \quad x = \begin{bmatrix} P_1 \\ P_2 \\ P_4 \end{bmatrix} \quad (5.9)$$

$$\hat{x} = (A^T A)^{-1} A^T y \quad (5.10)$$

The linear least-squares approach is fast, deterministic, and inherently free from local minima. However, fixing P_3 introduces a compromise: its analytical estimation is no longer obtained directly from the regression, but instead must be determined through a separate search loop. The parameter $P_3 = \dot{\theta}_{th}$ represents the velocity scale over which the friction torque transitions from the stiction value ($P_1 + P_2$) to the Coulomb value (P_1). Its lower bound should be chosen greater than the smallest measurable velocity step in the dataset (typically 0.05– rad/s 0.1 rad/s for high-resolution encoders), while its upper bound should be set below the maximum velocity reached during the ramp experiments (typically 0.5 rad/s – 1.0 rad/s). The optimal value of P_3 is then obtained by evaluating a small set of candidate values within this range and selecting the one that minimizes the residual norm of the fit.

$$\|r\|_2 = \|y - A\hat{x}\|_2 \quad (5.11)$$

In this final step, the best value of P_3 is determined by selecting the candidate X_i that yields the optimal regression solution, choosing the best $\dot{\theta}_{th,i}$ for optimal Stribeck transition velocity.

Dedicated friction estimation document at [22].

5.3 Experimental Results and Model Validation

The friction identification procedure was executed using a dedicated ROS2 node (*friction_estimator.py*) that performs the complete experimental sequence in accordance with the methodology described in Sections 5.1 and 5.2. The setup consisted of the right motor winch assembly driven by an ODrive Pro controller, commanded via an ESP32 interface. Since the test involved only a few turns of the rope around the drum, no variable-gear-ratio compensation was performed. For longer or higher-load experiments, however, this compensation becomes mandatory. The brake system remained disengaged throughout the test to avoid introducing additional resistive torque. Naturally left and right friction estimation have to be done separately.

Test procedure:

The node applies a predefined sequence of torque steps (ramps) to the motor (0.75, 1.25, 1.75 Nm), interleaving steady-state phases with rapid torque increases and decreases to traverse the full velocity range of interest. These commands are published to the */arganello/dx/target_torque* topic, while motor telemetry including angular velocity $\dot{\theta}$ (rad/s) and torque τ (Nm) is subscribed from */arganello/dx/telemetry* at 200 Hz. The telemetry is logged in CSV format with timestamps.

The following graph will represent the result of the calibration torque ramps.

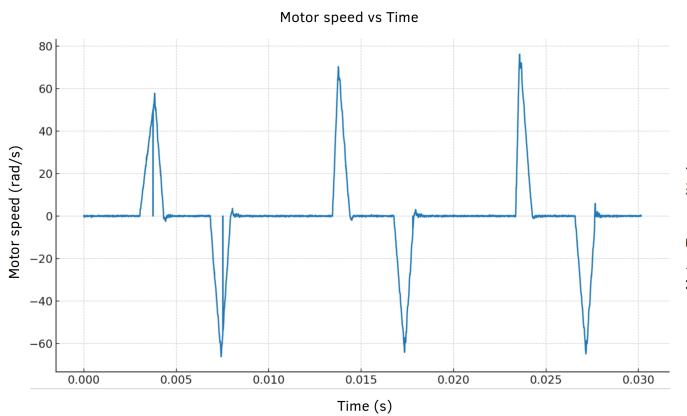


Figure 17: Plot of Measured torque vs time graph

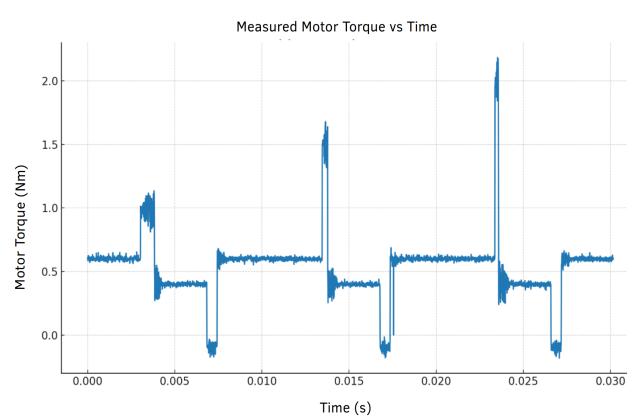


Figure 18: Plot of measured speed vs time graph

Once gathered the necessary feedback, parameter identification can start: The model parameters (P_1, P_2, P_3, P_4) were estimated by scanning a range of candidate Stribeck thresholds P_3 (0.1–1.0 rad/s) for 200 intermediate values and solving the least-squares problem. For each P_3 the regression matrix is built from the measured and the error norm $\|\hat{\tau}_{fr} - \tau_m\|_2$ is computed. In the end the preliminary identification of the friction model returned ($P_1 = 0.279$ Nm, $P_2 = -0.292$ Nm, $P_3 = 0.998$ rad/s, $P_4 \approx 0$) and did not provide a satisfactory fit to the measured torque–velocity profile. This discrepancy can be partially attributed to inertial effects: the velocity ramps in the dataset correspond to constant acceleration, implying a non-negligible constant inertial torque. Furthermore, the viscous coefficient P_4 was estimated as nearly zero, suggesting that the dataset did not contain sufficient information to capture velocity-dependent losses. Overall, the identified parameter set is consistent in capturing Coulomb and stiction effects but fails to represent viscous contributions. Both the dataset and the model structure therefore require refinement. Improved experiments such as compensating for passive torques, explicitly accounting for inertial torque during acceleration phases, acquiring higher-resolution torque measurements, and extending the velocity range to better excite viscous effects are necessary before the model can be considered validated for use in simulation and control.

6 Mechanical Design of the ALPINE Body

The ALPINE body constitutes the lower, tethered platform of the project: a lightweight yet durable chassis equipped with a pneumatic linear-actuator “leg” that generates controlled jumps and absorbs landing impacts. During flight and touchdown the robot remains tether-stabilized by the two winches that can pull from either direction, providing its main lateral thrust. A dual inertial measurement unit (IMU) setup underpins state estimation: the onboard IMU tracks the body’s orientation to enable automatic pitch compensation for consistent, energy absorbing landings, while the second external one (Fig. 22) estimates the rope angle with respect to the winches to aid spatial localization of the robot. Propulsion is enabled by a patented servo-valve actuation scheme that draws from a high-pressure air tank while modulating airflow and thus thrust so the jump impulse can be shaped precisely to the terrain and task. This section presents the end-to-end system architecture and implementation across mechanics, electronics, and software covering structural design, the jumping leg and tether interfaces, sensing and actuation subsystems, power and signal electronics, embedded software, and the integration steps that bind these elements into a functional robot.

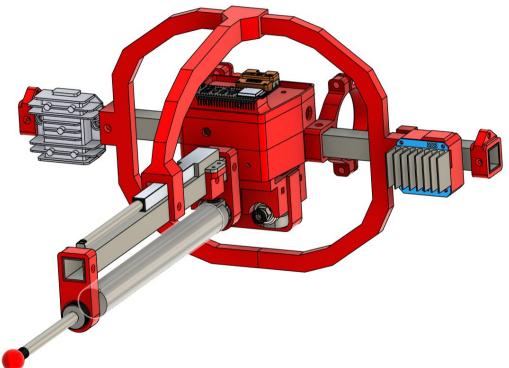


Figure 19: isometric view of the ALPINE body

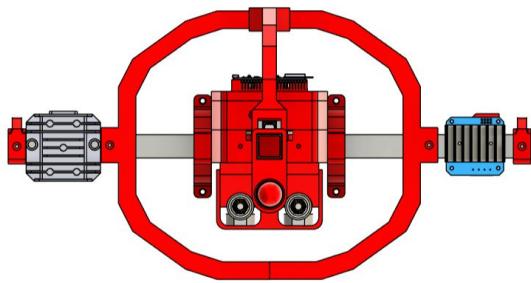


Figure 20: Alpine body front view

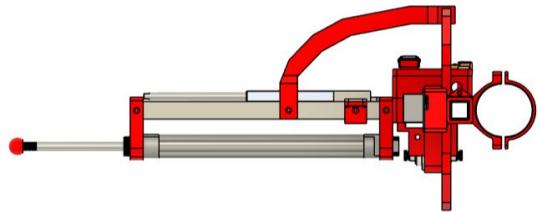


Figure 21: lateral view of the ALPINE body

6.1 CAD Design

The mechanical model and full assembly was developed in **Autodesk Fusion 360**. The chassis uses lightweight 20×20 mm aluminum extrusions joined by robust 3D-printed connectors, providing a stiff-but-light frame that is easy to iterate and manufacture. In the rear not shown in the CAD resides the **air tank**, which is the heaviest component; because the robot features left and right tether anchor points, the assembly must be balanced in pitch. To achieve balance by design, the pneumatic leg (piston) is arranged in a T-shaped layout that projects forward and counteracts the aft mass of the tank, while any remaining fine adjustment is handled electronically (see Section [9.1] for control details). All other relatively light components are mounted on the upper cross-member (“the top of the T from above”), located as close as possible to the overall center of mass to minimize inertial effects. The primary electronics are consolidated in an **internal bay**, and a halo-like protective hoop surrounds this area to shield cables, boards, and inertial measurement unit (IMU) from impacts during early jump tests. Refer to Figure 22 for a detailed layout.

6.2 ALPINE Body Components

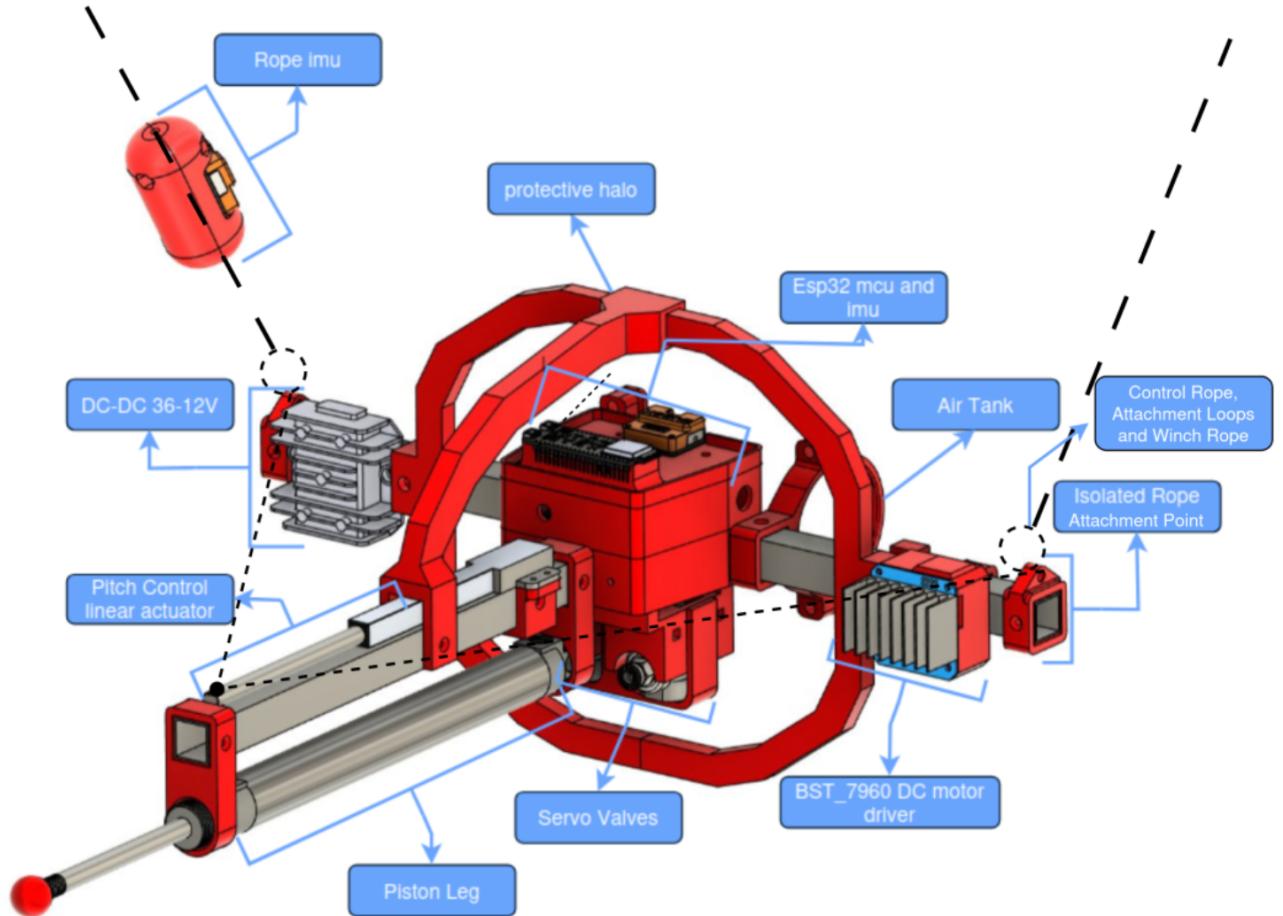


Figure 22: Alpine Jumping robot components.

The following subsections describe in detail the main components identified in Figure 22.

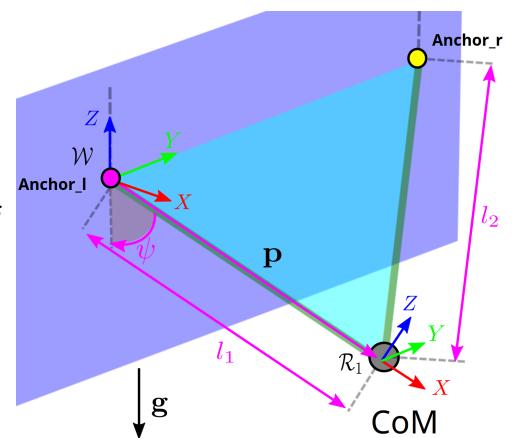
- **ESP32 micro-controller and IMU:**

The onboard micro-controller is a ESP32 hosting the low-level control loops, telemetry logging, and communication using the EspNow protocol (refer to chapter 9.2) provide a simple wireless link (see figure 16) to the base station, while the onboard IMU supplies body orientations and angular-rate data for stabilization and pose communication.

- **Rope IMU:**

An auxiliary inertial measurement unit (IMU) is mounted on the tether to measure rope orientation and its dynamics relative to the calibrated rope reference pose. The resulting rope direction vector, combined with the orientation of the rope plane from the exit point of the synchronous roller to the robot's center of mass (CoM) defines a segment whose vertices are those two points (W to CoM)(Diagram 1).The **IMU** provides the orientation for the reference frame

Diagram 1: Alpine body pose estimation.
R₁. By exploiting its own x-axis together with the **Diagram 1: Alpine body pose estimation.** rope length l₁, estimated from the synchronous roller encoder readings, it is possible to



determine the position p of the robot's CoM. A similar procedure can be applied to the second rope, yielding an independent estimate of the CoM position. Fusing the two estimates reduces the effect of IMU inaccuracies and improves robustness against measurement noise. In contrast, relying solely on the rope lengths l_1 and l_2 leads to an under determined problem: the CoM can only be localized within an infinite set of admissible positions, forming a semicircular locus. Incorporating IMU orientation measurements therefore eliminates this ambiguity and renders the odometry problem well-posed.

- **Protective halo:**

A rigid, lightweight 3D printed "halo" surrounds the central body acting as a roll cage. It protects critical electronics from impacts during landing and from accidental contact with rocks while also serving as a structural mounting ring for peripherals for future needs. The open-frame geometry minimizes structural mass and enhances serviceability.

- **Air tank:**

The pneumatic leg is supplied by a compact compressed air reservoir, pressurized to 200 bar. A primary integrated pressure regulator reduces this value to 4.5 bar, which constitutes the nominal supply pressure for the piston actuation during the jumping phase. This regulated pressure is further modulated by high-response servo valves (described in its own section later) to shape the pressure profile within the actuation chamber, thereby controlling both the impulse force and the timing of the jump. During landing, the system is subject to pressure peaks due to external impact loads. In this phase, the exhaust servo valve acts as the regulating element, releasing excess pressure in a controlled manner to achieve critical damping of the piston motion. This strategy ensures safe operation of the pneumatic circuit while enabling fine regulation of the high-pressure lines, allowing the robot to execute both jumping and landing in a controlled and repeatable manner. The complete system layout is reported in Chapter 8, Diagram 2.

- **Isolated Rope Attachment Point:**

The tether is secured to a robust 3D printed anchor point that is **galvanically isolated** from the chassis. The 3D-printed dielectric part provides the required insulation and mechanical compliance, ensuring that DC power is delivered to the robot exclusively through the conductive rope conductors and **not** via any return path through the robot frame. This isolation prevents ground loops and reduces radiated EMI (ElectroMagnetic Interference), thereby protecting low-voltage electronics.

- **Pitch-control linear actuator (BTS7960 driven):**

A compact electric linear actuator provides dynamic pitch adjustments by shifting a dedicated subassembly relative to the frame, compensating for mass asymmetries and external disturbances so that the robot's center of mass remains aligned with the rope anchors. Maintaining near-zero pitch ensures the body stays approximately (within 10°) from the normal to the wall, allowing the leg to absorb impact effectively during landing and avoids tipping over. The actuator is powered by a high-current BTS7960 H-bridge, which furnishes bidirectional, **PWM** based (Pulse Width Modulation) control of the **brushed DC** actuator supplying the current and voltage that exceed logic-level capabilities while enabling fine speed regulation. A closed-loop Proportional-Integral-Derivative (PID) controller running on the onboard micro-controller regulates the linear actuator velocity and thus the pitch angle to align with the wall normal direction (see Sec. 9.1 for PID

tuning and implementation), always yielding a stable configuration.

- **Piston leg:**

The main leg consists of a pneumatic cylinder that converts the energy stored in compressed air into linear impulse. During take-off, it extends rapidly to generate the jump force, while during landing it functions as an adjustable damper by modulating the exhaust flow through the servo valves (Section [9.1] for details on the control strategy).

- **Servo valves:**

Two linear servo valves regulate the airflow into and out of the pneumatic cylinder. By modulating the aperture of the internal ball element, the controller shapes the chamber pressure profile to generate thrust during take-off and to achieve critical damping during landing. Fast dynamic behavior and high repeatability are essential to ensure consistent jump trajectories; therefore, servo motors are employed to provide precise, electrically controlled valve openings. Preliminary tests with solenoid valves were also conducted, but their limited control bandwidth proved inadequate for the performance requirements of this project, particularly in the landing phase, where rapid pressure control (exceeding 20 Hz) is necessary to emulate critically damped leg behavior and prevent rebounds on the wall. Rendering of the servo valve is shown in Fig. 23.

- **DC-DC 36→12 V converter:**

A step-down DC/DC converter generates a regulated 12 V rail from the main 36 V bus. This 12 V line powers the valves and intermediate electronics and is further converted locally to 5 V and 3.3 V for logic-level circuitry. The dedicated converter ensures stable operation across mixed-voltage domains and streamlines power distribution among subsystems. Additional technical considerations are discussed in Chapter 7.

- **Control rope, winch rope, and attachment loops:**

The **control rope** is responsible for regulating the pitch orientation of the ALPINE body. It connects the **Isolated Rope Attachment Points** to the **Pitch-Control Linear Actuator** which in turn adjusts the tension of this rope and thereby shifts the center of mass. Since the robot is suspended by only two anchor points, active balancing through this mechanism is essential to maintain the desired orientation. The **winch rope**, previously introduced in Section 2.5, serves as the primary load-bearing element of the system. In addition to supporting the full weight of the robot, it also transmits electrical power from the winch to the ALPINE body. The PET sleeve is connected on either side at the **attachment loops** which are intentionally designed to be as small as possible and placed in close proximity to the **Isolated Rope Attachment Points** while keeping them mobile. This configuration provides slightly offset and mobile anchor points, which allow the Pitch-Control Linear Actuator to trim the pitch angle effectively. By contrast, if the loops were directly fixed to the rigid attachment points, pitch adjustment would be mechanically constrained and therefore impossible.

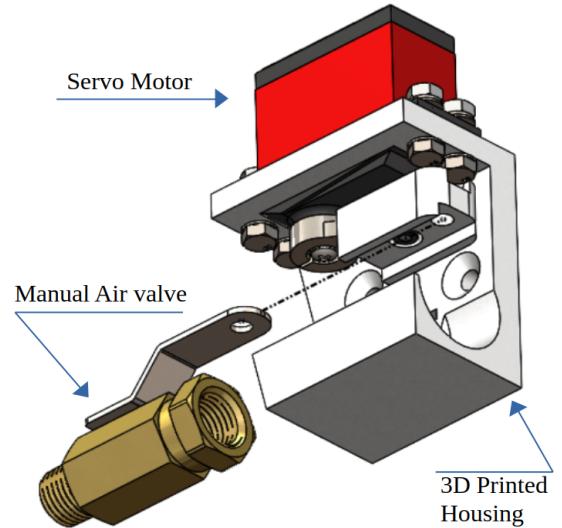


Figure 23: Servo valve assembly courtesy of [42] Halfcat Rocketry

7 Electronic Design of the ALPINE Body

7.1 Electrical Schematic

Figure 24 illustrates the complete electrical schematic of the onboard control system. The design

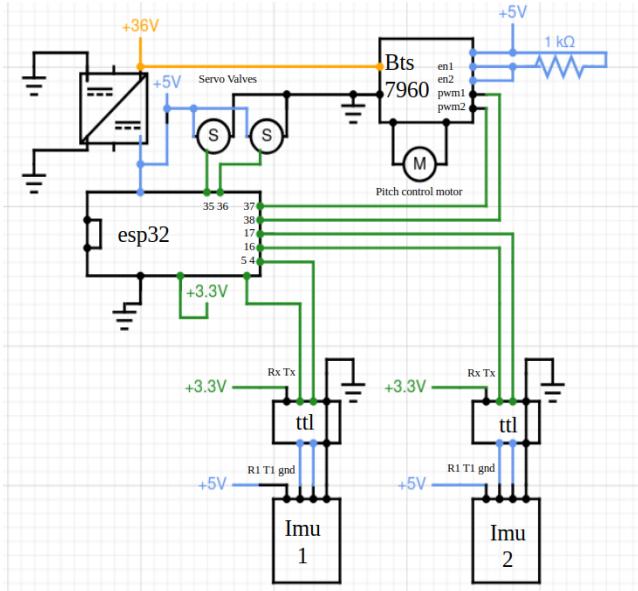


Figure 24: Alpine Jumping robot electrical schema.

is organized to clearly separate high-power and low-power domains, ensuring both safe operation and reliable signal integrity. The system is powered by a **36 V main bus** coming from the winches ropes, which supplies the high-power components. A step-down DC/DC converter derives a regulated **5V rail**, further reduced locally to **3.3 V** by the ESP32 board.

- **5 V** supplies the servo valves, **TTL** (Transistor-Transistor Logic) converters, and **IMUs**.
- **3.3 V** is reserved for the **ESP32** logic and **UART** (Universal Asynchronous Receiver-Transmitter) communication.

This staged distribution minimizes electrical noise, transmission losses, and enables the

use of thinner conductive ropes by transmitting higher voltage at lower current.

An in-depth discussion of the essential electrical components follows.

ESP32 micro-controller:

At the heart of the system lies the **ESP32**, which coordinates real-time operations. Its responsibilities include:

- Generating control signals for the servo valves.
- Driving the **BTS7960** motor controller via **PWM** to maintain stable pitch.
- Acquiring orientation and dynamic data from the **IMUs** through **UART** interfaces.
- Communicate telemetry via a wireless connection to the base station as seen in Chapter 4.

The **ESP32** operates entirely at **3.3 V** logic levels, ensuring low-power operation and compatibility with modern digital peripherals. It acts as the bridge between sensor feedback and actuators also sending telemetry back to the base station, forming the core of the robot's embedded control loop.

Servo Valves:

Two electrically actuated servo valves regulate airflow into and out of the pneumatic cylinder. Powered at **5 V** and controlled directly by the **ESP32**, they enable:

- **Thrust generation** during take-off, by admitting compressed air in.
- **Critical damping** during landing, by throttling exhaust flow.

Pitch Control Motor and Driver:

Pitch regulation is achieved through a dedicated **pitch-control motor**, a linear actuator, driven by a **BTS7960 H-bridge driver**. The BTS7960 receives high-power input from the **36 V** bus and control signals from the **ESP32**:

- **PWM inputs** modulate motor speed and direction.

- **Enable pins** ensure safe operation.
- A **pull-up resistor** stabilizes the enable pins at logic High connecting it to 5V so that the motor is always active in order to keep balance. (Fig. 24 top-right)

Inertial Measurement Units (IMUs):

Two IMUs are mounted on the system to measure orientation and dynamics. Each **IMU**:

- Operates from the **5 V supply**.
- Communicates over **RS232**, converted to 3.3 V logic by a dedicated **TTL level shifter** to **UART**.
- Transmits the orientation (quaternion) at high-frequency (100Hz) onboard processing and inertial data to the ESP32.

This configuration ensures accurate motion sensing and preserves electrical compatibility with the onboard micro-controller, thereby maintaining reliable integration of the IMUs within the control system.

8 Leg System of the ALPINE Climbing Robot

8.1 Piston Performance Calculations

Based on the simulation results presented in Chapter 2, the worst-case operating scenario requires a leg thrust of **280 N**. To meet this requirement, the pneumatic actuator must be dimensioned accordingly. The supply pressure is regulated to **4.5 bar**, and standard pneumatic cylinder sizes were considered during the selection process. A cylinder with a **bore diameter of 32 mm** and a **stroke length of 200 mm** was identified as suitable for the application. It is important to note that commercially available pneumatic pistons are manufactured in standardized sizes, which guided the selection process. The chosen bore dimension represents the smallest standard size capable of exceeding the required thrust, while the selected stroke length ensures sufficient extension to deliver the necessary impulse during take-off. This choice balances performance requirements, component availability, and system compactness, making it well suited for the intended application.

The piston **radius** and **area** are computed as:

$$r = \frac{d}{2} = \frac{32}{2} \text{ mm} = 0.016 \text{ m} \quad (8.1)$$

$$A = \pi r^2 = \pi(0.016)^2 = 8.04 \times 10^{-4} \text{ m}^2 \quad (8.2)$$

The **available thrust** at 4.5 bar (neglecting rod area and losses on the extend side) is:

$$F_{\max} = P_{\max} A = (4.5 \times 10^5)(8.04 \times 10^{-4}) = 3.61 \times 10^2 \text{ N} \approx 361.91 \text{ N} \approx 362 \text{ N} \quad (8.3)$$

8.2 Piston Stroke, Volume, Acceleration, and Timing

The **displaced volume** for a full extension is given the stroke **s** of 200 mm:

$$V = As = (8.04 \times 10^{-4})(0.20) = 1.608 \times 10^{-4} \text{ m}^3 = 160.8 \text{ cm}^3 \quad (8.4)$$

Assuming an equivalent accelerated mass of **10 kg** for the robot during push-off, the **maximum acceleration** produced by the cylinder thrust is:

$$a = \frac{F_{\max}}{m} = \frac{362}{10} = 36.2 \text{ m s}^{-2} \quad (8.5)$$

With uniformly accelerated motion from rest to the end of stroke has **extension time t**:

$$s = \frac{1}{2}at^2 \Rightarrow t = \sqrt{\frac{2s}{a}} = \sqrt{\frac{0.40}{36.2}} \approx 0.105 \text{ s} \quad (8.6)$$

The **piston end-of-stroke speed** is therefore

$$v = at = 36.2 \times 0.105 \approx 3.8 \text{ m s}^{-1} \quad (8.7)$$

8.3 Required Air Flow and Tube Sizing

The **peak volumetric flow** into the cap end (quasi-incompressible kinematics approximation) is

$$Q_{\max} = Av = (8.04 \times 10^{-4})(3.8) = 3.06 \times 10^{-3} \text{ m}^3 \text{s}^{-1} \approx 3.06 \text{ L s}^{-1} \quad (8.8)$$

Selecting a hose with **internal diameter**:

$$d_{\text{tube}} = 6.5 \text{ mm} = 0.0065 \text{ m}, \quad (8.9)$$

its **sectional area** is:

$$A_{\text{tube}} = \frac{\pi d_{\text{tube}}^2}{4} = \frac{\pi (0.0065)^2}{4} \approx 3.32 \times 10^{-5} \text{ m}^2 \quad (8.10)$$

The corresponding **air velocity** in the tube at peak is

$$u = \frac{Q_{\text{peak}}}{A_{\text{tube}}} = \frac{3.06 \times 10^{-3}}{3.32 \times 10^{-5}} \approx 92.2 \text{ m s}^{-1} \quad (8.11)$$

For the real prototype, a **6.5 mm internal diameter** tube was selected, corresponding to an **outer diameter of 10 mm**. This choice represents the largest tubing dimension that could be accommodated within the compact volume of the robot. A larger inner diameter would reduce the flow velocity, potentially improving laminarity and lowering pressure losses. However, it would also increase the outer diameter, making the tubing impractical for integration into the system, as larger hoses are more difficult to bend. Another important consideration is that the calculated flow corresponds to the **maximum demand condition**, which occurs only at the final instants of peak velocity during take-off. Under most operating conditions, the actual flow requirement is lower, reducing both the average air velocity and pressure drop in the tubing. Finally, it should be noted that the total tubing length in the prototype is less than **0.5 m**, just sufficient to connect all components. This short length further limits pressure losses, allowing them to be neglected in the calculations.

8.4 Pneumatic Diagram

The pneumatic circuit shown in Diagram 2 controls a double-acting cylinder "leg" using compressed air stored in the air tank. A pressure regulator reduces the supply from **200 bar** to **4.5 bar**, enabling its safe use since the tubing cannot withstand pressures above **10 bar**, while

ensuring consistent actuator performance. Two servo valves (S) operate independently and should never be opened simultaneously; when one admits pressurized air into the cylinder, the other vents the same chamber directly into the atmosphere. The inlet valve connects to the regulated air source, whereas the exhaust valve discharges the expelled air, completing a system that provides precise and monodirectional motion of the cylinder. Note that the retraction of the cylinder is not powered by

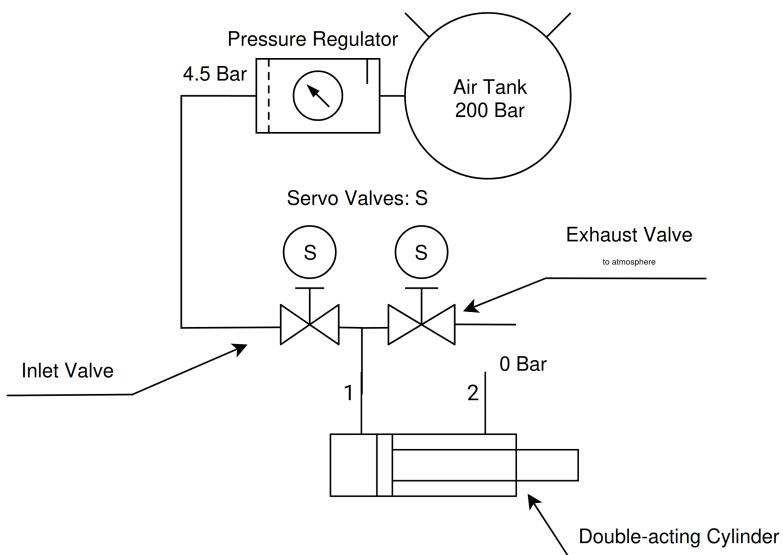


Diagram 2: Pneumatic diagram of the ALPINE jumping robot

compressed air, but by contact of the piston with the wall and the inertia of the robot during landing, so chamber (1) has to be vented for the retraction to occur. Chamber (2) is always vented and not used to assist retraction.

9 Software Architecture of the ALPINE Body

The software architecture of the robot comprises of **two** ESP32 micro-controllers. The first is mounted onboard the robot, where it handles low-level tasks such as sensor interfacing, valve actuation, and stability control (see Section 9.1). The second, referred to as the “**dongle**” serves as a ground station: it communicates with the robot via ESP-NOW and connects to the host computer via USB, exposing all functionalities to the ROS2 ecosystem. As discussed in Section 9.2, the dongle must remain within wireless range of the onboard micro-controller. For this reason, it may be connected directly to a nearby computer USB port, or alternatively placed at an optimal location using a USB extension cable to ensure robust communication.

9.1 Onboard Firmware for Real-Time Operation

The onboard micro controller unit (MCU) of the robot has been chosen for its processing capability and integrated wireless features. Its primary role is to handle time-critical operations, sensor interfacing, and low-level actuator control.

Two Movella IMUs [23] are connected to the ESP32 via **RS232** communication protocol. Since RS232 is not natively supported by the MCU because it operates at higher differential voltage levels (± 5 V) that are incompatible with the ESP32's 3.3 V UART interface a TTL-to-UART level converter is employed to ensure proper communication compatibility between the two standards (see Fig. 24). Each IMU provides a data stream composed of quaternions, raw accelerometer measurements, and gyroscope signals. The first IMU is rigidly aligned with the robot's frame and is used to estimate the robot's pose, linear accelerations, and angular velocities, which are essential for stabilization. The second IMU, as discussed in Chapter 6.2, is dedicated to measuring rope inclination with respect to the winch system. For pitch stabilization, the onboard **ESP32** interfaces with the **BTS7960 DC motor driver**. The driver is controlled via two PWM channels, providing bidirectional velocity control of the pitch-control linear actuator (Fig. 22). The actuator maintains the robot's pitch angle within $\pm 10^\circ$ from the normal of the wall. In principle, this reference corresponds to the normal of the landing surface, which under the idealized condition of a vertical wall coincides with the horizon. In practical applications, however, estimating the local surface geometry is required, a problem that falls outside the scope of this thesis. For the control part, a **Proportional-Integral-Derivative** (PID) controller regulates the pitch motion. The control error is defined as the difference between the measured pitch angle and the wall's normal. This error serves as the PID input, while the controller output is a PWM duty cycle that drives the H-bridge driver, thereby controlling the velocity and direction of the pitch-control linear actuator. The PID parameters were tuned experimentally, balancing stability with responsiveness to ensure that the pitch remains within range during dynamic maneuvers. In addition to stabilization, the ESP32 manages the servo-actuated inlet and exhaust valves of the pneumatic leg. These valves regulate chamber airflow during thrust generation and landing. During jumps, the onboard controller receives the high-level command F_{leg} representing the desired leg force. This value is combined with the known mass of the robot to compute the target acceleration. A PID controller then regulates the inlet valve position so that the measured acceleration follows the commanded value. By modulating the airflow into the piston chamber, the inlet valve provides the pressure input required to generate the commanded leg force, which is applied over a specified thrust duration to deliver the desired impulse. In flight, the onboard controller continuously monitors **IMU** data to detect acceleration spikes along the leg axis. When such a spike is observed, the firmware initiates the landing phase, it activates an open loop **valve-**

opening profile, experimentally tuned to release chamber pressure in a controlled manner (different profiles and implementation at [18]). This ensures that the impact energy is dissipated smoothly, preventing rebound. The result is a landing strategy that combines real-time IMU feedback with active valve modulation to achieve effective impact damping. Future works will involve the implementation of a closed loop strategy. Finally, the ESP32 transmits telemetry data from both IMUs via ESP-NOW to the ground station at 100 Hz, which was identified as the maximum frequency for stable bidirectional communication. Further details on this protocol are provided in Chapter 9.2.

9.2 ESP-NOW Dongle: Firmware and ROS2 Integration

A second ESP32 micro-controller is dedicated to managing the wireless communication link between the Robot and the host computer. This unit, referred to as the “dongle” (Fig. 16), functions as the ground station by receiving telemetry packets from the robot via ESP-NOW and forwarding them to the PC through a USB interface. The name “dongle” reflects its role as a compact USB bridge between the wireless embedded system and the ROS2 protocol.

ESP-NOW [4] is a proprietary wireless protocol developed by Espressif [6], operating in the **2.4 GHz ISM band**. It enables low-latency, peer-to-peer communication without the overhead of a full Wi-Fi connection. According to the official Espressif specifications, ESP-NOW supports a communication range of approximately **100 m in open environments** and **30–50 m indoors**, depending on antenna design and radio conditions. This wireless link is essential, as the robot receives power through conductive ropes but does not carry data in the cables, thus requiring a reliable bidirectional radio communication channel. To guarantee deterministic performance, ESP-NOW communication operates on a fixed Wi-Fi channel; in the current implementation, **channel 1** (2.412 GHz) is used, with the option to reassign channels in case of interference. Unlike shared time-slot protocols, where latency grows with the number of connected nodes, operating on a fixed channel with a dedicated link ensures consistent update rates and minimizes communication delays. However, this approach requires a dedicated pair of ESP32 devices for each radio bridge, increasing hardware complexity and cost. A time-division multiplexing (TDM) approach was initially tested but discarded, as adding multiple peers (two winches and one ALPINE body) introduced excessive latency, making the scheme unsuitable for real-time operation.

On the **telemetry side**, the dongle receives packets at **100 Hz** from the ALPINE onboard controller. Each packet contains sensor data, including the outputs of both IMUs (quaternions, linear accelerations, and angular velocities). These packets are sent to the host PC via USB and processed by a dedicated ROS2 node, **dongle_node.py**. This node publishes the telemetry topics `/alpine/telemetry` using the ROS2 communication protocol and additionally provides a custom ROS2 service, `/alpine/jump` which requires a single float32 argument corresponding to the desired leg force to initiate a jump sequence, this in turn is transmitted to the ALPINE onboard micro-controller, jumping as described in chapter 9.1.

In summary, the dongle establishes a **bidirectional 100 Hz wireless bridge** between the robot and the host PC (see Fig 16). On the host side, it integrates seamlessly with ROS2, exposing low-level functionalities for control while simultaneously providing continuous telemetry for estimation and higher-level decision-making. (see code implementation [18]).

10 Conclusions

This thesis represents my experience in developing a working prototype of the ALPINE Climbing Robot. What began as a conceptual design challenge, an unconventional idea aimed at demonstrating the feasibility of a novel robotic platform, was brought to life through the collective effort of our team. The process of translating the concept into a real-time prototype also revealed several overcomplicated or non-essential components, which were simplified or merged into more practical solutions. In particular, the articulated landing mechanism presented in Focchi's paper [7] was removed in favor of a single piston leg capable of performing both propulsion and landing functions. The current prototype must be regarded as a **proof of concept**, whose primary purpose was to validate the feasibility of the proposed architecture rather than to serve as a final product. Inevitably, many aspects remain open for improvement, and the lessons learned during this development will serve as the foundation for future iterations. Several potential directions for enhancement have already been identified. A major design improvement concerns the integration of **active stabilization mechanisms**, essential because the robot cannot currently control its orientation during flight and is therefore at risk of tipping over at landing. Technologies such as propellers or cold-gas thrusters, which are widely employed in aerial and space robotics to achieve precise attitude control through thrust vectors, working in accordance with Newton's third law, were considered and partially tested during this project but not implemented in the prototype due to time and resource constraints. Their adoption in the ALPINE Climbing Robot could significantly enhance robustness in real-world deployments, where environmental disturbances, such as wind or falling rocks, are far more severe than in the controlled laboratory environment. Other approaches, such as the use of a **gyroscopic wheel** for stabilization, were also explored. However, the demand of very high rotational speeds 30.000 RPM (revolutions per minute), additional tilting mechanisms, and strict mass constraints made this solution impractical for the current design. In conclusion, the ALPINE Climbing Robot presented in this thesis should be viewed as a **first successful prototype** in a broader line of research. While not without its limitations, it demonstrates that such an unconventional concept is technically feasible.

For transparency of the plagiarism requirements I need to add that the preparation of this document was partially supported by the use of generative artificial intelligence tools, which were employed exclusively for text drafting and refinement purposes. [32]

11 Bibliography

- [1] Aerones, “Robotics Revolutionizing the Wind Industry - Wind Turbine Blade Repairs” YouTube. [Online Video]. Available: <https://www.youtube.com/watch?v=pz9Wn2Kcj4A>. [Accessed: Aug. 23, 2025].
- [2] Alkalla M. G., Fanni M. A., Mohamed A. M., Hashimoto S., 2017, “Tele-operated propeller-type climbing robot for inspection of petrochemical vessels” Industrial Robot: An International Journal, vol. 44, no. 2, pp. 166–177, doi: 10.1108/IR-07-2016-0182.
- [3] Desmos, Motor Parameter Estimation Tool (online calculator). Available: <https://www.desmos.com/calculator/1bw85mchnu>. [Accessed: Aug. 27, 2025].
- [4] Espressif Systems, ESP-NOW Wireless Communication Protocol. Available: <https://www.espressif.com/en/solutions/low-power-solutions/esp-now>. [Accessed: Aug. 27, 2025].
- [5] Espressif Systems, ESP32-H2-DevKitM-1 User Guide. Available: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32h2/esp32-h2-devkitm-1/user_guide.html. [Accessed: Aug. 27, 2025].
- [6] Espressif Systems, Official Website. Available: <https://www.espressif.com/>. [Accessed: Aug. 27, 2025].
- [7] Focchi M., Del Prete A., Fontanelli D., Frego M., Peer A., Palopoli L., 2025, “ALPINE: A climbing robot for operations in mountain environments” Robotics and Autonomous Systems, vol. 190, art. 104999, doi: 10.1016/j.robot.2025.104999.
- [8] Focchi M., Bensaadallah M., Frego M., Peer A., Fontanelli D., Del Prete A., Palopoli L., 2023, “CLIO: A novel robotic solution for exploration and rescue missions in hostile mountain environments,” *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, pp. 7742–7748, doi: 10.1109/ICRA48891.2023.10160440.
- [9] Focchi, M., Climbing Robots – MATLAB Simulation (GitHub repository). Available: https://github.com/mfocchi/climbing_robots2?tab=readme-ov-file. [Accessed: Aug. 27, 2025].
- [10] Halfcat Rocketry, Servo-Actuated Ball Valve. Available: <https://www.halfcatrocketry.com/sabv>. [Accessed: Aug. 27, 2025].
- [11] Hawkes E. W., Christensen D. L., Cutkosky M. R., 2015, “Vertical dry adhesive climbing with a 100× bodyweight payload” *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, USA, pp. 3762–3769, doi: 10.1109/ICRA.2015.7139722.
- [12] Jensen-Segal S., Virost S., Provancher W. R., 2008, “ROCR: Dynamic vertical wall climbing with a pendular two-link mass-shifting robot” *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, USA, pp. 3040–3045, doi: 10.1109/ROBOT.2008.4543672.
- [13] KAIST Urban Robotics Lab, “Wall-Climbing Aerial RObot System (CAROS)”. [Online] Available: <https://youtu.be/EeWIkytu1vY>. [Accessed: Aug. 23, 2025].

- [14] Kim S., Spenko M., Trujillo S., Heyneman B., Santos D., Cutkosky M. R., 2008, “Smooth Vertical Surface Climbing With Directional Adhesion” IEEE Transactions on Robotics, vol. 24, no. 1, pp. 65–74, doi: 10.1109/TRO.2007.909786.
- [15] Lahouar S., Ottaviano E., Zeghoul S., Romdhane L., Ceccarelli M., 2009, “Collision free path-planning for cable-driven parallel robots” Robotics and Autonomous Systems, vol. 57, no. 11, pp. 1083–1093, doi: 10.1016/j.robot.2009.07.006.
- [16] Lebus International Engineers GmbH, 2023, *Lebus Brochure EN*, online: https://lebus-germany.com/wp-content/uploads/2023/09/00979_LebusBrochure_EN.pdf [Accessed: Aug. 27, 2025].
- [17] Malacarne R., 2025, “Multi-jump Trajectory Optimizer and Software Design of the ALPINE Climbing Robot”, Master’s Thesis (forthcoming), University of Trento.
- [18] MalaHard-RoboTech, Climb_arduino – Embedded Source Code (GitHub repository). Available: https://github.com/MalaHard-RoboTech/Climb_arduino. [Accessed: Aug. 27, 2025].
- [19] MalaHard-RoboTech, Climb_ros2 – ROS2 Jazzy Source Code (GitHub repository). Available: https://github.com/MalaHard-RoboTech/Climb_ros2. [Accessed: Aug. 27, 2025].
- [20] MalaHard-RoboTech, ALPINE Climb Robot Documentation (GitHub repository). Available: https://github.com/MalaHard-RoboTech/Documentation_Climb. [Accessed: Aug. 27, 2025].
- [21] MalaHard-RoboTech, ALPINE Climb Robot Variable Gear Calculation Documentation (GitHub repository). Available: [https://github.com/MalaHard-RoboTech/Documentation_Climb/blob/main/Hardware%20\(mechanical\)%20part/1_Winch_Part/Variable_gear_transmission_calculations.md](https://github.com/MalaHard-RoboTech/Documentation_Climb/blob/main/Hardware%20(mechanical)%20part/1_Winch_Part/Variable_gear_transmission_calculations.md) [Accessed: Aug. 27, 2025].
- [22] MalaHard-RoboTech, ALPINE Climb Robot Friction Estimation Documentation (GitHub repository). Available: [https://github.com/MalaHard-RoboTech/Documentation_Climb/blob/main/Hardware%20\(mechanical\)%20part/1_Winch_Part/friction_estimation.md](https://github.com/MalaHard-RoboTech/Documentation_Climb/blob/main/Hardware%20(mechanical)%20part/1_Winch_Part/friction_estimation.md) [Accessed: Aug. 27, 2025].
- [23] Movella, MTi-320 – Inertial Measurement Unit Datasheet. Available: <https://www.xsens.com/hubfs/Downloads/Leaflets/MTi-320.pdf>. [Accessed: Aug. 27, 2025].
- [24] Nesnas I. A. D., Matthews J., Abad-Manterola P., Burdick J., Edlund J., Morrison J., Peters R., Tanner M., Miyake R., Solish B., Anderson R., 2012, “Axel and DuAxel Rovers for the Sustainable Exploration of Extreme Terrains” Journal of Field Robotics, vol. 29, no. 4, pp. 663–685, doi: 10.1002/rob.21407.
- [25] Newdick S., Ongole N., Chen T. G., Schmerling E., Cutkosky M. R., Pavone M., 2023, “Motion Planning for a Climbing Robot with Stochastic Grasps” Proc. IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, pp. 11838–11844, doi: 10.1109/ICRA48891.2023.10160218.

- [26] Nishimura Y., Takahashi S., Mochiyama H., Yamaguchi T., 2022, “Automated Hammering Inspection System With Multi-Copter Type Mobile Robot for Concrete Structures” IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 9993–10000, doi: 10.1109/LRA.2022.3191246.
- [27] ODrive Robotics, Motor M8325S 100kV – Documentation. Available: <https://docs.odriverobotics.com/v/latest/hardware/odrive-motors.html#m8325s-100kv>. [Accessed: Aug. 27, 2025].
- [28] ODrive Robotics, ODrive Pro – Datasheet. Available: <https://docs.odriverobotics.com/v/latest/hardware/pro-datasheet.html>. [Accessed: Aug. 27, 2025].
- [29] ODrive Robotics, ODrive Pro – Encoder Datasheet. Available: <https://docs.odriverobotics.com/v/latest/manual/hardware-config.html#onboard-encoder>. [Accessed: Aug. 27, 2025].
- [30] ODrive Robotics, ODrive Pro – Regeneration Clamp Datasheet. Available: <https://docs.odriverobotics.com/v/latest/hardware/regen-clamp-datasheet.html>. [Accessed: Aug. 27, 2025].
- [31] ODrive Robotics, Official Website. Available: <https://odriverobotics.com/>. [Accessed: Aug. 27, 2025].
- [32] OpenAI, “Introducing ChatGPT,” OpenAI Blog, Nov. 30, 2022. Available: <https://openai.com/index/chatgpt/>. [Accessed: Aug. 27, 2025].
- [33] Parness A., Abcouwer N., Fuller C., Wiltsie N., Nash J., Kennedy B., 2017, “LEMUR 3: A limbed climbing robot for extreme terrain mobility in space” Proc. IEEE International Conference on Robotics and Automation (ICRA), Singapore, pp. 5467–5473, doi: 10.1109/ICRA.2017.7989643.
- [34] Pixabay, “Computer desktop keyboard monitor (vector graphic),” Pixabay. Available: <https://pixabay.com/vectors/computer-desktop-keyboard-monitor-160579/>. [Accessed: Aug. 24, 2025].
- [35] Polzin M., Centamori F., Hughes J., 2023, “Heading for the Abyss: Control Strategies for Exploiting Swinging of a Descending Tethered Aerial Robot” Proc. IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, pp. 5373–5378, doi: 10.1109/ICRA48891.2023.10160347.
- [36] Riskin D. K., Racey P. A., 2010, “How do sucker-footed bats hold on, and why do they roost head-up?” Biological Journal of the Linnean Society, vol. 99, no. 1, pp. 233–240, doi: 10.1111/j.1095-8312.2009.01362.x.
- [37] Stanford Biomimetics & Dexterous Manipulation Lab, “SCAMP: Stanford Climbing and Aerial Maneuvering Platform” Stanford University. [Online]. Available: <https://bdml.stanford.edu/Main/SCAMP>. [Accessed: Aug. 23, 2025].
- [38] Sukvichai K., Maolanon P., Songkrasin K., 2017, “Design of a double-propellers wall-climbing robot” Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, pp. 239–245, doi: 10.1109/ROBIO.2017.8324424.

- [39] Susanto W., Babuška R., Liefhebber F., van der Weiden T., 2008, “Adaptive friction compensation: Application to a robotic manipulator” Proc. 17th IFAC World Congress (IFAC’08), Seoul, Korea, pp. 2020–2024, doi: 10.3182/20080706-5-KR-1001.00343.
- [40] Wang L., Culha U., Iida F., 2014, “A dragline-forming mobile robot inspired by spiders” Bioinspiration & Biomimetics, vol. 9, no. 1, art. 016006, doi: 10.1088/1748-3182/9/1/016006.
- [41] Yang L., Li B., Feng J., Yang G., Chang Y., Jiang B., Xiao J., 2023, “Automated wall-climbing robot for concrete construction inspection” Journal of Field Robotics, vol. 40, no. 1, pp. 110–129, doi: 10.1002/rob.22119.
- [42] Zarebidoki M., Dhupia J. S., Xu W., 2022, “A Review of Cable-Driven Parallel Robots: Typical Configurations, Analysis Techniques, and Control Methods” IEEE Robotics & Automation Magazine, vol. 29, no. 3, pp. 89–106, doi: 10.1109/MRA.2021.3138387.