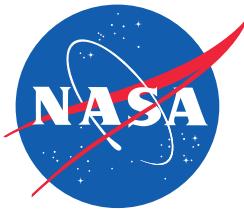


POLITECNICO DI TORINO

COLLEGIO DI INGEGNERIA INFORMATICA, DEL CINEMA E
MECCATRONICA

MSC IN MECHATRONIC ENGINEERING

Tether Management and Tension Control for Rappelling Rovers



Supervisor

Prof. Marina INDRI

Author

Alessandro STEFANINI

Supervisors at Jet Propulsion Laboratory:

Dr. Travis Brown

Dr. Issa A. D. Nesnas

April 2018

Abstract

Space robotics has always been source of innovation for all the robotic industry. In fact, the challenges that are faced in this field generate solutions for many other applications. The critical conditions of operation require new approaches to robotic mobility, especially due to the impossibility of a real-time control. Among the many investigations, tethered rovers could be able to traverse extreme terrains and reach new places where no human technology has been yet. As scientists believe these locations are interesting targets for novel planetary research, the development of such robots is active and progressing, with the aim to send one of them on a mission in the next years.

This work describes the design and testing of a tether control system for Axel, a prototype rover built at NASA's Jet Propulsion Laboratory. This is the first iteration of the component's design. The performance analysis of this initial version lays out guidelines for improvement, which will be necessary to make the control system ready for a field test, and, eventually, for its ultimate goal: the flight mission.

Acknowledgments

I really want to thank all the people and the institutions that made this work possible. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the Visiting Student Research Program (JVSRP) and the National Aeronautics and Space Administration. I thank Travis, who hosted and guided me with attention, Issa for being the excellent leader of Axel's team, which is made of amazing people (in no particular order): Evan, Jacek, Mike, Laura, Melissa, Joel, Bob, Patrick, Dean, Garreth, Teddy, Amanda, Nikola and many others. I learned a lot from them, and I had a great time at JPL! I also appreciated the support of my advisor at the Politecnico di Torino, prof. Marina Indri and the Politecnico for providing the funding.

Thanks to the people who made my stay in California an exciting adventure. I had so much fun at 1010 with Emma, Thomas, Kim and Eric. Thanks to all the friends who explored with me: in particular Liton, Andrew, Nathan, Charles, Andreas, Vince, Orazio, Filippo, Luigi, Federico, Luis, Bryan, Francesca, but also many others. I will always remember this beautiful time and will keep all the memories.

I also thank all the friends in Turin, for the good times we had during these two years (and more): I could never be bored, and I truly enjoyed all the moments together.

My most heartfelt thanks are for my family, because they have been always ready to support me in everything: from my parents Roberto and Francesca to Chiara and Matteo. I would not have done anything without their efforts.

Contents

1	Introduction	6
1.1	Thesis outline	7
2	Mobile robots for extreme terrain	8
2.1	Robotic mobility	8
2.1.1	Moon's Lava Tubes	9
2.1.2	Mars' Recurring Slope Lineae	10
2.2	Classification of space robotics mobility	11
2.2.1	Conventional mobility	11
2.2.2	Unconventional mobility	12
2.3	Tethered mobility	13
3	Axel & DuAxel rovers	15
3.1	Axel	15
3.1.1	Hardware	17
3.1.2	Software	18
3.1.3	CLARAty	19
3.2	DuAxel	20
4	Tethered mobility test bed	23
4.1	Overview	23
4.1.1	Subsystems	23
4.2	Specifications and capabilities	25
4.3	Hardware	25
4.3.1	Load motor	25
4.3.2	Sensors	28
4.3.3	Controllers and communication	31
4.3.4	Power supply	31
4.4	Software	32
4.4.1	Elmo Whistle module	32
4.4.2	Level winder motion module	36

4.4.3	Measurement Computing USB-204 module	38
4.4.4	Test bed module	38
5	Tether management system	41
5.1	Overview	41
5.2	System	43
5.2.1	Primary Tension Module	43
5.2.2	Slack buffer and spool	49
5.3	Control	52
5.3.1	PID control	54
5.3.2	H_∞ control	61
6	Results and conclusions	73
6.1	PID control	73
6.1.1	Reference input tests	73
6.1.2	Shock loading tests	74
6.1.3	Disturbance response tests	75
6.2	H_∞ control	75
6.3	Spooling and level winder test	79
6.4	Future work	80
	Bibliography	82

Disclaimer

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Chapter 1

Introduction

Space exploration has always been one of the most prominent fields where advanced robotics is applied in every aspect of its broad range. The critical conditions of operation require strict constraints and a strong resistance to adversities. Today, with the renovated energy towards the space science, many public agencies and private businesses employ lots of resources to develop reliable machines able to work partially or fully autonomously on other planets. A peculiar aspect of planetary exploration is the uncertainty and the geometry of the terrain, which requires special mobility systems. Difficult to be accessed places are often the favorite for doing novel research about geological history of planets. That is where the most interesting samples are, and their retrieval and analysis could bring outstanding advance to our knowledge of planetary formation. For this reason, space robotics is always developing new technologies to allow planetary rovers to move and operate in impervious terrain: in particular, the tethered mobility could have a strong impact in accessing craters and steep slopes. This work describes the process of designing, building and testing a Tether Management System (TMS) that enables a new level of control and safety of such rovers: they will be able to navigate autonomously, with an unprecedentedly low hazard level and precise movement. All the development has been carried out at Jet Propulsion Laboratory in Pasadena, CA (United States of America), under the funding of the NASA through the JPL Visiting Student Research Program (JVSRP), which allows students from all over the world to take part in the leading research in space industry. This project, that has been conducted over a six months period, aimed to advance the development status of Axel, a prototype of a planetary rover, which has been invented by team 347F (Robotics mobility for extreme terrain) in JPL and it has been submitted to IROS 2018 [1].

1.1 Thesis outline

This work is divided in sections that follow the temporal and functional development of the tether management system.

- In **Chapter 2** an outline of the state of extreme mobility is presented, with focus on the tethered mobility. This is compared with other types of unconventional approaches to robotic mobility investigated both inside and outside NASA's Jet Propulsion Laboratory.
- **Chapter 3** introduces the Axel and the DuAxel rover by NASA's Jet Propulsion Laboratory, and the target terrain configurations for which they have been conceived. This section, along with the overview of the rover, discusses the necessity of a TMS.
- The novel research follows in **Chapter 4**, which describes the process of designing and building a test bed for such a control system.
- **Chapter 5** contains the core of the work, that is the design of a suitable control algorithm for the TMS. Two approaches are proposed: a decentralized one based on the PID method and a theoretical implementation of H-infinity control.
- This dissertation ends with **Chapter 6**, which shows the results and conclusions, giving some guidelines for future development.

Chapter 2

Mobile robots for extreme terrain

2.1 Robotic mobility



Figure 2.1: NASA/JPL-Caltech, Curiosity Rover on Mars.

The goal of a space rover is to reach a specific destination on a space body. It can carry a payload, such as an instrument to be deployed, or use onboard sensors to take measurements. Conventionally, robots employ wheels or legs

to traverse graceful terrains; however, limits of this methodology are evident in the case of extra-planetary exploration. Uneven surfaces, obstacles, absence of energy sources, together with dust and sand constitute serious hazards, which are, sometimes, impossible to overcome with traditional types of mobility. Another important factor is the impossibility of a real-time remote control, due to the large transmission distances that characterize space missions. For example, a Mars rover like Mars Science Laboratory (known as Curiosity, shown in Figure 2.1) has a communication delay ranging from 3 to 21 minutes, depending on the position of the two planets in their orbits. For this reason, the robot has to adopt some autonomy functionalities to traverse certain types of terrain, and during these operations, it is necessary to guarantee a safety margin in case of unexpected events, like tipping over a steep slope.

Possible extreme environments can be easily found in the Solar System, basically in every space body: from satellites like the Moon or Enceladus with its global ocean to planets like Mars or Venus. The following are examples of interesting targets for science missions, but hard to reach for conventional space rovers.

2.1.1 Moon's Lava Tubes

Lunar sub-surface tunnels have been hypothesized since 1971 [2], following the observation of large linear rilles on the surface of Moon. The most supported theory about their nature is that they are stable cavities running for many kilometers, and they formed when the ancient basaltic lava flow drained and left an hollow passage. In 2009 JAXA' SELENE mission showed [3] the presence of vertical holes in the ground, as shown in Figure 2.2. They are located in proximity of the Marius Hills, which seem to be possible entrances to those tubes, where the ceiling has collapsed [4]. Further research has revealed a significant amount of holes spread all over the lunar surface. Lava tubes could be home for human bases, because they can be isolated from the lunar surface environment by closing the ends of the tunnels. Moreover, the high pressure and temperature of the lava flows has cristallized the walls, thus allowing to create an artificial atmosphere in the hollow spaces. Future missions [5] aim to reach those holes to evaluate their life supporting capabilities and study lunar formation: samples have to be taken from almost vertical walls and the bottom of the pits, requiring the rover to traverse such an unconventional type of terrain.

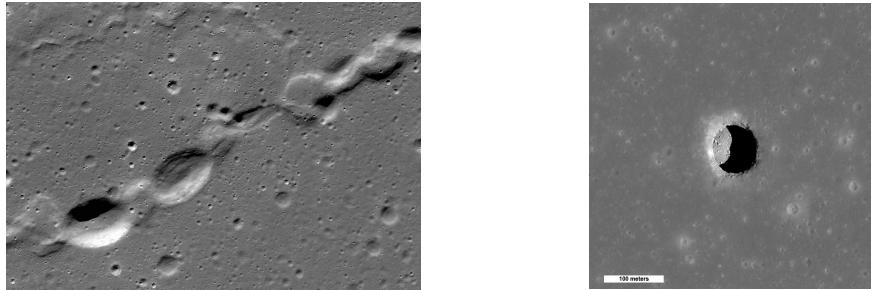


Figure 2.2: NASA/GSFC/Arizona State University, Details of lava tubes and lunar holes.

2.1.2 Mars' Recurring Slope Lineae

Mars has always been a controversial candidate planet for Earth-like life-hosting conditions: although the atmosphere is mainly composed by carbon dioxide and the surface is a cold desert, large permanent deposits of water ice have been discovered in both the poles. Scientists are still looking for other forms of water, even at larger latitudes. One of the most interesting, and, in this sense promising, feature of the Red Planet are the Recurring Slope Lineae (RSL): seasonal streaks of dust and small rocky particles collected along slopes [6]. Mars Reconnaissance Orbiter's (MRO) images (in Figure 2.3) suggest the presence of hydrated salts, which may indicate past or current availability of water [7]. In-situ measurements could shed light on the role



Figure 2.3: NASA/JPL-Caltech, Detail of RSL in Newton Crater

of those geological formations [8] and their possible connection with life able to cope with extreme environments. In this case, the terrain is particularly steep and unstable, definitely impossible to be traversed by a traditional space rover.

2.2 Classification of space robotics mobility

Space robot design started with the first space missions, and initially the robots were developed to be static. Their objective was to land roughly in a specific point and deploy instruments in-place. Technology improvements and the need of machines capable of doing multiple tasks in the same mission guided the transition to mobile robots. For some missions, static machines are still used, as in the case of JPL's InSight lander, which aims to investigate Mars' interior structure. However, most space rovers are mobile and employ wheels to travel.

2.2.1 Conventional mobility

The favourite configuration is by far the rocker-bogie (in Figure 2.4), where six wheels are distributed on a four joints structure that allows minimal displacement of the main chassis with respect to the surface contact points. This arrangement has been very successful for JPL's martian missions Mars Exploration Rover (MER) and Mars Science Laboratory (MSL), allowing them to travel long distances: Opportunity reached 43.1 km of total driving distance in August 2016 on Mars. Additional advantages of this system are the ability to surmount small obstacles without tilting the main body, thus increasing the range of traversable terrain configurations, and the possibility to turn the vehicle in-place thanks to individually steering motors in the front and rear wheels. However, as it can happen that a single wheel has to lift a significant portion of the total mass, motors usually have a high gear reduction, hence the speed is limited. Another issue that has emerged recently is related to control: while passing over an obstacle, the wheels can not run at the same speed, otherwise the middle wheel undergoes an intense radial stress by hitting the obstacle when the front wheel has just passed the obstacle. Without proper control the damage can be fatal to the rover mobility. Figure 2.5 shows Curiosity's wheels damage due to this.

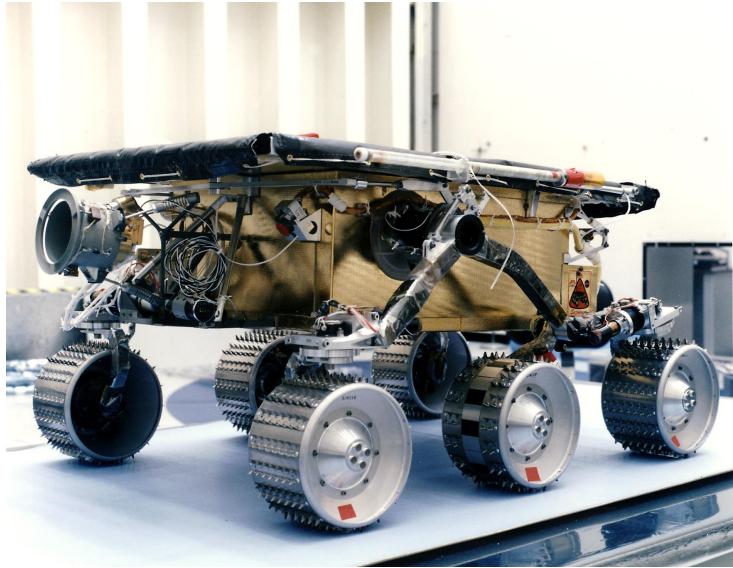


Figure 2.4: NASA/JPL-Caltech, Pathfinder rover showing the rocker-bogie mechanism.

2.2.2 Unconventional mobility

Other approaches involve the use of different propulsion systems and they can be divided in the following groups:

- *Aerial robots* can use propellers or passive mechanisms like blimps or gliders to move in the air. For this reason the use is limited to planets, as they have an atmosphere. Atmospheric composition and gravity make the design of such systems particularly difficult to be tested and validated. The first robot of this type is a drone carried and deployed by Mars 2020 rover and it has 1 m long blades to cope with the low atmospheric density (in Figure 2.6). Also, takeoff and landing can be hard to achieve on inclined terrains.
- Nature inspired mechanisms are the foundation of the *biomimetic robots*, which use arms, legs or even the body to imitate animal behaviors. They can walk, crawl, climb or swim. Usually the complexity of these systems and the high number of actuators make them not suitable as they are for space robotics, but research is still active in this field.
- *Hybrid mobility* is a combination of traditional systems with original techniques to produce more advanced rovers. Examples of this type are All-Terrain Hex-Legged Extra-Terrestrial Explorer (ATHLETE) from NASA, an hexapod with wheels as feet and VertiGo, a prototype of a

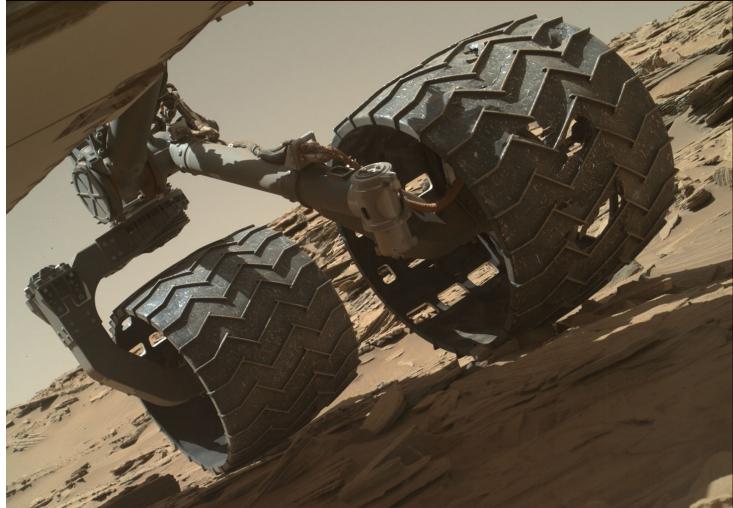


Figure 2.5: NASA/JPL-Caltech, Curiosity’s middle wheels are damaged (2017).

drone car that can climb walls thanks to propellers, developed by Swiss university ETH and Disney Research.

2.3 Tethered mobility

A possible solution to the challenges introduced by the described extreme environments are rappelling rover. The idea is to use a tether to add another degree of actuation to a wheeled robot. Most of the previous works involving a tether did not take advantage of it for mobility [9], but limited its use to supply the robot with power and long range communication capabilities. Instead, the force generated on the tether can be actively exploited for achieving precise mobility on these coarse terrains. The rover is connected to an anchor through the tether, and it can use it to descend a cliff or a steep slope. The anchor can be another rover, called mothership, that provides energy and communication to the orbiter. This configuration is called *mother-daughter* and has several advantages:

- The *daughter* can operate for long time without energy source nearby, because it can be supplied from the *mother* through the tether.
- The *mother* can host more than one *daughter*, providing redundancy in case of failure, or to support multiple missions at once.

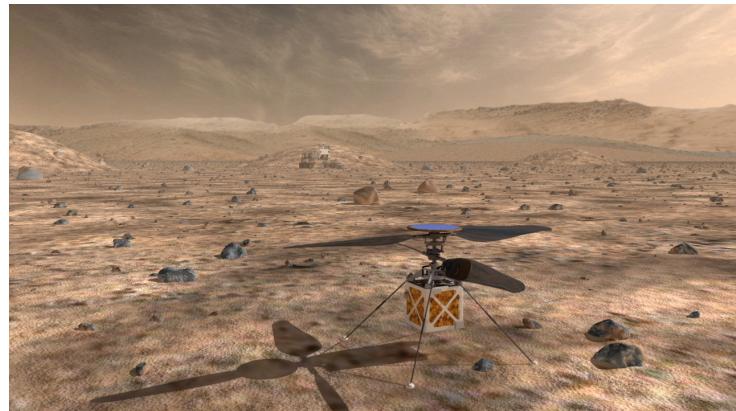


Figure 2.6: NASA/JPL-Caltech, Proposed Mars 2020 Elicopter.

- If mobile, the *mother* can move to different locations and deploy again the *daughter*.

This type of mobility also requires a limited amount of motors, which is preferable for low failure rate applications.

Chapter 3

Axel & DuAxel rovers

3.1 Axel

Axel has been developed with joint efforts from Jet Propulsion Laboratory and California Institute of Technology [10] as a possible extreme terrain rover, able to traverse cliffs, craters and steep slopes to take measurements in inaccessible places. The result is a low mass (between 30kg and 55kg), minimalist but versatile mobile robot that can be used in a *mother-daughter* configuration. It is composed by two groused wheels and a central body con-



Figure 3.1: NASA/JPL-Caltech, Axel on a field test.

taining the avionics and the spool. The tether comes out from a boom that can rotate freely with respect to the body. This serves many purposes: it can

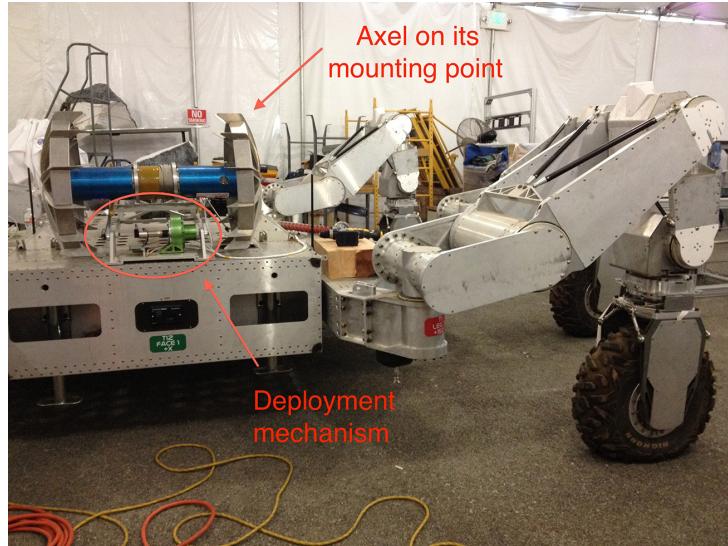


Figure 3.2: Second version of Axel prototype mounted on ATHLETE.

be used to move on flat terrains by generating a reaction force on the ground; it can substitute one of the wheels in providing motion or both of them for driving on a straight line; it can wind and unwind tether if the spool motor fails and it guides the docking to the *mother* robot. At the ends of the body, science instruments are packed in enclosures inside the wheels to save space and provide further protection. By rotating the central body, they can be oriented and deployed, also thanks to a four bar linkage that simultaneously lifts the panel and pulls out the instrument. A single Axel can carry up to 8 tools in its bays.

The choice of storing the tether inside Axel and not in the *mother* comes from the necessity to limit tether lifespan degradation. In fact, if the tether was fixed to the rover's end, it would undergo serious friction at each anchor point, i.e. contact point with the ground or obstacle. Instead, in this way, after a certain portion of tether is laid, it will not rub against the edges during Axel motion. The symmetric structure of the rover allows it to be fully operative when flipped over, making it particularly robust for uncertain geometry of the terrain. Extreme mobility is achieved by using the four degrees of freedom (two wheels, spool and caster arm) together. When on a cliff, the tether tension provides a vertical force, which combined with wheels traction, allows Axel to turn in-place. Moreover, the paddles on the wheels allow the rover to traverse obstacles having height up to the wheel's radius, without using any additional actuator.

3.1.1 Hardware

Axel has undergone many redesigns, and the latest one is the third iteration. In this version, four main motors provide mobility to the rover: they are brushless Maxon 167178, rated for 120 W. Gear reduction is different depending on the type of actuation. The spool motor and the boom motor have 840:1 while the wheel motors have 2353.3:1. There is also a motor for each instrument bay, that lifts the lid and extracts the tool, and they are slightly less powerful than the main ones. All the motors are driven by Elmo Whistle controllers rated 48V/10A, which provide also brake engagement and communication over a Controller Area Network (CAN) bus. Moreover, Axel's brakes are fail-safe because they engage without power, hold the motor in case of power failure. This kind of brakes uses permanent magnets to keep the rotor still, by attracting a locking cage. When current flows in the windings, the electromagnetic field pushes the cage away from the rotor, which becomes free to spin.

The body contains also many cameras to capture and map the environment and in the future a LIDAR might be added for improved accuracy. All the sensors on the body can be oriented by turning together the wheels and using the caster arm to produce a reaction force from the ground. This allows to reduce drastically the amount of actuators necessary for local traversing. Another onboard sensor is the Microstrain *IMU 3DM-GX3-25*, capable of outputting all the necessary values relative to inertia and accelerations.

All the sensors and actuators are mounted on the cylindric body's internal surface, leaving a void in the middle, along the axis between the wheels. This space is used for connection wires and airflow: in fact, at the ends, located in the center of the wheels, there are two fans pushing heat outside.

Axel's computation core is an Advantech *PCM-3362*, with an Atom *N450 1.67GHz PC/104+* processor and many communication interfaces, such as FireWire for cameras and CAN for motor actuation. Main power is provided through the tether, however a pack of 4 lithium polymer batteries (Thunder Power 22 V 8 Ah) can supply the rover for 8 hours, and they can be charged when not used. The motors power line has a switch that can be remotely controlled with a Beagle Bone board.

Although long range communication is supposed to happen on the system at the other end of the tether, Axel has three antennas for short range remote control and debug over WiFi. The tether, shown in Figure 3.3, includes four conductors: two for power and two for communication and an optic fiber, which is not used yet. Proposed usage involves obviously communication, however recent ongoing studies are investigating the ability to reconstruct the tether position in the space and the stresses on the anchor points thanks



Figure 3.3: Tether inner structure: its construction leads to an increase of stiffness when under strain.

to light diffraction in the fiber. The two communication conductors are used as two channels for a full-duplex, single-ended Ethernet transmission line. The power line runs at the Axel’s power bus voltage, but this is still a controversial decision: while a higher voltage would reduce the losses along the line, a lower voltage keeps the tether warm thanks to the thermal losses, and this could be vital in cold environments, where it could be severely damaged by extremely low temperatures.

3.1.2 Software

Although Axel’s software is centralized, and runs on the main computer, some low-level code is executed on the scientific instruments and motor drivers. In particular, Elmo controllers run an embedded software able to communicate over CAN bus with the *SimplIQ* protocol and drive the brushless motors. It is possible to set many values such as position, speed, torque or armature current depending on the required type of control.

SimplIQ protocol is built over the CANOpen stack: Elmo expose a particular register as a Process Data Object (PDO), both writable and readable on two different mappings. By writing on this register different parameters can be set on the controller, including starting and stopping signals. This topic is described in depth in Section 4.4.1.

The main computer’s operative system is Ubuntu Linux 12.04.2 LTS, but an upgrade being carried out at the time of writing brings the version number to Ubuntu Linux 16.04 together with more performant libraries and Robotic Operative System (ROS) support on more recent boards, the NVIDIA Jetson TK1 and NVIDIA Jetson TX2. The hardware upgrade involves also

a new CAN interface board, the PEAK-System PCAN-miniPCIe, that can drive two different buses at the same time. Moreover, the Linux kernel has been compiled with kernel preemption flag enabled to achieve soft real-time capabilities.

3.1.3 CLARAty

The core of the software is CLARAty [11] (Coupled Layer Architecture for Robotic Autonomy), an in-house backed and developed framework for high-performance robotic applications. It is mainly written in C/C++ in conjunction with small parts of Python, Perl, assembly and other minor languages. It is compatible with many hardware architectures (ARM, x86, x64 and others) and operative systems (Linux, Solaris, MacOS, VxWorks). CLARAty also provides an interface for ROS to use all its libraries, especially the navigation and planning algorithms.

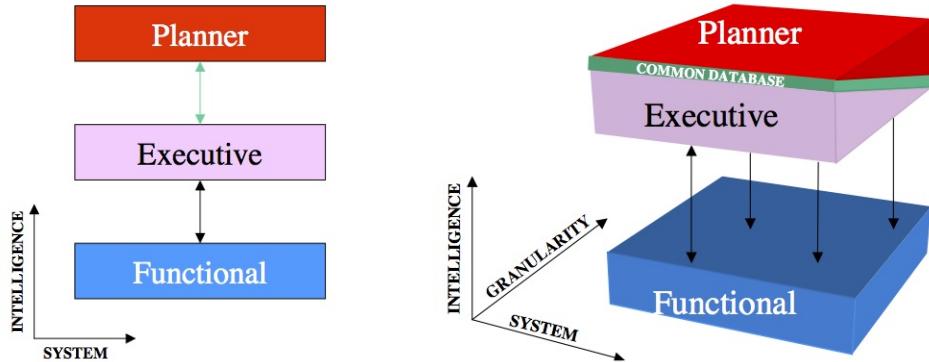


Figure 3.4: Robotic framework architectures: on the left a traditional framework structure, on the right CLARAty structure.

CLARAty structure, represented in Figure 3.4, aims to code reusability and layerization through modularization. While many other robotic frameworks organize functionalities in 3 levels of abstraction, CLARAty uses a 2 levels architecture:

- The lower layer is called *functional*, because it exposes all the functionalities of the hardware system. It aims to offer a standardized interface for different hardware platforms. This is achieved thanks to a further layerization: modules can be of *hardware* type if they define the specific way to handle a certain component, usually they include protocols; instead, modules are *software* if they provide access to the

relative *hardware* through standard methods, as an interface to a class of components.

- The higher layer is called *decisional* and it can include internal stratification of planning functionality. The peculiarity of CLARAty is that in this level, each one of the planning agents can access the whole *functional* layer, with the goal of minimizing inconsistent information between layers.

The role of this work does not involve any part of the *decisional* layer, because the goal is to provide a access to an hardware subsystem in Axel. However, ultimately *decisional* layer will benefit from these results.

Module categorization in CLARAty follows the level of abstraction, and it is possible to identify three groups within the whole CLARAty repository (Figure 3.5):

- The *hardware branch* includes all modules that provide access to specific hardware, by means of protocols and communication specifications. Modules in this group are usually tied to small families of components and they are the most critical in term of real-time requirements.
- *Robotic branch*, also called *functional branch*, collects all the abstraction functionalities underlying the higher levels. It includes many libraries containing mathematical definitions and algorithms, such as motion control, image transformation, etc.
- The *application branch* contains the modules required for specific applications, like the core modules for a certain robot, which start and manage all the other modules.

3.2 DuAxel

While Axel can be sent directly to the target and be deployed from the landing point, it might be preferable to use it as part of a bigger rover that can traverse graceful terrains faster and be the anchor during the descent. For this reason, the team developed a coupled configuration, where two Axels are connected by a central module, composing all together a four wheel system called DuAxel, whose prototype is shown in Figure 3.6. DuAxel is able to traverse plain terrain and mild slopes, thanks to the flexible mobility. In fact, the housing mechanism for Axel allows other degrees of motion: each Axel can roll and yaw around its mounting point. When close to the target, around

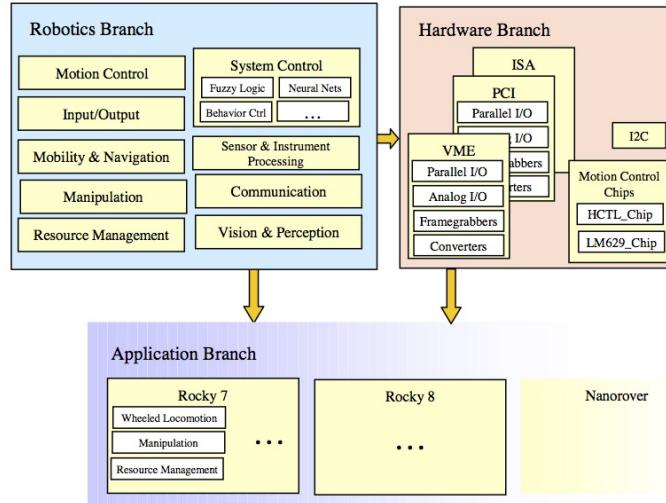


Figure 3.5: Organization of CLARAty modules.



Figure 3.6: NASA/JPL-Caltech: A prototype of DuAxel.

some tens of meters, DuAxel can stop and deploy one Axel, providing it power and long-range communication. In fact, Axel could traverse environments with few or no power sources, like a Moon's lava tube, and it can not carry all the transmission system on itself. The central body of DuAxel is supposed to host power systems such as solar panels, and many other components that can not fit in the slim body of Axel. However, the main advantage of this configuration is the redundancy: the second Axel can provide a fall-back rover in the case the first one is lost or broken. To preserve DuAxel mobility in such cases, it is possible to add castor wheels to the main body, close to the Axel's mounting points, so that the system becomes a differential driven mobile robot. Investigation on this system is still going on, and results are expected in the close future. Certain is that the choice of putting the winch

mechanism in Axel instead of the central body, makes the development easier and DuAxel more failure-proof.

Chapter 4

Tethered mobility test bed

4.1 Overview

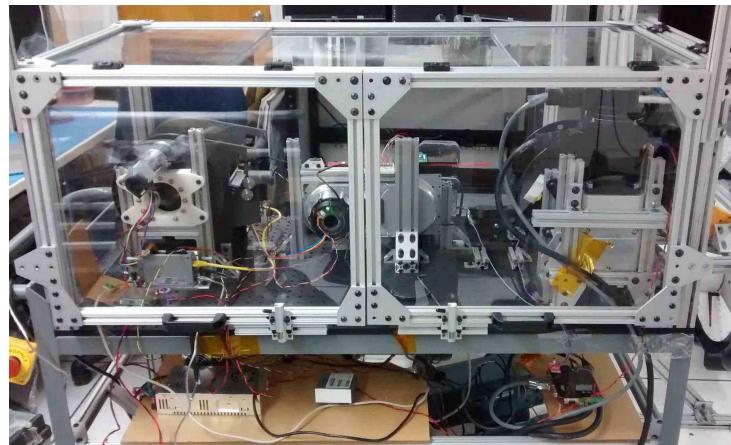


Figure 4.1: TMS test bed in its safety enclosure.

The validation of the system has been carried out on a purpose-built test bench rig capable of producing a controlled load on the tether and measuring all the relevant values. Figure 4.1 shows the complete build and Figure 4.2 reports a sketch of the system. After this in-lab validation, the TMS is supposed to undergo a redesign aiming at integrating it in the rover's body so that a field test can be done. The software is built on top of CLARAty framework, introduced in Section 3.1.3.

4.1.1 Subsystems

The test bed is composed by three subsystems:

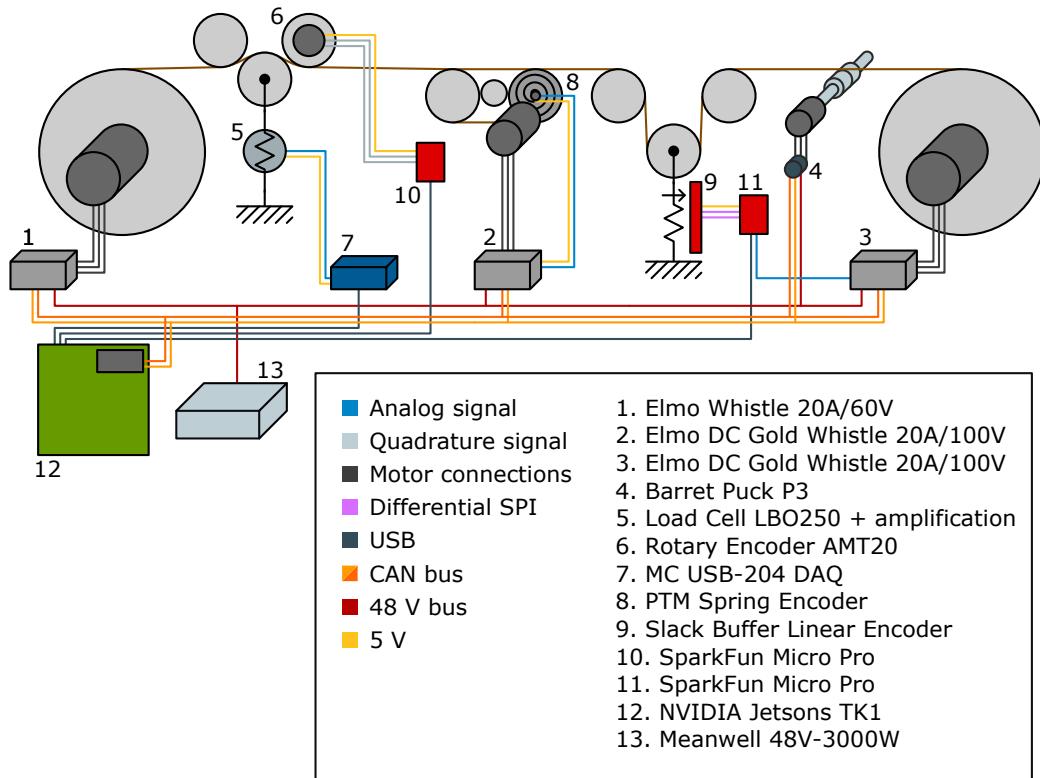


Figure 4.2: Test bed scheme.

- **TMS:** the actual system integrated in Axel.
- **External load:** a spool actuated by an industrial motor.
- **Sensor module:** an assembly of sensors placed between the load and the TMS.

It is worth mentioning that the modular nature of the system makes possible to rearrange the positions of the components freely; in fact, all the parts are mounted on a base plate with a grid of threaded holes. This is useful to do tests with any possible length of unreeled tether and to arrange components in a different way. For example, a box containing sand can be put between the load motor and the TMS to evaluate the degradation of the tether due to dust collected in thecurls.

4.2 Specifications and capabilities

Many experiments have been executed on the test bed; they can be divided in two groups:

- Axel's tether management system validation
- Tether testing and profiling

For both of them, limit values have been identified to constrain the selection of the components. Estimation of the tether's maximum stress requires to consider the most intense threat, that is a free fall of the rover. This can happen for many reasons, such as an anchor point failure, a slip between wheel and ground, or the overturning of the rover, as explained previously. Axel's current weight is approximately 50 kg, and gravity on target planets or satellites is less than Earth's. Hence, using a conservative approach and assuming Earth's gravity acceleration, a tension no higher than 500 N is expected, with the rover fully hanging. Moreover, a safety factor of 2 is introduced to have enough robustness, totaling 1000 N.

The long term mechanical resistance of the tether has to be evaluated, and this can be done by running the tether for a certain time through sand and dirt. The speed of the tether is limited to 0.50 m/s, which is the maximum spooling speed, but the load motor has to reach higher values to simulate a free fall. The faster the load motor is, the bigger the simulated free fall range is. Final speed v_f after a free fall of h meters is the following:

$$v_f = \sqrt{2gh}$$

Hence, for 1 m fall, the rover would reach a speed of 3.13 m/s, and this would require a mechanical power of 3130 W to simulate the upper limit of the tension range because

$$P = Fv$$

where P is the power, F the linear force and v the speed.

4.3 Hardware

4.3.1 Load motor

Since the load motor has to satisfy the requirements for the testing and has to run with a 48 V bus (for safety reasons), the selected motor is a Trust Automation SE650-1000. Although the SE650 is rated for a much higher armature voltage, around 165 V, it naturally runs at low speed when on 48

Mode	Empty spool (4 cm)	Full spool (8cm)
Continuous mode	600 N, 0.5 m/s	300 N, 1 m/s
Intermittent mode	1800 N, 0.45 m/s	950 N, 0.9 m/s

Table 4.1: Maximum torques and speeds for the load motor.

V, and requires a gearbox of 7:1 to produce about 1 m/s speed on the tether. With a small ratio gearbox, the reflected inertia is small, allowing a more precise control of the load motor. This has been the deal breaker that made the SE650 preferable over other motors with 48 V rated armature windings, such as the BG95X80-CI from Dunkermotoren.

In fact, the reflected inertia I_r is

$$I_r = I_l N^2$$

where I_l is the load inertia and N is the gear reduction from Axel system to the load.

While SE650 requires a 7:1 reduction for 1 m/s maximum speed, the BG95X80 necessitates a 35:1 reduction. Respective rotor inertias are $1.29 \cdot 10^{-6}$ kg m² and $1.89 \cdot 10^{-7}$ kg m², resulting in reflected inertias of $6.32 \cdot 10^{-5}$ kg m² and $2.31 \cdot 10^{-4}$ kg m².

Another important aspect to be considered is the load spool radius, which acts as a variable ratio gearbox: it is expected a radius ranging between 4 cm and 8 cm from the center of the shaft. Hence, the capabilities of the load motor are the ones in Table 4.1. The intermittent mode is up to 3 seconds, to not overheat the motor. However, this is not an issue because the maximum tension is needed only for free fall simulations, which last a short time. The gearbox is a 7:1 Neugart WPLE120 with low backlash to maximize efficiency.

An Elmo Whistle 20A/60V drives the motor, and communicates with the central computer through the CAN bus. The control parameters are derived with the *Composer* software from Elmo, using its automatic tuning procedure. Priority has been given to performance: to obtain the minimum possible rise time.

To compute the motor profiles, plotted in Figure 4.3, DC motor equations can be used, because apart from the commutation brushless and DC motors have the same electrical behavior.

$$T_m = k_t I_a \quad (4.1)$$

$$V_{emf} = k_v \omega \quad (4.2)$$

$$V_a = R_a I_a + V_{emf} \quad (4.3)$$

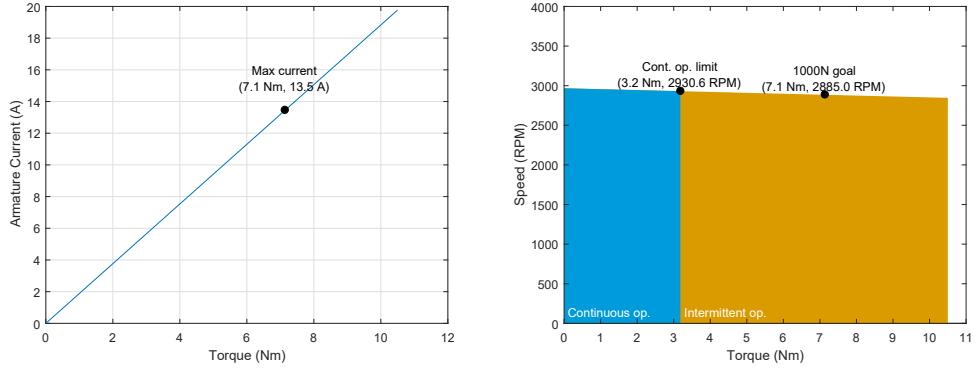


Figure 4.3: Load motor profiles: on the left armature current, on the right the motor characteristic.

where T_m is the mechanical torque produced on the rotor, I_a , V_a are the armature current and voltage respectively, V_{emf} is the voltage drop on the armature due to magnetic field induced currents flowing in the opposite direction to the applied current and ω is the speed of the rotor. k_t and k_v are the torque constant and voltage constant respectively.

Although they have different SI units of measure, they represent the same physical quantity. In fact, it is possible to multiply the first member of (4.1) with the second of (4.2) and the second member of (4.1) with the first of (4.2) to obtain:

$$k_v \omega T_m = k_t I_a V_{emf}$$

then substitute V_{emf} from (4.3)

$$k_v \omega T_m = k_t (I_a V_a - R_a I_a^2)$$

but the left member is the product of rotational speed with respective torque, which is the mechanical power produced by the motor, P_{out} . Instead, the left member is the net electrical power by the motor:

$$k_v P_{out} = k_t (P_{in} - P_{losses})$$

and considering the balance equation of the power

$$P_{in} = P_{out} + P_{losses}$$

it is trivial that

$$k_v = k_t$$

These constants are used to compare motors, because they are a measure of how the power is transformed in the motor. Basically, a lower k_t is

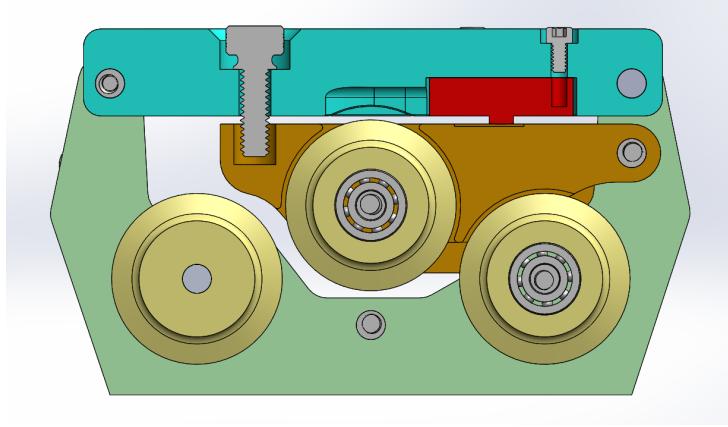


Figure 4.4: The load cell sensor assembly: when the tether is taunt, the middle sheave pushes on the load cell (in red) through a lever.

characteristic of efficient and fast motors: the inertia of the rotor is usually low and so is the voltage rating. This is also limited by the insulation of the armature windings, which cannot go below a certain thickness, otherwise it would fail.

4.3.2 Sensors

The sensor package contains two sensors:

- **Load cell LBO250 from Transducer Techniques** to measure the tension on the tether through a pulley system
- **Quadrature encoder AMT20 from CUI Inc.** to measure the position of the tether

The load cell has a maximum capacity of 114 kg, and considering the lever in the sensor assembly the capacity is around 200 kg (Figure 4.4 shows its geometry). This means it can take measurement up to 2000 N, vastly over the expected range of tension. As interface to the computer the Measurement Computing USB-204 DAQ has been selected, providing 12-bits of resolution over 5 V, which is also the supply voltage for the load cell. The output of the load cell can go up to 10 mV, so it required a *INA125* instrumentation amplifier to comply with the DAQ input range.

An instrumentation amplifier is the circuit represented in Figure 4.5. Its function is to amplify a voltage difference and the input/output relation is

$$V_{out} = \left(1 + \frac{2R_1}{R_{gain}}\right) \frac{R_3}{R_2}$$

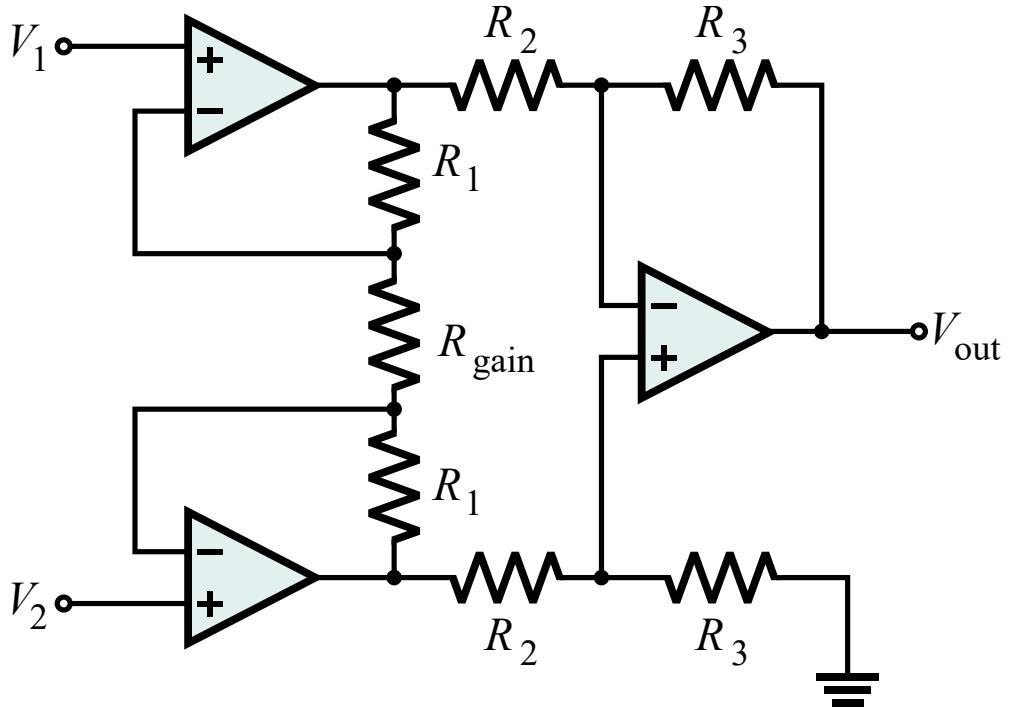


Figure 4.5: Wikipedia: Instrumentation amplifier circuit.

In the *INA125* package the R_{gain} is not present, but its terminals are exposed, so that the gain can be customized depending on the design's need. In this case a potentiometer has been used to test different gains, and consequently, resolutions. After calibration is done, the potentiometer can be replaced with a resistor having the same resistance. With a gain of 500x, a 4.8 V amplified output corresponds to 980 N on the tether, and 0.25 N resolution after the DAQ.

Position measure can be taken with encoders (Figure 4.6), which are called *linear* or *rotary* depending on the type of geometry. Moreover, encoders are *absolute* if their measure is taken with reference to a certain set-point, or *incremental* if they measure with reference to the last measurement. For this application, an incremental encoder seems to be more suitable because first, it is difficult to define a set-point on a tether, and second, it may be possible to end with huge values even when working on a small fraction of tether that is far from the set-point.

However, the use of linear encoder is not possible in this application, because they usually require a sensor or a susceptible component mounted

on the object whose position has to be measured, that is, in this case, the tether and it can not be modified.



Figure 4.6: Encoders: on the left a linear encoder, on the right the rotary encoder that has been used (AMT20).

In conclusion, the choice is constrained to a rotary incremental encoder, and the relation between rotational position and linear position is trivially derived. This type of sensor produces two square waves that represent the output of internal switches. These switches can have different working principles, but basically there are two types: some have magnets and Hall effect sensors, others have an opaque disk with transparent parts and a sensor which measures occlusion.

Switches are located along a circumference of a certain radius around the shaft, and there is a small displacement between them, so that one is activated with a small delay with respect to the other. The output signal is a pair of square waves having a phase difference: their frequency is proportional to the rotational speed of the shaft and the switching sequence defines the direction.

The quadrature encoder is interfaced to the computer by a Sparkfun Micro Pro, running a program that reads the quadrature signals and stores the counts in the memory. The program relies on interrupts that are activated when the edge of the square wave on each pin changes: the interrupt routine first checks the direction by comparing the couple of signals with the previous measurement, then increases or decreases the counter in a register. The main computer can query the register or reset it to zero. The resolution is 0.31 mm and the maximum speed, limited by the interrupt frequency of the microcontroller, is 30 m/s.

4.3.3 Controllers and communication

The main computer board is a Nvidia TK1 running Ubuntu Linux 14.04 with a PEAK CAN mini PCIe interface. The load motor controller is connected through the CAN bus, together with all the Elmo controllers of the system. Both the USB-204 DAQ and the Micro Pro communicate by USB connection, with HID drivers for the former and in serial mode for the latter.

4.3.4 Power supply

Power is provided through separate and independent buses at different voltage levels. There are multiple reasons for this:

- There is better noise rejection, because the noises on the three buses interfere.
- In case of an emergency stop only the high power line is switched off, keeping all the controllers safe ad running.
- When the motor loads decelerate, the shunt power can not be transferred to logic supply lines and destroy them by overvoltage.

The three buses with the respective power transformers are the following:

- **Meanwell RSP-3000, 48 V:** supplies the motors through their respective motor controllers
- **Meanwell WDR-120, 12 V:** supplies some motor controllers logic circuits, the amplification stage for the PTM spring's encoder and the main computer
- **Meanwell HDR-15, 5 V:** supplies the load motor encoder and hall sensor because the Elmo Whistle is not capable of outputting enough power for them

The 48 V bus is expected to receive the regenerative energy from the motors when they are decelerated by an external force. This problem can be solved either by putting a shunt resistor with the corresponding switching circuit for each motor controller or adding a battery to the bus. In this case, the battery solution has been used because of its simplicity and to recover the energy.

For safety reasons, an emergency-stop button has been added for the 48 V line. When closed, it supplies current from the 12 V bus to a solenoid that mechanically closes the high power line.

4.4 Software

The test bed is managed by a program executed on the TK1. Responsibilities of the central software are to start the Axel's TMS, drive the load motor and log all the necessary data. Obviously, a fast logging is required to have a good validation of the system. The main limit is the CAN bus bandwidth, together with the CAN software protocol SimplIQ, which has a roundtrip time of 2 ms, on average.

As presented in Section 3.1.3, Axel' software is built on top of CLARAty robotic framework, so it seemed natural to follow the same path for the testbed, to minimize code production and reuse many functions that have been coded for the same hardware. During the development of the control software, many CLARAty modules have been written and added to the source repository at JPL.

4.4.1 Elmo Whistle module

CLARAty already offered an implementation of a software stack for Elmo controllers through CAN bus. However, the code was not yet updated to work with the updated class called *Bus Manager*.

The need of a *Bus Manager* emerges when on the CAN bus there are many controllers working at the same time: as the TK1 has a multitasking operative system, many agents can require to access the CAN bus concurrently. The previous version of the Elmo module worked with a request-response fashion: the request for the controller was prepared, then sent over the CAN bus, finally the module waited in a busy way for the response. Busy wait means that the process does not release the CPU during the wait. Of course this is first a waste of resources, but also it keeps the bus from being used while waiting for the response. Despite the old version had a sort of protection for multiple agents sharing the bus resource with a mutex lock so that only one agent could use the CAN bus at once, it was only limited to the Elmo class, so it worked when on the bus there were only Elmo controllers.

A recent improvement to CLARAty framework has been the addition of the *Bus Manager*: it has exclusive access to the CAN bus and manages the read and write queues on it. *Bus Devices* are subscribed to an instance of *Bus Manager*, which handles packets produced by the *Bus Devices* and sends them at the right time. Meanwhile, it continuously reads the CAN inbound buffer and provides the received packets to the respective *Bus Device*.

The *Bus Manager* has basically two always-running threads with respective queues: one for sending packets and one for receiving packets. The queues are circular and their size depends on the effectively used bandwidth

of the CAN bus. It means that they have to be scaled accordingly to the number of devices on the bus and the frequency of the messages. With 4 controllers on the network and a rate of 600 Hz each one (100 Hz position, speed, torque, set-point, 50 Hz current, fault register, control mode, status register), the total bandwidth is 2400 CAN messages per second. Average message length is 4 bytes and each message requires a total of 77 bits to be transferred, thus the CAN bus utilization percentage is 18.5%. In fact the bandwidth of the CAN bus is 1 Mbit/s for short lengths (<1 m), as in this case.

The controller area network (CAN) bus was invented in 1983 by Robert Bosch GmbH, with the purpose of connecting all the electronic devices inside vehicles. The goal was to create a system able to work in noisy environments, and to be resilient to network faults, such as a node not able to communicate. After some years, this approach turned to be successful and has been increasingly employed in and outside the car industry, generating many versions of it and even some international standards.

The CAN standard (ISO 11898) defines the lower 2 layers of the OSI model. The physical mean of transmission is a twisted pair of conductors, which carry a differential signal to cancel out the electromagnetic noise. Neglecting the differential aspect, the signal has two voltage levels: low and high. High voltage is used for the *dominant* bit, that is 0. On the contrary, low voltage is 1, called *recessive*. The meaning of *dominant* and *recessive* comes from the fact that if two or more line drivers are trying to set two different levels, the resulting voltage is high. Transmission voltages usually range between 0 and 5 V.

The higher level defines how data is encapsulated in frames, which are indivisible and only one frame can be transferred at a time. As many other multi-point network types, information on the bus is available to all the nodes: it is their responsibility to use only the relevant ones and discard the others. The frame is a sequence of bits and contains different parts.

The *arbitrariation field* is the first part of the frame: it contains a sequence of bits with fixed length. The first CAN standard has 11 bits in this field, but a newer revision, called *extended CAN*, has, instead, 29 bits. Its function is to decide which frame has to be transferred: all the nodes that have to transmit a message start with pulling the line to the voltage defined in the most significant bit (MSB) of the arbitrariation field. When a node pulls the line high, due to a 1 bit in its frame, if there is any other transmitted frame with a 0 bit in that position, then the node stops the transmission and listens the message, following the principle of the *dominance* and *recessivity* pair. The consequence is that this rule prioritizes the frames having a low *arbitrariation field*, while it penalizes high ones.

After the *arbitrariation field*, there is a *header*, containing information about the length of the message, message type and integrity, communication status like acknowledgements or requests for response. At the end of the frame there is the actual data of the message.

On the top of this convention many software stacks have been produced for different purposes. Between those, the CANopen protocol was developed for industrial applications and has been standardized by CiA-CAN in 1994. CANopen considers each device on the network an individual entity with an address and an *object dictionary*(OD), which is a collection of registers that can be read or written through the CAN bus. Each of these objects has its address, composed by index and sub-index.

In CANopen, the *arbitrariation field* is used as a combination of the device address and the object address, so that there are more important devices and more important objects. With an 11 bits field, the first 4 are for the object and the last 7 are for the device. This rule limits the total number of nodes in the network to 127, with the standard CAN.

There are many families of CANopen devices, and each family defines some mandatory objects to be supported by every device. The most important family is *DS301*, and it is the basement for other families. The *DS301* divides the objects in two groups: service data objects (SDO) and process data objects (PDO). SDOs are used directly access the register represented by the object, thus they can be *downloads* in the case of writing, or *uploads* in the case of reading. The use of SDOs requires two messages for each access: one for request and the other for response. The downside of SDOs is the overhead introduced for each access. For this reason PDOs have been implemented: they are a faster way to set or query a register in the node. Writing can be done with only one message, but the node has to be previously instructed on how to map the specific PDO with a register, and this can be done with a SDO download. Another advantage of PDOs is that they can be sent not only with an explicit request but with an external trigger like a timer, a synchronization message, or a physical event.

Elmo controllers fully comply with the *DS301* standard, however they also expose two particular objects to control the motor in a more human-friendly way: the *OS interpreter* and the *binary interpreter*.

The *OS interpreter* is basically a text console: when writing to the register, Elmo interprets the text as a command of its scripting language and executes it, then it sends the response back. Unfortunately this way of communication is characterized by an huge overhead, and it is not preferable for fast control. However, it is necessary to start embedded programs on the Elmo and to request certain data, like text sequences, to the controller.

On the other hand, the *binary interpreter* uses short payloads (8 bytes for

set and 4 bytes for query) to modify parameters in the drive. Each parameter is identified by a pair of letters and an index: for example, UI[1] accesses an integer variable stored in the RAM, that can be used by programs running on Elmo. In this case, UI is the array register of the 'user integers' and 1 is the index (Elmo uses indexing starting from 1; if the register is scalar, then this number is 0). The *binary interpreter* follows the convention of *SimplIQ*, which is the scripting language of Elmo controllers.

The previous Elmo code base in CLARAty was written basing on the *binary* and *OS interpreters*, then, to reduce the amount of code to be rewritten, the new code recycles the functions that encode and decode data for those protocols. In the future, a transition to the CANopen standard can be done, using PDOs to achieve maximum performance.

Elmo control stack is composed by two different CLARAty modules: the *hardware module* and the *software module*. Underlying reason is to preserve the differentiation between the specific hardware and the motor interface. In fact, by its nature, CLARAty provides an interface class which represents a generic motor and exposes attributes and methods like positions and set-point commands respectively. This allows code reusability and portability, easier updates, and a more layerized level of control.

As the *hardware module* represents the physical controller, it has been natural to define a table of registers: they are the bricks of the *SimplIQ* protocol. The table tries to replicate the status of the registers in the Elmo: a set of parameters, that can be chosen during the class instantiation, is periodically updated. In this way, the table acts as a cache for the periodically updated values, and the time to access the information becomes short.

However, timing is critical, and the implementation of the table plays an important role in this aspect. To reduce access time, the table has been implemented as an *HashMap*, whose index is an enumerator corresponding to each register. The result is a $O(1)$, because all the possible entries are already in the table, as they are predefined by the protocol.

The *hardware module* inherits directly the *Bus Device* class, so that it can use the CAN bus. The father class is characterized by a virtual method which is called by the *Bus Manager* after it receives a packet ready to be parsed by the device. It follows that this method is time-critical, because the reading pipeline of the bus can get stuck or miss packets if the parsing requires too much time.

To correctly insert the received information in the respective register of the *HashMap*, the class reads the first bytes of the packet, because they contain the two letter code and the index of the data. Then, register access is again critical, but this time the enumerator indexing can not be used, as the only reference available is the combination of characters and the index.

Solution of this problem is another `HashMap`, having as keys hashes of all the fields of the address and as elements the enumerator values. This allows an update of the registers' `HashMap` with $O(1)$ complexity. Both the `HashMaps` are built during class instantiation and it makes the loading slightly longer, but they guarantee good performances on run-time.

The retrieval for motor's data can be periodic or aperiodic, as explained before. Periodic requests rely on one of the *Bus Manager*'s features: it exposes a scheduling function, allowing to set the frequency, the priority of the message and the behavior in case of a missed deadline. Then, for some important parameters, such as encoder position and encoder speed this saves time, because the values are cached in the *hardware class*' registers. Instead, aperiodic requests generate a sporadic query packet and they wait to receive the response from the drive with the updated value.

Performance optimization of the code has required a deep analysis of the communication stack, from the socket interface, offered by Linux kernel, to the specific class. The resulting class has introduced an asynchronous behavior and a concurrent access to the CAN bus through the *Bus Manager*, whereas the old implementation used in an exclusive way the bus.

4.4.2 Level winder motion module

The fair lead motion is constrained by the movement of the spool's drum: for a certain displacement of the drum, there is a certain position of the fair lead. This relation is only depending on position, and no other value comes into play.

By instinct, it could be natural to assert that the position of the fair lead should depend also on the amount of tether in the spool, because with a bigger radius the fair lead moves faster. This is true, at least in the basic assumption that the tether moves faster when the radius increases, with the same angular speed. In fact this holds because the linear speed is product of radius and angular speed

$$v_{tether} = r_{spool}\omega_{spool}$$

However, the linear position of the fair lead depends on the number of turns in one level only, and this is always the same. Thus a possible open-loop control can take the spool position as input and output the fair lead position. There are many options to build this relation: it can be done either with an *incremental* or *absolute* approach. The former one handles the positions with reference to the previous positions, so the algorithm has to save the previous state of the system. The latter, instead uses a static function that does not

involve previous states to determine the current set-point. In comparison, the *relative* version is more sensitive to errors because the error cumulates over time in the states, then the *absolute* approach is to be preferred.

This type of relation can be described by the plot in Figure 4.7.

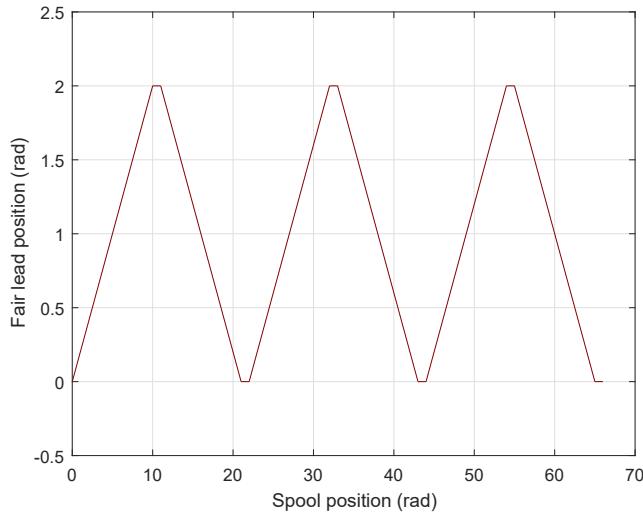


Figure 4.7: Fair lead position depending on spool position.

The produced CLARAty module is built on the *Periodic Task* class: in this way, a certain function can be executed with a precise frequency over time. In this case, the function first reads the position of the spool motor in radians, then it computes the respective position of the fair lead. It can be noted that the function represented in Figure 4.7 is periodic, and the period is the double of the sum of all the windings in one turn (because the fair lead goes back and forth).

To make the algorithm more flexible, many parameters have been exposed in the constructor to customize the behavior depending on the starting conditions, which are position and winding direction.

The algorithm has two parts: one for initialization and one for run-time calls. During the initialization the offset of the first period of the trapezoidal wave is found. The period starts with the tether being on one edge and ends when the fair lead reaches the other edge. The time spent on the edge is considered at the beginning of the period. It is worth to mention that there are two types of alternating periods depending on the direction of the fair lead. Assumed an initial direction, the two types of period occur in alternate fashion, so to compute the type of period from the current position, it is necessary to find the amount of the periods from the starting one. If it is an

even number the direction is different, otherwise it is the same.

The bounding positions of the fair lead are defined during the calibration and homing procedures: the fair lead travels with low speed and torque to the edges until it is stopped and saves the position in radians. A file containing these values helps to avoid the recalibration every time the system is repowered.

4.4.3 Measurement Computing USB-204 module

The USB-204 Data Acquisition (DAQ) device connects to the TK1 through a high speed USB port and provides full functionality with the Measurement Computing software. It is able to perform both single and multiple readings with different frequency. On the other hand, such application can not be used in CLARAty for two reasons: first it is made for Microsoft Windows, and second it does not provide any way of interfacing to another program (API).

The producer of the USB-204 provides on its website an open-source library for Linux systems, which has been used to build the CLARAty module. Although the library is flexible and could theoretically support all the capabilities of the device, only the relevant features have been developed.

In fact, the module exposes two functions: one is for a single read and the second is a fixed amount of reads with a certain frequency. In the first case the frequency limiting factor is the USB bandwidth, that imposes a cap at 100 Hz. With the second function, the DAQ is able to buffer the measurements on its internal memory and transmit them over USB at a slower pace. In this case the limit is at 200 kHz.

4.4.4 Test bed module

The test bed module is the point of access to the whole test bed and tether management system. It provides human-machine interaction through a text interface. The goals of this module are the following:

- Perform the required experiments and gather data about the system
- Automate repetitive tasks, reduce the amount of calibrations and set-ups
- Expose relevant data for debugging and inspecting the system
- Guarantee flexibility of operations, without the need of writing new code

The main limitation is the absence of a terminal: there is neither a screen nor an input device. Also, given the necessity to stay away from the machine during the operation, remote control would be preferable. Thus, because graphical interfaces are usually heavy and difficult to adapt with many types of screen, the need of a text interface is motivated. In fact, this interface can be shown through a remote terminal in a connection like secure shell (SSH). The module employs the standard C input/output library to communicate with the user.

During the initialization process, the software checks the status of the system: in case of a missing device, like a motor controller that has not been powered on, the program halts and notifies the user. In case of an optional component, such as the load cell for tension measurement, it gives only a warning.

The core of the module is the *profiles* system. It is a solution that tries to be a flexible framework to define and run experiments on the test bed. A *profile* is a text file containing a timeseries of set-points for the motors.

The following is an example of *profile*:

```
LOAD_MOTOR
# -name- controlmode controlparams
MODULE -First module- P 0.1 10 -3
# time, value
0,0.99965
0.001,0.99965
0.002,0.99964
0.0030001,0.99963
0.0040001,0.99962
SPOOL_MOTOR
MODULE -Second module- T 0.2 9 -3
0,1
0.1,0
MODULE -Third module- S 1 0.1 0
0.2,0.1
0.5,0
```

The file is composed and organized by *modules*: each one of them is characterized by a control mode and a set of control parameters and it is defined for one specific actuator. The possible actuators are:

- *LOAD_MOTOR*
- *SPOOL_MOTOR*

- *PTM_MOTOR*
- *LEVEL_WINDER_MOTOR*

Instead the control modes are:

- *P*: position in *rad*
- *S*: speed in *rad/s*
- *T*: torque in *A*

The set of control parameters is used to calibrate the motor control loop, which is based on a Proportional Integral Derivative (PID) control. The structure of the *profile* has always at least one actuator with one module, but more can be inserted. Inside each module there are set-points, composed by a time and a reference value, separated by a comma. The script system supports also comment lines starting with "#".

When a *profile* is loaded by the program, this creates a table including all the set-points in a single timeseries array, which is ordered, obviously, by time. The table has then a sequential access available to the software. Then, the *profile* starts running, another thread with a busy loop is created, and the busy loop continuously checks if it is time to send the next command to the respective motor driver over CAN bus. At the same time, the logging happens on a different thread.

Among all the tests, the one that has been favored by this feature is the sine sweep test, in which the load motor follows a sinusoidal trajectory in the position, but the frequency of the sinusoid changes over time, from slow to fast.

Chapter 5

Tether management system

5.1 Overview

Previous research has shown that active tether control is necessary for precise control over the trajectory of the rover [12]: slackness can cause entanglement, instead Axel could be flipped over in case of a too tight tether. Other work investigated knifing as a failure mode of the spool mechanism [13]. It happens when a layer of tether goes under a lower level and this holds the tether while unwinding. The solution to knifing seems to be a constant tension applied to the tether; experiments were carried out in the range of 50N, and they proved higher quality of spooling with increasing tension. However, the conclusion is that requirement of controlling spooling tension is mandatory for a safe and long lasting usage of the winch. Another important failure mode related to the tether can occur if an anchor point slips: Axel would fall freely until the tether goes slack, then both the rover and the tether would undergo a significative shock load. Because the tether has a limited compliance as it hosts signal lines, this could result in a serious damage on either the tether or the spooling mechanism. As linear stiffness is inversely proportional to the amount of laid out tether, it is possible to define a value of safety for the system, called *fall factor*, F :

$$F = \frac{h}{l_t}$$

here h is the free-fall height difference and l_t is the *effective length* of the tether, that is between Axel and the last anchor. It is not necessary to consider the other part, as the anchor point can be considered fixed by the friction. Moreover, assumption of this work is that the maximum distance of an anchor point is likely 10 m, basing on terrain geometry data from Mars' orbiters. The allowable fall factor for a certain type of tether depends on its

compliance: stiffer tethers are more sensitive to shock loads and they can not handle high impulsive forces. This is described by the following equation:

$$F_{max} = \frac{T_{max}^2}{2gmEA}$$

where g is the gravity acceleration, m is the mass attached to the tether, T_{max} is the tension limit, E is the Young's modulus and A is the cross sectional area. In its current version, Axel weights 54 kg, and the tether has an axial rigidity of $EA = 44500$ N/% and it is capable of supporting a maximum tension of 4454 N. Without any compliance between the tether and Axel's body, the maximum fall factor would be 0.42. However, research on the elasticity of the ropes shows that their dynamic stiffness is usually three times the static stiffness. With this consideration, the admissible fall factor drops to 0.143, which is too poor to guarantee safety margins during autonomous operation. Some work has been already done to address this issue: the spool mechanism already includes a rotational spring between the drum and the motor, but this is only passive and does not increase significantly the fall factor. Moreover, previous research [12] aimed to build a closed loop control using a load cell that measured the tension on the tether. Results showed that the main limit to the approach was the tension coupling between the spool mechanism and the external environment. The tether management system adds a further compliance in series with the tether, and this improves the fall factor. Another advantage is that the inertia of the rover is partially decoupled from the tether by the means of an actuated spring: it is possible to control the dynamic behavior of the rover mass acting directly on the spring. Despite the large amount of factors affecting the bandwidth of such a system, estimation of the dynamic performances can be done following a certain approach [14]. If the motor back-emf is assumed as a damping force, and a 100 Nm/rad spring stiffness is chosen following the requirements, the resulting behavior is shown in Figure 5.1.

The stiffness crossover point is around 3.35 m, giving the most advantage to the system for short lengths of tether. Also, the natural frequency of the mechanical system is around 25 Hz, however, when considering also the electrical dynamics of the motor, the large-force bandwidth seems to be roughly 3.5 Hz. On the other hand, control bandwidth seems to be from one to four times more, then the control goal bandwidth is around 3-15 Hz. This is enough for Axel, because it is a slow moving robot.

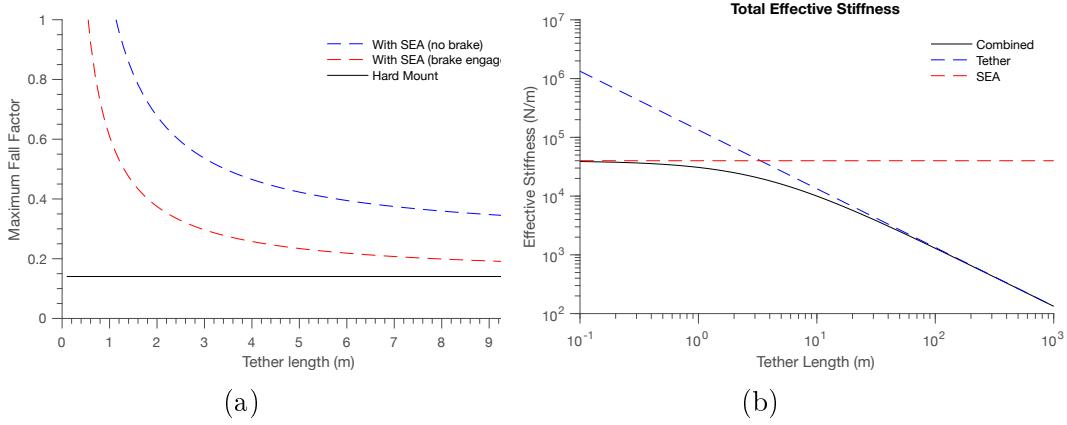


Figure 5.1: (a) Fall factor comparison. (b) Stiffness in relation to lenght of tether.

5.2 System

The system can be considered as two sub-systems working in cascade mode:

- *Primary Tension Module (PTM)* is the most external part and interfaces the high stresses produced by the environment with the low-tension internal spooling mechanism. Its goal is to decouple the two behaviors.
- The *spooling system* stores the tether and unwinds or winds it following the PTM.

These two modules work together and in a coordinated fashion to allow a precise control over the trajectory of the rover. They are packaged in a square prism enclosure, which is located on the Axel's boom. Design of the package is tricky because the size can cause issue to driving and the docking procedure. In the current design, the limiting factor is the bending radius of the tether, which is, by specifications, 10 cm.

5.2.1 Primary Tension Module

The PTM working principle comes from the nautical world: when motors were not still invented, sailors used a particular mechanism to lift significant weights such as anchors, cargos, sails. That force multiplication device, called *capstan* (reported in Figure 5.3), relied on friction and levers: the rope attached to the load was winded around a big vertical-axed cylinder and the free end was kept tight. Then, seamen pushed on bars mounted on the upper

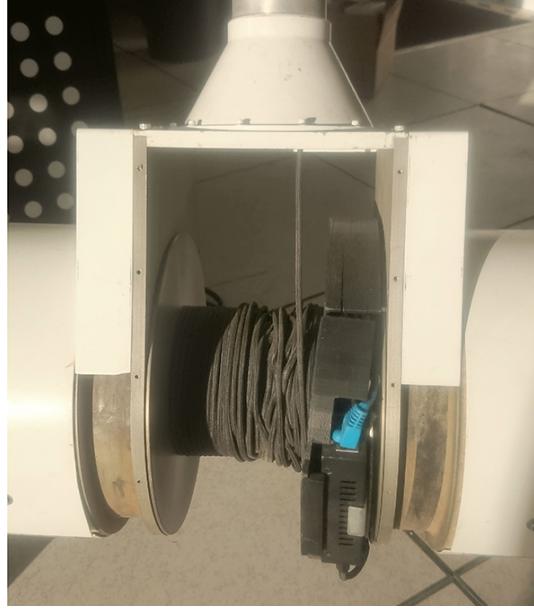


Figure 5.2: Previous condition of the Axel' spooling system

part of the cylinder so that it could rotate and move the load at the end of the rope.

In the PTM, there are two capstans constrained by a gear, that makes them move together. The capstan closer to Axel's center is actuated by a motor, which provides the force to be multiplied. The diameter of the capstans is 10 cm, and this is the lower limit according to tether specifications. However, with a different tether having a smaller bending radius, more compact designs could be possible.

The number of tether windings around the capstans is 5 as it is the best tradeoff between the amount of friction needed for the tether to stick to the capstans and the packaging size. Moreover, the tether runs inside cut-off slots to avoid any failure due to entanglement inside the device.

The actuated capstan is not directly connected to the motor, but there is compliancy provided by a spring. This type of actuator is called Series Elastic Actuator (SEA), which has been increasingly used in robotic applications, since the mid 1990's. Among the advantages, the most valuable are a more precise control of the load and a reduced weight of the actuator [15]. On the other hand, the main downside is a reduction of the bandwidth, taken up by the dynamics of the spring.

Although SEAs are characterized by always having a motor, a load and a spring, there are many possible designs, with different arrangements of these components [16]. The configuration used in this case is called compact Plan-



Figure 5.3: Wikipedia: A capstan.

etary geared Elastic Actuator (cPEA): it is smaller than the others and the spring deflection can be measured directly on the spring itself, because one of its ends is attached to the housing. Otherwise it would have been necessary to measure both load and motor position and compute the difference. Spring, motor and load are interfaced thanks to a differential planetary gear that has been custom designed and produced.

Recent work in this field prefer the use of stiffer springs to preserve the bandwidth of the overall system. However, this requires a fast and high performance control system, that is not available in space applications, where computational power is limited.

The design of the spring has been carried out through two opposite and competing objectives:

- on one hand, as it has been shown that the bandwidth of the system is proportional to spring stiffness, it is necessary to increase the stiffness to achieve increased bandwidth;
- on the other hand, a soft spring is able to better absorb shock loads and avoids that impulsive forces make their way inside more delicate parts of Axel.

The result of this optimization process is a titanium rotational spring with a stiffness coefficient of 100 Nm/rad. The peculiar geometry of the spring has required a production process based on 3D metal printing, that has been executed by another NASA facility. In fact, the spring is made of two planar

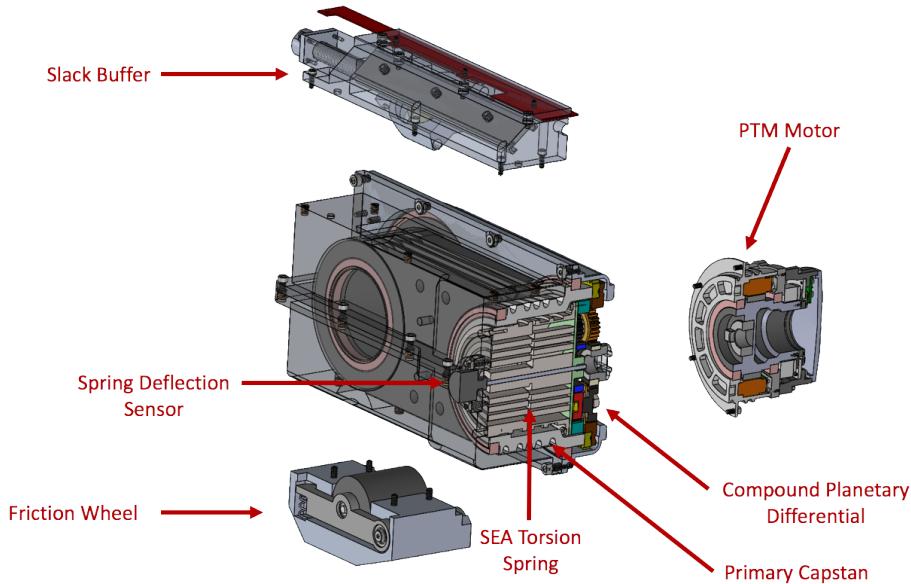


Figure 5.4: PTM and slack buffer assembly exploded view.

rotary springs, which have in common the same cylindrical core and are connected in series. The two sub-springs have twice the required stiffness, so that the resulting combination has the right one. In the design, the main limiting factors have been the locations of the mounting points, because, for compactness requirements, the spring is housed inside the capstan, which has a constrained radius. By inserting mounting points on the circumference of the spring, the most compact possible design has been achieved.

The maximum load expected on the spring is 50 Nm, which is caused by 1000 N of tension on the tether, and this generates a spring deflection of 28.6° , for a total of 57.2° if considering forces on both directions.

As previously said, the spring has one end attached to the housing, and the other is connected to a planetary differential gearbox (Figure 5.7). It is composed by a sun gear, a support sun gear, compound planets, stationary ring gear and an output gear ring as shown in Figure 5.5.

Although the gearbox is built with modified commercial gears and components, the particular assembly of the planetary gears allows to have a zero backlash. It has a 35:1 ratio from motor to load and 36:1 ratio from motor to spring. The absence of supporting bearings and backlash keeps the efficiency high.

The spring deflection is measured by a high accuracy potentiometer: the *Copal JT22-120*, which is inserted in the hollow tube in the spring, and locked

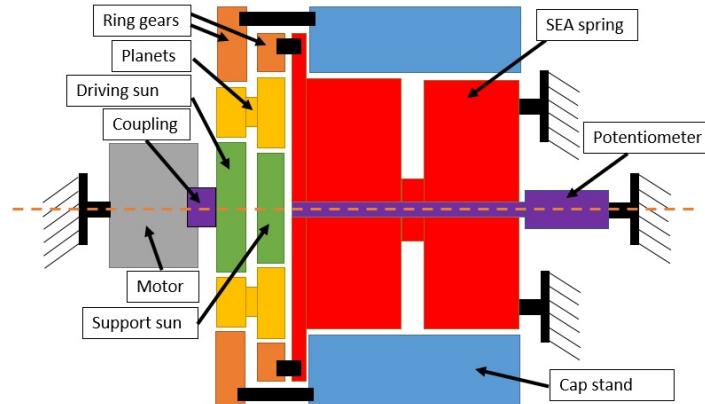
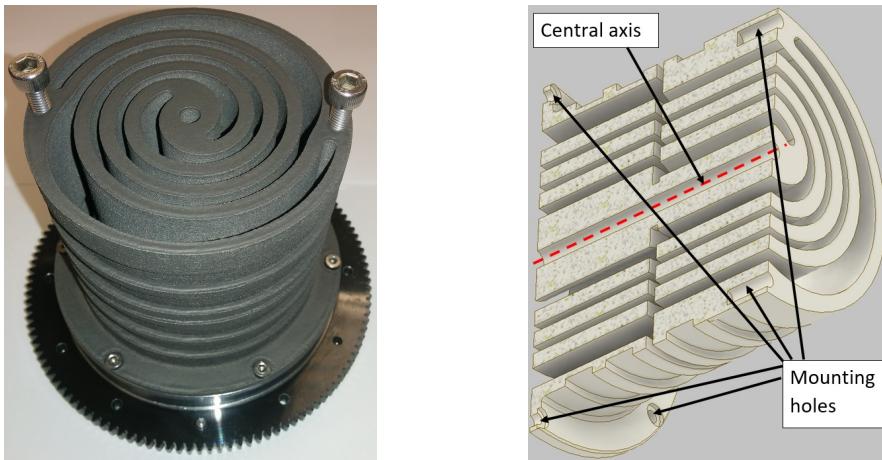


Figure 5.5: Diagram of the cPEA in the PTM.



(a) Prototype of the spring.

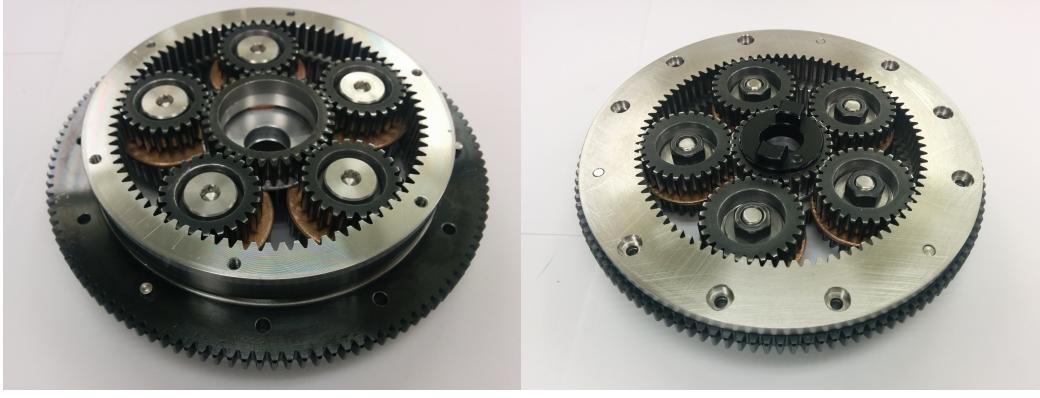
(b) Cross section view.

Figure 5.6: PTM spring.

by a screw from the gearbox side. It has a 120° range, outputs a 5 V range signal and it is supplied with 12 V. Its input-output characteristic is shown in Figure 5.8. It can be noticed that only one third of the full circle is usable for sensing, as the other two output a saturated value around 1 V.

For this reason, to allow a simple installation and use, a mounting plate has been produced. The mounting plate is a circle with holes on the circumference, so that after being inserted in its location, the sensor can be rotated to set the zero point to be in the last part of the characteristic.

It could be straightforward to use 2.5 V as the zero output, however it can be noticed that the deflection of the spring is not expected to be symmetric, because strong forces can only pull the tether from the outside up to 1000



(a) Motor-side view.

(b) Spring-side view.

Figure 5.7: PTM differential planetary gearbox.

N. The maximum pulling force from the inside of Axel is the regulated spool tension, which by design requirements is lower than or equal to 50 N.

Moreover, there is another issue that had been encountered during the testing of the first iteration of design: the analog signal was not strong enough to be sensed by the Analog to Digital Converter (ADC) for very small deflections. This has been solved by amplifying the signal four times: two times to reach the ADC range, which is 10 V, and two times because the sensor range is twice the expected deflection of the spring, so the sensed range is practically cut in half. Another improvement can be done to better account this problem, that is to amplify and offset the signal over the full range of the ADC, between -10 V and 10 V.

In conclusion, reference has been set around 0.6 V before amplification to allow a full excursion of the ADC input range. The signal is amplified by a *INA125* instrumentation amplifier (Figure 4.5) with a big resistance as R_{gain} , 1 MOhm.

In a successive version of the circuit, a low pass filter has been added before the amplifying stage, because the high frequency noise was picked up by the ADC. Also, a linear voltage regulator has been included in the supply line to filter out the noise on the bus.

Actuation of the PTM is done by a *Robodrive ILM85x13* servo motor, whose housing has been custom designed to fit in the package of the overall system. Some parts, like the rotor support, have been 3D printed with a carbon-injecting procedure to obtain a strong, but slightly flexible material.

Figure 5.10a shows, from left to right, the cutaway view of the motor, gearbox, spring and capstan assembly. The motor armature is rated for 48 V and its maximum speed is 2900 RPM which is roughly 42 cm/s on the

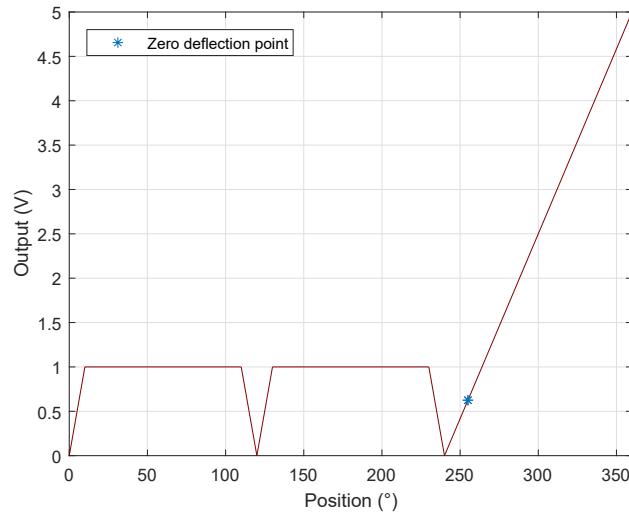


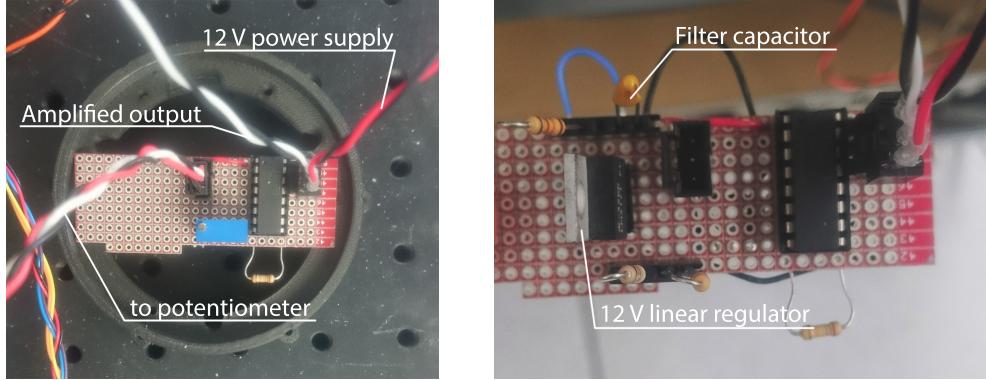
Figure 5.8: JT22-120 characteristic.

tether, so this is the limit speed for the system. Instead, the produced torque is 0.43 Nm in rated conditions and 1.2 Nm for intermittent operation. These torques translate to 310 N and 864 N, respectively. The housing also contains a magnetic absolute encoder, *RD70-AKSIM*, which interfaces directly with the motor controller, and a power-failsafe brake.

5.2.2 Slack buffer and spool

The spool mechanism is a redesign of the previous system. It has been upgraded with a lower reduction gearbox, passing from 840:1 to 81.7:1. The motor is a *Maxon EC 40 393025*, which is the revision of the previous motor. In this way the spool actuation is weaker than before, but able to always keep up with the PTM speed: in fact, the speed of the tether with an empty spool (5 cm radius) is 58 cm/s, that is more than the PTM (42 cm/s). The lower torque is not an issue, because instead of having tension control in the spool as was before, the tension control is done by the PTM, and the spool has only to keep a small tension on the tether.

To overcome the problem of knifing a level winder has been added to the spool. Despite most spooling applications make use of a diamond screw to constrain the motion of the fair lead with respect to the spool drum, this design employs a worm screw. In fact, the diamond screw could be directly connected and rotated by the means of a gearing system to the drum, reducing the number of actuators by one. But this configuration has



(a) First version with only the amplification stage.
 (b) Redesign with a linear voltage regulator and a low pass filter.

Figure 5.9: PTM potentiometer signal conditioning circuit.

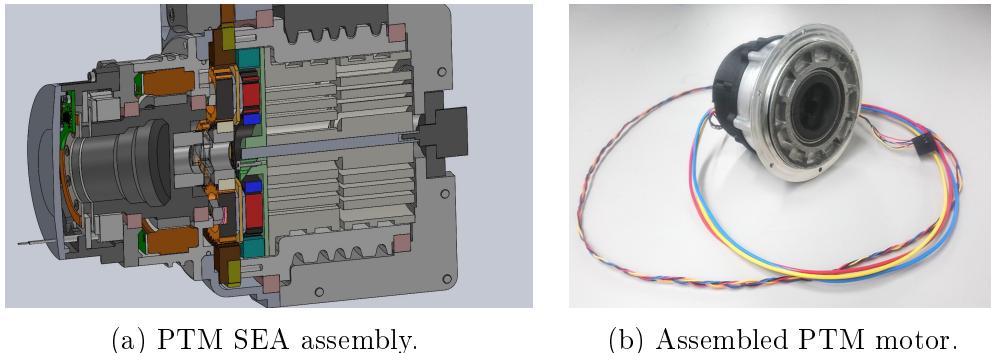


Figure 5.10: PTM motor.

also one degree of motion less in comparison, and this could be really useful in case of spooling fault: the fair lead could move independently from the drum and fix the issue.

Another advantage of the worm screw is that it enables a very fine threading, and this allows high lateral forces to be managed by the fair lead. On the other hand, the use of a worm screw requires a particular trajectory generation which is described in Section 4.4.2. The worm screw is actuated by a small *Maxon* motor, which is driven with a *Barret PUCK P3* controller. The *P3* communicates with the main computer through the CAN bus, and it has been chosen for its small footprint: it can fit at the end of the motor on the opposite side of the load.

Between the spool and the PTM a slack buffer provides compliance if the tether goes slack. This can happen if the PTM has to respond to an impulse or a far set-point. The slack buffer is shown in the upper part of Figure 5.4.

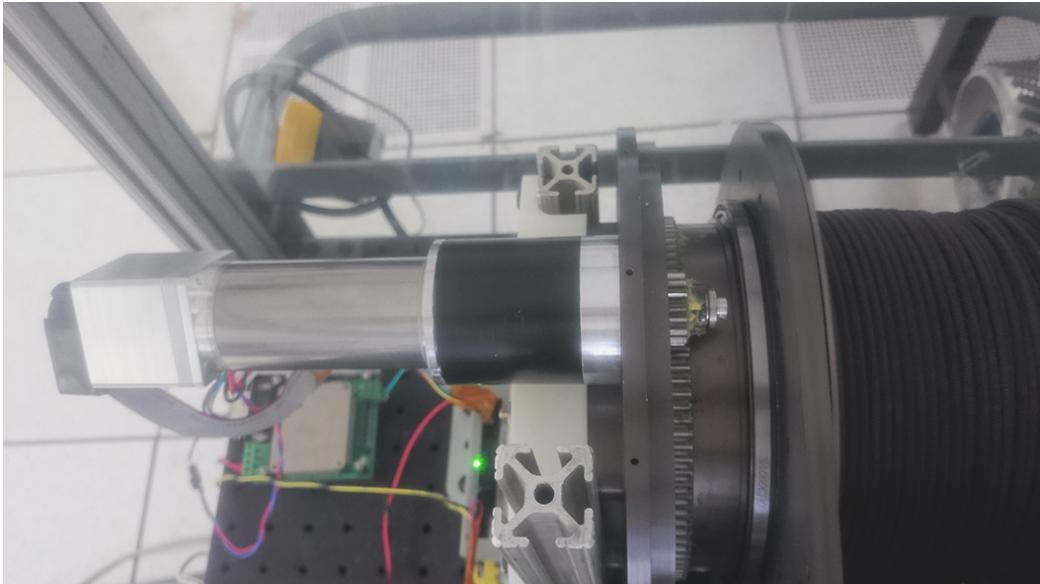


Figure 5.11: Spool motor assembly.

It is basically a pulley where the sheave in the middle can ride on a linear bearing, and it is constrained by a spring, which displacement is sensed by an absolute linear magnetic encoder, the *Zettlex 3711_V2*. The encoder has 0.016 mm resolution, and it is able to read changes of tension down to 8 mN; the total run of the slack buffer is 80 mm. The spring is off-the-shelf and it has been measured to verify its stiffness. Spring stiffness selection is a trade-off between performance and robustness of the system:

- a *soft* spring provides more decoupling between the PTM and the spool, and it leaves to the spool motor more time to catch up with the PTM dynamics,
- instead, a *stiff* spring enables more bandwidth to the overall system, and reduces the chance of a transient of no tension on the PTM from the spool side, which, as explained before, prevents completely the functionality of the PTM.

Plot in Figure 5.14 shows that the maximum measurable tension on the slack buffer is 42 N, which is slightly less than the required tension.

A possible workaround is to preload the spring and move the resting position, but the maximum deflection of the spring is 90 mm, thus only a 10 mm preloading can be achieved. In this way, the full range would be incremented by 8 N, reaching a total of 50 N. This could seem enough for the

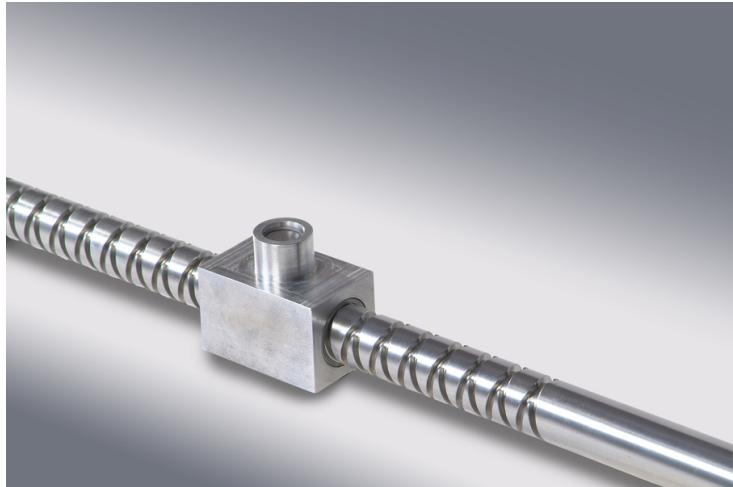


Figure 5.12: Rempco: A diamond screw with its fair lead.

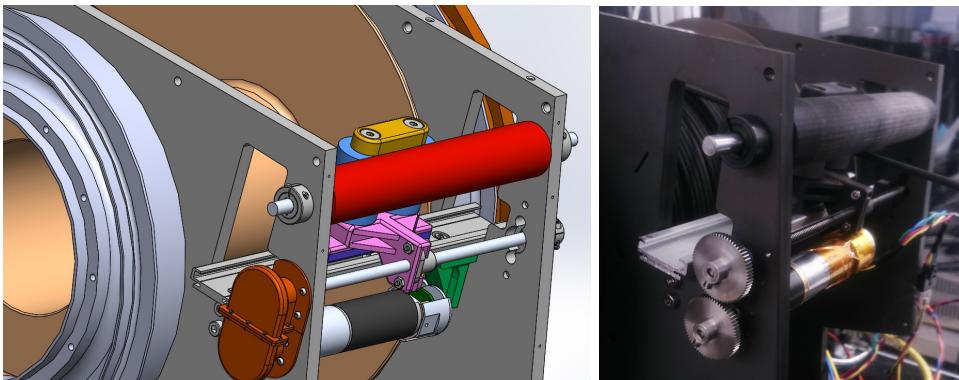


Figure 5.13: Design of the level winder on the left, and actual level winder on the right.

purpose, however it would not still be possible to measure tensions close and higher than 50 N, and that is necessary to for a closed loop control system.

In conclusion, the slack buffer has been tested with reference tensions around 20 - 30 N for this reason, but it does not affect the validation of the control system.

5.3 Control

Control design for the Tether Management System can be approached in many ways: as discussed previously, it is composed by two subsystems working in cascade: the PTM and the spool. The main challenge is to preserve

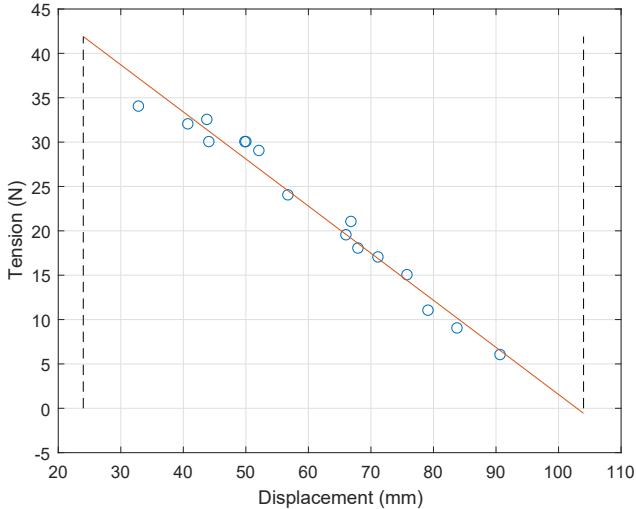


Figure 5.14: Slack buffer spring characterization.

the bandwidth even with a significant amount of tether outside Axel. Requirements and goals of the design are the following:

- Ability to control the tension on the spool, by setting a reference tension, which is usually a step input. This input command is supposed to be around 50 N, but further work could prove that higher tension is even better to have a good quality of spooling.
- Robustness to external tension: the system has to be resilient to changes in tension outside Axel. The maximum disturbance that can occur in this case is a free fall, and the PTM could undergo a force impulse which is the full weight of Axel itself.
- Control of tether position, like winding and unwinding. For this goal, the performance required is limited by the bandwidth of the system.

Two approaches have been followed and developed. The first is based on a Proportional Integral Derivative (PID) control: as the Elmo motor controllers can host and run embedded code, it has been natural to deploy a software PID controller on them. The resulting control is expected to be fast thanks to the decoupling, and to add some robustness, the algorithm allows to change the PID gains during run time. The second is based on H_∞ control, and it is supposed to run on the main computer or on one added to the system for the purpose. With this control, the two systems are expected to "cooperate", because the controller can access the full state of the system

at the same time. However, the complexity of this approach lies in the model of the system, which has to be accurate to get a good performance.

While the PID control has been deployed successfully on the system and validated, the H_∞ control has not been tested on the real system, due to lack of time. However, some simulation results will be presented, which can give some insight on the comparison of the two control systems.

5.3.1 PID control

PID control is one of the most basic closed loop control, and its performance depends on the choice of its parameters. The process of selecting the parameters set is called *tuning* and there are many procedures to obtain a satisfactory set.

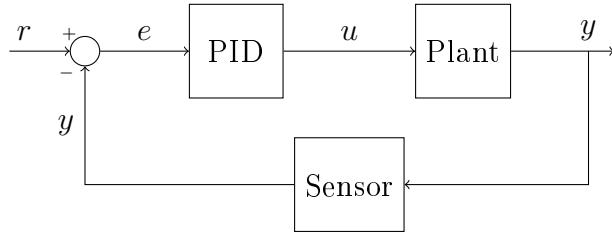


Figure 5.15: Block diagram of the PID control.

In this type of control, the controller takes as input the reference value that the system should output and the actual output of the system, then it computes the difference. It is called the *error*, and for a given time t it is:

$$e(t) = r(t) - y(t)$$

where $r(t)$ and $y(t)$ are respectively the set-point and the system output for a given time.

The controller can act on the controlled system, also called *plant*, through a signal $u(t)$, which is usually an input to an actuator. The control variable, $u(t)$, is the sum of three different types of action that aim at correcting the error. These three values are summed by the means of weights, that are called *gains* (K_P , K_I , K_D).

- The *proportional* action produces a control signal that directly scales with the error in a linear way. Its derivation is

$$u_P(t) = K_P e(t) \quad (5.1)$$

The underlying idea is simple: the more is the error, the more is the corrective action imposed by the controller to the system. However, using only this action does not guarantee that the output follows the reference, instead there will be an error between the reference and the output value. Because without error, the controller outputs a zero command.

- To solve the steady state issue introduced with the proportional action, the sum of the past errors can be used to produce another correction that grows when the error is not decreasing.

$$u_I(t) = K_I \int_0^t e(\tau) d\tau \quad (5.2)$$

In this way, the proportional effect decreases with the error, but it is compensated by the integral action.

- Usually both the proportional and integral actions produce a closed loop system with underdamped dynamics, and oscillations occur after the transient until they are damped.

$$u_D(t) = K_D \frac{de(t)}{dt} \quad (5.3)$$

By adding a derivative correction, the controller tries to predict the future error state linearly, by using the error's rate of change. This action produces a damping directly proportional to the gain, K_D , which is limited by the noise on the error. In fact, sensor noise has usually high frequency, and the derivative action amplifies the dynamics.

When the set point is far from the output of the system, an *integral windup* can occur if the integral action is present. In fact, the integrator could accumulate a value that is bigger than the maximum allowed control command, then the system would overshoot and even become unstable. A possible fix to this problem is to introduce saturation to the integral action, so that it can only increase (in both signs) up to a certain value. This value has to be tuned together with all the gains of the PID.

Another improvement that can be added to the PID algorithm is the *deadband prevention*. The deadband is a certain range of error values around 0, and the controller output for all of them is 0. This is useful when the PID generates an unwanted small control action when the error is negligible. In such cases the system could oscillate around the reference state.

The complete control law is the sum of (5.1), (5.2) and (5.3):

$$u(t) = u_P(t) + u_I(t) + u_D(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (5.4)$$

It is easy to implement in a software as it has a small footprint in terms of time and space.

Listing 5.1: Pseudocode of PID algoritm.

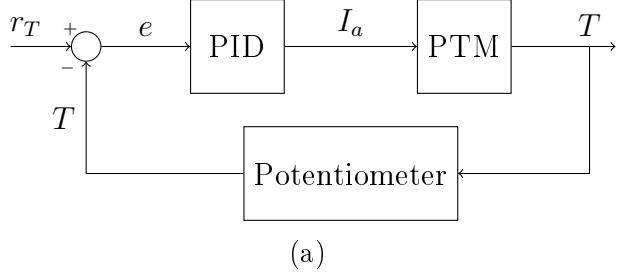
```

error sum is 0
past error is 0
while true
    get the reference
    get the current output of the system
    compute the current error as the difference
    compute proportional action from current error
    add the current error to the error sum
    compute integral action from error sum
    apply anti-integral windup and deadband policies
    compute the error rate as the difference between
        ↪ current error and past error
    compute derivative action from the error rate
    set current error as past error
    compute control action
    apply control action to the plant
end

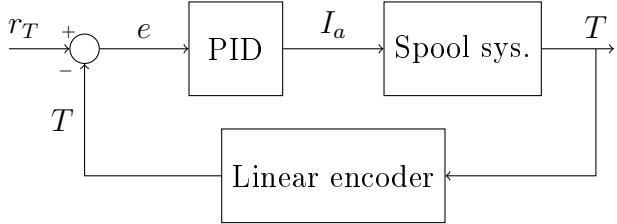
```

In the case of the TMS, two control loops are necessary: one for the PTM and one for the spool. The two available actuators are the PTM and spool motors, and the two available sensors are the PTM spring potentiometer and the slack buffer linear encoder. Thus the first control loop sets the deflection of the spring in the PTM, that can be either in position or torque as they are proportionally linked by the stiffness constant of the spring. The same holds for the spool and the slack buffer where the feedback is taken from the linear encoder. The two resulting control loops are shown in Figure 5.16.

The control variable for both the subsystems is the armature current of the motors and this is due to performance purposes. Elmo controllers include a microprocessor that executes embedded programs. The computational time is shared between user code, such as the PID controller, and the producer's code, which is responsible for low level control of the motor. Elmo Whistle controllers provide many control modes: position, speed, torque or armature current. Depending on the selected control mode, the Elmo driver runs less or more control loops, because they are nested.



(a)



(b)

Figure 5.16: PID control schemes of the TMS subsystems: (a) PTM loop, (b) spool loop.

For example, the execution of a position control loop uses an internal speed control loop, which, again, relies on the armature current control loop. This structure is shown in Figure 5.17.

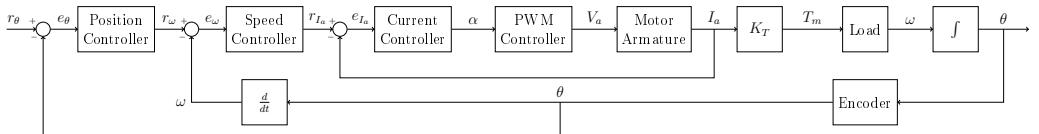


Figure 5.17: Simplified block diagram of Elmo control.

As inner loops are characterized by faster dynamics, the bandwidth is not the same for the three loops: Elmo allows to set the sampling time of the innermost control loop, which is the *current loop*. This can be set through CAN bus when Elmo controller is not driving the motor. If T_c is the sampling of the current loop, the sampling period of the position and the speed loops are $2T_c$ (in microseconds). Selection of T_c is a compromise between high servo performance and the scan loop (background) operations, which are the user program execution and the interpreter agent. When T_c is low, the drive has more control bandwidth, however less computing power remains for the user program and the responsiveness of the interpreter.

For this reason, the tradeoff has resulted in the selection of $70 \mu\text{s}$ as T_c , which gives to the PID user program a sampling period of 10 ms, and the

chosen control mode is current control. In this mode, the Elmo drive can set a certain current on the armature circuit of the motor, thus enforcing a specific torque on the load as described by (4.1). Empirical experiments also showed a better performance when the motor is controlled through its armature current, instead of both position and speed control, which have slower transient.

By running the two control loops on the Elmo drives, instead of the main computer, a significative load is removed, and there are more computational resources for high level control, such as a tether position control loop wrapped around the two PID loops.

Elmo drives' general purpose inputs are particularly limited: they can get values on their internal registers, written by the CAN bus controller; or they can read analog voltages through two ADCs. While reference values for the PIDs can be set easily through the CAN bus, it has been a difficult challenge to interface the PTM spring potentiometer and the slack buffer encoder to the motor controllers.

As explained in Section 5.2.1, the PTM spring potentiometer outputs an analog tension between 0 and 5 V, but a signal conditioning circuit has been added to improve the quality of the readings, so that Elmo driver can read a signal ranging from 0 to 10 V. Moreover, as the signal from the potentiometer was noisy, a RC low-pass filter has to be included before the ADC. The bandwidth of the filter is around 30 Hz, which is enough to preserve the dynamics of the signal.

On the other hand, the slack buffer encoder, which is shown in Figure 5.18, sends its readings through differential SPI. Serial Peripheral Interface (SPI) is a communication system made for digital data transmission between integrated circuits on the same board. There is a *master* and one or more *slaves*, and they share three logic lines: the *clock*, the signal from master to slaves (*MOSI*) and the signal from slaves to master (*MISO*). Moreover, there is another logic line for each one of the slaves, used to select which slave has to communicate with the master. In the case of the encoder, the *Zettlex 3711_V2*, all these signals are differential, to improve noise rejection. The *MOSI* line is not present in the sensor, because it does not need any kind of information from the master.

To request data from the sensor, it is needed to produce a square wave signal on the clock line. At each tick of the clock, the sensor sends a bit on the *MISO* line. There are many possible configurations for this behavior: in fact, the slave could send a bit either when the clock is high or when is low, and the reading from the master can happen either on the rising of the edge or the trailing one. For this reason, there are four different modes of SPI communication depending on these factors, as shown in Table 5.19.

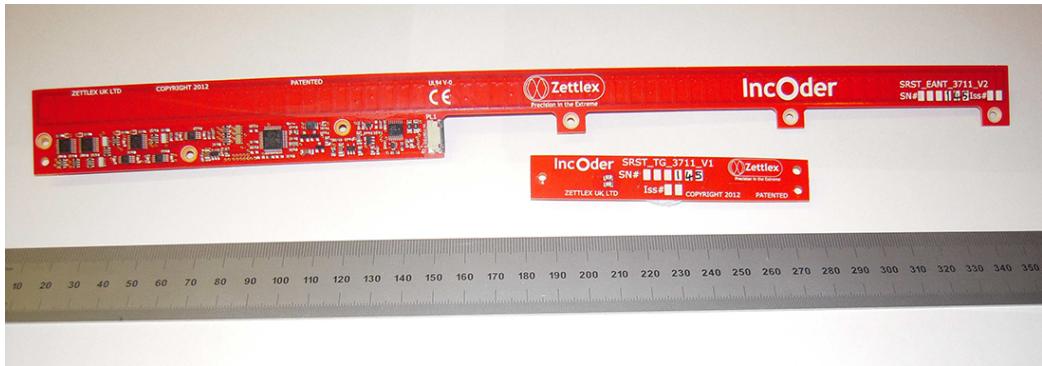


Figure 5.18: The Zettlex 3711_V2 linear absolute encoder.

Mode	Clock	Sampling
0	0	rising edge
1	0	falling edge
2	1	rising edge
3	1	falling edge

Figure 5.19: SPI modes.

It is important that master and slave use the same mode, otherwise communication can not happen. In this specific case the SPI mode used by the sensor is 0. Unfortunately Elmo controllers share the SPI pins with the motor encoder ones, then it is not possible to connect another encoder and use its reading as a general purpose input. The solution is to insert a signal conversion circuit that transforms the digital reading to an analog tension that can be read by the Elmo through its ADC input.

A simple microcontroller, the *Sparkfun Micro Pro*, has been used for this purpose. It can run code at 16 MHz, and there are many libraries already available that provide SPI functionality to it. However, the Micro Pro can not output a variable regulated analog voltage, because it lacks a proper Digital to Analog Converter (DAC). As the microcontroller exposes Pulse Width Modulated (PWM) pins, it is possible to output an high frequency square wave, having different periods for high and low state. The ratio of the two periods defines the mean value of the signal, and it can be varied by changing the ratio. However, this is not sufficient, because the ADC on Elmo is fast enough to read the complete square wave, and not just the mean value. There has to be an averaging filter that produces the mean value: it can be either software or hardware. While software implementation would add load to the Elmo controller, the hardware solution is a low-pass RC filter

that does not decrease the performances of the system.

On the other hand, the low-pass filter has a limited bandwidth, because of the possible selection of the components. Thus, to have a reliable measurement, a dedicated DAC circuit has been used. The *Adafruit MCP4275* can receive a 12 bits digital input through I²C and output an analog value between 0 and 5 V. I²C is another transmission system, developed for communication between integrated circuits. The main difference with SPI is that the data line is shared, while the SPI has two lines that allow a full-duplex transmission. The Micro Pro supports also the I²C, so its function is to relay the data from the SPI bus to the I²C bus.

At the end of the chain, the analog voltage from the MCP4275 can be read from the Elmo Whistle drive. The final system is shown in Figure 5.20.

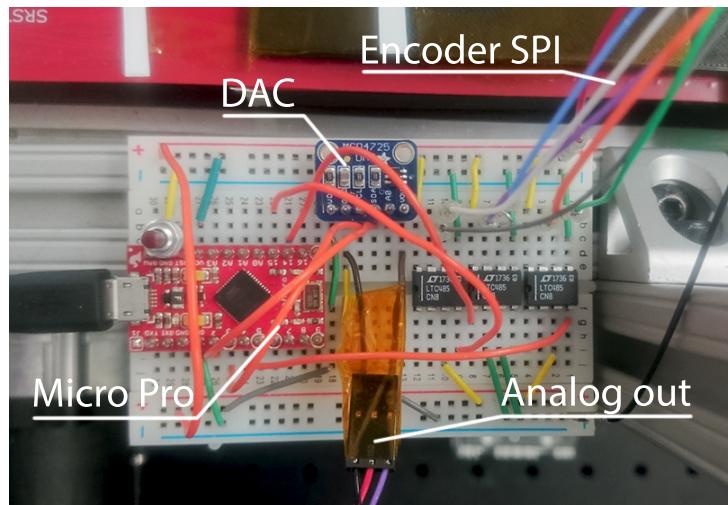


Figure 5.20: The signal conditioning circuit for the slack buffer encoder.

Tuning of PID systems is definitely the most challenging part in the implementation, because there is not a unique and straight procedure to obtain the gains. Among all the methods, the most used is the Ziegler-Nichols, and requires to find the minimum proportional gain that produces a stable oscillatory response of the closed loop system, then the other gains are derived through empirical formulas.

In this work, the method followed for tuning the controllers is slightly different from Ziegler-Nichols'. At the beginning, both K_I and K_D are 0, and the system is controlled only with the proportional correction. K_P is increased until the system becomes unstable and produces an oscillatory response. Then K_D is increased until the oscillations are damped. The process is repeated again and again until the derivative gain starts to amplify

the noise of the sensor: this is an upper limit. At the end, a tradeoff between K_P and K_I is found by decreasing the first one and increasing the second one until the performances are satisfactory.

5.3.2 H_∞ control

The controlled system, or plant, is a multiple input multiple output (MIMO) system with two inputs and two outputs. Inputs are the PTM motor and the spool motor, that can be activated with one of their control modes; instead, the two outputs are the PTM spring deflection and the slack buffer spring deflection. Moreover, a disturbance is present: the load on the external side of the tether, that acts on the capstans of the PTM.

As explained previously, it is possible to distinguish two separated systems working in cascade: the first one is the PTM, which is stiff but slow, and the second one is the spool with the slack buffer, fast and soft. The goal of the control system is to coordinate the two subsystems and set, at the same time, the tension inside the spooling system and the tension of the PTM spring.

H_∞ is a class of controllers which result from an optimization procedure and guarantee the satisfaction of imposed performance requirements. While some control methodologies do not require knowledge about the controlled systems (such as the PID described in Section 5.3.1), this controller is built on the system model, although uncertainties can be included as constraints in the optimization.

Hence, the stability and the performances of the H_∞ controller depend on the fidelity of the model to the real system. The procedure of obtaining a model of the system is called *identification* and there are several approaches to do it. Depending on the knowledge of the intrinsic behavior of the system it is possible to distinguish three types of methodologies:

- *Black box*
- *White box*
- *Grey box*

The *Black box* model assumes no prior information about the working principles of the system. The identification procedure obtains coefficients for certain types of transfer functions that approximate the real system's behavior. The level of approximation and the non-linearity of the analyzed system are used to select a parametric model, which is a mathematical representation of the system, but with tunable parameters. Then, a process called modeling uses existing input/output data from the real system to find the

values of these parameters, such that the output of the model follows the data with the same input. The validation of the model can be done by comparing its behavior with new data from the real system, that was not used for identification.

Sometimes, the system to be controlled is completely known in terms of behavior: in case of a simple system, it is possible to build the model starting from the mathematical rules that characterize its basic components. This approach usually generates complex models and requires more time to be carried out. Models of this type are called *White box* models.

Grey box models result from a combination of black box and white box models: the system is partially modeled with the underlying equations, however there are still some parameters that have to be tuned.

In order to build a model for the system, a grey box approach has been followed: most of the components have simple characteristic equations, which correspond to first or second order transfer functions.

The PTM system is made up by a differential gear connected with a motor, a spring and the capstan. The motor is a DC brushless, and its behavior is described by the equations of the DC motor (in Section 4.3). The block diagram built from these equations is shown in Figure 5.21.

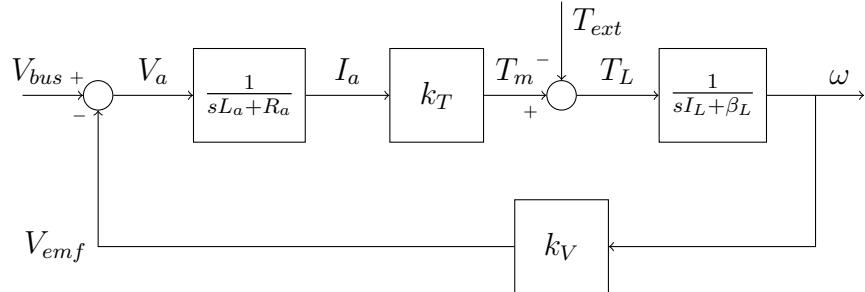


Figure 5.21: DC motor block diagram. The voltage imposed by the controller bridge, minus the back electro motive force (emf) passes through the armature windings, and the respective current defines the mechanical torque produced by the motor T_m . The external torque acting on the rotor is subtracted to obtain the actual torque on the rotor. The rotor speed is given by the rotational Newton's laws from the applied torque. In conclusion, the loop is closed by the means of the back emf, which is proportional to the speed.

However, the model lacks the control part of the motor: it is not directly actuated through the bus voltage, because the Elmo DC Whistle takes care of producing the precise voltage with the respective timing. Then, to simplify the modeling, it is possible to assume that Elmo is able to set a certain

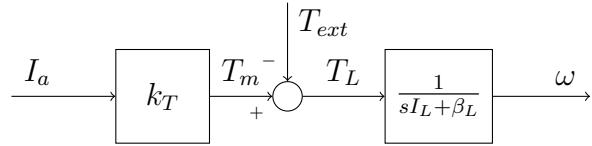


Figure 5.22: DC motor block diagram with current input.

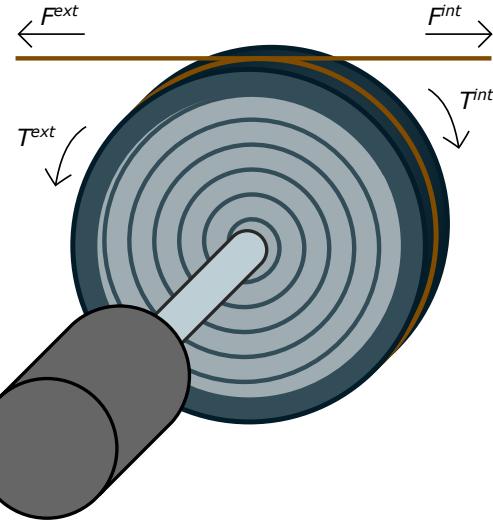


Figure 5.23: Diagram of forces acting on the capstan.

current on the armature with a negligible delay: the system becomes then the one represented in Figure 5.22.

Another advantage of this type of model is that the control could be faster because Elmo would use less control loops, and more bandwidth can be achieved.

In terms of dynamics, the PTM spring links a position with a torque, and can be used in both ways as it is a scalar and invariant factor.

For what concerns the capstan, there are three different torques acting on it: the first torque comes from the planetary gearbox and provides actuation by the motor, the other two torques are generated by the tension of the tether. In addition, the two respective forces have different nature: one is the external tension, T_{ext} , and it is related to the environment, the other is the internal tension, T_{int} , the same felt by the slack buffer. This last tension is not the spooling tension because the slack buffer decouples the two systems, thanks to the pulley and the spring.

By using the equations provided in a previous research paper about SEAs [16], which exploit a three bodies interaction approach, the full model of the PTM can be derived, as shown in Figure 5.24.

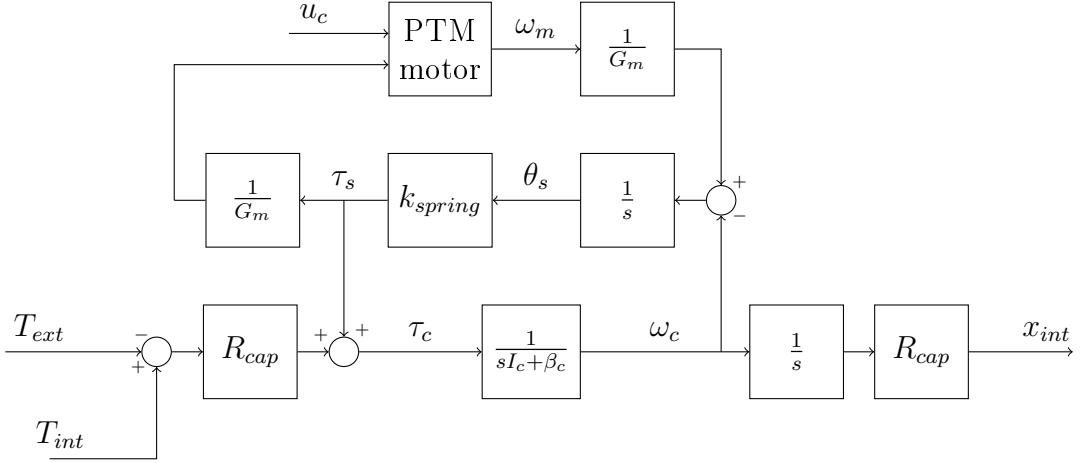


Figure 5.24: PTM motor block diagram.

The PTM system has the control input, which goes to the motor, the two tensions described previously and as output the displacement of the tether in the slack buffer. Friction in the system has only been considered in the rotational dynamics: viscous damping has been included for both the rotor of the motor and the capstan, but not for the spring.

The parameters that appear in the system diagram have been taken from the datasheet, and assumed as accurate. However, an identification of all the parameters could be done to improve the robustness of the model.

The only parameter that has been identified is the spring constant, k_{spring} in Figure 5.24.

To obtain a value that can be inserted in the model equations, a series of increasing torques has been applied to the spring and its deflection has been measured. Taking this kind of measurements has been challenging, because of the lack of specific instrumentation made for this purpose: usually these tests require particular machines to be employed, which can produce a precise torque on the spring and measure its displacement with accuracy.

The procedure used in this case started with locking the capstan, otherwise the torque applied from the motor to the differential gearbox would be split between the capstan and the spring. The capstan locking has been achieved by tensioning the tether, and the tension has to be equal on both sides. If the tension is different, then the difference would be "felt" by the spring due to the differential gear.

After the capstan is locked, the motor can be controlled in position mode and has to be moved with increasing displacement from the initial position: this will produce a torque on the spring through the gears, thus causing

a deflection. The potentiometer inside the spring can directly output the displacement of the free end. Figure 5.25 shows the resulting data.

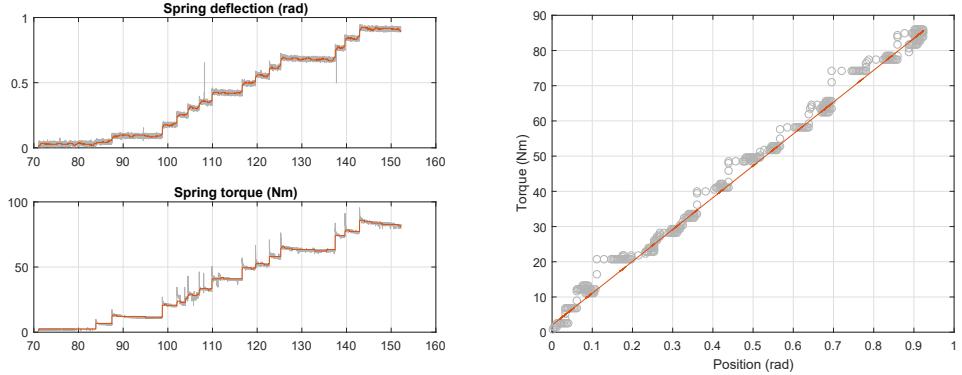


Figure 5.25: Data used for PTM spring identification.

It is possible to note that the relation between the deflection and the torque is linear as expected, hence, a linear regression has been used to determine the stiffness coefficient.

The relation is expressed by the Hooke's law:

$$t = k_s \theta \quad (5.5)$$

where t is the torque on the rotational spring, k_s is a constant that depends on the spring, and θ is the deflection.

The resulting stiffness is 90.47 Nm/rad, which means that the 3D printed spring is slightly softer than expected (100 Nm/rad).

Then, the remaining part of the system, that includes the slack buffer and the spooling mechanism is modeled following the same principles, and the full model is shown in Figure 5.26.

As in the case of the PTM spring, the slack buffer spring coefficient has been derived from data. From the data shown in Figure 5.14, the resulting spring constant is 530 N/m, which is slightly lower than the design parameter (>600 N/m).

The H_∞ optimized controller is a closed loop controller that is placed before the plant in the direct branch of the loop. An example that reflects such system is represented in Figure 5.27.

Let $G_C(s)$, $G_P(s)$, $G_S(s)$ be the transfer functions of controller, plant and sensors respectively. For sake of simplicity, the sensor transfer function will be supposed $G_S(s) = 1$, but all the discussion still holds in the general case. The following tranfer functions can be defined:

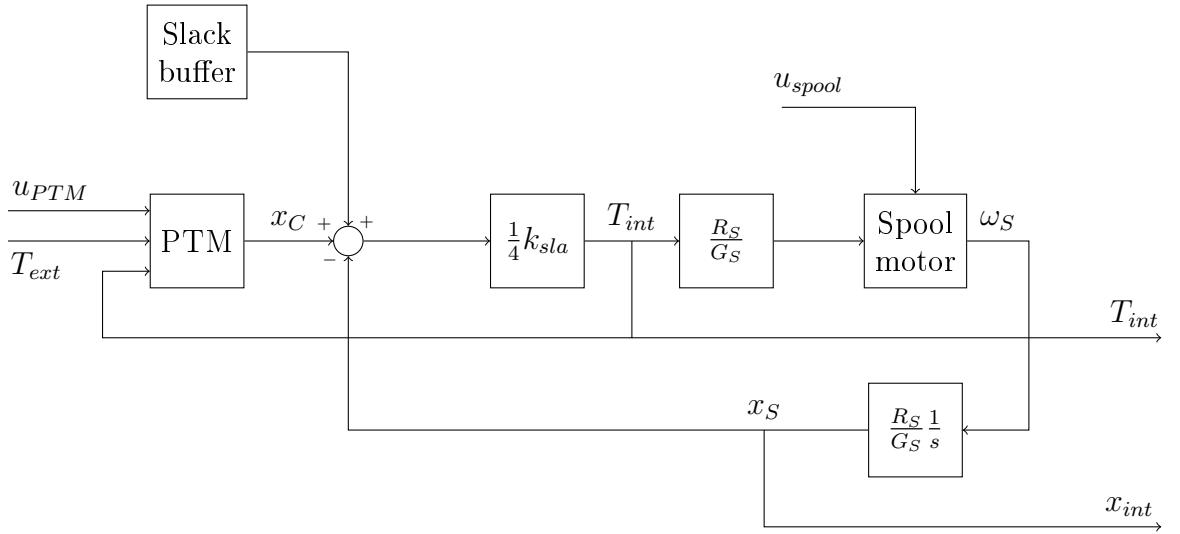


Figure 5.26: The full block diagram of the TMS.

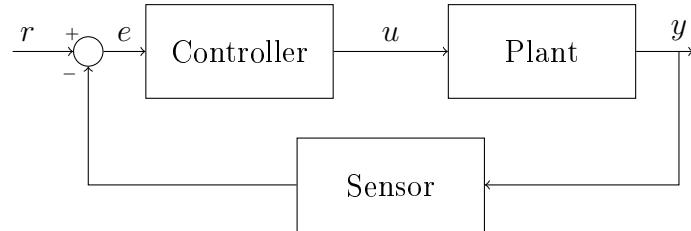


Figure 5.27: The generic feedback control loop.

- *Direct loop function:*

$$G(s) = G_C(s)G_P(s) \quad (5.6)$$

- *Feedback function:*

$$H(s) = G_S(s) \quad (5.7)$$

- *Open loop function (or just loop function):*

$$L(s) = G(s)H(s) \quad (5.8)$$

- *Closed loop function:* the transfer function between the control input to the whole system and the output.

$$W(s) = \frac{G_C(s)G_P(s)G_S(s)}{1 + G_C(s)G_P(s)G_S(s)} = \frac{G(s)H(s)}{1 + G(s)H(s)} \quad (5.9)$$

- *Sensitivity:*

$$S(s) = \frac{1}{1 + L(s)} \quad (5.10)$$

- *Complementary sensitivity:*

$$T(s) = 1 - S(s) = \frac{L(s)}{1 + L(s)} \quad (5.11)$$

The last two transfer functions are very important, because they can be related to stability and performance indicators, thus allowing to set constraints through their shaping. Objective of the control problem is to determine a proper $G_C(s)$ that

- stabilizes the closed loop system
- respects the performance criteria
- is robust against model uncertainty

It is necessary to relate these aspects to some variables of the control system.

If a polynomial reference input is considered, the transfer function from r to e has to contain a certain number of poles in $s = 0$ for the system to be stable, as shown in (5.12) (assuming $H(s) = 1$ ad previously stated).

$$\begin{aligned} e(s) &= r(s) - y(s) \\ &= r(s) - \frac{G(s)}{1 + G(s)H(s)}r(s) \\ &= r(s) - \frac{L(s)}{1 + L(s)} \frac{1}{H(s)}r(s) \\ &= S(s)r(s)^* \end{aligned} \quad (5.12)$$

The poles can either be in the plant or in the controller, and if the reference is a step, $S(s)$ must have at least 1 pole in $s = 0$ to have null error at steady state.

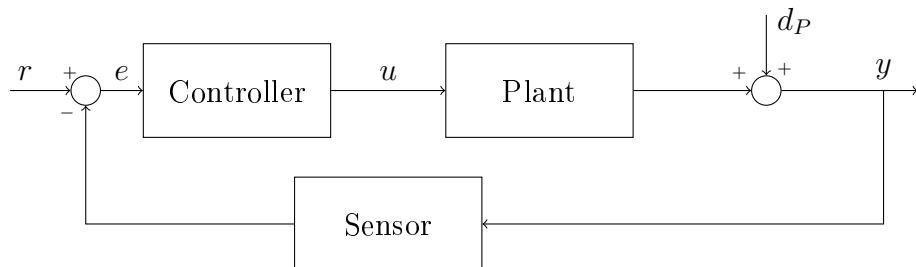


Figure 5.28: Disturb acting on the plant.

Also, specifications on the plant disturbance response, whose block diagram is shown in Figure 5.28, become constraints on $S(s)$:

$$e^{d_p}(s) = y^{d_p}(s) = \frac{1}{1 + L(s)} d_p(s) = S(s) d_p(s) \quad (5.13)$$

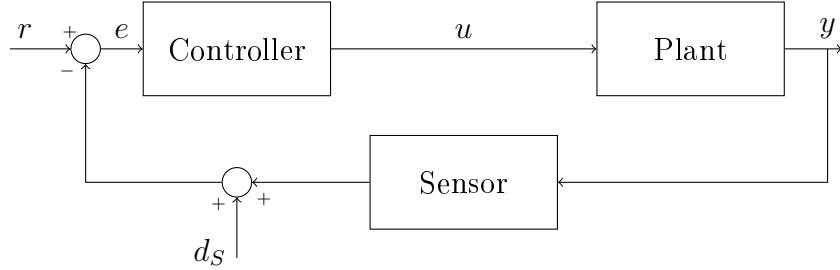


Figure 5.29: Disturb acting on the sensor.

On the other hand, specifications on the sensor disturbance response, whose block diagram is shown in Figure 5.29, become constraints on $T(s)$:

$$e^{d_s}(s) = y^{d_s}(s) = \frac{G(s)}{1 + G(s)H(s)} d_s(s) = T(s) \frac{1}{G_S(s)} d_s(s) \quad (5.14)$$

In terms of performance, it is possible to set requirements on the transient response. The most common specifications are the following:

- *Overshoot* $\hat{s} \leq \bar{\hat{s}}$
- *Rise time* $t_r \leq \bar{t}_r$
- *Settling time* $t_s \leq \bar{t}_{s,\alpha}$ with settling percentage α

Figure 5.31 shows how the requirements both in the steady state and transient response are transferred into the frequency domain, in particular they become specifications on the sensitivity and complementary sensitivity functions. As shown before, the numbers of poles in $s = 0$ of the sensitivity function defines the type of the closed loop system: it means that the more poles are in $s = 0$, the higher order of input can be followed as reference. In the majority of cases, and also this one in particular, the assumption is that the reference is given with step variations. This means that the sensitivity function must have at least one pole in $s = 0$ to have a zero steady state error. The two red-dotted areas are called *frequency masks*, and they are used to make the closed loop system resilient to disturbances. In the case of a low-frequency disturb acting on the plant, with maximum frequency ω_p

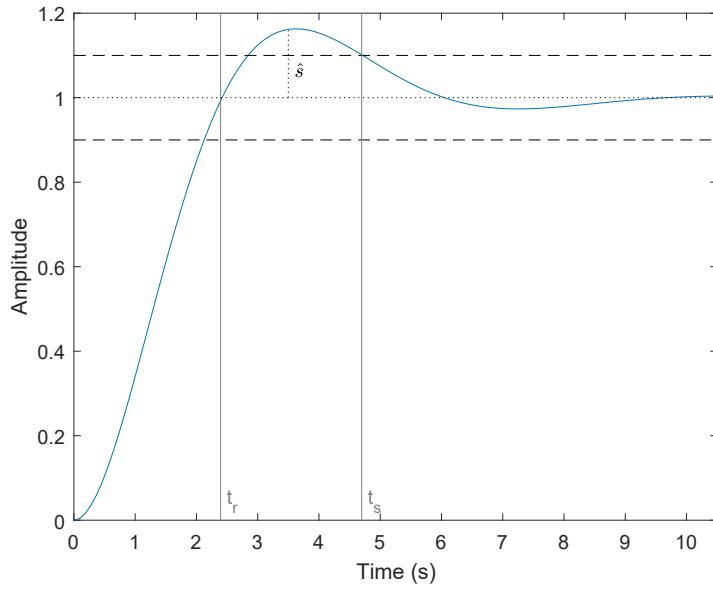


Figure 5.30: The step response of a second order system: in this case, the settling range, α , is 10%.

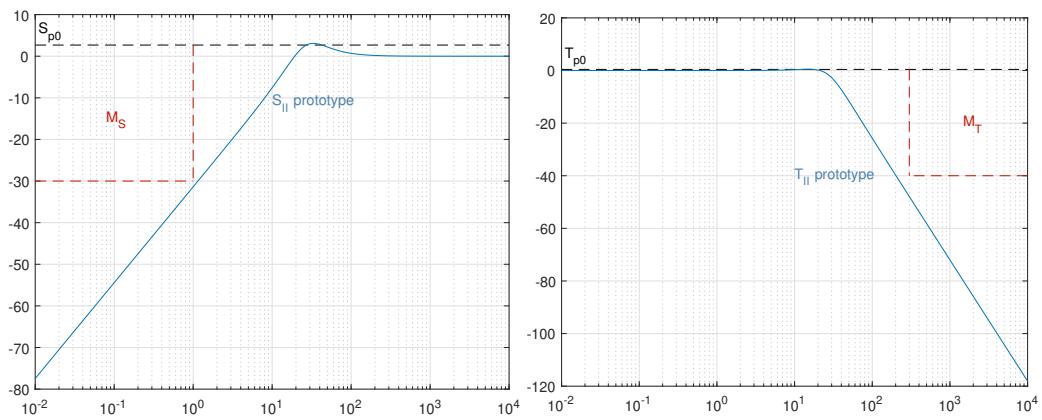


Figure 5.31: Requirements on the closed loop system are translated into constraints on the sensitivity and the complementary sensitivity functions.

and amplitude a_p , its output can be expressed in relation to $S(s)$, by using (5.13). The goal is to bound the output of the disturbance in a certain range, which can be assumed symmetric with zero mean and radius ϵ_p . This yields to

$$|S(s)| \leq \frac{\epsilon_p}{a_p} = M_S \quad \forall \omega \leq \omega_p \quad (5.15)$$

A disturbance in the feedback branch, usually related to the sensor that measures the outputs of the plant, can be limited in the same way. However, the translation of this requirement is not on the sensitivity function, but on $T(s)$. Sensor noise is usually characterized by high-frequency spectrum, so the mask is on the right side of the Bode diagram. The peak values of $S(s)$ and $T(s)$ are S_{p0} and T_{p0} , respectively. Together with the crossover frequencies, they are defined by the transient response constraints as follows

$$T_{p0} = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad (5.16)$$

$$S_{p0} = \frac{2\zeta\sqrt{2+4\zeta^2+2\sqrt{1+8\zeta^2}}}{\sqrt{1+8\zeta^2}+4\zeta^2-1} \quad (5.17)$$

The damping ζ reflects the damping of a second order system that satisfies the performance requirements.

The procedure follows with the derivation of *weighting functions* that incorporate any possible transfer function (of S and T) that is in the constraints (masks, peaks, poles, etc.). Weighting functions are assumed rational, because they can be inverted with minor effort, but this is not limiting. A mathematical framework is necessary to express the idea of "inclusion" of transfer functions. This is strongly related to norms, as they can represent a distance. In particular, the ∞ norm is considered in this work.

Given a SISO linear time invariant system with transfer function $F(s)$, the H_∞ norm is defined as

$$\|F(s)\|_\infty = \max_\omega |F(j\omega)| \quad (5.18)$$

From the design constraints weighting functions are obtained in the following form:

$$|T(j\omega)| \leq |W_T^{-1}(j\omega)| \quad |S(j\omega)| \leq |W_S^{-1}(j\omega)| \quad \forall \omega \quad (5.19)$$

In equivalent way

$$|W_T(j\omega)T(j\omega)| \leq 1 \quad |W_S(j\omega)S(j\omega)| \leq 1 \quad \forall \omega \quad (5.20)$$

In terms of H_∞ norms

$$\|W_T(j\omega)T(j\omega)\|_\infty \leq 1 \quad \|W_S(j\omega)S(j\omega)\|_\infty \leq 1 \quad (5.21)$$

This new set of constraints can be used for controller synthesis. Many approaches are available:

- traditional loop shaping techniques
- H_∞ norm minimization
 - LMI approach
 - Riccati equations approach
 - Youla-Kucera parametrization approach

With the norm minimization approach, the goal is to keep the norm of the product between each transfer function and the respective weighting function below or equal to 1. If either one of the two norms in (5.21) are greater than 1, then the closed loop system with the respective controller does not satisfy the requirements.

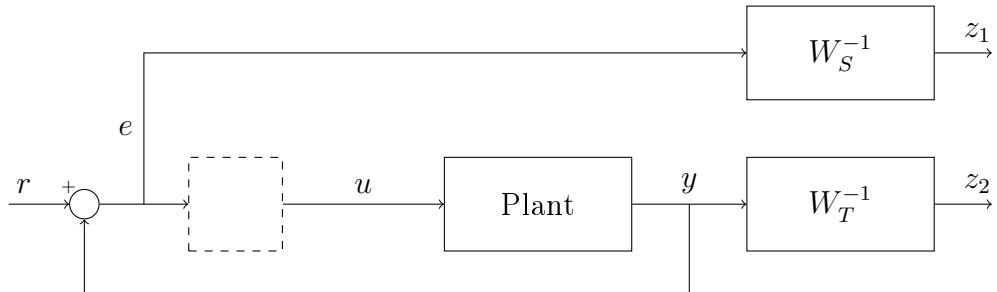


Figure 5.32: A block diagram that visualizes how the constraints are inserted in the control system transfer function.

Figure 5.32 shows the scheme of the closed loop system without the controller, but with the weighting functions. They are placed so that they output exactly the norms that have to be minimized. In fact, $W_S^{-1}(s)$ has as input the error whose transfer function is $S(s)$, and $W_T^{-1}(s)$ takes as input the output of $T(s)$. This type of structure is called *generalized plant* (diagram in Figure 5.33).

Thus the optimization problem that has to be solved is

$$G_C(s) = \arg \min_{G_C \in G_C^{stab}} \|M_{rz}(s)\|_\infty$$

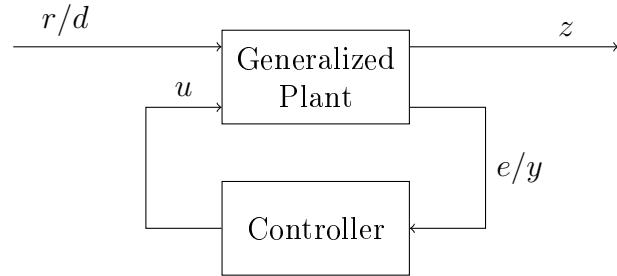


Figure 5.33: The generalized plant.

Where G_C^{stab} is the set of all the controllers that stabilize the nominal plant and $M_{rz}(s)$ is the transfer function of the generalized plant from the inputs/disturbances to the generalized variables.

$$M_{rz}(s) = \begin{bmatrix} W_S^{-1}S \\ W_T^{-1}T \end{bmatrix}$$

MATLAB's *Robust Control Toolbox* has been used to solve such problem.

Chapter 6

Results and conclusions

While the PID control method has been tested on the physical system, the H_∞ controller has only been simulated on MATLAB with the model used for the synthesis. Obviously, this prevents a complete comparison of the two approaches. However, preliminary results offer an interesting insight of the intrinsic performances and capabilities of the TMS.

6.1 PID control

6.1.1 Reference input tests

The validation of the decoupled control system has shed light on the quality of the TMS' first design and many improvements have been implemented during the testing. For example, the signal from the PTM spring potentiometer had initially a low signal to noise ratio (snr). With the addition of the amplifying stage and the low-pass filter, the quality of the measurements has increased significantly.

During the tuning procedure, the PID gains in both loops have been limited by the noise on the feedback branch. Higher accuracy could be achieved with better hardware. The step response (in Figure 6.1a) showed a settling time of around 0.2 seconds, corresponding to a positioning bandwidth of 5 Hz (consistent with earlier estimates). A sine sweep was then used to directly measure the bandwidth (in Figure 6.2). These results showed a resonant peak and phase reversal at 7 Hz. The PTM was able to faithfully track references up to around 5 Hz. Comparison with the external load cell, however, revealed an unexplained discrepancy. While the PTM spring is faithfully tracking the desired cyclic tension profile, the actual tension seen at the output does go through the same magnitude of variation (in Figure 6.1b).

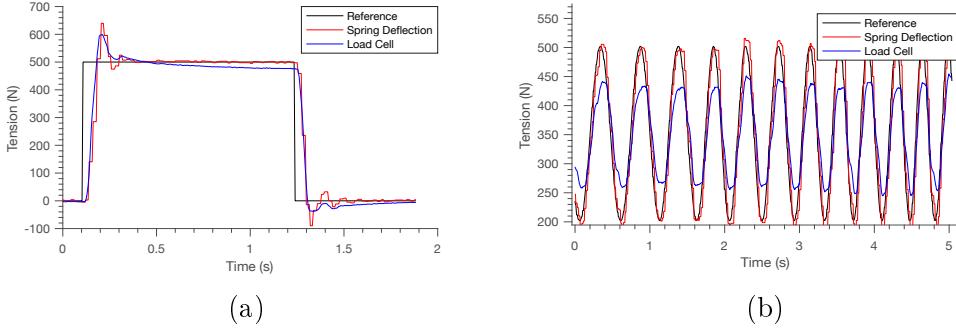


Figure 6.1: (a) Step response at full Axel weight. External load cell used to validate tension sensing from spring deflection. (b) Comparison of PTM-measured tension and external load-cell tension under cyclic loading. Discrepancy is thought to be due to the unusual strain-stiffening property of the Axel tether.

It's not immediately clear what is leading to this effect, but it may be related to previously undocumented mechanical property of the tether. Specifically, the tether exhibits pronounced "strain stiffening". The tether is typically similar to rope in flexibility, but when exposed to high stress, it becomes semi-rigid until it is flexed under low tension. This may be due to the unique construction of the tether (Figure 3.3), which includes woven layers of high-strength material surrounding a central helix of copper conductors. As the tether is stretched, the woven layers tighten around the copper, resulting in high bending resistance. Alternatively, this may come purely from plastic deformation of the copper wires. It is theorized that these effects could lead to large levels of "rolling-resistance" in the capstan when under high tension or potentially alter the forces applied by the tether on the load cell pulley.

6.1.2 Shock loading tests

A major goal of this work is to improve the drop tolerance of Axel under small incidental falls. Several drop simulations were performed with the testbench at its maximum load velocity of 0.5 m/s, with a free tether length ranging from 50cm to 30m. Slack tether was placed between the load spool and the PTM, and both the TMS controller and the load spool were activated. The plot from the 50cm test is shown in Figure 6.3. Since there is no tension on the tether, the PTM starts reeling at maximum rate to increase the tension. This is followed by a point where the tether becomes taut, simulating the drop. The PTM and the load spool then synchronize with the PTM enforcing the desired tension value. Meanwhile, the spool tension controller is able

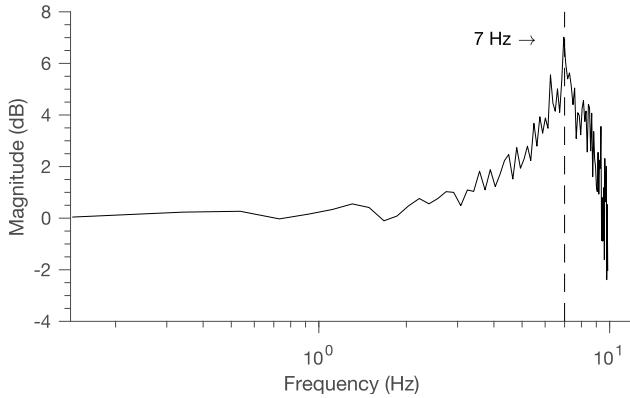


Figure 6.2: Bode plot from experimental frequency sweep. System is usable up to 5 Hz, with a resonant peak at 7 Hz.

to maintain almost perfect tension both before and after the fall. More extensive tests are needed to test the performance of the system under higher drop velocities not currently possible on the testbench due to the load motor specifications.

6.1.3 Disturbance response tests

In addition to shock loading, the system was tested for lower frequency disturbance response behavior. A sinusoidal disturbance profile was applied by the load motor while the PTM was commanded to maintain a desired reference tension (Figure 6.4). Under these conditions, the PTM was able to regulate the tension to within 50 N up to 1 Hz and 120 N of the reference value over the full frequency range (0.1 to 2 Hz). Once again, a large discrepancy was observed between the PTM-estimated tension and the external measured tension. This was consistent across the full frequency spectrum.

6.2 H_∞ control

Simulation of the designed controller has been done with the same model used for controller synthesis, but a minor modification has been introduced to include the saturation of the motors. In fact, their input signal, which is a current set-point, is limited by the maximum current allowed on the motors. The time-step of the simulation, 1 ms, is a tradeoff between a value that guarantees likelihood to the reality and a value that allows a short simulation time.

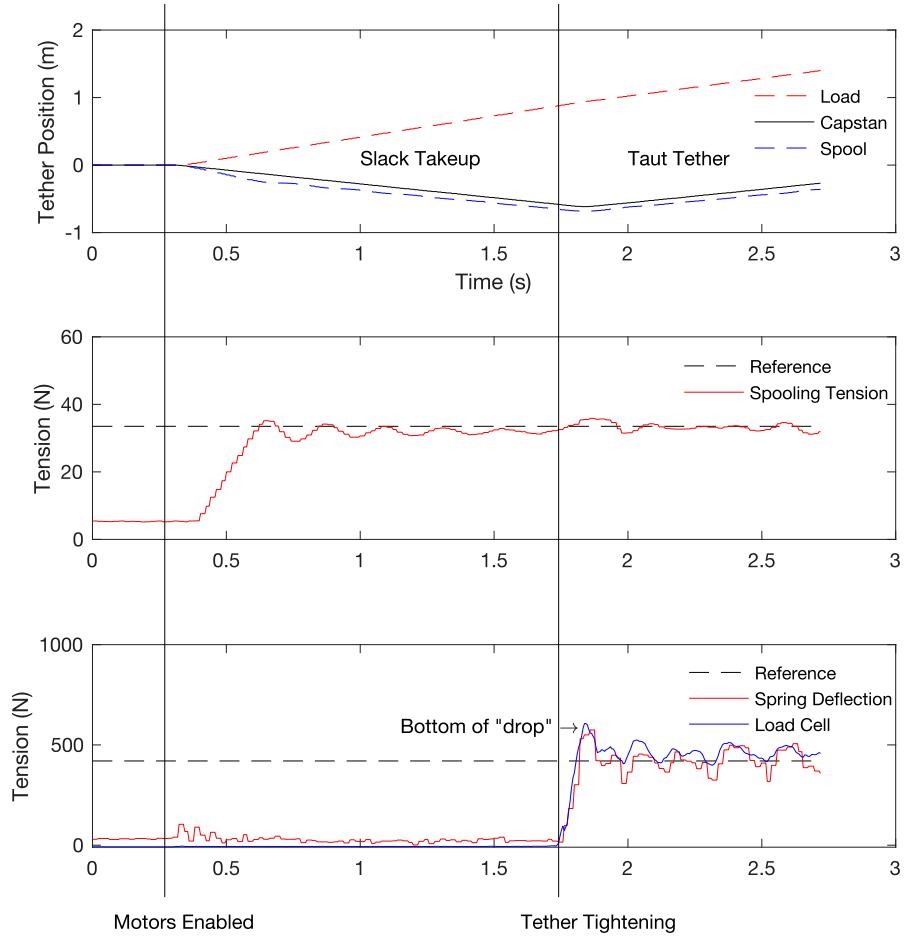


Figure 6.3: Performance of the TMS under a simulated drop. Proper spooling tension is maintained throughout the fall and the output tension is controlled to be close to the desired reference. The offset between the spool and capstan due to the tether stored in the slack buffer.

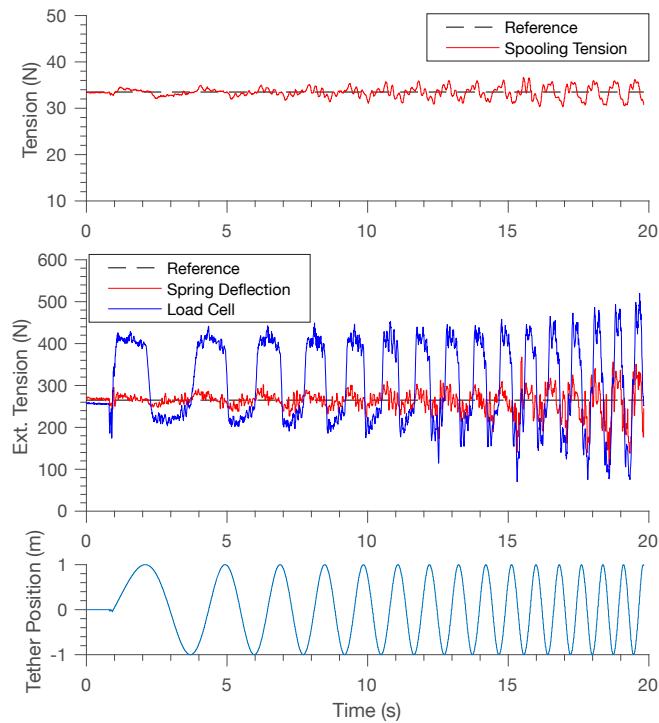


Figure 6.4: Behavior of the TMS under a cyclic 1 meter external position disturbance. Spooling tension is almost entirely isolated from the effects. External tension as measured by the SEA is well controlled, but the load cell again indicates a strong friction-like effect from the tether.

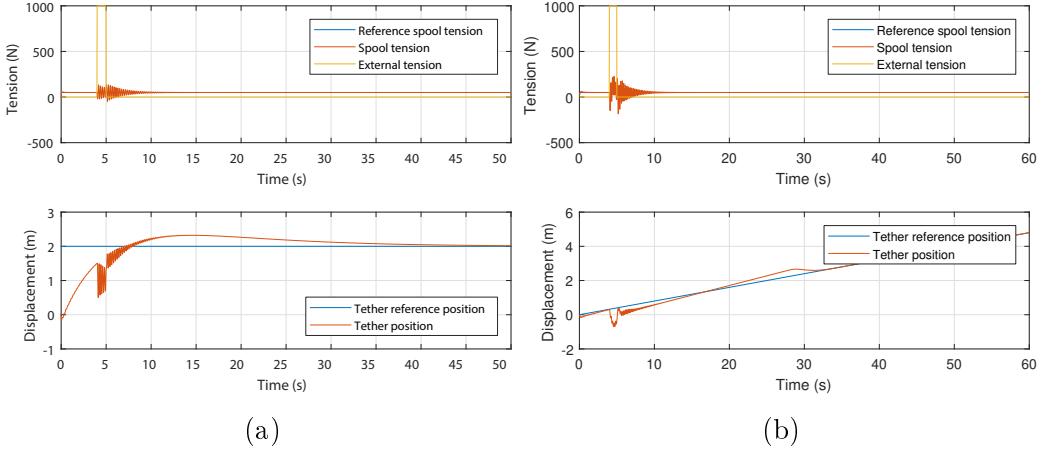


Figure 6.5: (a) The response of the system to a step input on the tether position reference and an external disturbance of 1000 N at 4 s. (b) The response of the system to a ramp input on the tether position reference and an external disturbance of 1000 N at 4 s.

In Figure 6.5a the result of a step input simulation is shown: the system is able to correctly track both the internal tension reference set-point (50 N) and the position set-point of the tether (2 m). The external tension is a step disturbance acting at 4 s for 1 s: its effect on both output is canceled by the control system in a short time, around 5 s. However this is still not satisfactory, because the steady state in the position control is reached after a long time, 45 s, and this too much even for a slow robot like Axel. Improvement could be obtained with a better model and the addition of feed forward component that speeds up the rise time. Also, the damping of the positioning system seems to be too low. In fact the oscillations around the steady state value last for several seconds. Figure 6.5b presents the result of the previous simulation set-up, with a different type of reference for tether position. In this case, a ramp with coefficient 0.08 has been used as input. In other words, it has been enforced a speed of 8 cm/s of the tether towards inside of Axel, that is the sense of winding. The transient period seems to be much shorter than the previous case, however there is a glitch happening at around 30 s. The exact reason of this is not yet clear, but it might be some latent dynamics in the system that are really slow and their output shows with significant delay. Alternatively, it may be possible that, due to the disturbance, the control of the position is firstly managed by the PTM subsystem (during the disturbance), and then the spool handles the positioning of the tether. Further investigation is necessary to understand this behavior.

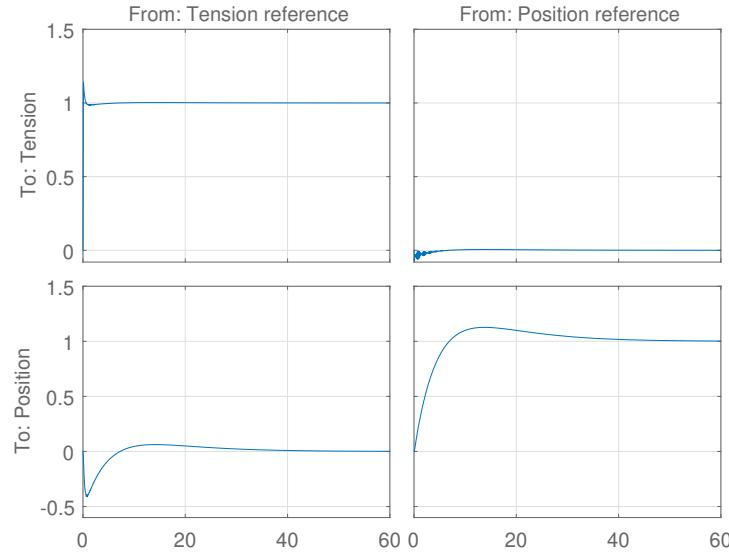


Figure 6.6: MIMO step response of the control system.

Finally, Figure 6.6 shows an interesting property that is often intrinsic in H_∞ controllers: the decoupling of the two subsystems. In fact, it is possible to distinguish separated actions of the two inputs on the two outputs. During the transient state they are not independent, as each input can affect both outputs, but during steady-state the two systems are completely decoupled. Following the insight on performance given by the previous simulations, it can be noted that the positioning of the tether is characterized by a really slow dynamic behavior, that happens in range of seconds.

6.3 Spooling and level winder test



Figure 6.7: Two images of the spooling: before and after hitting the edge.

The result is a well performing control system, however, as the width of

the drum is not an integer multiple of the tether diameter, there is some empty space in the edges, which creates some glitches in the windings, after some meters of spooling. One possible solution is to add some kind of compliant edges that can bend to allow tether in without leaving gaps, otherwise the spool drum has to be exactly a multiple of the tether diameter in width. Spooling has been tested up to the maximum speed of the system, and it does not seem affected by how fast the system runs. On the other hand, a better performing level winder system could use a closed loop control to position the fair lead. In fact, a compact spooling is required to avoid knifing and compactness could be guaranteed by enforcing a *lead angle*. It means that the fair lead is always in advance with respect to the tether in the drum when winding: in this way the tether will be placed as close as possible to the previous winding. On the other hand, this is impossible to achieve with an open loop system like the one described, because there is no way to know the position of the tether on the drum. There are many ways to measure it: for example, computer vision could be used to reconstruct the status of the drum. However, this solution could be affected by dust and absence of internal light, then a mechanical measurement device would be preferable. A joint placed on the fair lead could move with the tether's angle and an encoder could measure it, thus providing the required value.

6.4 Future work

The work described in this document has obtained good results in exploring an innovative application of robotic mobility. Despite the limited time available for testing, interesting results have been achieved. They give an useful insight on the TMS, and it is already possible to suggest some guidelines to improve this system and the respective test bed.

- For what concerns the test bed, the load cell assembly design should be revisited to better constrain transform the tether load into a force that compresses the load cell. Probably a different geometry is required to avoid unwanted bending. Also, the load motor with its gearbox is not able to provide enough speed on the tether to simulate a realistic free-fall drop from more than 50 cm. Suggestion is to swap the reduction or the load motor so that the test bed is capable of producing the right stresses on the TMS.
- PID control could be satisfactory with a better tuning and the introduction of a feed-forward action. However, there are some intrinsic properties of the tether and the system that limit its performance.

Further investigation on these factors is needed to understand how a mechanical redesign could remove them. Finally, a position control loop can be designed around the two control loops to command tether spooling and unspooling.

- Despite H_∞ controller works in simulation, its performances are not in line with what wanted. Probably, an improved model of the system that includes friction and tether strain could increase the controller's performances. The implementation of the H_∞ controller can be done easily via software on the TK-1 and it is necessary to compare the two control systems.

Bibliography

- [1] T. Brown, A. Stefanini, N. Georgiev, J. Sawoniewicz, and I. A. D. Nesnas. Series elastic tether management for rappelling rovers. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.
- [2] R. Greeley. Lava Tubes and Channels in the Lunar Marius Hills. *Moon*, 3:289–314, December 1971.
- [3] Junichi Haruyama, Kazuyuki Hioki, Motomaro Shirao, Tomokatsu Morota, Harald Hiesinger, Carolyn H. van der Bogert, Hideaki Miyamoto, Akira Iwasaki, Yasuhiro Yokota, Makiko Otake, Tsuneo Matsunaga, Seiichi Hara, Shunsuke Nakanotani, and Carle M. Pieters. Possible lunar lava tube skylight observed by selene cameras. *Geophysical Research Letters*, 36(21), 2009. L21206.
- [4] T. Kaku, J. Haruyama, W. Miyake, A. Kumamoto, K. Ishiyama, T. Nishibori, K. Yamamoto, Sarah T. Crites, T. Michikami, Y. Yokota, R. Sood, H. J. Melosh, L. Chappaz, and K. C. Howell. Detection of intact lava tubes at marius hills on the moon by selene (kaguya) lunar radar sounder. *Geophysical Research Letters*, 44(20):10,155–10,161, 2017. 2017GL074998.
- [5] L. Kerber, I. Nesnas, L. Keszthelyi, J. W. Head, B. Denevi, P. O. Hayne, K. Mitchel, J. Ashley, J. Whitten, A. Stickle, M. Paton, K. Donaldson-Hanna, R. Anderson, D. Needham, P. Isaacson, , and L. Jozwiak. Moon diver: A discovery mission concept for understanding the history of the mare basalts through the exploration of a lunar mare pit. In *49th Lunar and Planetary Science Conference*, 2018.
- [6] Alfred S. McEwen, Lujendra Ojha, Colin M. Dundas, Sarah S. Mattson, Shane Byrne, James J. Wray, Selby C. Cull, Scott L. Murchie, Nicolas Thomas, and Virginia C. Gulick. Seasonal flows on warm martian slopes. *Science*, 333(6043):740–743, 2011.

- [7] Lujendra Ojha, Mary Beth Wilhelm, Scott L. Murchie, Alfred S. McEwen, James J. Wray, Jennifer Hanley, Marion Massé, and Matt Chojnacki. Spectral evidence for hydrated salts in recurring slope lineae on Mars. *Nature Geoscience*, 8(11):829–832, 10 2015.
- [8] G. Meirion-Griffith, I. Nesnas, L. Kerber, R. Anderson, T. Brown, F. Calef, J. Burdick, , and M. Tanner. Accessing Mars recurring slope lineae: Mobility systems analysis. In *2018 IEEE Aerospace Conference*, March 2018.
- [9] P. Abad-Manterola, I. A. D. Nesnas, and J. W. Burdick. Motion planning on steep terrain for the tethered Axel rover. In *2011 IEEE International Conference on Robotics and Automation*, pages 4188–4195, May 2011.
- [10] Issa A.D. Nesnas, Jaret B. Matthews, Pablo Abad-Manterola, Joel W. Burdick, Jeffrey A. Edlund, Jack C. Morrison, Robert D. Peters, Melissa M. Tanner, Robert N. Miyake, Benjamin S. Solish, and Robert C. Anderson. Axel and duaxel rovers for the sustainable exploration of extreme terrains. *Journal of Field Robotics*, 29(4):663–685, 2012.
- [11] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, and H. Das. The claraty architecture for robotic autonomy. In *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, volume 1, pages 1/121–1/132 vol.1, 2001.
- [12] E. Ferrentino. Tethered mobility and control for extreme terrain rovers Axel and Du-Axel rovers. Master’s thesis, Politecnico di Torino, 2014.
- [13] A. Balossino. Design and implementation of a tether management system for the Axel rover. Master’s thesis, Politecnico di Torino, 2013.
- [14] D. W. Robinson, J. E. Pratt, D. J. Paluska, and G. A. Pratt. Series elastic actuator development for a biomimetic walking robot. In *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399)*, pages 561–568, Sept 1999.
- [15] G. A. Pratt and M. M. Williamson. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 399–406 vol.1, Aug 1995.

- [16] Chan Lee, Suhui Kwak, Jihoo Kwak, and Sehoon Oh. Generalization of series elastic actuator configurations and dynamic behavior comparison. *Actuators*, 6(3), 2017.