POLITECNICO DI TORINO

FACOLTÀ DI INGEGNERIA

Master's Degree in Mechatronic Engineering
(Ingegneria Meccatronica)

Master's Degree Thesis

Design, implementation and control of an electromechanical perturbation device for postural control evaluation

Supervisor:
Prof. Daniela Maffiodo

Candidate:
Daniel Pacheco Quiñones

Co-Supervisors:
Prof. Carlo Ferraresi
Prof. Walter Franco
Prof. Giovanni Gerardo Muscolo

OTTOBRE 2020

"The greatest challenge to any thinker is stating the problem in a way that will allow a solution."

Bertrand Russell

Abstract

Orthostatic equilibrium conservation is one of the most critical features of the human body. Nevertheless, the biological control system behind it can be subject to reduced performance due to physical incidents, neurological pathologies, or aging. For this reason, recent researches on the biomedical field have focused on a proper method able to evaluate it in a univocal and replicable way for medical checkup reasons.

Born as a collaboration between Politecnico di Torino Mechanics and Aerospace (DIMEAS) department and the Università degli Studi di Torino Neuroscience department, the Perturbation Generation Analysis System (PGAS) project revolves around the study of an alternative, more flexible way to evaluate human postural control. The system entails the use of a perturbative system, able to introduce a definite, unidirectional, and plannable source of disequilibrium on the subject's body, and a rigid ground platform, to calculate the subject's reaction to said distress. During the years, the perturbation system's various prototypes have focused on the biomedical correlation between the disequilibrium entity and its effects, but, as the PGAS research kept developing, the main problem has always involved the distress generation control. This is not only because manual or, later, pneumatic actuations introduce strong dynamic complications to the overall system, but also since human body reaction and general contact force dynamics are both highly complex, uncertain and nonlinear phenomena.

This study aims to solve the control problem and refine the overall device design by implementing a linear electrical motor to the system, able to damp out or even delete many criticalities of the previous prototypes. The distress full characterization and design has revolved around planning a control algorithm that introduces stepped force profiles to body-machine contact interactions. This is because both force amplitude and impulse duration values were deemed critical to the study for a good biomedical characterization of the phenomenon. Considering that, as a basic approach to the problem, the system was approximated to a linear one, different straightforward closed-loop control methods were tested. Nevertheless, a serendipitous approach to the problem was deemed unable to adequately perform due to the electric motor's severe control input saturation. Therefore, the study has focused on a control algorithm able to consider this problem in advance, maintaining good overall performance, especially during the actuator's strike phase.

As for the device implementation, the system has been reproduced and tested in a simulation environment, then put into service in a workbench for hardware-in-the-loop testing. The use of the electric motor frequency converter, which also contains a practical pico-PLC for finite state machine development, has permitted a smoother and more compact global design with respect to the previous pneumatic PGAS device, whose solenoid valves required an air compressor to work. This physical domain reduction is considered as a key flexibility enhancement since the final system aims to a portable device that can be handled by the operator.

As for the simulation and workbench testing, the used control algorithm has proved good first approach performances, but still requires robust control features further enhancements able to consider and damp out the many nonlinearities and uncertainties involved in the phenomenon. Nevertheless, electromechanical actuation proved to be a far more efficient, straightforward and compact solution than the previous ones, and the prototype will serve as a good starting point system for future development.

Index

Index of Figures

Index of Tables

Chapter 1: State of the art

1.1 Overall introduction

Postural control can be undoubtedly enlisted as one of the most critical features of the human body. Not only balance regulation is essential for accomplishing autonomous and semi-autonomous movement, but also for managing random - and possibly abrupt - interactions with the external environment. Since postural control is so crucial, not only one, but three different sensorial systems partake in achieving its correct regulation: the somatosensory system, the vestibular system, and the visual system. Considering that the upright position denotes an unstable equilibrium point - roughly like in the inverse pendulum case -, the aforementioned systems are continuously working together, correcting the body's posture to reach the specific task (static equilibrium or movement), and reducing the risk of falling at a minimum. Balance recovery is then managed through three different strategies, which are, in order of the entity of the disequilibrium:

- Ankle strategy (Figure 1.1*a*), commonly used to maintain orthostatic (i.e., upright) position, useless for higher balance perturbations since it holds an irrelevant moment of inertia;
- Hip strategy (Figure 1.1*b*), for moderate balance perturbations and undersized or loose support surfaces;
- Step strategy (Figure 1.1*c*), used for high or too abrupt balance perturbations when the other two strategies are deemed useless.

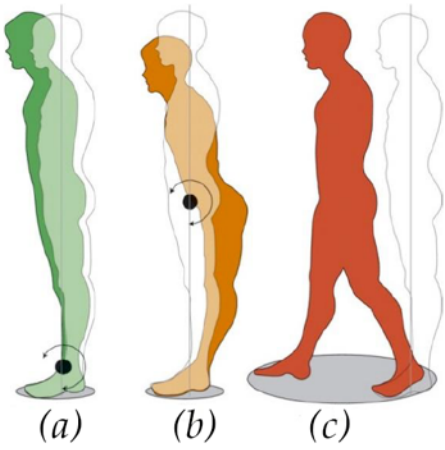

Figure 1.1: Ankle, hip and step strategies

Orthostatic equilibrium is achieved when the body Center of Mass (CoM) is aligned with the foot-ground Center of Pressure (CoP). [1]

Nonetheless, postural control can be easily subject to reduced performance and, in worse cases, failures mostly due to physical traumas, neurological pathologies, and aging. Achieving an objective postural control analysis, able to surely quantify and compare over time the subject's balance skills, is consequently becoming a matter of interest in the biomedical and bioengineering fields. Studies on the matter have achieved too few standardized protocols, and they all focus on the interaction between the patient and an oscillating platform as the perturbing system. We can take as an example the following three methodologies:

- Motor Control Test (MCT) (Figure 1.2*a*);
- Adaptation Test (ADT) (Figure 1.2*b*);
- Sensory Organization Test (SOT) (Figure 1.3).

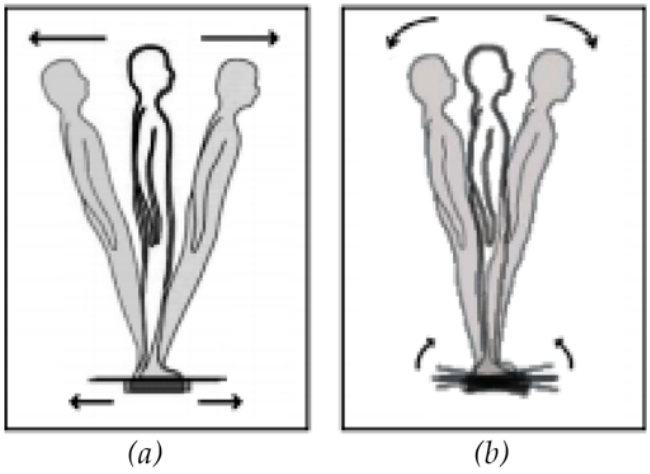

Figure 1.2: Motor Control Test (MCT) and Adaptation Test (ADT)

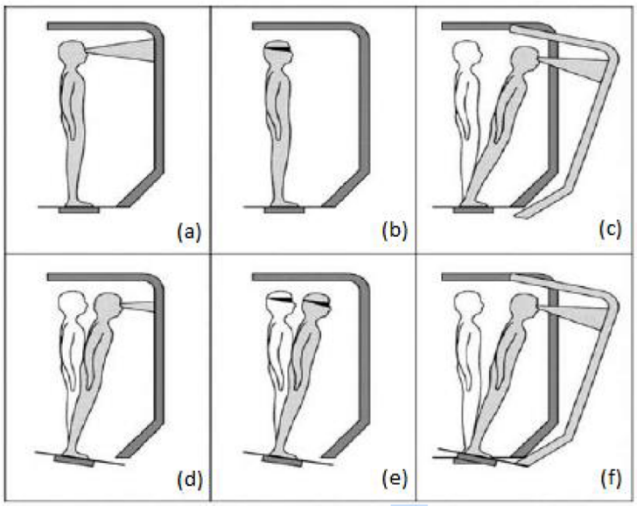

Figure 1.3: Sensory Organization Test (SOT)

In each of these protocols, the subject is instructed to maintain the balance on the platform after perturbing it with anteroposterior translational (MCT) or pitch rotational (ADT) oscillations. In addition to adapting to the platform's movement, the patient could be subject to a conflictual visual stimulus on a front screen (SOT) [1]. Balance skills can thus appropriately be measured through the displacements of the CoM. However, a standing platform system is far from being a realistic portrayal of an individual's most common life activities, in which external interactions could potentially take place on, and consequently perturbate, any part of the human body. Moreover, these methodologies could produce biased results due to the rather predictable source of the perturbation. These matters, along with generally big overall dimensions of the aforementioned designs, have compelled the bioengineering field to seek for a more flexible and compact solution.

1.2 Further postural control analysis devices

To achieve a non-biased and realistic response from the subject, many studies developed different approaches to the matter, regarding different subject positions and environment interactions.

A noteworthy experiment, shown in Figure 1.4, studies the subject in a movement condition. The patient is instructed to move on a walkway equipped with 14 hidden ground obstacles, arranged to perturb the subject balance and study their recovery from tripping. In addition to this, subjects are asked not to land on a specific projected forbidden zone, if present. A random number of normal walking trials is added in between the experiments to achieve a non-predictable subject-obstacle interaction. Focus is thus set on the patient's foot trajectory correction, and balance skills quantification is based on feet position with respect to body overall angular momentum during and following the recovery foot landing. [2]

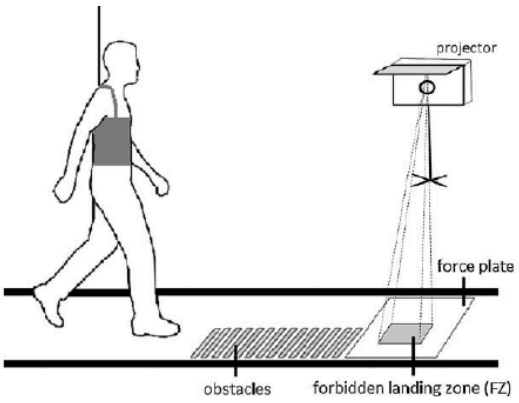

Figure 1.4: Tripping correction analysis

A more complex perturbation environment during gait is shown in Figure 1.5. In this experiment, the patient has to walk on a treadmill and is subject to lateral balance perturbations, given by a computer-controlled pneumatic device. The study analyzes the subject's different reactions concerning the same known perturbation occurring in different phases of the stride cycle. Data acquisition focuses on different foot and muscle strategies to restore balance. [3]

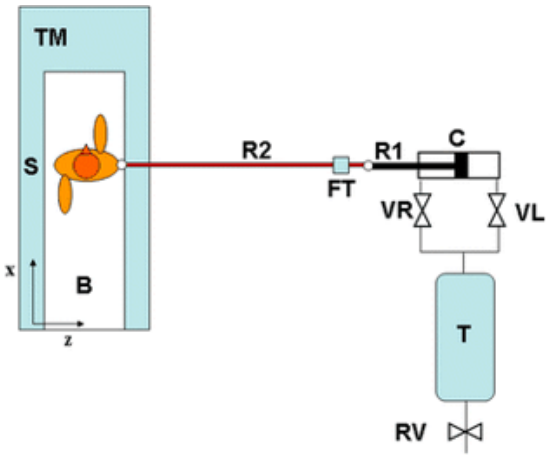

Figure 1.5: Lateral perturbation analysis system

Another research focuses on the subject's ability from recovering to unexpected loading of the spine and studies it in terms of the trunk neuromuscular and kinematic response. The patient is placed in a semi-seated position through slings, connected to a perturbation system shown in Figure 1.6. The study analyzes the effects of four parameters: preload, sudden load, initial trunk flexed posture, initial abdominal antagonistic activity. Data are collected through a potentiometer and a load cell. [4]

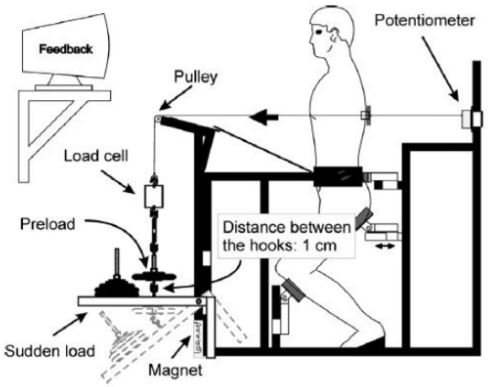

Figure 1.6: Trunk response analysis system

To reduce the predictability of the interaction with the environment, other studies developed more complex experimental systems, like the one shown in Figure 1.7. In this particular research, a hidden system of slings and pulleys makes the direction of the force, which is applied through the potential energy of a weight-drop, unpredictable to the patient. The subject's response is thus, to some extent, more realistic with respect to the previous experiments, at the expense of developing a more complex and bulkier design. [5]

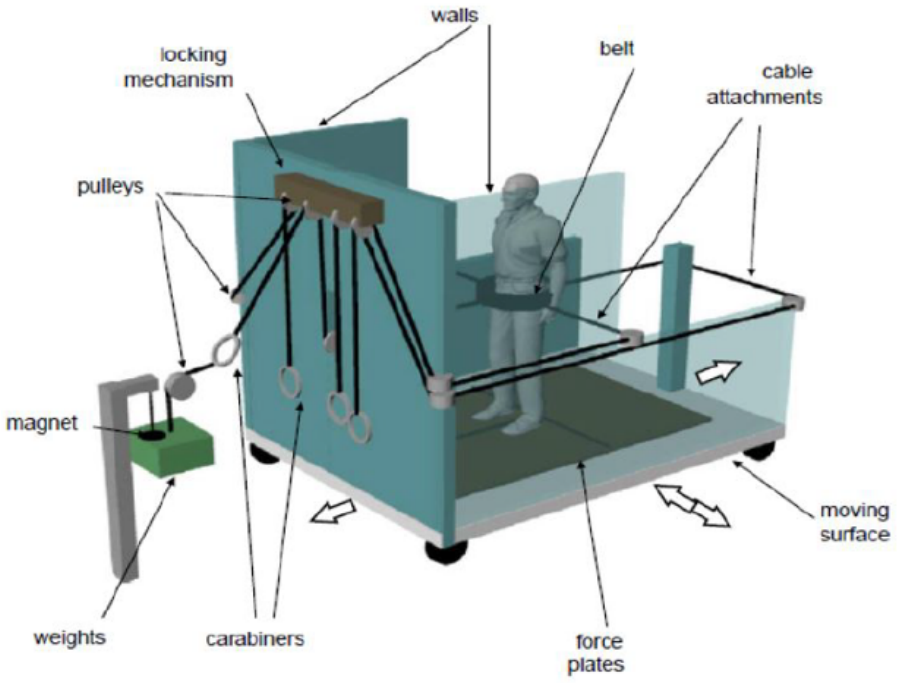

Figure 1.7: Weight-drop cable pulls system (CPs)

1.3 PGAS, first prototype

The Mechanics and Aerospace Department (DIMEAS) of Politecnico di Torino, in collaboration with the Neuroscience department of Università degli studi di Torino, began new research based on a manual postural perturbator, named Perturbation Generation Analysis System (PGAS), shown in Figure 1.8 and Figure 1.9. The study focuses on a way of overcoming the problematics of flexibility and compactness presented in Paragraph 1.1, designing a device that can be easily held by the operator through a handle. The balance-perturbing blow can then be administered on any point of the patient's body in an unpredictable way directly chosen by the operator. A clamping plate connects the handle of the device to a cylinder containing a uniaxial load cell. Said cell, adequately mounted under the polyethylene foam contact surface, measures the impulsive force of the strike. The postural control analysis system is complemented by a rigid ground platform, measuring, through force plates, the CoP displacement of the patient's body response, and allowing a comparison with the strike entity.

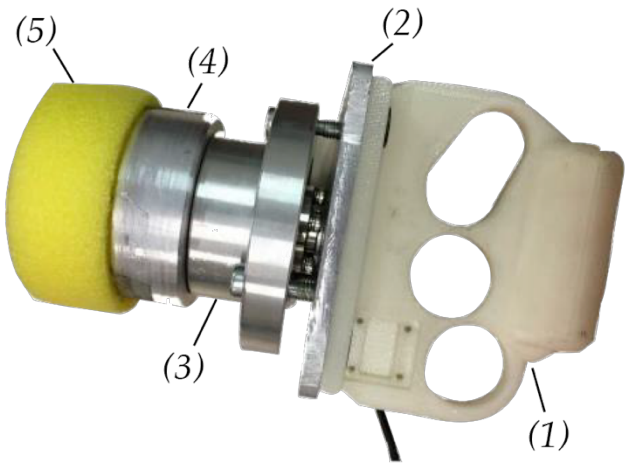

Figure 1.8: PGAS, first prototype

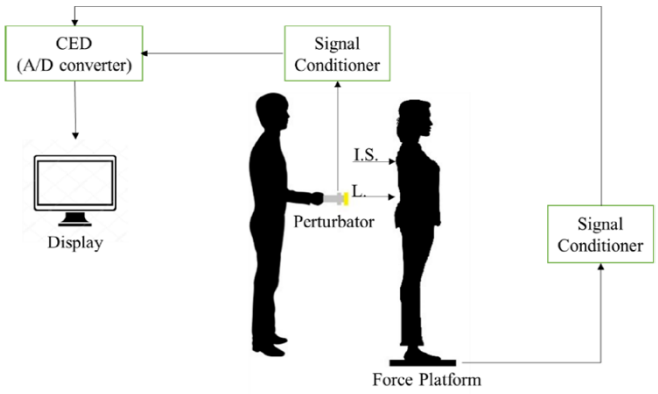

Figure 1.9: PGAS, first prototype experiment setup

#	**Component**
1	Handle
2	Clamping plate
3	Cylinder with uniaxial load cell
4	Support cylinder
5	Polyethylene foam interface

Table 1.1: PGAS, first prototype components list

However, the economic affordance of the relatively simple design of this solution is achieved at the expense of unmanageable experiment repeatability and control, due to lack of physical maneuverability of the manual blow. Operator training gives little to no advantage, and unbiased data acquisition is hard to collect. [1]

1.4 PGAS, second prototype

Research about PGAS has concentrated on the way to solve the repeatability problem of the prototype described in Paragraph 1.3. To achieve this, the design has revolved on device automatization through a linear pneumatic perturbator. The semiautomatic device can either be directly held by an operator through handling, as in Figure 1.10, or mounted on a fixed frame. The handled configuration's components are listed in Table 1.2.

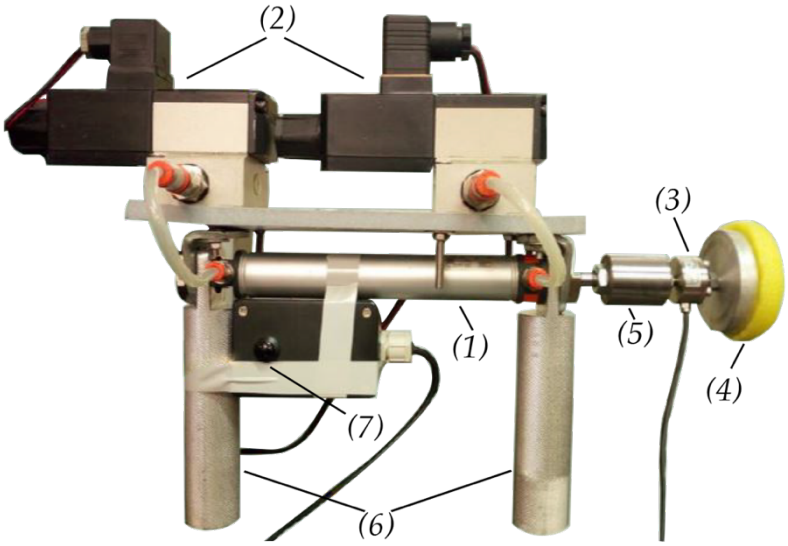

Figure 1.10: PGAS, second prototype, handled configuration

#	**Component**
1	Double-acting cylinder MetalWork ISO 6432
2	Flow proportional valves CKD, 3AF2 with respective control boards CKD APC-23
3	Uniaxial load cell Dacell UMM
4	Polyethylene foam interface
5	Screw adapter
6	Aluminum handling cylinders
7	Release button

Table 1.2: PGAS, second prototype components list

The control environment has been developed in the *MathWorks Simulink* environment, using a *dSpace* machine as the real-time controller of the proportional valves. Strike ensues once the release button is pushed, and the machine self-retracts at the end of every impact. Control system hardware features are shown in Figure 1.11.

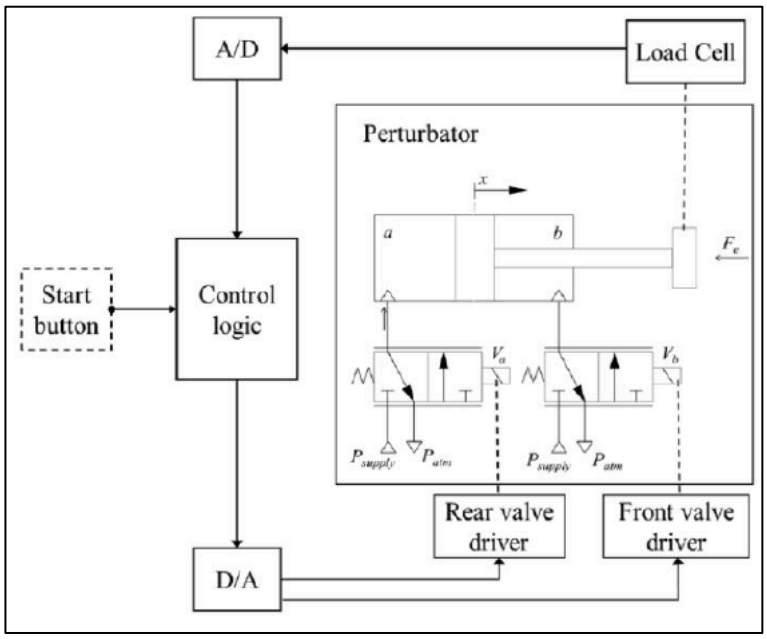

Figure 1.11: PGAS, second prototype control system

This new setting allows proper control of the most relevant variables involved, e.g., the impact force and duration, reference valued at 50 N and 250-500 ms. In fact, research on postural control with this device discovered that there's a direct correlation between human body reaction and force impulse, i.e., the time integral of the force, rather than force itself. This is to say that also impact duration is a crucial parameter to be controlled and properly set. [6]

Nonetheless, control optimization is still a difficult task. The PGAS prototype faces a series of uncertainties and nonlinearities which are difficult to manage, including:

- The stick-slip phenomenon, happening between the rod and the inside guideways of the double-acting cylinder;
- The fluid dynamics involved inside the valves and the front and rear chambers of the cylinder, due to compressed air being a real gas and flux losses, which are still too high even if tube length is minimized by putting the valves directly on the cylinder;
- The high nonlinearity involving the impact phenomenon and the relative contact force, which is also remarkably complex to model. Being the uniaxial load cell the only sensor device involved in the control feedback loop, uncertainties involving the impact first instant and perpendicular dynamics entity are added to the problem [7];
- General uncertainties involving the subject's body dynamics;
- In the handled configuration, as in the first prototype, the operator's body dynamics could potentially compromise the experiment producing biased results. This means that this uncertainty has to be added to the problem.

In addition to the control difficulties, a significant problem involving the adoption of a pneumatic system is the necessity of an air compression system, which makes the overall design bulkier and more costly than the previous prototype.

1.5 PGAS, third prototype

To solve a few of the problems of the previous prototype presented in Paragraph 1.4, the DIMEAS division developed a third design, using the same electro-pneumatic approach of the second one. The device is shown in Figure 1.12 and its components are listed in Table 1.3.

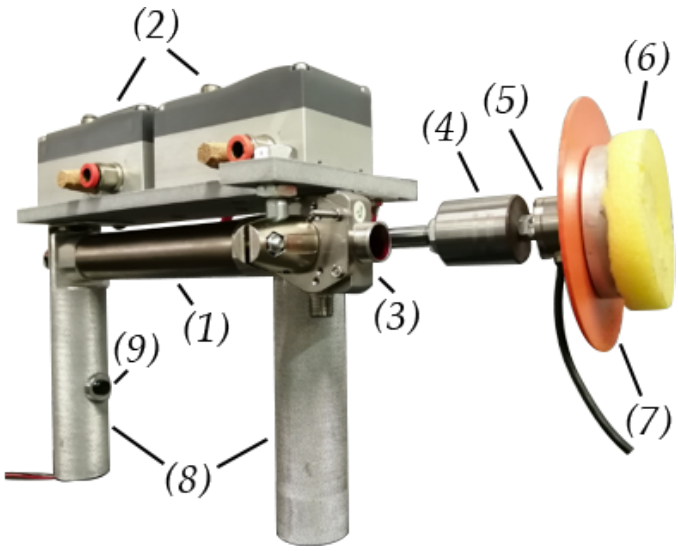

Figure 1.12: PGAS, third prototype, handled configuration

#	**Component**
1	Double-acting cylinder SMC MQMLB25H-100D
2	Flow proportional valves Camozzi LRWD2-34 with relative controller boards
3	Laser sensor Banner Q4XTULAF300-Q8
4	Screw adapter
5	Uniaxial load cell Dacell UMM
6	Polyethylene foam interface
7	Target disk for optical sensor
8	Aluminum handling cylinders
9	Release button

Table 1.3: PGAS, third prototype components list

The third prototype implemented:

- An optical laser sensor as a rod position transducer. This addition makes the control of the approach and retraction phases of the rod much more manageable, closing the feedback loop in position or in velocity;
- Much more performing valves, with lower response time and weight;
- A low-attrite cylinder with a metal-to-metal-sealing system;
- A real-time *Speedgoat* system, which is more performant and compact than the previous *dSpace* environment. This implementation allows more fitting output curves (i.e., lower output-reference errors) during the strike phase and less bulky design.

A proportional-integral (PI) controller was chosen as the controller structure in both approach/retraction and strike phases. Regarding the latter, the PI values were accurately selected through bench testing and were designed in the MathWorks Simulink environment to change according to the entity of the blow, in order to achieve more fitting output curves.

Due to the improved performance of the new prototype, more accurate testing could be accomplished. On the matter, two on-subject tests were elaborated to obtain relevant biomedical results:

- A constant reference force ($F = 40\ N$) test, in which strike's time duration varies from 50 to 250 ms;
- A constant reference impulse ($I \cong F * t = 5\ Ns$) test, in which strike's force amplitude and time duration vary, respectively, from 20 to 100 N and from 50 to 250 ms;

Nonetheless, the nonlinearity problems due to fluid dynamics make the control task still challenging to manage. This is because, during the strike phase, high performance is requested in very small amounts of time. Considering also that the controlled system is highly nonlinear, as mentioned in Paragraph 1.4, the adopted PI controller isn't able to obtain suitable minimizations of the impact force output-step reference error, resulting in too high values of overshoot, delay, and ripples' amplitude as shown in Figure 1.13.

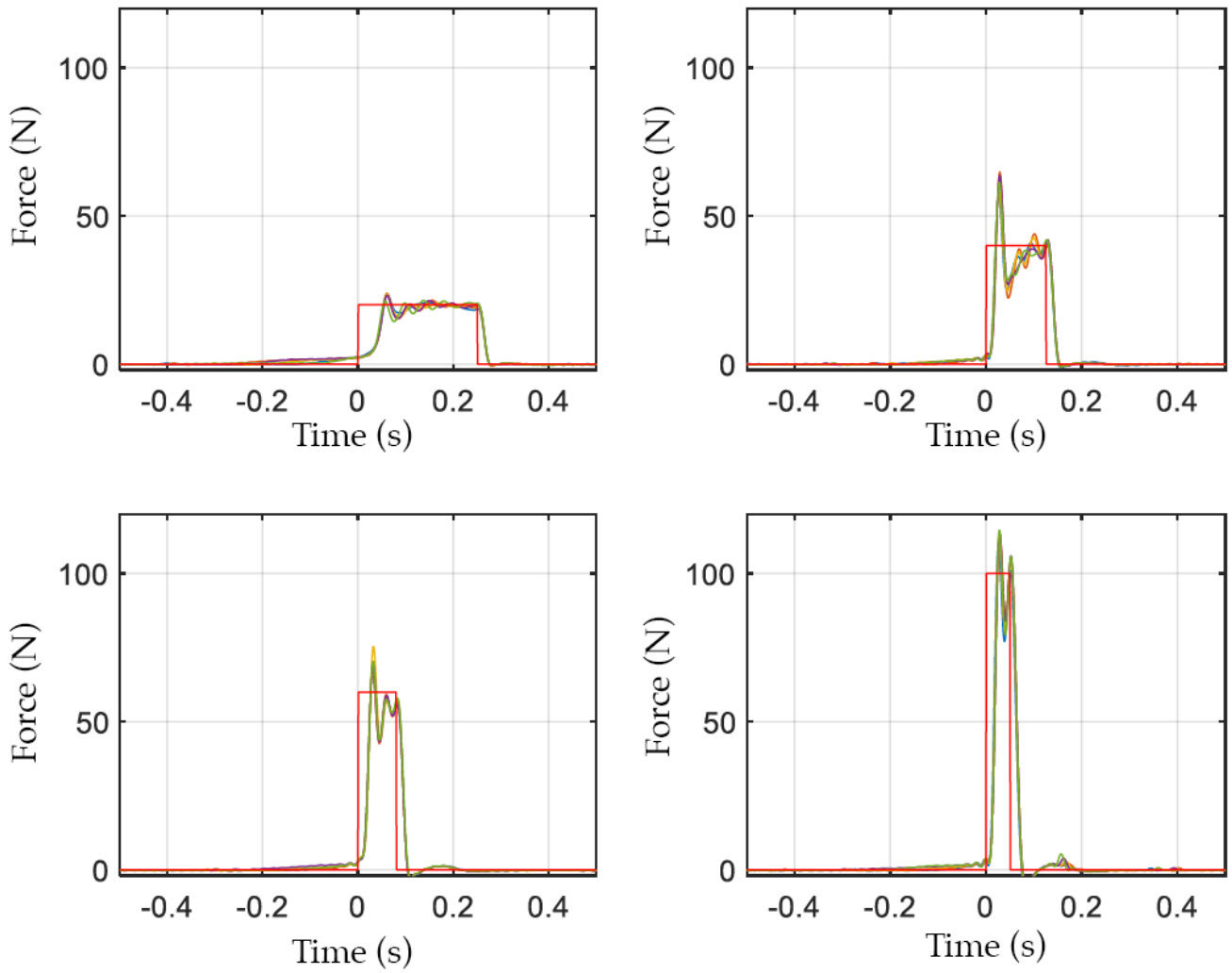

Figure 1.13: PGAS, third prototype impact force profiles (step reference in red)

Chapter 2: PGAS, fourth prototype design

2.1 Premise

The fourth prototype of the PGAS perturbator revolves around an entirely new approach, involving the adoption of a linear synchronous electric motor (LSM) as the balance perturbation striking system. As a matter of fact, using a linear electric motor provides many benefits, including:

- The multi-domain system to be controlled is only electro-mechanical and does not involve pneumatic subsystems as in the second and third prototypes. This solution produces benefits involving:
  - Control strategies, since nonlinearities and uncertainties generated by the fluid dynamics of the compressed air entering in the pneumatic piston chambers are no longer present;
  - System compactness, economic affordance, and portability, since the system no longer needs an air compressor;

- More efficiency than using a rotational electric motor and rotational-to-linear mechanical couplings (e.g., screw-nut, gear-rack) due to direct power transfer. This translates in:
  - Fewer power losses due to attrite;
  - Decreased mechanical parts' wear;
  - Faster response and higher precision;
  - Little to no need for lubricant oils.

Given these premises, the DIMEAS division seeks to overcome the problems presented in Paragraphs 1.4 and 1.5, developing a much more performative and reliable model.

To achieve an accurate comparison of the fourth and third prototypes, the same testing characteristics as the ones described in Paragraph 1.5 are operated.

To model and test both the prototype and the control algorithm used, Model-In-The-Loop (MIL) and Hardware-In-The-Loop (HIL) testing are adopted. Since the Speedgoat Real-Time target machine selected is expressly designed to work with *Simulink*, and the control environment is then managed online via PC, no embedded controller is needed, and Software-In-The-Loop (SIL) testing is unnecessary.

2.2 Plant model design

As mentioned in Paragraph 1.1, orthostatic equilibrium is reached if and only if the body's CoM vertically aligns with the foot's CoP, i.e., the overall weight force and the vertical component of the support reaction lie on the same straight line, so that the moment arm of the couple is canceled. This condition expresses a dynamic balance since the erect position is always subject to body sway, associated with the variation of the position of both the centers due to external and internal micro-perturbations. Since CoM position is more complicated to determine given that it requires the precise position of different body segments, posturography refers only to the CoP displacement for body sway characterization, although the two are not the same. With the aim of analyzing, as described in Paragraph 1.5, only low valued perturbations, the human body orthostatic position can be reached mainly through the ankle strategy described in Paragraph 1.1. The human body orthostatic position can consequently be roughly schematized as an inverse pendulum, as in Figure 2.1. [8]

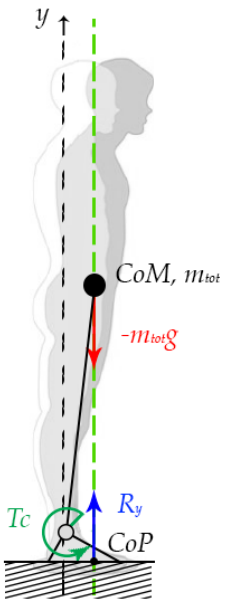

Figure 2.1: Inverse pendulum human body schematization, orthostatic equilibrium

Literature offers many models to examine human responses to balance perturbations. Among them, a study schematizes the human reaction as mastered by a PID controller, whose parameters K_p, K_d, and K_i are related to, respectively, stiffness and damping properties and the steady-state error value, inside a delayed closed loop on the angular displacements, as in Figure 2.2. As a matter of fact, once the body receives a perturbation, for a time instant equal to a certain τ_d, depending on many factors involving the subject but roughly of the order of magnitude of $0{,}1 \div 0{,}2\ s$, the body responds as a non-controlled one. This is because the nervous and muscular systems take time to process, elaborate the appropriate reaction signal and intervene on the balance perturbation in a proper way. [9]

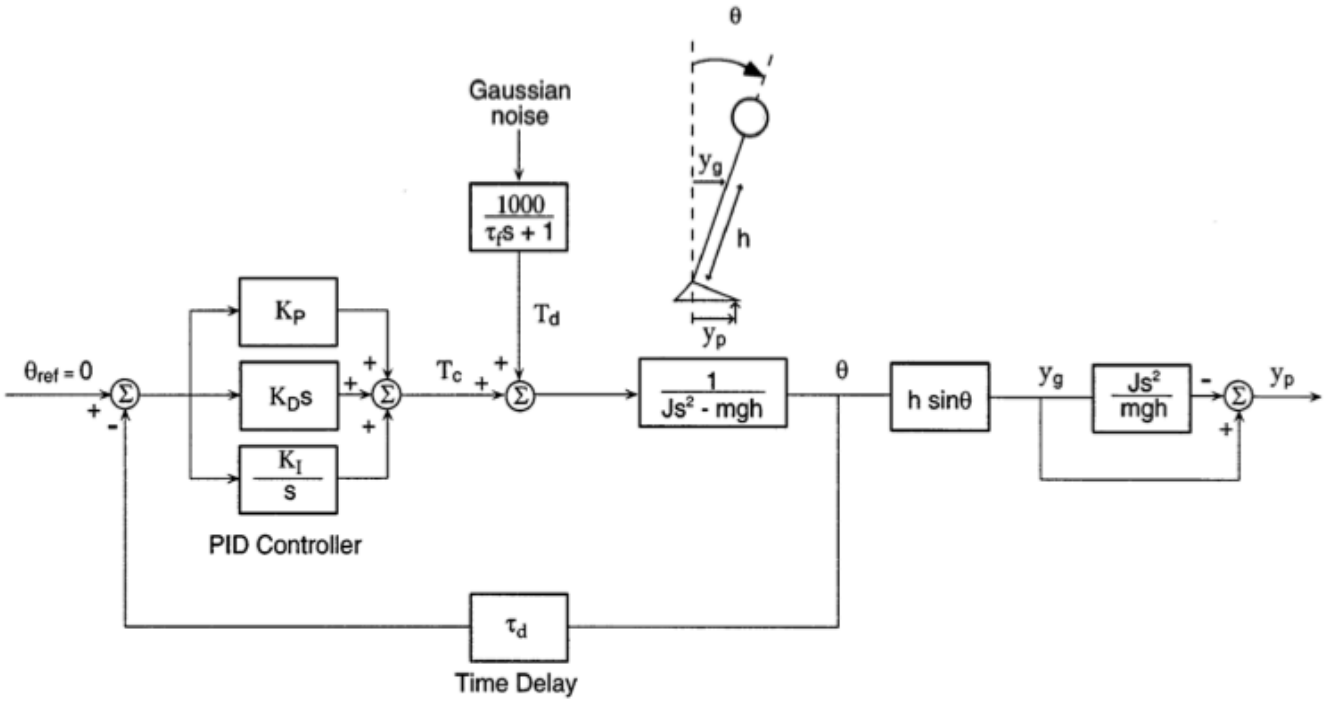

Figure 2.2: Delayed PID model describing the control of human body upright stance

Due to the property mentioned above, since the target strike time duration is below 250 ms if we're referring to the same testing parameters of the third prototype described in Paragraph 1.5, we can operate under the assumption that the subject's body reacts to the blow as a non-controlled object. We can also assume that since the strike phase works with little overall displacements, the angular displacement $\theta \cong 0$, and the rotational problem can then be transposed in a linear one via Taylor's Series truncated at the first order. These hypotheses introduce a beneficial simplification to our model, that can now be represented by a standard linear mass-spring-damper system, at the expense of introducing some error inside the plant model that the robustness of the control algorithm is able to damp out.

During other time intervals than the strike phase, the subject's behavior can be neglected since there's no physical body-machine interaction. Thus the model of the further postural response is omitted since it's not the focus of this study. Furthermore, the approach and retraction phases' speed can be directly imposed on the specific actuator using the driver (see Paragraphs 0 and 3.3): this, added to the fact that their dynamics are not of significative importance, permits to neglect plant (i.e., prototype's dynamics) modeling for these phases.

2.3 The NiLAB GmbH GD160Q linear motor

The linear actuator chosen for the system is the *NiLAB GmbH* Green Drive series Tubular synchronous linear motor 160Q, i.e., having a shaft diameter of 160/10 = 16 mm and Q = 4 windings in the primary. The standard GD160 class motor is shown in Figure 2.3, while the reasons for choosing a linear motor are illustrated in Paragraph 2.1. For short time duties as in the prototype's case, the motor shows a peak force of 105,94 N and a maximum velocity of 6,82 m/s, which are suitable parameters for the system design. The stator and chosen slider longitudinal dimensions are, respectively, of 219 mm and 377 mm. This allows enough space for having a constant slider's velocity.

Figure 2.3: GD160 class linear motor

As Figure 2.3 shows, the linear motor used is double-sided. This means that the chassis of the prototype must consider additional space for the retracted position without safety problems for the final user in the handled configuration.

Since the shaft end hole is an M6x8, the actuator is now compatible with the load cell (see Paragraph 2.5.1) and doesn't need a screw adapter as in the previous prototypes mentioned in Paragraphs 1.4 and 1.5, resulting in a more compact design.

Thanks to the presence of t-slots along three sides of the stator, the system's mounting and overall complexity are improved with respect to the said previous prototypes, which needed the use of angle brackets to be mounted on the chassis. In fact, only three t-nuts are used for chassis mounting, while an additional one is used for the optical sensor, as described in Paragraph 5.2.4.

The actuator contains an encoder sin/cos, whose resolution is of 1 sine period per polar pitch (= 60 mm).

The GD160Q key parameters are reported in Table 2.1.

Parameter name			Value	Unit
Mechanical	Stator	mass	1,75	kg
		length	219	mm
	Slider	mass	0,53	kg
		length	377	mm
		diameter	16	mm
	Peak force (service type S3 and duty cycle=5%)		105,94	N
	Continuous stall force (passive cooling)		23,7	N
	Maximum velocity		6,82	m/s
	Number of poles		2	
	Polar pitch (N to N)		60	mm
	Virtual radius (see Paragraph 3.2)		10	mm
Electrical	Servo Drive Power Supply		325,22	V
	Peak Current (duty cycle=5%)		2,48085	Apk
	Continuous stall current (passive cooling)		0,555	Arms
	Force constant		42,83	N/Arms
	Back EMF constant (phase-phase)		34,97	Vpk/(m/s)
			24,73	Vrms/(m/s)
	Resistance (phase-phase)	At 25°C	58	Ω
		At 135°C	83,0734	Ω
	Inductance (phase-phase)		33,46	mH
	Electrical time constant		0,58	ms
	Motor constant		3,83	N/W^(1/2)
Thermal	Maximum winding temperature		135	°C
	Maximum peak current duration		1	s
	Maximum power dissipation		38,46	W
	Thermal resistance	Case-environment	1,210	°C/W
		Winding-Case	1,680	°C/W
		Winding-environment	2,890	°C/W
	Thermal time constant		1424	s

Table 2.1: GD160Q linear motor key parameters

2.4 The strike interface

The strike interface is covered by a 20 mm layer of expanded polyethylene with negligible mass, 10 N*s/m damping, and 10 kN/m stiffness. The polyethylene interface is fixed to an aluminum disk with a central threaded hole.

2.5 The sensory devices

2.5.1 *The Dacell UMM load cell*

The *Dacell* UMM load cell, shown in Figure 2.4, is the sensor used to measure the contact force during the strike phase to generate a feedback signal for the controller algorithm. The sensor is mounted directly between the interface and the actuator's slider, thus reporting compressive loads. The sensor's key parameters are reported in Table 2.2.

Figure 2.4: The Dacell UMM load cell

Parameter name			Value	Unit
Performance	Rated capacity (R.C.)		50	kgf
	Rated output (R.O.)		0,7÷1,2	mV/V
	Safe overload		120	% of R.C.
	Non linearity		0,1	% of R.O.
	Hysteresis		0,1	% of R.O.
	Repeatability		0,1	% of R.O.
	Zero balance		±2	% of R.O.
Electrical	Recommended excitation		5	V
	Input resistance		350±30	Ω
	Output resistance		350±2	Ω
Thermal	Temperature range	Compensated	-10÷60	°C
		Safe	-10÷80	°C
	Temperature effect	On R.O.	0,1	% of load/10°C
		On zero balance	0,05	% of R.O./10°C

Table 2.2: Dacell UMM load cell key parameters

The load cell sensor connects with the *Dewetron* DEWE-RACK-4 described in Paragraph 5.3.2.

2.5.2 *The BANNER Q4XTULAF300-Q8 optical sensor*

The *BANNER* Q4XTULAF300-Q8 laser sensor, shown in Figure 2.5, is the optical sensor chosen for the prototype implementation. The sensor is a visible red class 1 laser of 655 nm, and is protected against reverse polarity and transient overvoltage. The optical sensor's key parameters are reported in Table 2.3.

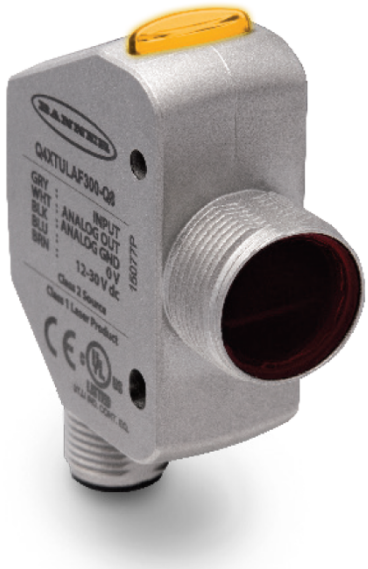

Figure 2.5: Q4X Series optical sensor

Parameter name			Value	Unit
Performance	Working range		25÷300	mm
	Accuracy	25 to 100 mm range	<0,3	mm
		100 to 300 mm range	<1	mm
	Safe overload		120	% of R.C.
Electrical	Supply voltage		12÷30	V
	Output voltage		0÷10	V
	Power consumption		<675	mW
Thermal	Temperature range		-10÷50	°C

Table 2.3: BANNER Q4XTULAF300-Q8 optical sensor key parameters

The optical sensor also implements a 2-point teach feature. In this way, it automatically offsets and rescales the signal by associating the fully retracted rod position to 0 V output signal, the fully extended rod position to 10 V and saturating the out of range positions. In this way:

$$Rod\ displacement\ [mm] = feedback\ signal * \frac{range}{10}$$

2.6 The Parker SLVD1N driver

2.6.1 *Introduction*

The *Parker's Small Low Voltage Drive – New* (SLVD-N) Series, shown in Figure 2.6, is the frequency converter used as the link between the *Speedgoat* controller, the actuator and its resolver. The size of the converter, i.e., the nominal output current the driver can supply, is the lowest of the series available, the SLVD1N, which can provide 1,25 Arms.

Figure 2.6: The SLVD-N Series driver

The converter can work with and automatically recognizes both mono-phase and tree-phase alimentations coming from the power grid, using them for energizing the electric motor. As for the logic circuit, the alimentation needs to be of 24 Vdc. All the alimentation electrical cables are protected by proper fuses, prescribed in the driver's datasheet.

The converter is provided of all the relevant parameters of the linear motor, to perform a control action on the rod's speed, position, or produced force using internal or external, digital or analogic reference signals. First run setup and speed control calibration were performed by the supplying company and are not discussed on this thesis.

The SLVD1N driver works with two types of data entities:

- Parameters (PrX), chosen a priori, describing the actuator's relevant quantities, the control strategy adopted, the analogic scaling values, filter definition, threshold values, etc.;
- Bits (bX.X), that can be used for pico-PLC design, signal monitoring, error reporting, and encoder signal processing.

Parameters and bits can be modified either manually, through the display's buttons, or through the Parker's *MotionWiz* Software (Paragraph 2.6.3), specifically designed for quick design setups and pico-PLC implementation (Paragraph 2.6.4).

2.6.2 *Speed and force regulation operating mode*

The operating mode adopted for the design is the speed regulation one (Pr31=0), shown in Figure 2.7. The mode uses an analog external speed reference (b40.2=0) coming from the *Speedgoat* machine, calculating the input according to its amplitude (b40.12=0) or frequency (b40.12=1).

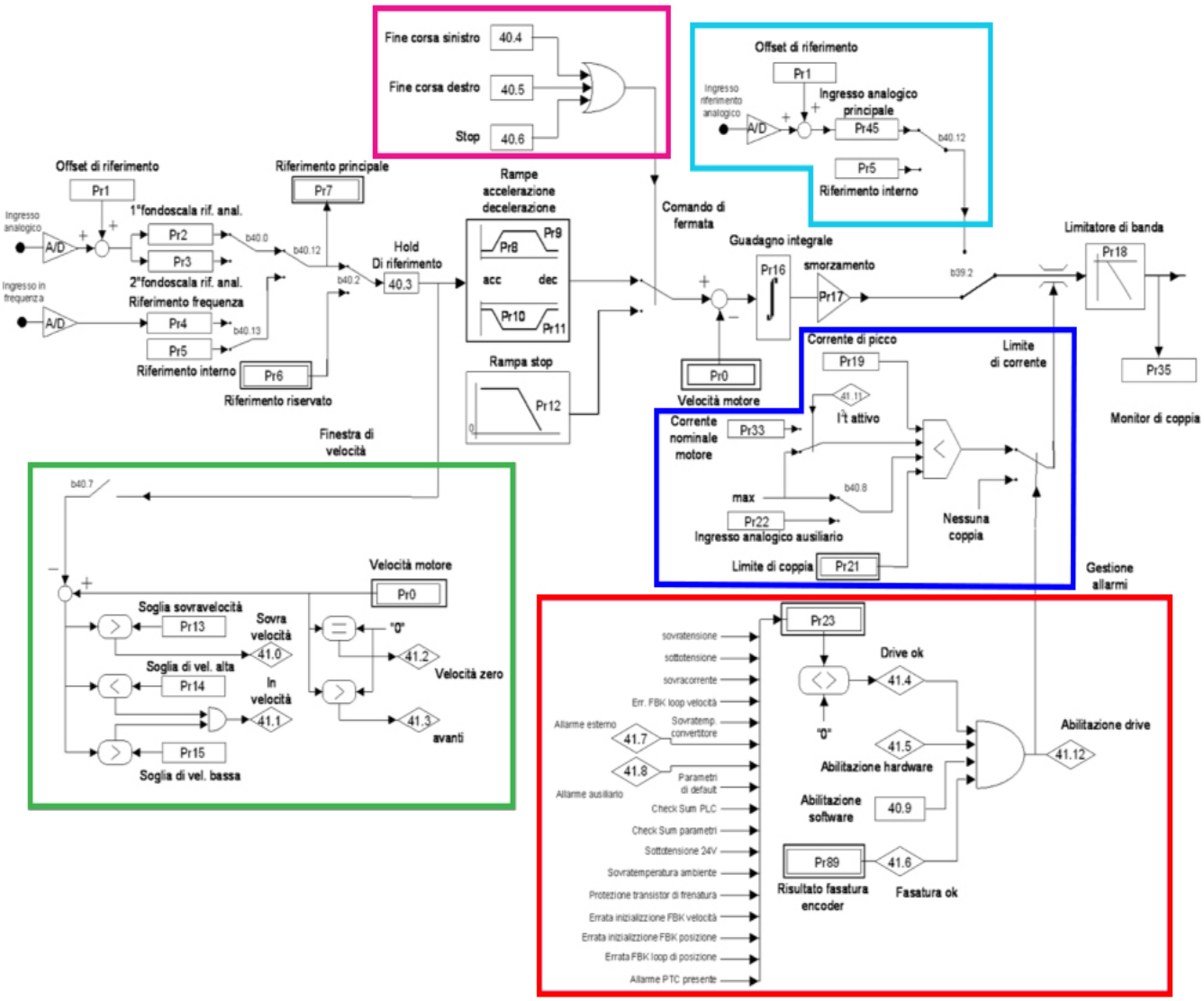

Figure 2.7: Speed regulation SLVD1N operating mode

After the signal is acquired and stored in Pr7, the mode restrains it following acceleration and deceleration ramps whose trapezoidal profile is modulated by parameters Pr8, Pr9, Pr10, and Pr11. The mode uses a Proportional-Integral (PI) controller, (integral gain k_i=Pr16 and proportional gain k_p=Pr17), on the speed reference error, calculated using the encoder signal. The requested output current, i.e., force, is object to threshold and filtering limiters.

As Figure 2.7 shows, the converter's mode also implements:

- A speed monitoring system (in green), in which bits 41.0, 41.1, 41.2 and 41.3 report, respectively, overspeed, in speed, zero speed, forward speed conditions on the difference between the actual and reference velocities of the linear motor;

- A stop system (in purple), that turns the speed reference processed signal into a deceleration ramp regulated by Pr12 if the logic sum (OR) of the following bits is 1:
  - Left end of stroke bit b40.4;
  - Right end of stroke bit b40.5;
  - Stop function bit b40.6;

- A current limiting system (in blue), that limits the output current of the converter to the minimum value of the following:
  - Peak current (Pr19), set a priori, normally always under three times the motor's nominal current;
  - The motor's nominal current (Pr33), set a priori, only present in the said minimum comparation if the motor is overheating (b41.11=1). The motor thermal image is estimated through i2t calculations and monitored by Pr36, which is the percentage value of the temperature of the motor's innermost windings with respect to the nominal temperature. Once Pr36=100%, b41.11 switches to 1;
  - A possible signal analogic force limit signal, only present in the minimum comparation if the bit b40.8=1;
  - Force limit (Pr21), a reserved and read only parameter;

- Drive habilitation (in red), monitored by the status of bit b41.12, which is the logic product (AND) of the following bits:
  - "Drive ok" bit b41.4, monitoring the presence of any software or hardware error inside the system;
  - Hardware habilitation bit b41.5;
  - Software habilitation bit b40.9;
  - Proper encoder phasing bit b41.6;

  If any of the said bits is null, b41.12=0, imposing a null current limit on the current limiting system. Therefore, the motor cannot run.

Inside the speed regulation operating mode, by switching the bit b42.2 from 0 to 1, the converter turns in current mode (i.e., force regulation) (in light blue), excluding the closed loop on the speed and assuming the analog reference as current (i.e., force) reference. If the nominal temperature threshold is reached, the motor's nominal current is set as the current's limit as in the speed regulation mode.

The speed and force regulation operating modes contain an input signal scaling and offset performed by parameters Pr1, Pr2, Pr3, Pr4, and Pr5.

Table 2.4 shows all the relevant parameters and bits employed in the driver. If not otherwise specified, all the parameters can be read and written inside the converter.

PrX	Parameter name	Description	Unit
0	Motor's speed (read only)	Calculated by the encoder.	rpm
1	Analog reference offset	Its unit value depends on b42.2.	rpm or Nm
2	Analog speed reference gain	If b40.0=0 and b40.12=0, Pr7=Pr2*Vin/9.76	rpm/V
7	Main speed reference (read only)	The desired speed of the driver.	rpm
8	Positive speed acceleration ramp	Time necessary for a speed positive modification of 1000 rpm, if the speed is positive.	s
9	Positive speed deceleration ramp	Time necessary for a speed negative modification of 1000 rpm, if the speed is positive.	s
10	Negative speed acceleration ramp	Time necessary for a speed positive modification of 1000 rpm, if the speed is negative.	s
11	Negative speed deceleration ramp	Time necessary for a speed negative modification of 1000 rpm, if the speed is negative.	s
12	Stop function deceleration ramp	Time necessary for a speed absolute modification of 1000 rpm towards null speed.	s
13	Overspeed limit	Absolute value threshold for overspeed checks.	rpm
14	High speed limit	Absolute value upper threshold for in speed checks.	rpm
15	Low speed limit	Absolute value lower threshold for in speed checks.	rpm
16	Integral gain of the speed regulator	Used in the driver's embedded PI.	

17	Damping of the speed regulator	Used in the driver's embedded PI.	
18	Band width limiter	Used to set the cut-off frequency of the first order filter used on the digital signal of the torque request. f=1240/Pr18 Hz.	
19	Peak current	Maximum current providable by the driver expressed as percentage of the peak current of the converter.	0.1% Ipeak
21	Torque limiter (read only)	For the current limiting system block.	%Rated torque
23	Alarm code	Reports all the possible software and hardware errors inside the driver.	
31	Operating mode	Value 0 is the speed and force regulating mode shown in Figure 2.7.	
33	Rated current of the motor	Rated current to which the motor must be set.	A
35	Torque monitor (read only)	Indicates the percentage of the torque that the motor is supplying.	% Torque peak

Table 2.4: SLVD1N driver's key parameters

Note that, due to the SLVD1N general purpose, all the parameters involving the analog input are in rpm and in Nm. This problem is solved in Paragraph 3.2.

2.6.3 *The MotionWiz software*

All the *Parker* drivers interface with the *MotionWiz* software, whose objective is the fast insertion, modification and monitoring of all the driver's parameters, bits and possible alarm sources without the need of the device's embedded keypad. The software also implements the possibility of intervening on the pico-PLC in plain text format, and an oscilloscope function to check the input and output signals.

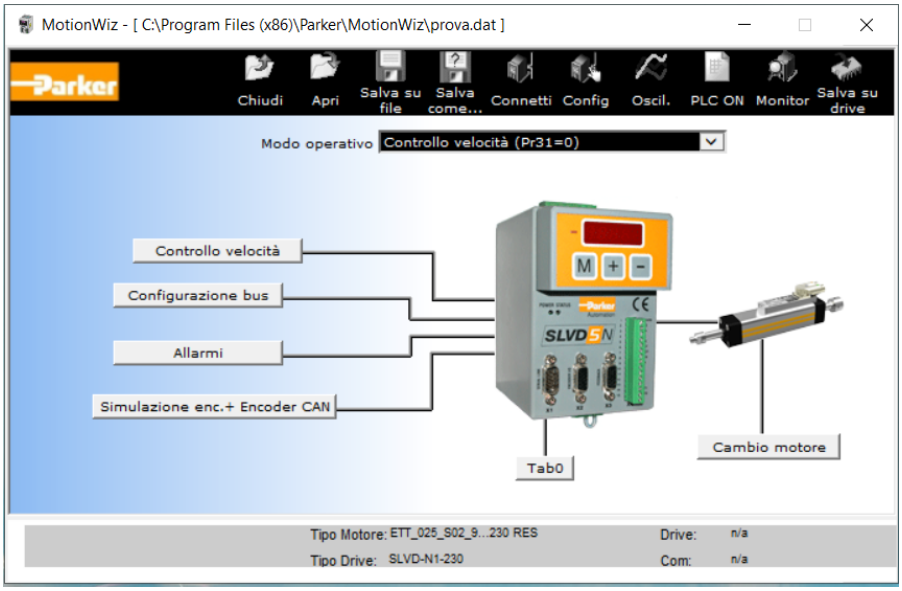

Figure 2.8: MotionWiz software home page

In order to properly communicate with the system, the device needs a USB to serial RS485/RS422 cable, plus a 9 to 15 pin serial adaptor cable. The latter was soldered in order to carry out the pinout in Figure 2.9.

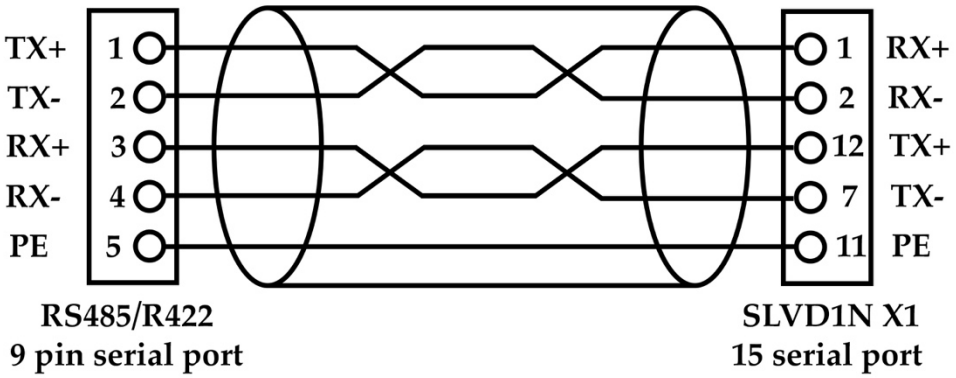

Figure 2.9: Serial to SLVD1N 9 to 15 pin adaptor cable pinout scheme

2.6.4 *The pico-PLC*

The pico-PLC inside the SLVD1N machine is the device connecting the outward world with the software inside the converter. The driver is indeed provided of 4 digital input pins, whose properties are shown in Table 2.5.

Parameter	Value	Unit
Input resistance	20±1	kΩ
Input Low voltage range	0÷5	V
Input High voltage range	10÷24	V
Reaction time	<2,5	μs
Type of driving required	PNP	

Table 2.5: SLVD1N's digital inputs properties

The PLC is able to convert the digital signal in a binary parameter and vice versa, and to solve simple mathematical and Boolean operations. The bits corresponding to the 4 digital inputs are b90.0, b90.1, b90.2, and b90.3. The structure of the PLC can either be modified manually, through the embedded keypad, or digitally, through the *MotionWiz* software, shown in Figure 2.10.

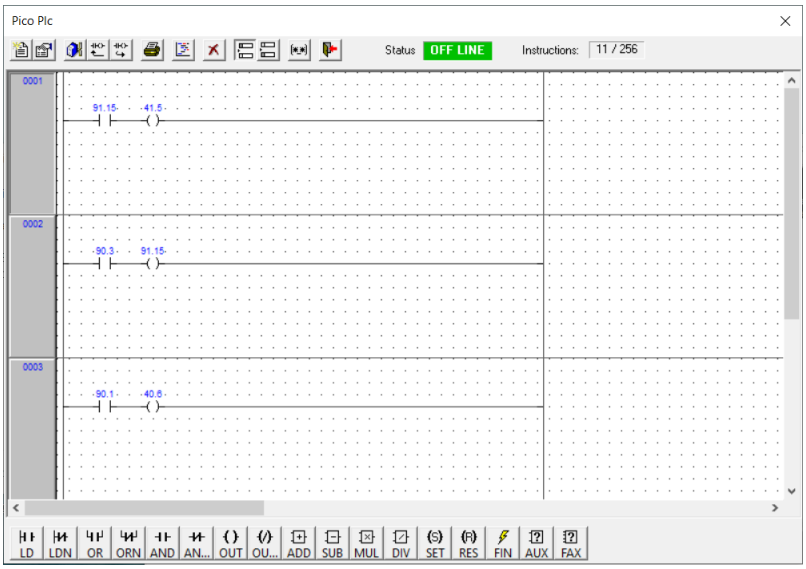

Figure 2.10: The pico-PLC in the MotionWiz environment

The scan period of the PLC is 6,144 ms. During this time, the converter reads the inputs, updates its two timers, runs the user's program, and updates the outputs. All the PLC instructions, with the exception of the arithmetical ones, operate on the single bit.

2.6.5 *The Schaffner FN 2010-20-06 filter for the driver*

As prescribed by the datasheet, the SLVD1N driver requires an electromagnetic interference (EMI) filter, in order not to inject noise in the power grid. The filter must be connected as close as possible the common ground and have a distance not greater than 30 cm from the converter. The choice fell on a Schaffner FN 2010-20-06 EMI filter, shown in Figure 2.11, whose key parameters are listed in Table 2.6.

Figure 2.11: The Schaffner FN 2010 filter

Parameter name	**Value**	**Unit**
Rated voltage	250	Vrms @ 50/60 Hz
	250	Vdc
Operating frequency	0÷400	Hz
Rated current	20 (23)	A @ 40°C (25°C)
Leakage current	0,66	mA
Power loss	3,8	W
Inductance	0,6 -30/+50%	mH
Input resistance	1000	kΩ
Input capacitance	0,1	µF
Output capacitance	4,7	nF
Temperature range	-25÷100	°C

Table 2.6: Schaffner FN 2010-20-06 key parameters

2.7 The Speedgoat real-time target machine

Due to its high flexibility and user-friendliness, a *Speedgoat* real-time target machine was chosen in the design's simulation and Hardware-in-the-loop (HIL) testing phases of the prototype. The *Speedgoat* machine, shown in Figure 2.12, contains 2 FPGA IO183 and 1 IO397 modules, whose operative parameters are shown in Table 2.7.

Figure 2.12: The Speedgoat machine

Parameter		IO183	IO397
Sampling frequency		1 kHz	1 kHz
Analog input channel	Ports number	4	4
	Bit width	16-bit	16-bit
	Voltage range	±10 V	0÷10,24 V
Analog output channel	Ports number	4	4
	Bit width	16-bit	16-bit
	Voltage range	0÷5 V	±10,8 V

Table 2.7: FPGA relevant parameters

One of the *Speedgoat* machine's main advantages is that it does not require the user any code generation since the device directly reads and uploads the *Simulink* code of the specific design and efficiently interfaces with the PC environment in terms of data acquisition and analysis. Another important feature of the *Speedgoat* machine is its expandability. The device can support more CPUs and Simulink programmable FPGAs without any loss of performance if the system requested it.

The *Speedgoat* machine's data acquisition system used in the analysis is described in Paragraph 5.3.1.

2.8 Nominal plant model

2.8.1 *Introduction*

With all the above hypotheses and data, we can design a nominal plant model for the striking phase, as in Figure 2.13.

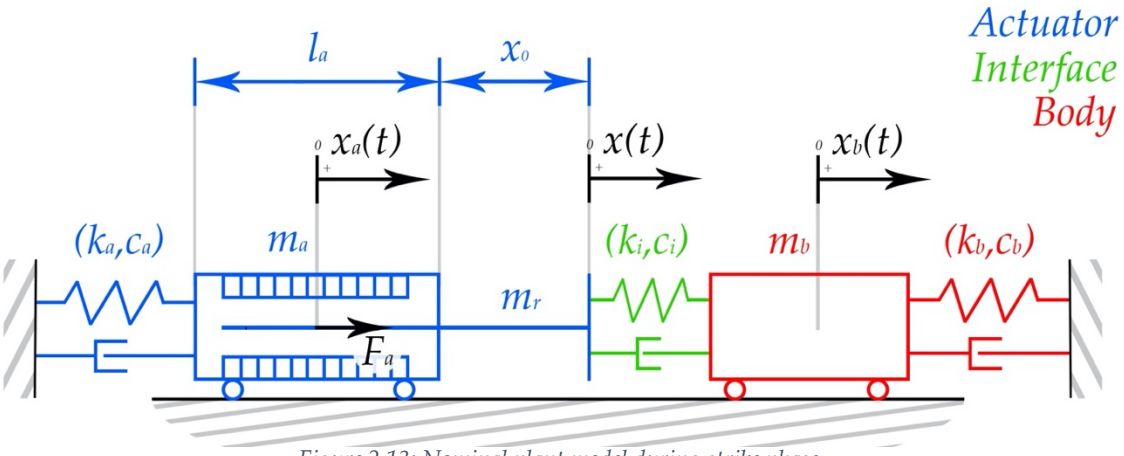

Figure 2.13: Nominal plant model during strike phase

Parameter/Variable		Symbol	Value	Unit
Actuator	Stator mass	m_a	1,75	kg
	Stiffness	k_a	30000	N/m
	Damping	c_a	1000	N*s/m
	Longitudinal dimension	l_a	219	mm
	Displacement	$x_a(t)$		m
	Rod length	l_r	408	mm
	Rod mass	m_r	0,57	kg
	Rod first contact extension	x_0	100	mm
	Rod first contact velocity	v_0	0,08	m/s
	Rod displacement after first contact	$x(t)$		m
	Produced force	F_a		N
Interface	Stiffness	k_i	10000	N/m
	Damping	c_i	10	N*s/m
Body	Mass	m_b	70	kg
	Stiffness	k_b	30000	N/m
	Damping	c_b	1000	N*s/m
	Displacement	$x_b(t)$		m

Table 2.8: Nominal plant model during strike phase key parameters and variables

As the Figure 2.13 and Table 2.8 show, the plant model is composed of three elements: the actuator's system, the (dimensionless) body's system, and the (massless) interface between the two. This model can work with both handled and mounted configuration by merely modifying the actuator's parameters (e.g., posing a very high valued stiffness representing the actuator's interlocking in mounted configuration), and for bench testing.

2.8.2 *Plant state space model through Lagrange approach*

To achieve the State-Space (SS) representation matrices to insert in the Simulink environment, Lagrange energetic approach is adopted. We have that:

$q \stackrel{\text{def}}{\equiv} \{q_i\}_{i=1,2,3} = \{x_a(t), x(t), x_b(t)\}^t$	Degrees of freedom (dofs)
$\mathcal{K}(q,t) = \frac{1}{2}m_a\dot{x}_a^2(t) + \frac{1}{2}m_r\dot{x}^2(t) + \frac{1}{2}m_b\dot{x}_b^2(t)$	Kinetic Energy
$\mathcal{U} = \frac{1}{2}k_a x_a^2 + \frac{1}{2}k_i\big(x_b(t) - x(t)\big)^2 + \frac{1}{2}k_b x_b^2$	Potential Energy
$\mathcal{D} = \frac{1}{2}c_a\dot{x}_a^2(t) + \frac{1}{2}c_i\big(\dot{x}_b(t) - \dot{x}(t)\big)^2 + \frac{1}{2}c_b\dot{x}_b^2$	Damping function
$\delta L = F_a(t)\big(\delta x(t) - \delta x_a(t)\big)$	Virtual Work of non-conservative forces
$\mathcal{L} = \mathcal{K} - \mathcal{U}, \qquad \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} + \frac{\partial \mathcal{D}}{\partial \dot{q}_i} = \frac{d\delta L}{d\delta q_i}$	Lagrange approach

Arriving at, omitting the dofs dependency on the time t, the dof equations:

$$\begin{cases} x_a)\ m_a\ddot{x}_a + c_a\dot{x}_a + k_a x_a = -F_a \\ x)\ m_r\ddot{x} + c_i\dot{x} - c_i\dot{x}_b + k_i x - k_i x_b = F_a \\ x_b)\ m_b\ddot{x}_b - c_i\dot{x} + (c_i + c_b)\dot{x}_b - k_i x + (k_i + k_b)x_b = 0 \end{cases}$$

$$\begin{bmatrix} m_a & 0 & 0 \\ 0 & m_r & 0 \\ 0 & 0 & m_b \end{bmatrix}\ddot{q} + \begin{bmatrix} c_a & 0 & 0 \\ 0 & c_i & -c_i \\ 0 & -c_i & c_i + c_b \end{bmatrix}\dot{q} + \begin{bmatrix} k_a & 0 & 0 \\ 0 & k_i & -k_i \\ 0 & -k_i & k_i + k_b \end{bmatrix} q = \begin{Bmatrix} -F_a \\ F_a \\ 0 \end{Bmatrix}$$

$$\stackrel{\text{def}}{\equiv}$$

$$[M]\ddot{q} + [C]\dot{q} + [K]q = \{F\}$$

Which, applying the reduction to first order technique, becomes:

$$v = \dot{q}, \qquad z = \begin{Bmatrix} v \\ q \end{Bmatrix} = \begin{Bmatrix} \dot{x}_a \\ \dot{x} \\ \dot{x}_b \\ x_a \\ x \\ x_b \end{Bmatrix}, \quad \begin{cases} [M]\dot{v} + [C]v + [K]q = \{F(t)\} \stackrel{\text{def}}{\equiv} \{T\}u(t) \\ \dot{q} = v \end{cases}$$

In which $\{T\} = [-1\ 1\ 0]^t$ is the selection vector.

This allows:

$$\dot{z}(t) = \begin{bmatrix} -[M]^{-1}[C] & -[M]^{-1}[K] \\ I_{3x3} & O_{3x3} \end{bmatrix} z(t) + \begin{bmatrix} [M]^{-1}\{T\} \\ 0_{3x1} \end{bmatrix} u(t)$$

$$\stackrel{\text{def}}{=}$$

$$\dot{z} = [A]z + [B]u(t)$$

In which [A] and [B] are, respectively, the dynamic and the input gain matrices of the system. We thus have computed the differential part (first matrices' couple) of the SS system.

As for the initial conditions, we have that:

$$z_0 \stackrel{\text{def}}{=} z(t=0) = \begin{Bmatrix} \dot{x}_a(0) \\ \dot{x}(0) \\ \dot{x}_b(0) \\ x_a(0) \\ x(0) \\ x_b(0) \end{Bmatrix} \cong \begin{Bmatrix} 0 \\ v_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

With x_0 and v_0 the first contact rod's displacement and speed respectively, considering null all the other entries of the initial state vector for simplicity.

2.8.3 *Plant output equations*

To achieve the full quadruplet of the form, and hence derive the algebraic part, we have to define the feedback system. Using the same sensors as in the third prototype (Paragraph 1.5), we can identify the feedback quantities as the rod displacement *x(t)*, detected by the laser sensor, and the interface force $F_i(t)$, detected by the load cell, as highlighted in Figure 2.14.

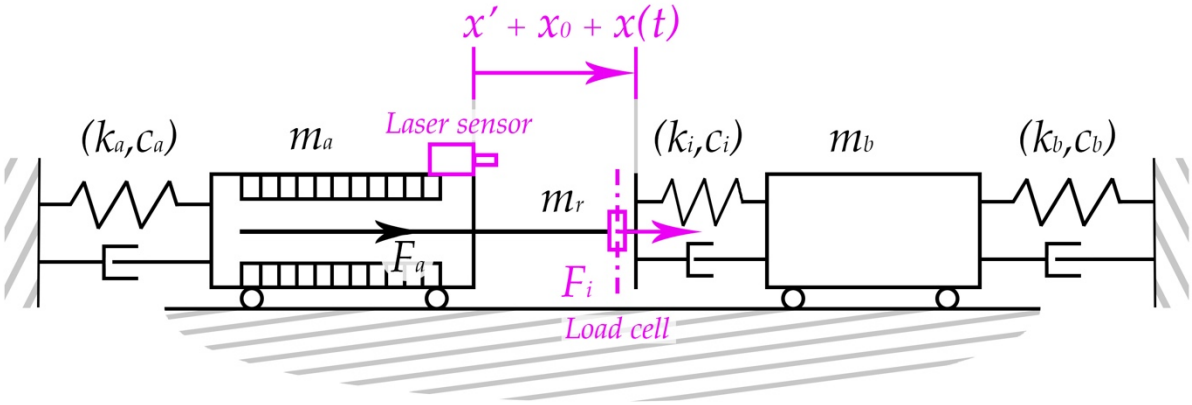

Figure 2.14: Plant sensory system

As shown, the laser sensor also reports the rod first contact extension x_0 and an additional length x' due to its mounting alongside the actuator's case, as described in Paragraph 5.2.4. Thanks to the BANNER teach function described in Paragraph 2.5.2, x_0 is cancelled out by the sensor's scaling.

We can consequently define the output equations as:

$$y(t) = \begin{Bmatrix} F_i(t) \\ x(t) \end{Bmatrix} = \begin{Bmatrix} k_i(x_b - x) + c_i(\dot{x}_b - \dot{x}) \\ x \end{Bmatrix} = \begin{Bmatrix} k_i(z_6 - z_5) + c_i(z_3 - z_2) \\ z_5 \end{Bmatrix}$$

$$= \begin{bmatrix} 0 & -c_i & c_i & 0 & -k_i & k_i \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} z + [0_{2x1}]u$$

$$\stackrel{\text{def}}{=}$$

$$y(t) = [C]z + [D]u$$

In which [C] and [D] are, respectively, the output gain and the direct link matrices of the system. Due to the fact that the laser sensor could be removed from the system in next prototypes, the adopted control design uses only the first row of C.

The nominal plant equations can then be written in the SS representation as a linear time-invariant (LTI) model:

$$\begin{cases} \dot{z}(t) = [A]z(t) + [B]u(t) \\ y(t) = [C]z(t) + [D]u(t) \end{cases} \quad in\ which\ z_0 = \begin{Bmatrix} 0 \\ v_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Chapter 3: Control strategies

3.1 Control logic

As in the previous prototypes described in Paragraphs 1.4 and 1.5, the control logic bases on a finite-state machine (FSM) concept and is arranged in the following four phases, as in Figure 3.1 and 3.2:

- Idle phase, default state in which the piston is fully retracted and still;
- Approach phase, in which the piston has to follow a positive speed reference signal as described in Paragraph 3.2. The phase stays on until the target or the upper end of stroke are reached;
- Strike phase, ensuing only if the target is reached, in which the interface contact force piston/body has to follow to a constant force reference signal of amplitude and duration imposed by the initial callback function, as described in Paragraph 3.3;
- Retraction phase, ensuing immediately after the strike phase or if the upper end of stroke is reached, in which the piston has to follow a negative speed reference signal and return in a fully retracted position to idle phase through a lower end of stroke signal. It is described in Paragraph 3.2.

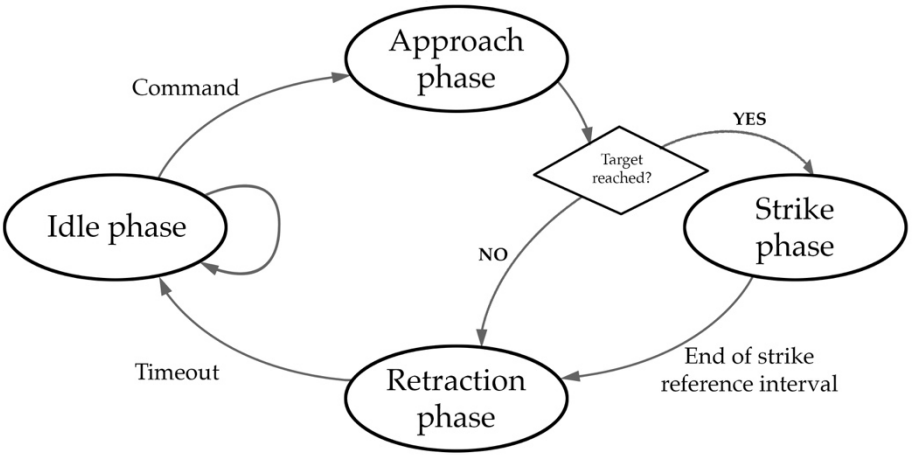

Figure 3.1: Control logic diagram

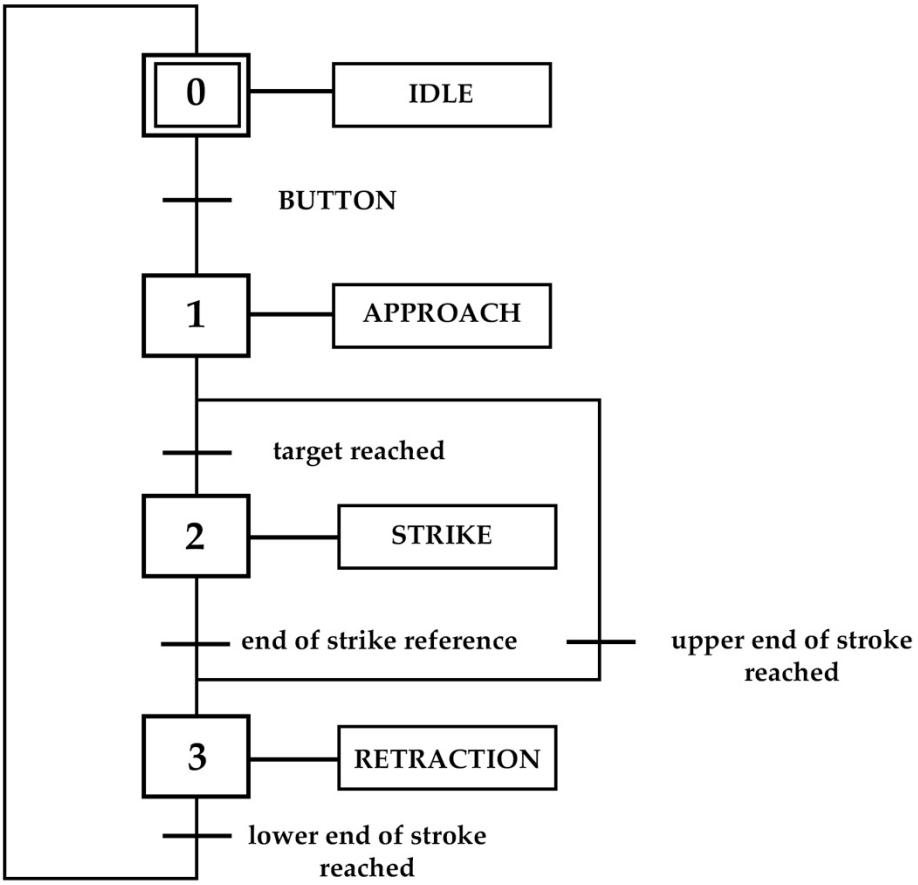

Figure 3.2: Descriptive GRAFCET

3.2 Idle, Approach and Retraction phases control

The adopted control strategy for Idle, Approach and Retraction phases is, as mentioned in Paragraph 2.6.2, a closed loop control with a PI controller directly performed inside the SLVD1N converter. Feedback signal is supplied by the motor's resolver, while proportional and integral gains for the controller are set inside the converter. The chosen values for the controlling algorithm are reported in Table 3.1.

Parameter	Symbol	Value
Proportional gain	k_p	500
Integrative gain	k_i	25

Table 3.1: Approach and retraction control parameters

As for the reference signal to use, the main problem is that, as shown in Paragraph 2.6.2, the SLVD-N drivers work with rpm speed measure unit, while a linear speed unit is needed to control the actuator more fittingly. We know from the driver's datasheet that:

$$v_r \stackrel{\text{def}}{=} Pr7 = V_{in} * \frac{Pr2}{9.76}\,[rpm]\ (*)$$
$$= V_{in} * \frac{Pr2}{9.76} * \frac{2\pi R}{60}[m/s]$$

with: $v_r = speed\ reference\ signal$
$V_{in} = driver's\ input\ voltage$
$Pr2 = reference\ gain\ \ [rpm/V]$
$R = actuator'virtual\ radius$

Since R is not provided in the actuator's datasheet, it is computed through the following formula:

$$(v_r)_{m/s} = (v_r)_{rpm} * \frac{2\pi R}{60} \Rightarrow R = \frac{30}{\pi} * \frac{(v_r)_{m/s}}{(v_r)_{rpm}}$$

To accurately measure the linear velocity, a speed test of the actuator's retraction is performed with the aid of the BANNER optical sensor described in Paragraph 2.5.2 in the test bench environment defined in Chapter 4.

Posing Vin=±1V and Pr2=50, the linear speeds is the slope of the actuator's displacement curve in Figure 3.3, while the angular speed is computed through (∗).

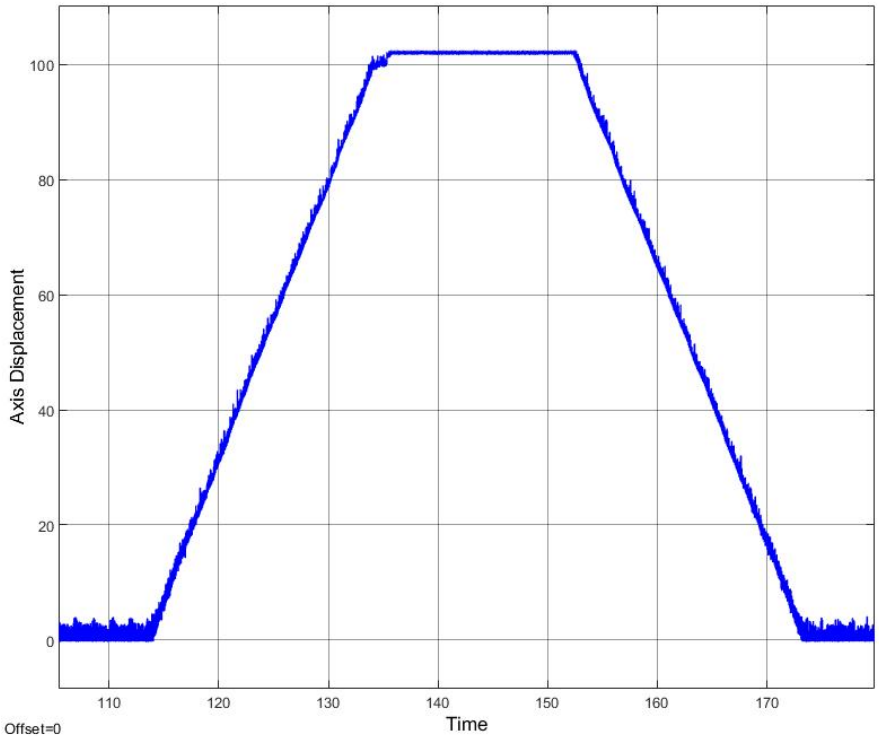

Figure 3.3: Virtual radius test, actuator's displacement curve

Thus:

$$\begin{cases} (v_r)_{m/s} \cong 0.005\ m/s\ (measured) \\ (v_r)_{rpm} = 5{,}122\ rpm\ (computed) \end{cases} \Rightarrow R = 0.01\ m$$

Therefore:

$$V_{in}|_{Pr2=50} = (v_r)_{m/s} * 186{,}5 = (v_r)_{cm/s} * 1{,}865$$

The value of the gain 1,865 V*s/cm is suitable for both the Speedgoat resolution and the operative conditions of approach and retraction. An absolute value of $v_r = v_0 = 0{,}08$ m/s is decided for both the phases. As for the idle mode, a null reference speed is adopted. In this way, the actuator's rod is forced to stay in the desired position during the idle phase by the energized phases inside the motor.

3.3 Strike phase control

3.3.1 *Introduction*

One of the main problems of the strike phase control, as already mentioned in Paragraph 1.4, is the highly nonlinear phenomena happening during the strike phase. Even if the electric linear actuator, as said in Paragraph 2.1, reduces them, the system nonlinearities still require to be damped out by the control algorithm, which refers to a linear system detailed in Paragraph 2.2. Another difficulty is related to the variability of the parameters involved, due to the fact that the prototype must work with different subjects and react accordingly. Furthermore, the GD160Q actuator, as shown in Table 2.1, is only able to reach 105,94 N peak force, i.e., a limited control input that could cause saturation problems especially when a 100 N force is required.

The control input limitation difficulty can be resolved mainly by adopting one of the following approaches:

- Cautious approach, relaxing performance requirements so the saturation constraints of the control input are met.
  In the PGAS case, due to the fact that the strike output signal requires to be as much similar to its reference for experimental data worthiness, this approach cannot be used;
- Serendipitous approach, designing a controller without taking into account the problem and thus allowing constraint violations.
  One of the drawbacks in using this approach is that, once the controller saturates, the control system works in saturation, possibly causing bad performance and, in the worst case scenario, instability, a situation to be avoided in the PGAS;
- Evolutionary approach, beginning with a serendipitous approach, then add a posteriori embellishments to mitigate the effects of saturation.
  In the PGAS case, this approach could cause problems related to other nonlinearities inside the system which are not taken into account;
- Tactical approach, taking into account the control input saturation constraint a priori, designing a controller able to properly act inside the saturation limits.

A tactical approach consists in, for example, making the control algorithm aware of the constraints by implementing the ability to predict the behavior of the system. This is the case of the Model Predictive Control (MPC), one of the most used algorithms in modern control design.

3.3.2 *The Model Predictive Control (MPC) working principles*

The MPC computes the controlled reaction of the plant of a finite number of steps in advance, i.e., its prediction horizon, through the Quadratic Program (QP) optimization algorithm. As a premise, in finite horizon optimization, to highlight the fact that everything is computed on the basis of the knowledge of the system in a certain *k*-th time, the standardized procedure is to adopt the following punctuation:

$$z(k+n)\ computed\ in\ time\ k, n\ \epsilon\ \mathbb{N}_0 \rightarrow\ z(k+n|k)$$

The QP bases its computations in minimizing the following cost function:

$$J\big(U(k|k)\big) = J_y\big(U(k|k)\big) + J_u\big(U(k|k)\big) + J_{\Delta u}\big(U(k|k)\big) + J_\varepsilon\big(U(k|k)\big)$$

In which:

- k current control prediction discrete time interval;
- n_y number of output variables;
- $n_u$number of control inputs, i.e., manipulated variables;
- H_p prediction horizon;
- $U(k|k) = \big\{u(k|k)\ u(k+1|k)\ \dots u\big(k+H_p-1\big|k\big)\ \ \varepsilon_k\big\}$, the QP decision;
    - $u(k+i|k)$ predicted control input of i steps forward at the k-th discrete time;
    - ε_k slack variable at the k-th discrete time, biasing soft constraint violation, that in certain cases could be unavoidable;
- $J_y\big(U(k|k)\big) = \sum_{i=0}^{H_p-1}\big[e_y^T(k+i|k)Qe_y(k+i|k)\big]$, the output reference tracking cost function;
    - $e_y(k+i|k) = r_y(k+i|k) - y(k+i|k)\ \epsilon\mathbb{R}^{n_y x\ 1}$ predicted output error of i steps forward at the k-th discrete time;
    - $Q\epsilon\mathbb{R}^{n_y x\ n_y}$ weight matrix, usually diagonal, which refers to the importance of the output error vector in the cost function;

- $J_u\big(U(k|k)\big) = \sum_{i=0}^{H_p-1}[e_u^T(k+i|k)R_u e_u(k+i|k)]$, the manipulated variable tracking cost function;

  - $e_u(k+i|k) = u_{target}(k+i|k) - u(k+i|k)\ \epsilon\mathbb{R}^{n_u x\ 1}$ predicted manipulated variables target error of *i* steps forward at the *k*-th discrete time;
  - $R_u\epsilon\mathbb{R}^{n_u x\ n_u}$ weight matrix, usually diagonal, which refers to the importance of the manipulate variables target error in the cost function;

- $J_{\Delta u}\big(U(k|k)\big) = \sum_{i=0}^{H_p-1}\big[\Delta u^T(k+i|k)R_{\Delta u}\Delta u\big(k+H_p\big|k\big)\big]$, manipulated variable move suppression cost function;

  - $\Delta u(k+i|k) = u(k+i|k) - u(k+i-1|k)\ \epsilon\mathbb{R}^{n_u x\ 1}$ predicted control input rate of *i* steps forward at the *k*-th discrete time;
  - $R_{\Delta u}\epsilon\mathbb{R}^{n_u x\ n_u}$ weight matrix, usually diagonal, which refers to the importance of the control input rate in the cost function;

- $J_\varepsilon\big(U(k|k)\big) = \rho_\varepsilon\varepsilon_k^2$, constraint violation cost function;

  - ε_k slack variable at the *k*-th discrete time, considered as object of optimization, biases soft constraint violation, that in certain cases could be unavoidable;
  - $\rho_\varepsilon\epsilon\mathbb{R}^+$ constraint violation penalty weight;

Therefore, the QP optimization algorithm does the following computation:

$$\begin{cases} \min\limits_{U(k|k)} J\big(U(k|k)\big) \\ U(k|k) = \big\{u(k|k)\ u(k+1|k)\ \dots u\big(k+H_p-1\big|k\big)\ \ \varepsilon_k\big\}\ s.t.\ \begin{cases} x(k+1) = Ax(k) + Bu(k) \\ G_u U(k|k) \le h_u(1+\ \varepsilon_k) \end{cases} \end{cases}$$

In which:

- Q, R_u, $R_{\Delta u}$, ρ_ε are design parameters. The higher the entries of a matrix with respect to the others, the more emphasis the system puts on the referred values' optimization;
- $G_u = \begin{bmatrix} I \\ -I \end{bmatrix}$, $h_u = \begin{Bmatrix} u_{max} \\ u_{max} \\ \dots \\ u_{min} \\ u_{min} \\ \dots \end{Bmatrix}$ input constraint structure system;

In the PGAS case, there are no soft constraints, and:

$n_y = 1; n_u = 1;\ u_{max} = -u_{min} \le f_{peak}$ saturation due to peak force.

Once the optimization has been done and $U(k|k)$, the MPC algorithm discards all its elements with the exception of the first entry and feeds it to the plant:

$$U(k|k)(1) = u(k|k) \rightarrow u(k)$$

This method, called receding horizon (RH) technique, closes the feedback loop in a fashion shown in Figure 3.4, which consider, as first approximation, the actuator's and sensors' transfer functions as unitary.

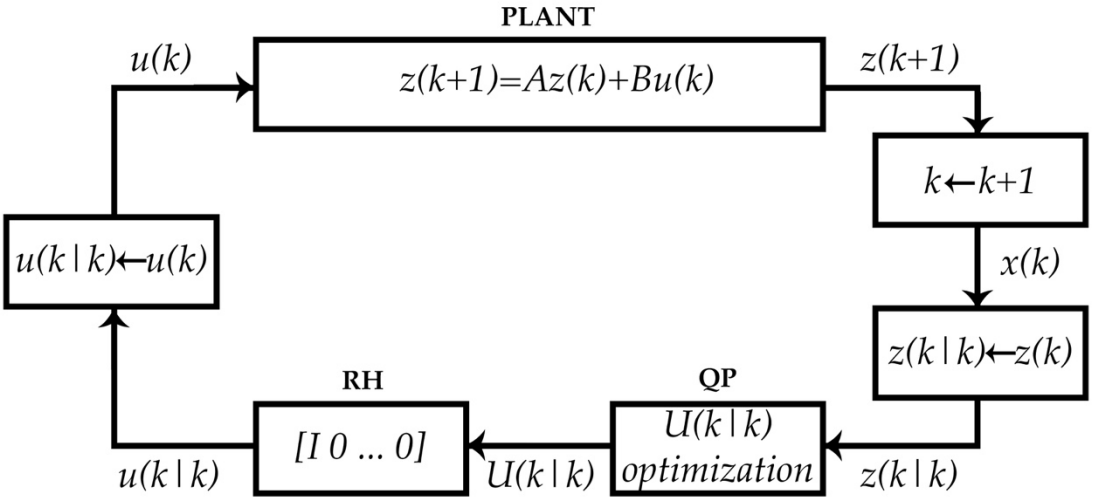

Figure 3.4: MPC algorithm loop

Due to the fact that the PGAS output system, described in Paragraph 2.8.3, does not measure all the states of the system, the MPC algorithm requires the implementation of state observers, the standard one being the Kalman filter. This consequently leads to an estimation error introduction caused by measurement noise and process disturbance.

To simplify the algorithm calculus, the prediction horizon H_p can be used in combination with the control horizon $H_c \leq H_p$. In this evolved procedure, the MPC algorithm predicts the system behavior until H_p steps forward, but optimizes $U(k|k)$ only for H_c steps. The remaining non optimized steps can be set, for example, in the following two ways:

$$\begin{cases} u(k+i|k) = u(k+Hc-1|k) \\ u(k+i|k) = -Kz(k+i|k) \end{cases} for\ i = Hc \ldots . Hp-1$$

In which K is a matrix computed by the LQ program through the weight functions and the plant system knowledge.

The algorithm could also work with a minimum prediction horizon H_w, not considering the first H_w steps due to delays, but in the PGAS case it is not required. [10]

3.3.3 *Model-in-the-loop testing for strike phase*

Model-in-the-loop (MIL) testing only focused on the Strike phase, due to the fact that Idle, Approach and Retraction phases controls are not of primary importance, as stated in Paragraph 2.2. The *Simulink* environment used in this phase, shown in Figure 3.5, only focused on the worst case reference, i.e., as described in Paragraph 6.2, the one with 100 N force amplitude and 50 ms impulse duration. Load cell noise was modelled to resemble the noise of the high pass filtered signal in the test bench, described in Paragraph 4.3, while nonlinear uncertainties were left unmodeled for simplicity.

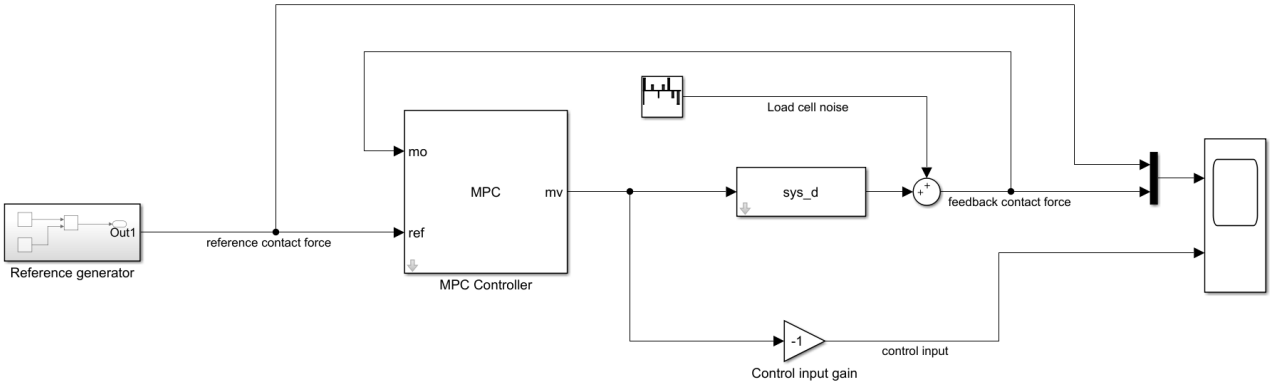

Figure 3.5: MIL testing for strike phase Simulink environment

Parameter		Symbol	Value	Unit
Prediction horizon		H_p	10	steps
Control horizon		H_c	2	steps
Control input constraint		$u_{max} = -u_{min}$	105	N
Weights	Manipulated variables	$R_u \epsilon \mathbb{R}^{1x1}$	0	
	Manipulated variables rate	$R_{\Delta u} \epsilon \mathbb{R}^{1x1}$	10	
	Output variables	$Q \epsilon \mathbb{R}^{1x1}$	9	

Table 3.2: MIL testing MPC parameters

Imposing the values in Table 2.8 and Table 3.2 and running the simulation, the contact force feedback signal profile and performance are shown in Figure 3.6 and Table 3.3. The model also considered that the force reference signal is issued once the force surpasses *LC_th*, a value imposed for the engineering reasons explained in Paragraphs 4.3 and 4.4.

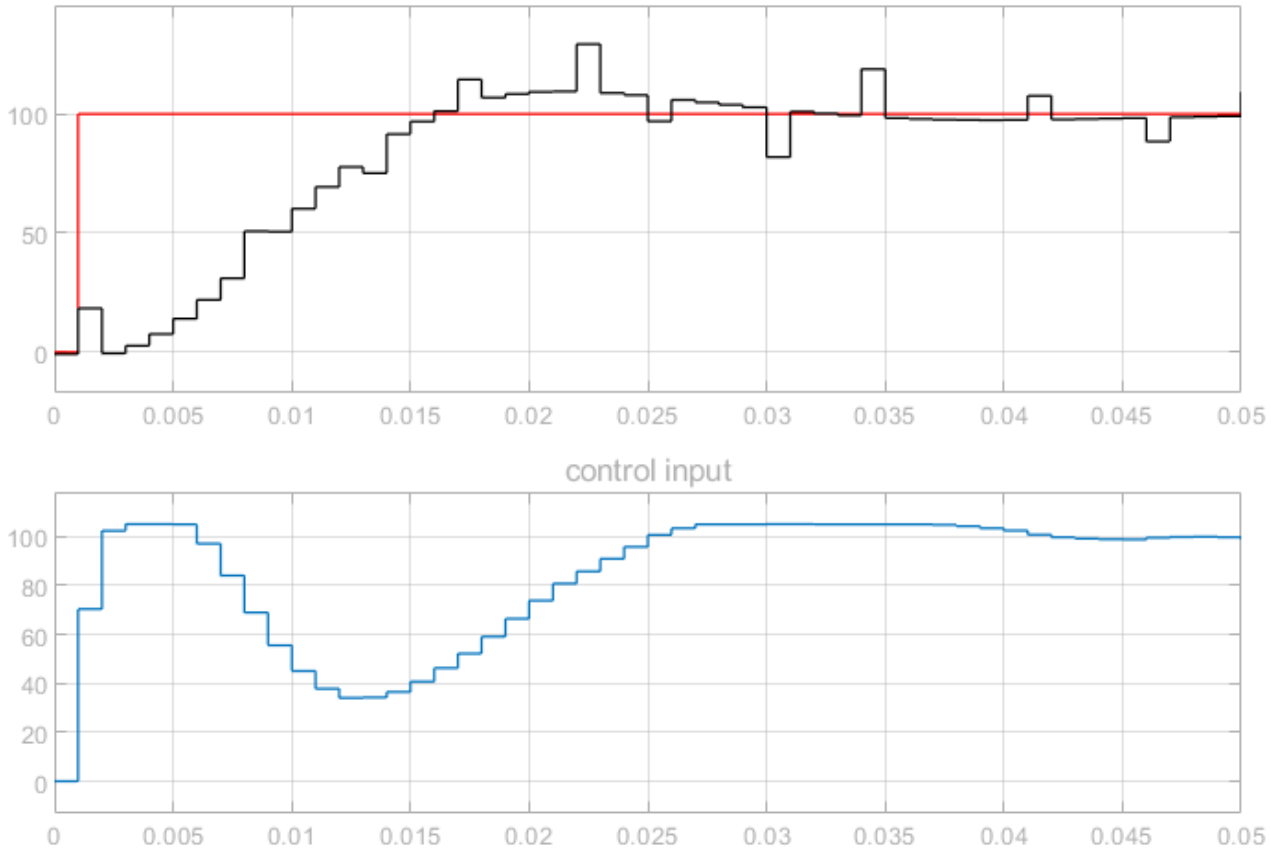

Figure 3.6: MIL reference (in red) control input (in blue) and force output (in black)

Performance index	Symbol	Value
Overshoot	$\hat{s}$	10%
Standard rise time	t_r	0,015 ms
Settling time	$t_{s,5\%}$	0,026 ms

Table 3.3: MIL output signal performance indices

As Table 3.3 shows, the MPC algorithm assures good performance in terms of overshoot and decent performance in terms of rising and settling time. Nevertheless, it must be considered that the system nonlinearities play a huge role in the system's dynamics, as shown in Chapter 6.

3.4 Pico-PLC ladder diagram

As stated in Paragraphs 2.6.1 and 2.6.4, the pico-PLC plays a huge role in the driver's habilitation and system controlling. The overall ladder diagram for the PGAS implementation is shown in Figure 3.7.

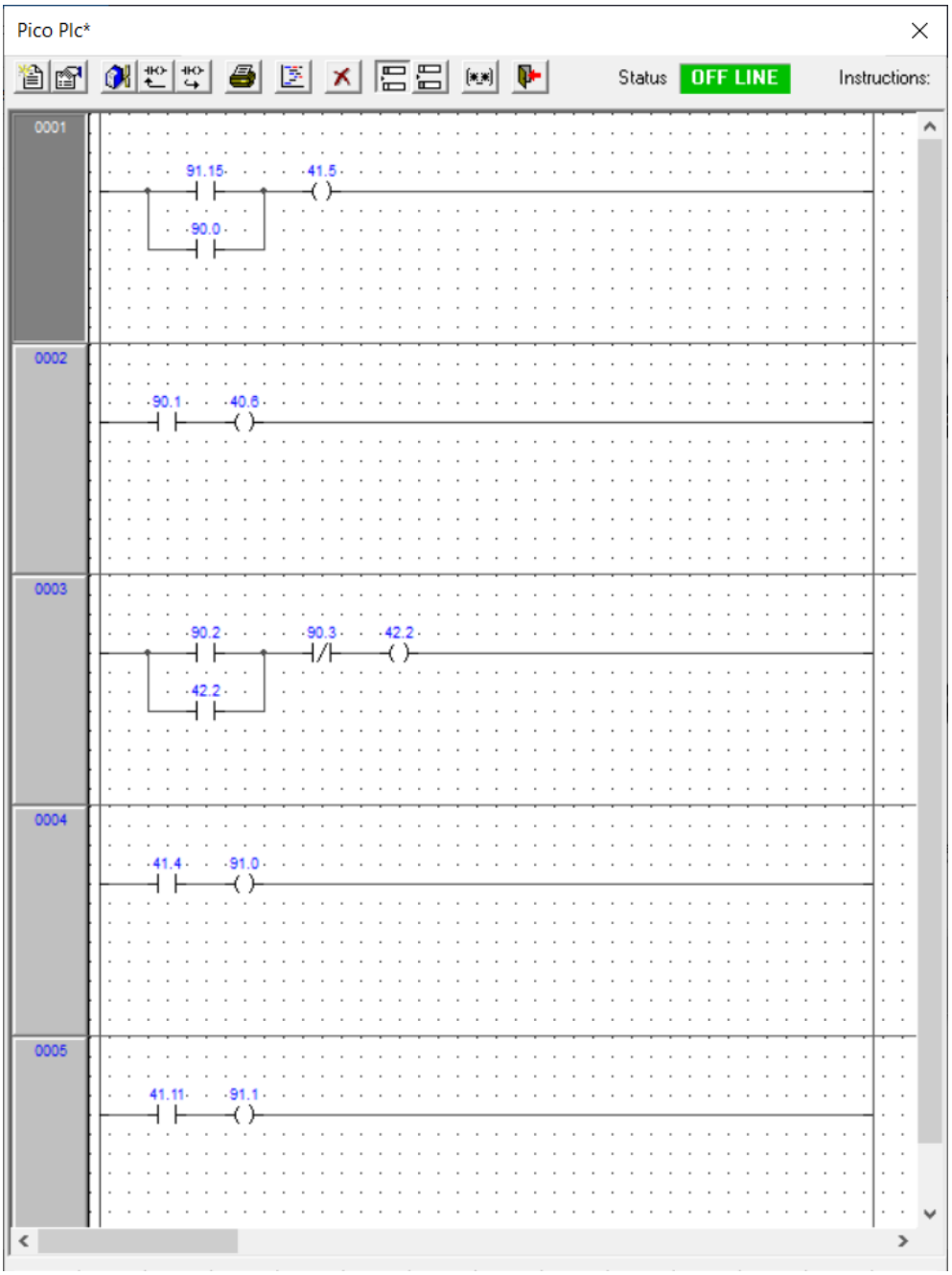

Figure 3.7: Pico-PLC overall ladder diagram in the MotionWiz environment

The ladder diagram is divided into the following five branches:

- *0001*: Hardware enabling branch. Bit b41.5, which is responsible or energizing the phases of the electrical motor, is alimented by the logic sum (OR) of b91.15, the default habilitation bit accessible from the driver's display, and b90.0, the first digital input port, which can be physically switched on by the power supply described in Paragraph 5.2.7;

- *0002*: Stop function branch. Bit 40.6, which enables the deceleration ramp as states in Paragraph 2.6.2 is accessed by the second digital input port b90.1 by default. This branch is not directly used in the PGAS implementation due to the lack of output ports in the *Speedgoat* data acquisition system, described in Paragraph 5.3.1. The stop function is thus indirectly implemented in the *Simulink* Transition Manager block shown in Paragraph 4.4. Nevertheless, the branch could be used in future implementations for redundancy;

- *0003*: Control strategies toggling branch. As stated in Paragraph 2.6.2, b42.2 is responsible for switching from speed regulation mode (b42.2=0), used in Idle, Approach, and Retraction phases, as described in Paragraph 3.2; and force regulation mode (b42.2=1), used in Strike phase, as depicted in Paragraph 3.3. Therefore, the bit must be automatically enabled by the Speedgoat software in order to work efficiently. The problem is that, as respectively stated in Paragraphs 2.6.4 and 5.3.1, the SLVD1N driver's Input High voltage range is from 10 to 24 V, while the Speedgoat FPGAS outputs can only reach approximately 10V. This uncertainty produces the continuous toggling of b90.2, which is thus unfit to directly alimenting b42.2. The problem is solved by using b90.2 and b90.3 as, respectively, SET and RESET control signal, and adopting a RESET dominant memory, whose formula is:

$$X = (SET + x) * \overline{RES} \;\rightarrow\; b42.2 = (b90.2 + b42.2) * \overline{\overline{b90.3}}$$

In which, according to Boolean algebra:

- $A + B$ is the logical sum (OR) of A and B;
- $A * B$ is the logical product (AND) of A and B;
- $\bar{A}$ is the logical negation (NOT) of A.

In this way, b42.2 auto-aliments until the RESET signal from b90.3 switches on, opening the branch. Consequently, the PGAS implementation needs to feature the logic diagram shown in Table 3.4.

Phase	b90.2 (SET signal)	b90.3 (RESET signal)	b42.2
Idle	0	1	0
Approach	0	1	0
Strike	1	0	1
Retraction	0	1	0

Table 3.4: Operating mode SET and RESET signals logic table

The $b90.3 = \overline{b90.2}$ strategy adopted is for software design simplification, as shown in Paragraph 4.3, and due to the additional features of the Transition Manager block described in Paragraph 4.4. A RESET dominant memory was preferred with respect to other strategies due to safety reasons;

- *0004*: Driver's status check branch. The absence of errors in the driver switches on b41.4, which directly aliments the digital output b91.0. This feature can be used in the Simulink via the *Speedgoat* acquisition for monitoring the driver's status;

- *0005*: I2T control status check branch. The i2t control ensuing switches on b41.11, which in turns aliments the digital output b91.1. As for the previous branch, the signal can be used in the Simulink implementation.

Chapter 4: Simulink environment implementation

4.1 Overall system

We can define, as in Figure 4.1, six main blocks composing the proposed *MathWorks Simulink* environment:

- I/O signals management block (in green), which is devoted to processing and converting the signals coming from the sensors and the *Speedgoat* machine and going to the SLVD1N driver (Paragraph 4.3);

- Transition manager block (in purple), which is dedicated to switching the actuator among the various stroke phases (Paragraph 4.4);

- Platform calculations block (in brown), registering data to be used for biomedical considerations (Paragraph 4.5);

- Controllers block (in light blue), containing the different controller transfer functions to be applied to the driver according to the stroke phase (Paragraph 4.6);

- End of simulation block (in red) (Paragraph 4.7);

- User interface (in purple) (Paragraph 4.8).

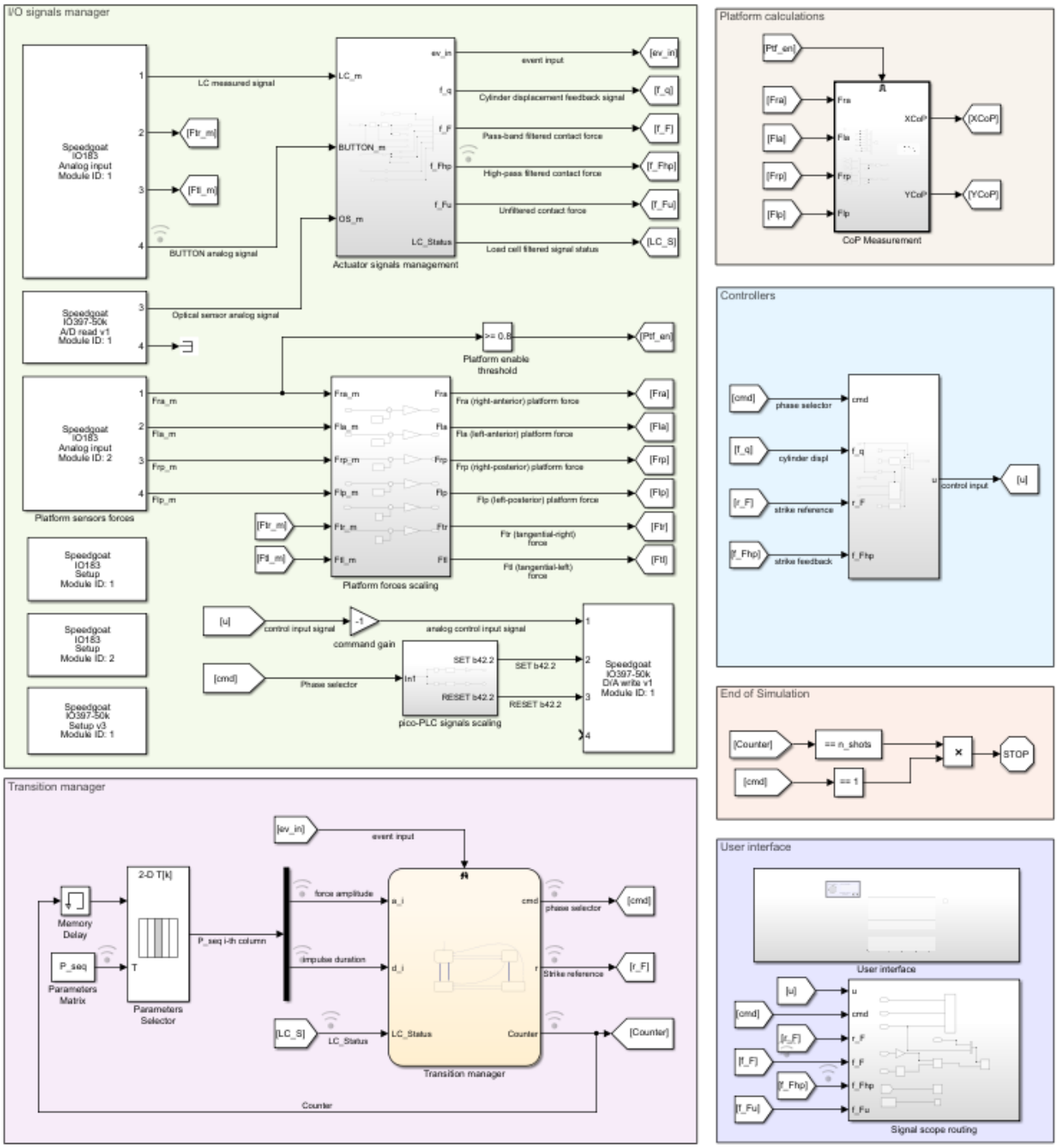

Figure 4.1: Simulink overall environment

4.2 Initial function callback

Once the *Simulink* structure is successfully downloaded into the *Speedgoat* machine and run, an initial function callback prompts a display message asking whether the user wants to upload an existing protocol or directly insert data. Direct insertion involves entering a total number of strikes n, and the force amplitude and duration of each number n_i of equal blows. The software reads (existing protocol option) or writes (direct insertion option) a matrix P_{seq} into the simulation environment of the kind:

$$P_{seq} = \left[\begin{bmatrix} a_1 \\ t_1 \end{bmatrix} \quad \cdots \quad \begin{bmatrix} a_1 \\ t_1 \end{bmatrix} \quad \cdots \quad \begin{bmatrix} a_m \\ t_m \end{bmatrix} \quad \cdots \quad \begin{bmatrix} a_m \\ t_m \end{bmatrix}\right] \in \mathbb{R}^{2xn_{shots}},$$

In which:

$$\begin{bmatrix} a_i \\ t_i \end{bmatrix} \quad \cdots \quad \begin{bmatrix} a_i \\ t_i \end{bmatrix} \in \mathbb{R}^{n_i x 2}, \qquad n_{shots} = \sum_{i=1}^{m} n_i$$

- n_{shots}total number of shots;
- m number of distinctive shots;
- a_i i-th force amplitude;
- d_i i-th impulse duration;
- n_i number of shots defined by a_i and d_i.

Hence the software asks the reader whether to shuffle the matrix's columns, in order to achieve a random sequence of the strikes, or not. Random sequence leads to a more realistic environment for balance perturbation, i.e., less biased results. In fact, the subject could unconsciously prepare their posture to receive the blow once acknowledging its entity after $k < n_i$ strikes if they are not shuffled.

The initial function callback also builds the MPC algorithm used in the *Controllers* block, described in Paragraph 4.6, using the parameters reported in Paragraph 2.8.

4.3 I/O signals manager block

The I/O signals manager block, shown in Figure 4.2, is devoted to the collection, elaboration and transmission of the input and output signals of the *Speedgoat* machine.

We can divide the inputs' kind into three classes:

- Event inputs, which serve for the switching of the various phases as described in Paragraph 3.1, and processed in the Transition management block;
- Feedback inputs, coming from the load cell (Paragraph 2.5.1) and the optical sensor (Paragraph 2.5.2) and can be used for closed-loop control strategies;
- Extra inputs used for biomedical analysis, i.e., tangential and vertical force inputs coming from the platform's sensors.

Every analog input and output coming from and going to the physical wiring harness of the *Speedgoat* is properly rescaled to be processed in the *Simulink* software and the SLVD1N driver as described in Paragraph 2.6.

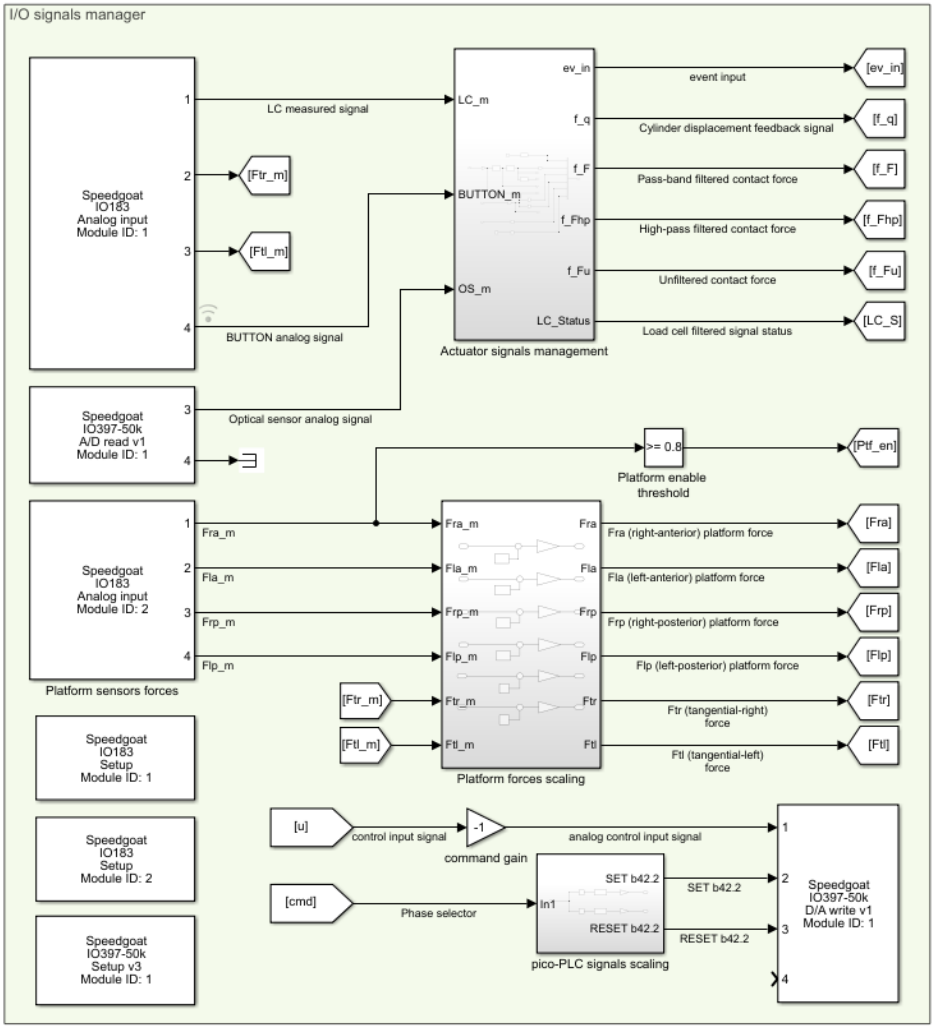

Figure 4.2: I/O signals manager block

The Actuator's signals management subsystem is shown in Figure 3.3.

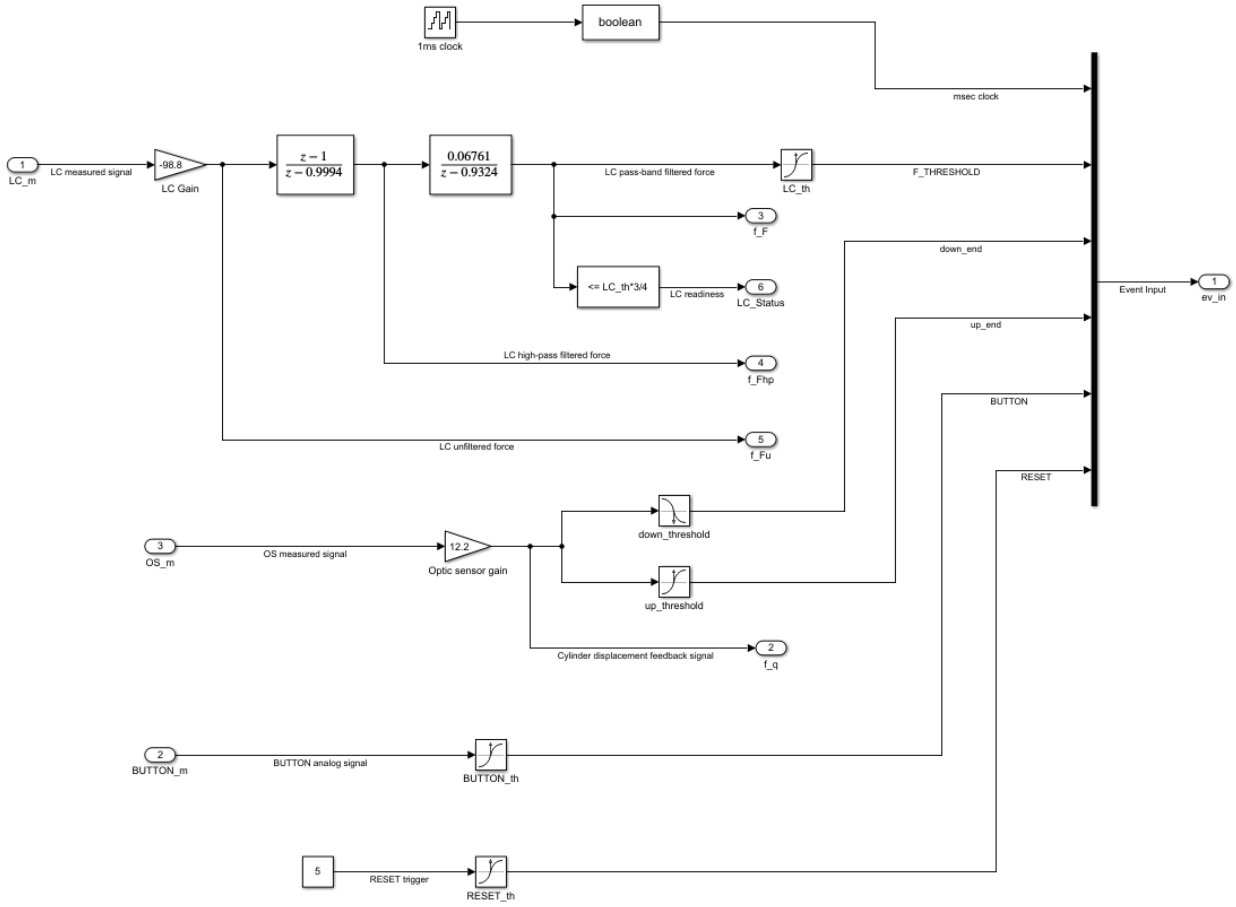

Figure 4.3: Actuator's signals management subsystem

The subsystem implements:

- A Boolean 1 ms clock used in timed events in the Transition manager block described in Paragraph 4.4;

- The scaling of the optical sensor signal in the modalities described in Paragraph 2.5.2. The signal is also used in the Transition manager signal for up and down limit switches through rising and falling threshold blocks;

- The BUTTON analog signal thresholding and the RESET virtual button implementation in the event input feeding the Transition Manager. The RESET virtual button is inside the user interface described in Paragraph 4.8.

- The scaling and filtering of the load cell signal. The pass-band filter adopted was divided in a in the following two blocks:

    - A high-pass filter, whose first order transfer function in Laplace domain is:

$$F_{hp}(s) = \frac{s}{s + 0{,}6283}$$

with a cut-off frequency of 0,1 Hz. The filter aims to remove the offset coming from the load cell sensor without affecting its dynamics. This is because the signal is used not only for thresholding, but also as feedback signal for closed loop control. The high pass filter was preferred with respect to a simple signal difference due to the fact that the load cell offset is not constant during the operation but could move due to the inhomogeneous interface element.

The Laplace domain transfer function has been translated in the discrete z-domain using the so called "zero order hold" technique, using as $t_s = 0.001\ s$, which is the sampling time of the *Speedgoat* machine.

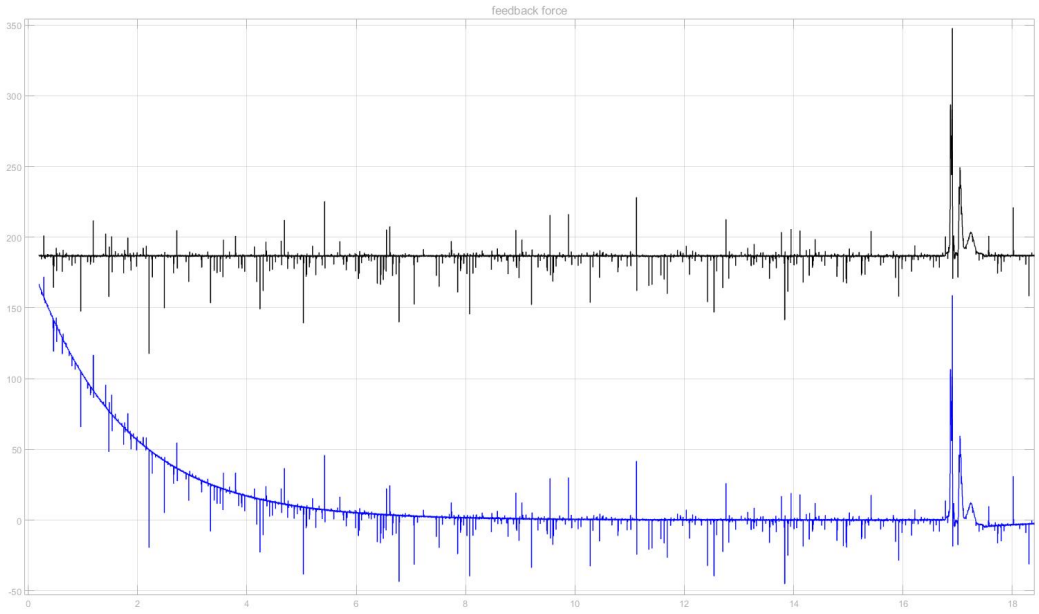

Figure 4.4: High-pass filtered (in blue) and unfiltered (in black) load cell signals

As Figure 4.4 shows, the damping of the signal's offset is not instantaneous. This means that the load cell signal is not immediately ready for comparation in thresholding and feedback as reported, respectively, in the *Transition manager* block (Paragraph 4.4) and the *Controllers* block (Paragraph 4.6). In order to reduce the initial time gap, an offset can be added to the unfiltered signal.

- A low-pass filter adopted, which has the following first order transfer function in the Laplace domain:

$$F(s) = \frac{70}{s + 70}$$

The filter, converted in the z-domain like the previous one, has been selected as a proper trade-off between noise damping and signal delay. In fact, due to the fact that the strike's time duration is of the order of tens of ms, as described in Paragraph 1.5, the filter cannot introduce a relevant delay on the signal. The filter aims to remove all the electromagnetic noise produced by the electric motor.

The resulting signal pass-band filtered signal with respect to the unfiltered one is shown in Figure 4.5.

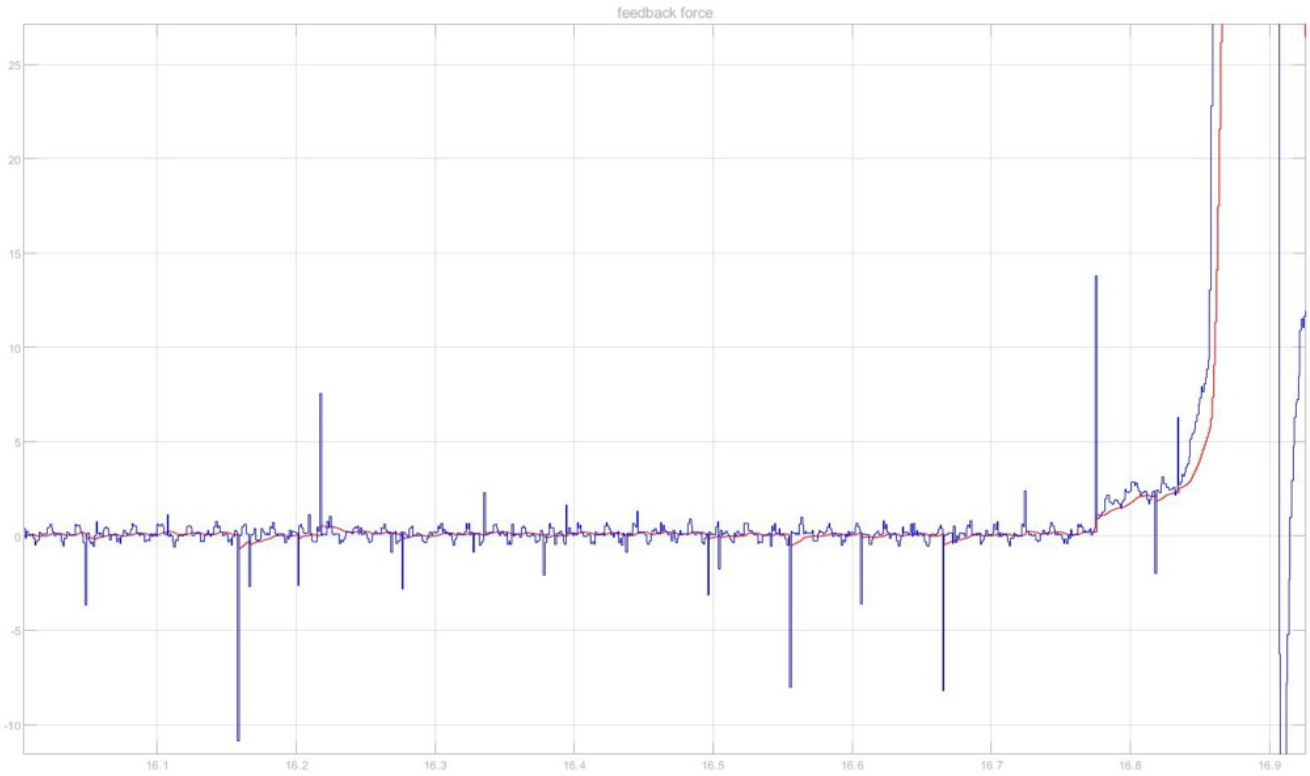

Figure 4.5: Pass-band (in red) and high-pass filtered (in blue) load cell signals

The load cell branch thus outputs:

- The unfiltered contact force *f_Fu* for displaying reasons;
- The high-pass filtered contact force *f_Fhp* to be used as feedback signal in the *Controllers* block, described in Paragraph 4.6. In fact, as Figure 4.5 shows, the pass-band filtered signal *f_F* delay is too high for using the signal as feedback;
- The pass-band filtered contact force *f_F* for displaying reasons. The signal is additionally subject of the following blocks, shown in Figure 4.2:

    - *LC_th= 4 N* threshold, producing the signal *F_THRESHOLD* used in the *Transition manager* block, described in Paragraph 4.4, to switch from the Approach to the Strike phase;
    - *<=LC_th*3/4= 3 N* comparation, producing the signal *LC_Status* that checks for the load cell signal readiness (*LC_Status=1)*. The signal is used in the *Transition manager* block, as a requirement for switching from the Idle to the Approach phase, and in the *User interface* block, described in Paragraph 4.8.

The Platform scaling subsystem is shown in Figure 4.6. The symbols' legend is shown in Table 4.1. The offset and gain values of all the forces had been found experimentally.

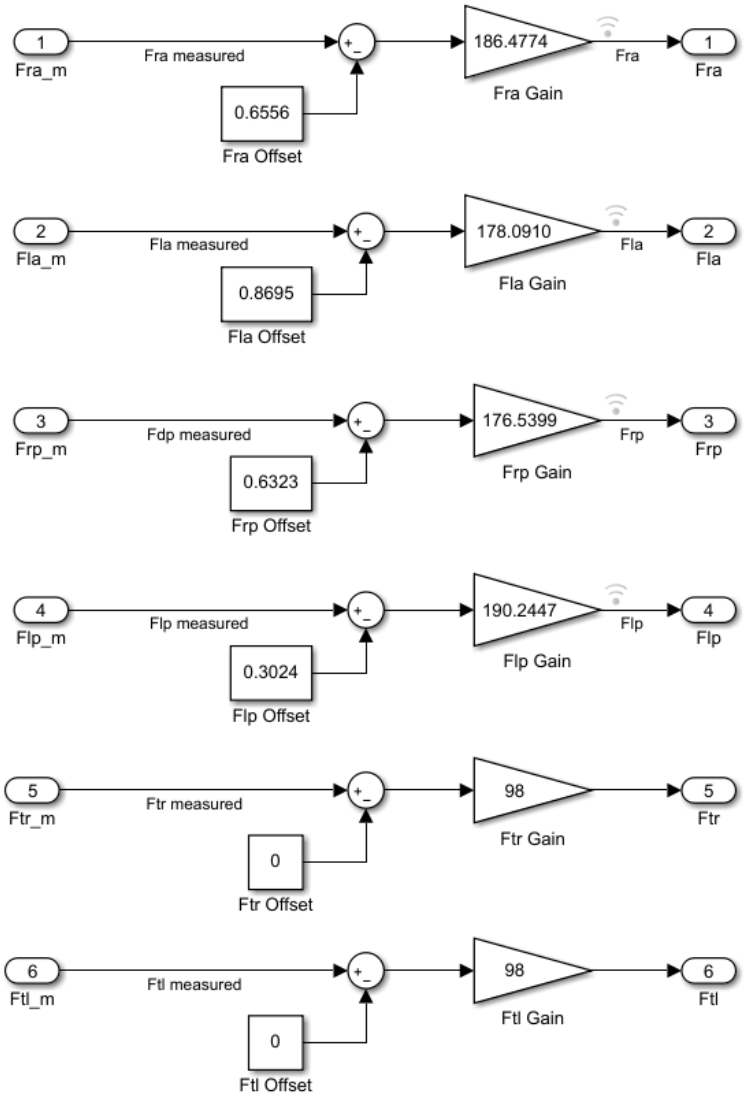

Figure 4.6: Platform scaling subsystem

Symbol	Meaning
Fra	Right-anterior force
Fla	Left-anterior force
Frp	Right-posterior force
Flp	Left-posterior force
Ftr	Tangential right force
Ftl	Tangential left force

Table 4.1: Platform forces legend

There are two kinds of output signals:

- The analog control input signal, which is the output of the *Controllers* block, described in Paragraph 4.6. The signal serves as speed reference in the Idle, Approach, and Retraction phases and force proper control input during the Strike phase;

- The digital SET and RESET signals, described in Paragraph 3.4, interfacing with the SLVD1N driver. The pico-PLC signals scaling subsystem is shown in Figure 4.7.

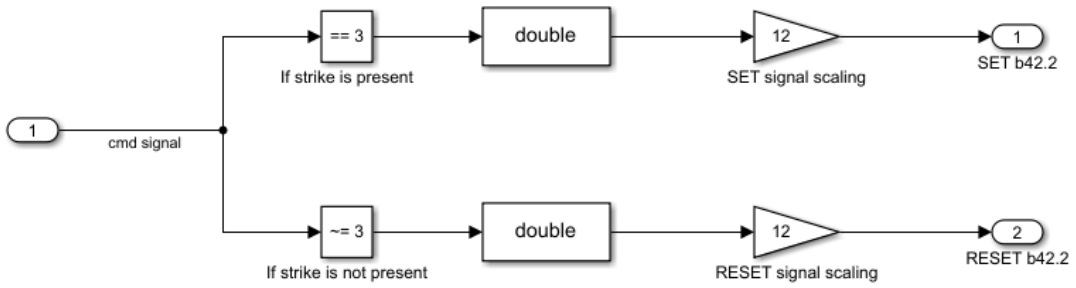

Figure 4.7: The pico-PLC signals scaling subsystem

The signals are derived by the comparation of the phase selector *cmd* signal, described in the Transition Manager block Paragraph 4.4 in order to achieve the logic diagram shown in Table 3.4.

4.4 Transition manager block

The transition manager, whose block is shown in Figure 4.8, communicates with the Initial function callback, through the *Parameters Matrix P_seq* block and the Input signals block, through the *Event input* and *LC_Status* signals.

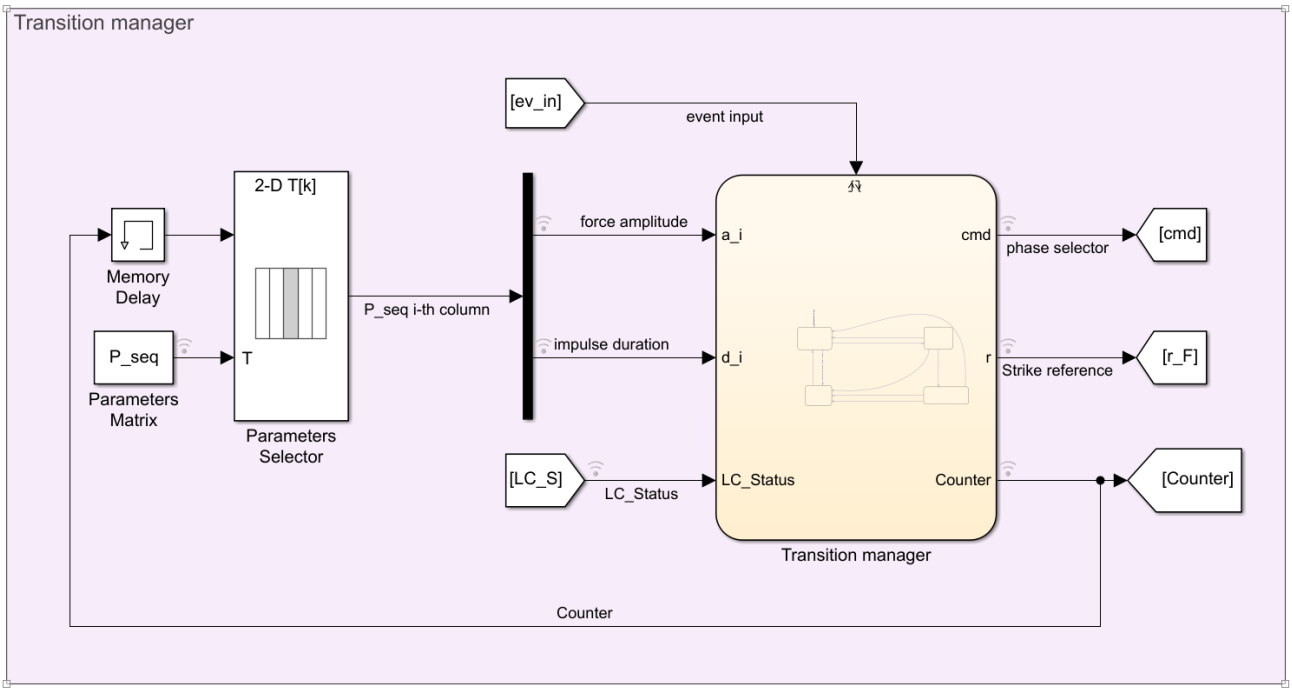

Figure 4.8: Transition manager block

The *P_seq* matrix feeds the Parameter Selector block, which, basing on the memory delayed parameter *Counter*, one of the outputs of the Stateflow block, outputs the corresponding column of the matrix *P_seq*. This data, divided in *a_i* (force amplitude) and *d_i* (impulse duration) is used in the Stateflow block to build up force reference signal *r_F* during the strike phase.

As Figure 4.9 shows, the Event inputs coming from the Input signals block (in brown) are processed in the *Stateflow* chart and used as a trigger for switching between the various phases of the system, as described in Paragraph 3.1. The chart, other than outputting the strike reference force amplitude during the proper amount of time, increases of 1 the *Counter* value after every successful strike, and outputs the parameter *cmd*, which is used as a reference in the *Controllers* block to select the proper controller to be applied according to the phase.

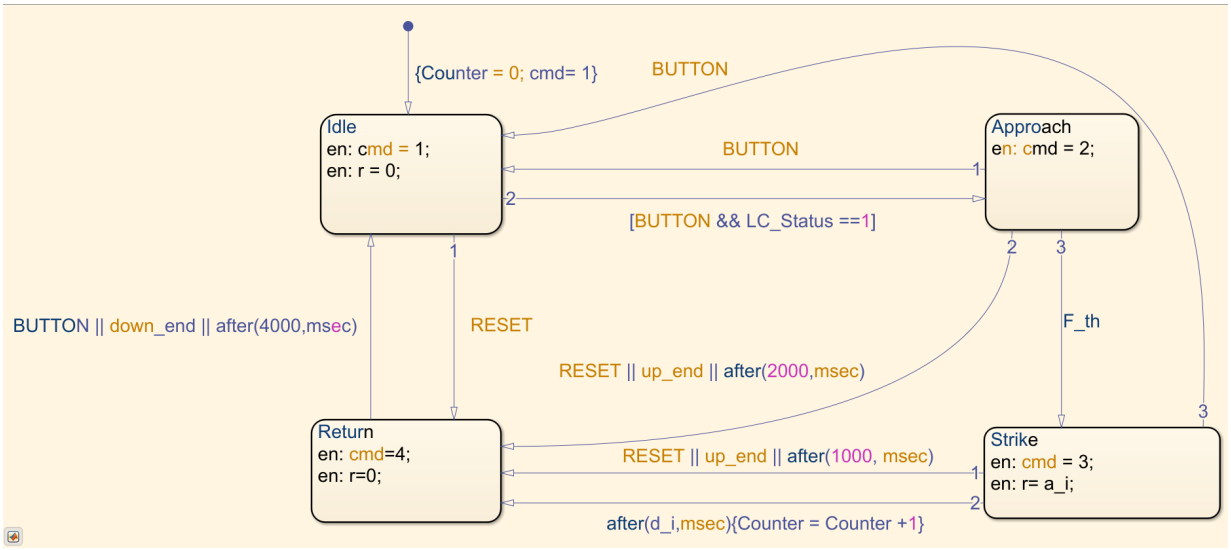

Figure 4.9: Transition manager Stateflow chart

The following additional features have been implemented:

- As described in Paragraph 4.3, the load cell filtered signal is not immediately ready due to transients introduced by the filters. For this reason, a *LC_Status* == 1 check is performed in the switch between the Idle and Approach phases. If this feature wasn't implemented, the Strike phase would immediately follow the Idle phase due to the wrong *F_th* comparation, prompting an incorrect behavior of the actuator;
- For safety reasons, the operator can push the button again to stop the shaft movement at any time;
- The virtual *RESET* button inside the User interface, explained in Paragraph 4.8, issues the rod retraction if the actuator is running or has been energized or stopped in a halfway position;
- Up and down limit switches prevent the rod to knock against the cylinder's case;
- If *F_th* value isn't reached in 2 s or the upper limit switch is activated, i.e., the subject is missed, the Strike phase is skipped, and the Return phase ensues;
- If a too high value of the strike time interval is input in the Initial Callback function or data is corrupted, the Strike phase still ends in maximum 1s or if the actuator has reached the upper limit switch. In this case, as in the situation in which the user pushes the button during the phase, the *Counter* is not increased and the shot can be repeated.

4.5 Platform calculations block

The Platform calculations block, as in Figure 4.10, inputs the scaled platform forces coming the Input management block and the digital enable signal which cuts off indetermination problems when the platform isn't used.

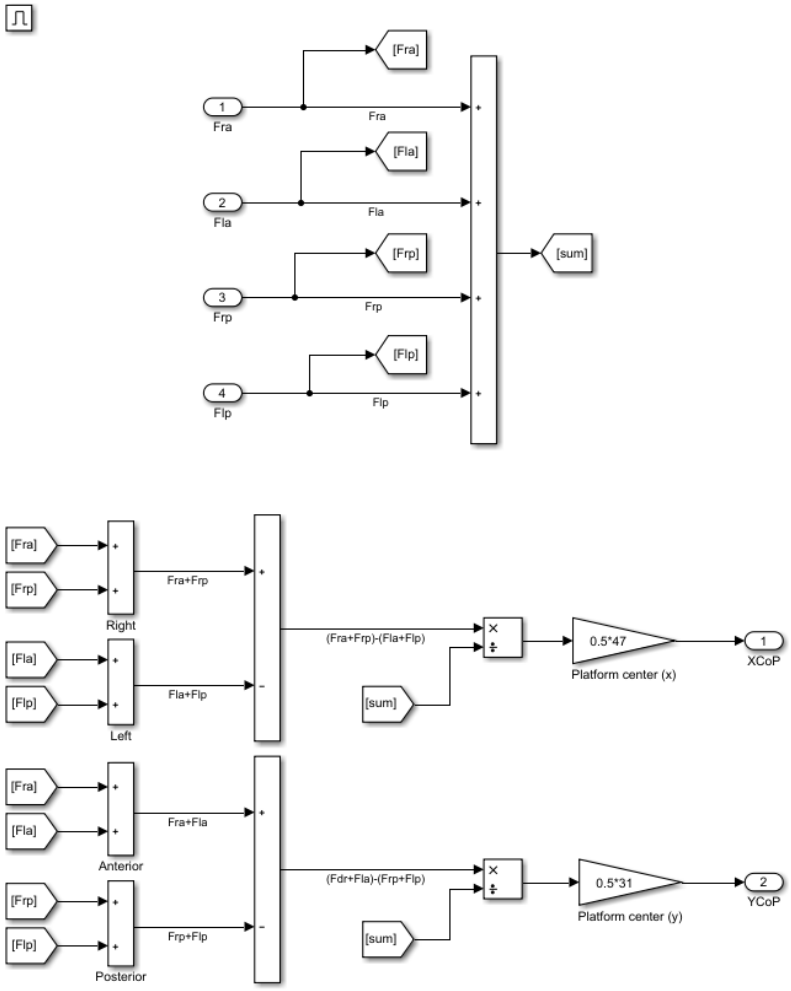

Figure 4.10: Platform calculations block

The enabled block makes and outputs the following computation to find the CoP coordinates (the symbols' legend is reported in Table 4.1) :

$$X_{CoP} = \frac{(F_{ra} + F_{rp}) - (F_{la} + F_{lp})}{F_{la} + F_{ra} + F_{lp} + F_{rp}}$$

$$Y_{CoP} = \frac{(F_{ra} + F_{la}) - (F_{rp} + F_{lp})}{F_{la} + F_{ra} + F_{lp} + F_{rp}}$$

The coordinates are then registered in a proper format and subject to biomedical analysis.

4.6 Controllers block

Directly in contact with the transition manager block, the Controllers block, shown in Figure 4.11, contains all the control transfer functions used for the control logic and strategies described in Chapter 3.

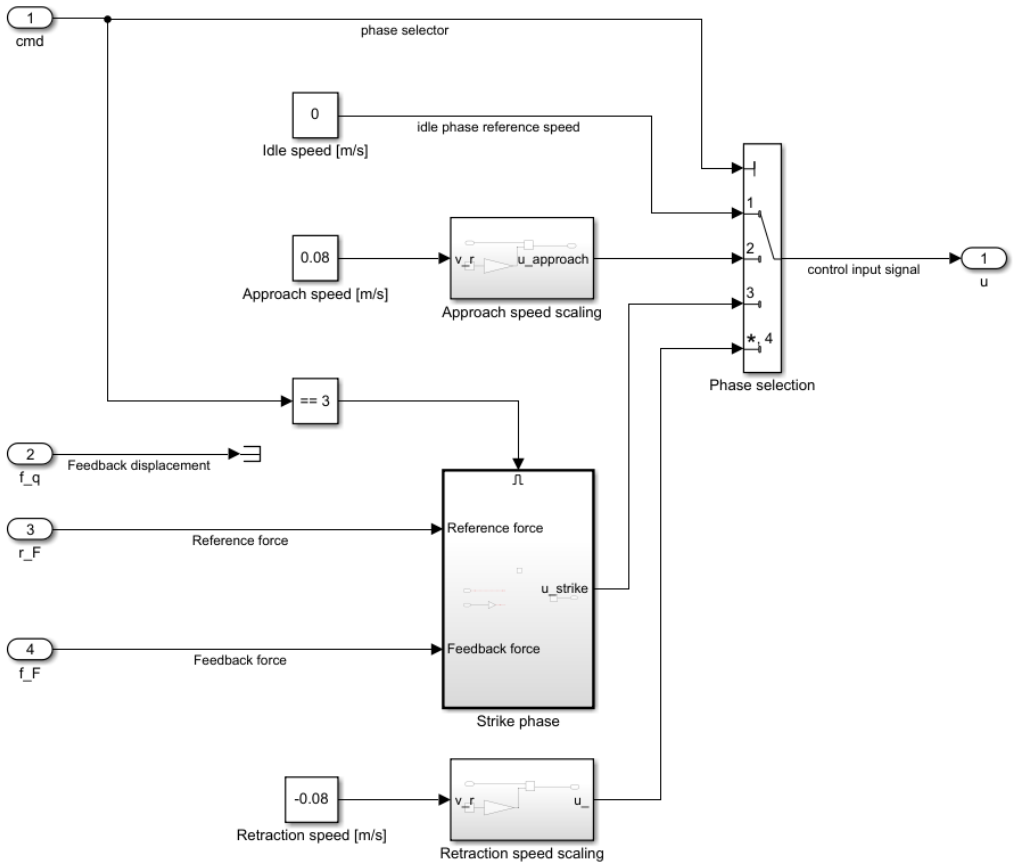

Figure 4.11: Controllers block

The phase selector *cmd* signal, built in the transition manager block, enables each control signal individually, feeding their outputs to the control signal *u* through a proper *Phase selection*. Each block properly scales *u* in order to be readable by the SLVD1N's software structure, as described in Paragraphs 2.6.2, 3.2 and 3.3 through the physical wiring harness. The Controllers block also inputs the feedback signals of the sensors entering through the *Speedgoat* to perform the requested closed-loop control.

Due to the fact that the NiLAB actuator contains an embedded encoder and the speed closed loop is performed in the SLVD1N software, the feedback displacement signal from the optical sensor isn't used in any control algorithm but the *Transition manager*.

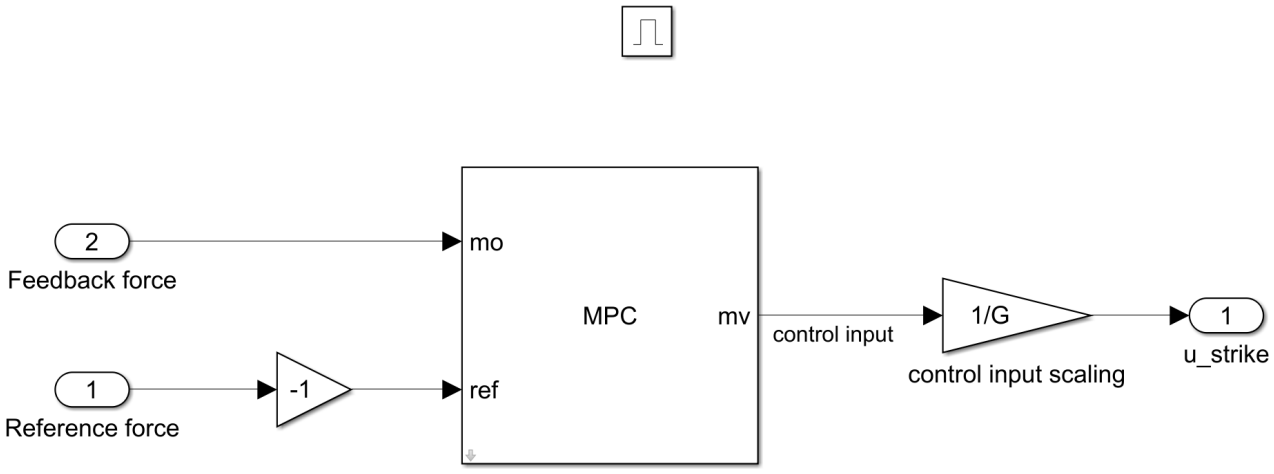

Figure 4.12: Strike phase subsystem

The strike phase block, containing the *MPC Controller* block shown in Figure 4.12, includes an enable block supplied by the *cmd==3* comparison signal. This makes the strike phase algorithm to work only if required, not affecting the *Speedgoat* machine calculus power during the other stages.

4.7 End of simulation block and Stop function callback

Once the desired number of strikes is reached (*Counter == n_shots* check) and the system is in idle phase (*cmd == 1* check) the End of simulation block, shown in Figure 4.13, stops the *Simulink* environment run configuration automatically.

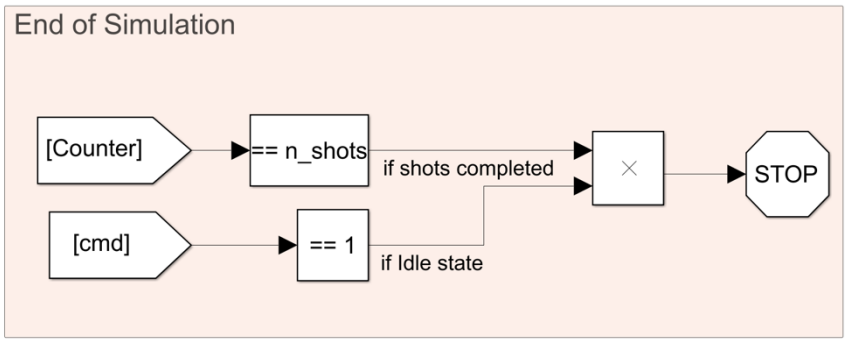

Figure 4.13: End of simulation block

Once the simulation ends, the Stop function callback saves all the relevant data inside a proper file for further biomedical analysis. The *n_shots* parameter has been introduced in Paragraph 4.2.

4.8 User interface

The user interface block, shown in Figure 4.14, is designed in order to contain all the useful information and commands needed during the PGAS's operations.

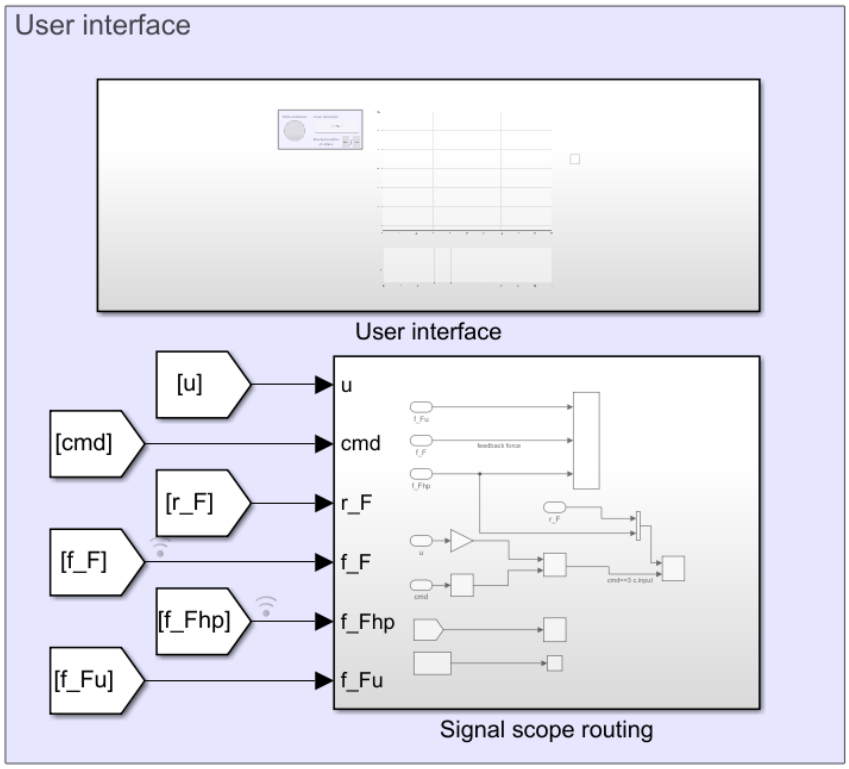

Figure 4.14: User interface block

The user interface subsystem, shown in Figure 4.15, implements:

- The strike readiness check lamp, supplied by the *LC_Status* signal, turning from red to green once the *<=LC_th*3/4* comparation is respected and the user can prompt the Approach phase through the push button, as described in Paragraph 4.3;

- The virtual RESET button, which prompts the immediate retraction of the actuator's rod, issued in Paragraph 4.4;

- The *Counter* display, which monitors, as described in Paragraph 4.4, the number of strikes done by the actuator;

- The following dashboard scopes:
  - The strike reference *r_F* (in red) and the high-pass filtered force *F_hp* (in blue);
  - The phase selector *cmd* scope;

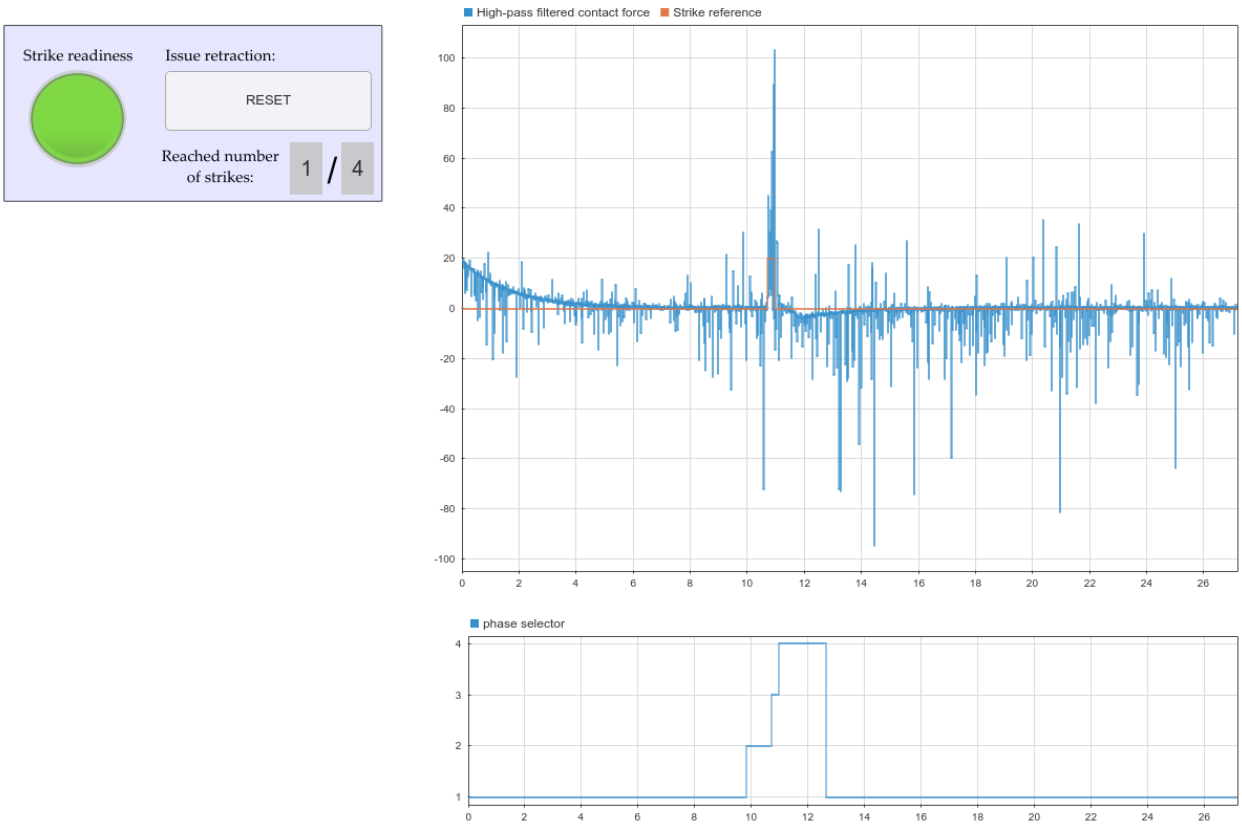

Figure 4.15: User interface environment

The signal scope routing subsystem, shown in Figure 4.16, is used for simple *Simulink* scopes and for user interface subsystem's dashboard scopes connections.

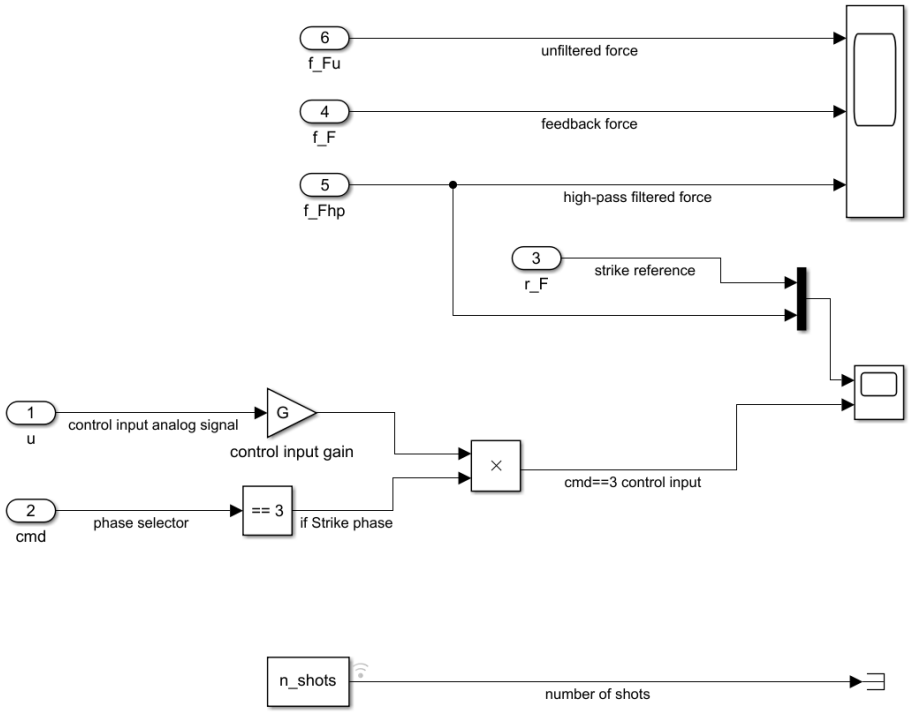

Figure 4.16: Signal scope routing subsystem

Chapter 5: Test bench implementation

5.1 Overall introduction

Test bench analysis is one of the critical steps in the validation phase of the prototype, as real subject testing cannot be performed without proper unit and integration testing. Test bench analysis shall verify the software architectural design's and software-hardware interface's compliances, along with proving control performance and robustness.

The test environment chosen for the PGAS fourth prototype is similar to the previous two ones and is shown in Figure 5.1. *SolidWorks* software was used to model and validate the solution adopted before implementing the necessary modifications on the previous configuration. The overall system in *SolidWorks* environment is shown in Figure 5.2 and its components are listed in Table 5.1.

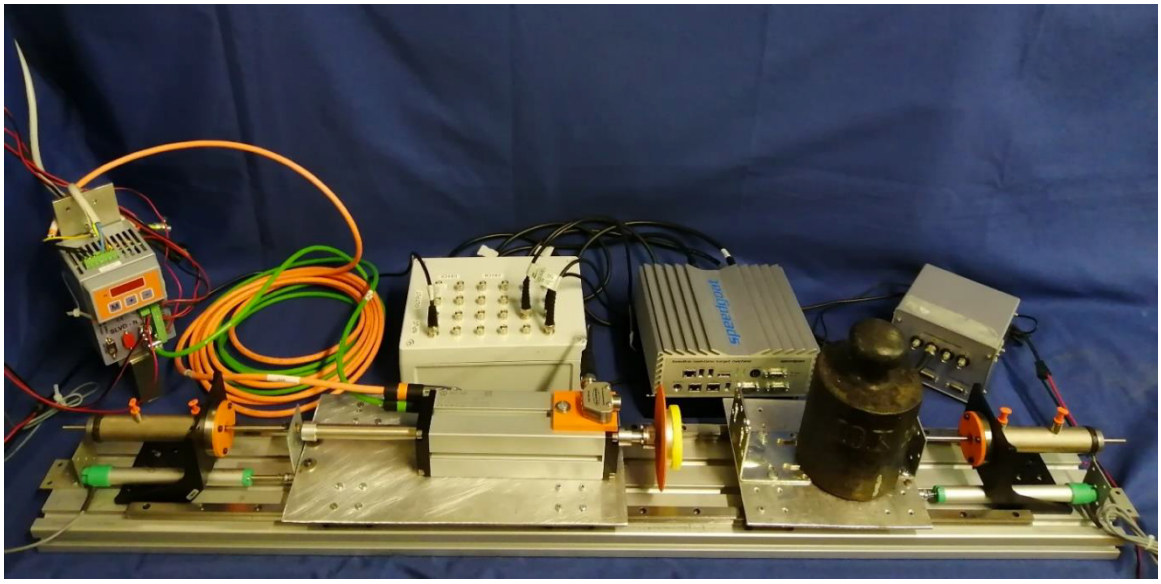

Figure 5.1: PGAS fourth prototype bench testing elements

Two sliding plates connected to bumpers were chosen to simulate the subject's and user's strike response, and coupled as in Figure 2.13. The quoted chart of the actuator's sliding plate is displayed in Appendix A, Chart 1.

Every parameter of Table 2.8 can be easily modified to test control robustness, which is a key requirement in the implementation, as mentioned in Paragraph 3.3. To properly verify every system associated dynamics in this phase, two position transducers, checking the actuator and body displacements, were added.

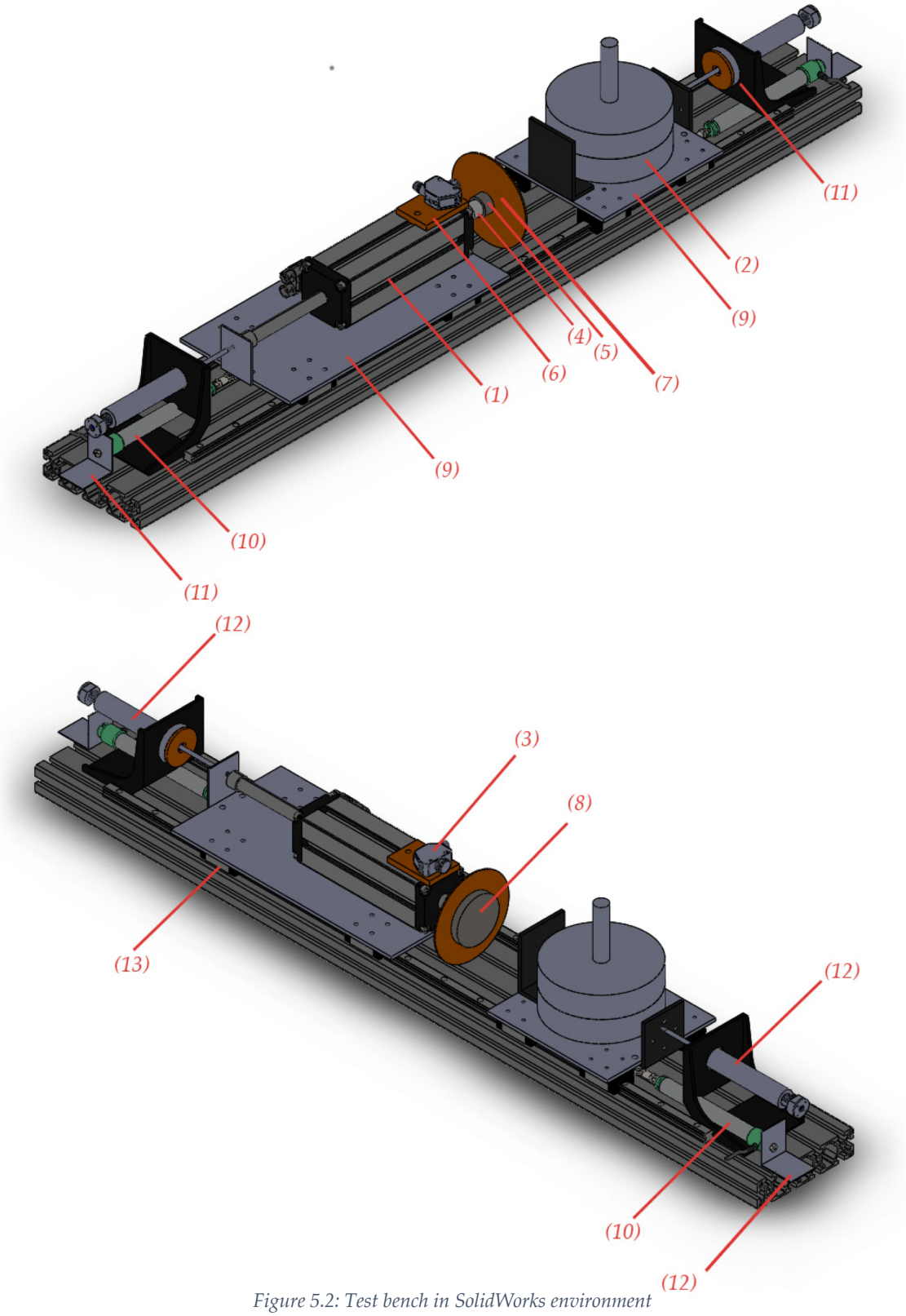

Figure 5.2: Test bench in SolidWorks environment

#	**Component**
1	NiLAB GmbH GD160Q linear motor (Paragraph 2.3)
2	Weights
3	BANNER Q4XTULAF400-Q8 optical sensor (Paragraph 2.5.2)
4	Load cell electrical insulation element (Paragraph 5.2.6)
5	Dacell UMM load cell (Paragraph 2.5.1)
6	Optical sensor's support (Paragraph 5.2.4)
7	Optical sensor's target disk (Paragraph 5.2.5)
8	Strike interface (Paragraph 2.4)
9	Sliding plates
10	GEFRAN PZ-34-A-100 displacement transducers (Paragraph 5.2.2)
11	Displacement transducers' L brackets
12	Linear spring bumpers (Paragraph 5.2.3)
13	MISUMI C-SHR28-1000-B4 linear guides (Paragraph 5.2.1)

Table 5.1: Testbench system components

5.2 Additional elements used in bench testing

5.2.1 *The MISUMI C-SHR28-1000-B4 linear guides*

A *MISUMI* C-SHR28-1000-B4 linear guide per plate corner per sliding element was used, for a total of eight sliders. The guides, specifically designed for heavy loads, are filled with lithium soap based grease and guarantee low-attrite performances through two rows of contacting steel balls, whose recirculation system is shown in Figure 5.3.

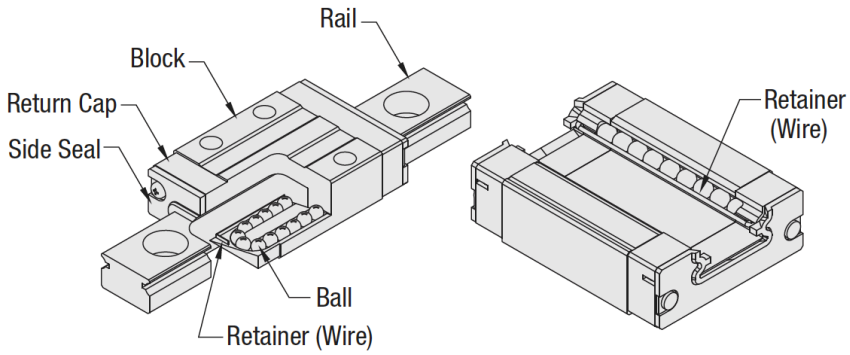

Figure 5.3: The linear guides adopted for bench testing

The adopted solution guarantees good performances in terms of friction force, i.e. required thrust force, and dynamic attrite coefficient μ = 0.004~0.006. The two parallel rails used in the implementation are 1 meter long.

Each sliding element is screwed to the relative sliding element's iron plate through four threaded holes.

5.2.2 *The GEFRAN PZ-34-A-100 displacement transducers*

The test bench environment also implements two *GEFRAN* PZ-34-A-100 displacement transducers, shown in Figure 5.4, to have a more proper system insight during the strike phase. The transducers are mounted through a self-aligning ball joint to each sliding plate and secured to ground through fitting L-shaped brackets as in Figure 5.1. They feature a 100 mm useful stroke and 0.1 V/mm analog gain.

Figure 5.4: The displacement transducer adopted for bench testing

5.2.3 *Linear spring bumpers*

Two custom made linear spring bumpers, realized by the DIMEAS department and shown in Figure 5.5, were used in order to model the stiffness and damping parameters of the actuator and the body. The element is composed by two parallel springs mounted on a central inner piston fixed to the contact rod. In this way one spring works in tension and the other in compression as shown in Figure 5.6. The bumpers are secured in place through L-shaped brackets as shown in Figure 5.1.

Figure 5.5: Linear spring bumper

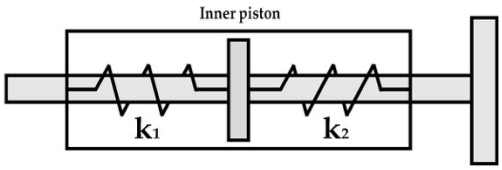

Figure 5.6: Linear bumper spring's position

The total stiffness of the element is then the sum of the two spring stiffnesses. Assuming equal values, thus:

$$k_1 = k_2 = \frac{k_{tot}}{2}$$

Three different types of linear springs are used to test the robustness of the control algorithm. The values are reported in Table 5.2.

Parameter	**Symbol**	**Stiffness Value [N/m]**	**Free length [m]**
Bumper stiffness	k_a, k_b	720	0.094
		1340	0.0935
		1610	0.0855

Table 5.2: Linear spring bumper stiffness values

The control algorithm assumes a value of 1500 N/m. As for the linear damping coefficients c_a and c_b, a common value of 1000 N*s/m is used.

5.2.4 *Optical sensor's support*

The optical sensor's place and mounting structure have been redesigned with respect to the previous prototype test bench due to the actuator's change.

As already mentioned in Paragraph 2.3, three of the four sides of the prismatic stator's surface contain properly fitted t-slots to use in order to mount the actuator to the chassis and to optional elements. Therefore, the upper side of the actuator is used as the bearing surface of the optical sensor support. The sensor's support underside, i.e., the contact surface with the actuator, is shown in Figure 5.7, and its features are listed in Table 5.3. The quoted chart of the piece is displayed in Appendix A, Chart 2.

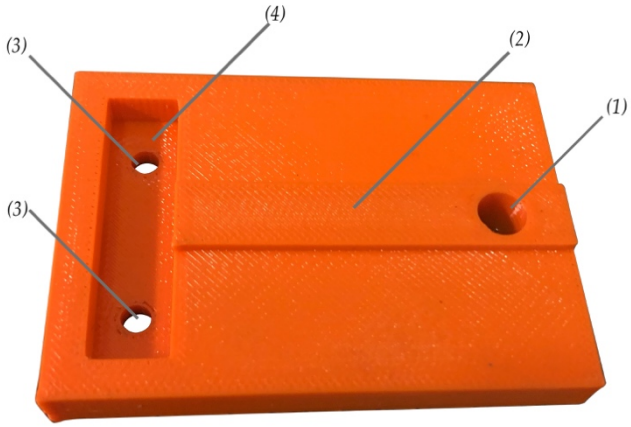

Figure 5.7: Sensor's support underside

#	**Coupling**	**Feature**
1	Support-actuator	through hole
2		contact surface extrusion
3	Support-optical sensor	through holes
4		Screw head slot

Table 5.3: Sensor's support features list

The extrusion on the underside of the support is designed to have clearance with the actuator's upper t-slot to place the piece in proper contact with the actuator's anterior cap. In this way, not only the support is fixed in rotation, but the sensor is not required to be recalibrated at every installation.

The optical sensor coupling is performed with two screw-nut couplings. A slot is designed to contain the two screw heads.

Due to the fact that the piece is not subject to heavy structural tensions, the piece was realized in ABS material through 3D-printing. The used 3D-printer is the *Stratasys'* Uprint SE Plus model, shown in Figure 5.8.

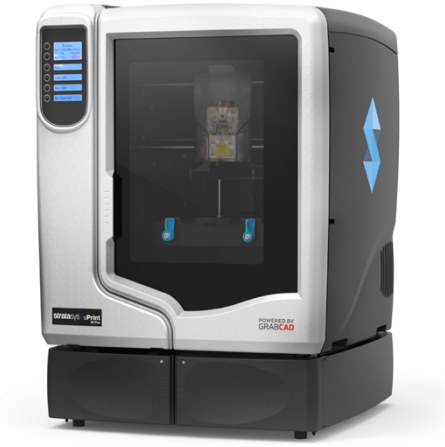

Figure 5.8: Stratasys' Uprint SE Plus

The printer operates with the so-called fused filament fabrication (FFF) technology, in which both the fabrication and support fibers are melted at a temperature of 300 °C and extruded on a disposable plate. The machine can work with files directly obtained in a *SolidWorks* environment. The object's orientation was printed upside-down in order to have no undercuts.

5.2.5 *Optical sensor's target disk*

Due to its position atop the actuator, the laser sensor requires a target disk to pinpoint the slider's displacement. The disk, shown in Figure 5.9, was realized with the same 3D-printing technique as the optical sensor's support described in Paragraph 78 and is put between the interface and the loadcell.

Figure 5.9: Target disk

The quoted chart of the piece is displayed in Appendix A, Chart 3.

5.2.6 *Load cell electrical insulation element*

One of the main problems encountered on test bench implementation was caused by the direct contact between the energized rod of the NiLAB electrical motor and the Dacell UMM load cell, described, respectively, in Paragraphs 2.3 and 2.5.1. This connection produced huge noises coming from the load cell signal, caused by both the actuator's electromagnetic field and, to a more considerable extent, eddy current. This is because the load cell sensor ordinarily works with low amounts of current, incompatible with the parasite current generated by the motor's energized phases.

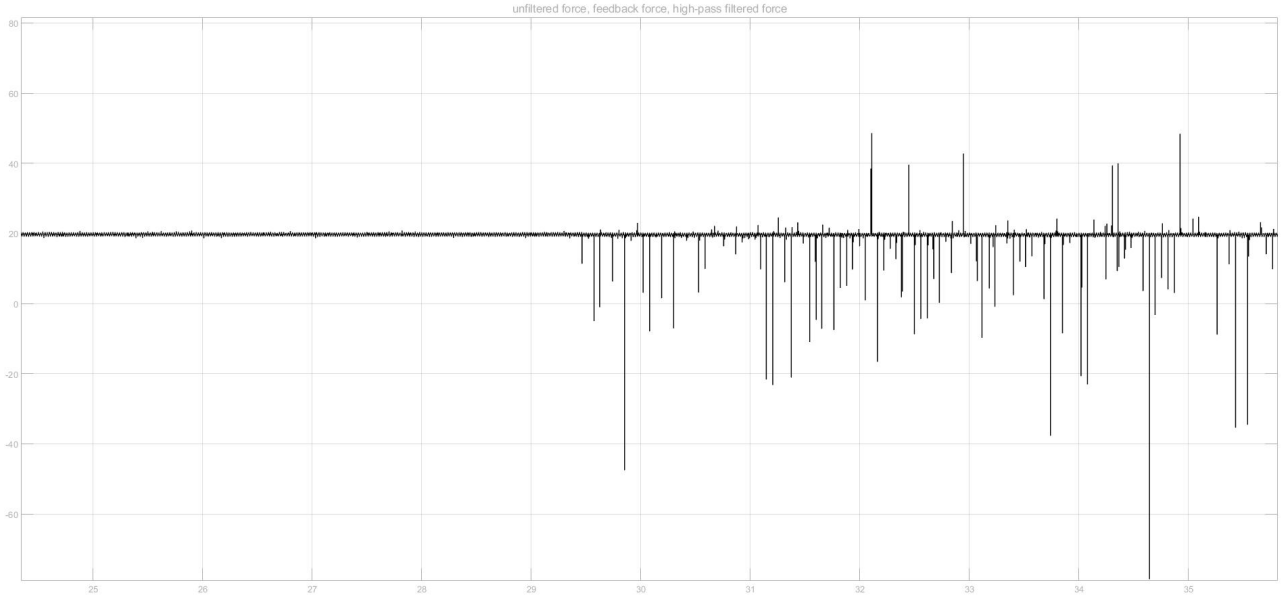

Figure 5.10: Electromagnetic noise effect on load cell signal

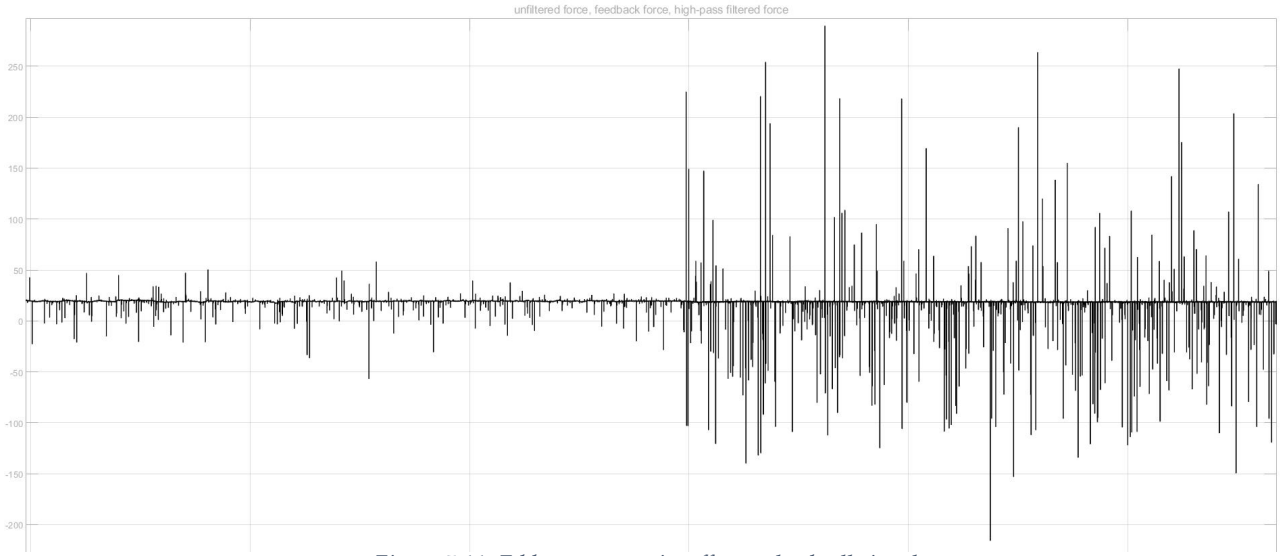

Figure 5.11: Eddy current noise effect on load cell signal

For this reason, other than shielding the acquisition system wire, an electrical insulation element was interposed between the load cell and the rod. The piece, shown in Figure 5.12, was obtained from the turning of a PTFE (Teflon) cylinder, as a simple Male-to-Female screw adapter.

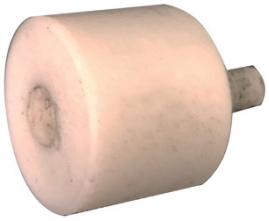

Figure 5.12: Load cell electrical insulation element

The quoted chart of the piece is displayed in Appendix A, Chart 4.

5.2.7 *Power supply*

The test bench implements a 24V power supply, shown in Figure 5.13. The element is connected to:

- The driver's DC alimentation port (Paragraph 2.6.1);
- The driver's first digital input port in order to enable the energization of the phases of the linear motor through the pico-PLC (Branch 0001 of Paragraph 3.4);
- The BANNER optical sensor (Paragraph 2.5.2);

Figure 5.13: Test bench power supply

5.3 The data acquisition system

5.3.1 *The Speedgoat acquisition*

The *Speedgoat* acquisition box is made of 4 BNC ports per input or output per FPGA module, for a total of 24 ports. The box directly communicates with the *Speedgoat* machine described in Paragraph 2.7 through three (one per FPGA module) 17-pin M12 round cables.

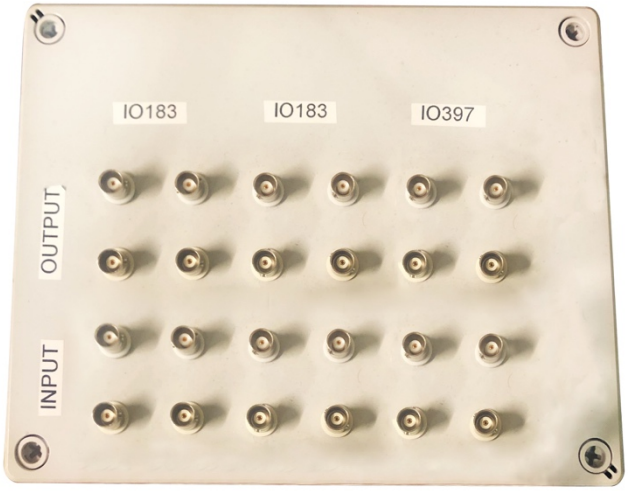

Figure 5.14: The Speedgoat data acquisition system

5.3.2 *The Dewetron DEWE-RACK-4 data logger*

A *Dewetron DEWE-RACK-4* is used to convert the load cell serial port to a BNC connector. The rack aliments the load cell with its proper voltage and can implement a low-pass filter.

Figure 5.15: The Dewetron DEWE-RACK-4 data logger

Chapter 6: Bench testing

6.1 Premise

Test bench data acquisition was the final core task included in the PGAS project research. The activity focused on all the variables involved in the actuator's closed loop control, without concentrating on the platform system. Due to the fact that Idle, Approach, and Retraction phases did not require any complex control activity, the bench testing primarily aimed on the MPC algorithm tuning.

As described in Paragraph 3.3, the MPC offers a unique flexibility in terms of all the parameters involved. By directly accessing the modelled system and control algorithm structure through the *Initial function callback*, reported in Paragraph 4.2, the user can easily modify all the parameters involved, which are:

- The initial state vector and the mass, stiffness, and damping values of the actuator, the body and the interface elements, which characterize the plant system according to the equations shown in Paragraph 2.8;

- The MPC control algorithm parameters, which are:

    - The prediction and control horizon;
    - The weight functions, i.e., as stated in Paragraph 3.3;
    - The control input saturation.

The direct access on the modelled system could be used in the operative circumstances to improve the control performance: for example, the final user could upload the actuator and body masses case by case. Straightforward data modification also plays a huge role in simplifying the MPC algorithm tuning phase, which is described in Paragraph 6.2. This is one of the many reasons discussed in Paragraph 3.3 which make models predictive control one of the most used techniques in modern design of control systems.

6.2 The MPC algorithm tuning

As stated in Paragraph 2.1, the main objective of the PGAS's fourth prototype research is the direct performance comparation with respect to the previous electro-pneumatic device. This also implies the determination of all the benefits and drawbacks of the two actuations in terms of impulse generation during the strike phase. Therefore, bench testing adopted the same contact force reference signals, which are divided in the two biomedical protocols shown in Table 6.1.

Protocol name	**#**	**Contact force [N]**	**Impulse duration [ms]**
*Constant impulse (I = 5000 N*s)*	*1*	20	250
	2	40	125
	3	60	83
	4	100	50
Constant force (F = 40 N)	*5*	40	50
	6	40	125
	7	40	150
	8	40	250

Table 6.1: Protocol reference contact force and impulse duration

As it can be noted, the various profile differ from each other considerably. Profile #4 is the most critical one, requiring the lowest impulse duration and the highest force amplitude values, which is near to the input saturation. Vice versa, profile #1 is the least critical to achieve for the opposite reasons. As experienced in test benching activity, all the other profiles show an in-between behavior with respect to profiles #1 and #4. Therefore, for sake of simplicity and report clearness, all the subsequent charts only focus on the two.

As described in Paragraphs 3.3.2 and 3.3.3, the tunable variables of the MPC algorithm are:

- The Prediction and Control horizons H_p and H_c;
- The cost function weight matrices, in the PGAS, one-dimensional:
  - The output error weight matrix Q;
  - The modified variable target error weight matrix R_u;
  - The modified variable rate weight matrix $R_{\Delta u}$.

All the charts are styled in the following fashion:

- Reference input (in red) and high-pass filtered contact force (see Paragraph 4.3) (in black) on the upper chart;
- Control input (in blue) in the strike phase (*cmd*==*3*, see Paragraphs 3.3 and 4.4) on the lower chart;

The contact force undershoot and overshoot in the Retraction phase shown in all the charts is due to inertia force due to high accelerations, or unwanted second contact. The latter condition can be easily removed by increasing the absolute value of the retraction speed in the *Controllers* block shown in Figure 4.11. This is not done in all the charts due to the fact that the phenomena involving the rod retraction are highly influenced of initial, i.e., end of Strike phase, conditions.

As one can note, the force reference profile in all the charts does not issue at exactly *LC_th*=4 N. This is due to internal delays in *Speedgoat* calculus, *Speedgoat*-driver communications, and driver calculus.

6.2.1 *Default conditions control*

Starting to the MPC toolbox default conditions, shown in Table 6.2, and imposing the prediction and control horizons as in said table, the MPC algorithm shows bad performance in both #1 and #4 profiles.

Parameter		Symbol	Value	Unit
Prediction horizon		H_p	10	steps
Control horizon		H_c	10	steps
Control input constraint		$u_{max} = -u_{min}$	105	N
Weights	Manipulated variables (MV)	$R_u \epsilon \mathbb{R}^{1x1}$	0	
	Manipulated variables rate (MVR)	$R_{\Delta u} \epsilon \mathbb{R}^{1x1}$	0,1	
	Output variables (OV)	$Q \epsilon \mathbb{R}^{1x1}$	1	

Table 6.2: MPC default conditions parameters

In detail:

- #1 Profile (Figure 6.1): the MPC algorithm cannot face the nonlinearities involved in the phenomenon, outputting an increasing and saturating oscillation. The control input also suffers the noise spikes from the load cell, adopting a different and unnecessary control strategy once it senses them;

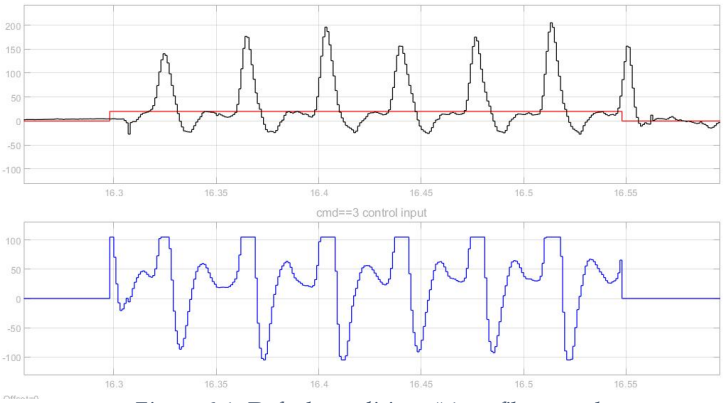

Figure 6.1: Default conditions #1 profile control

- #4 Profile (Figure 6.2): the MPC algorithm completely saturates the control input, outputting a too high overshoot (nearly 50%).

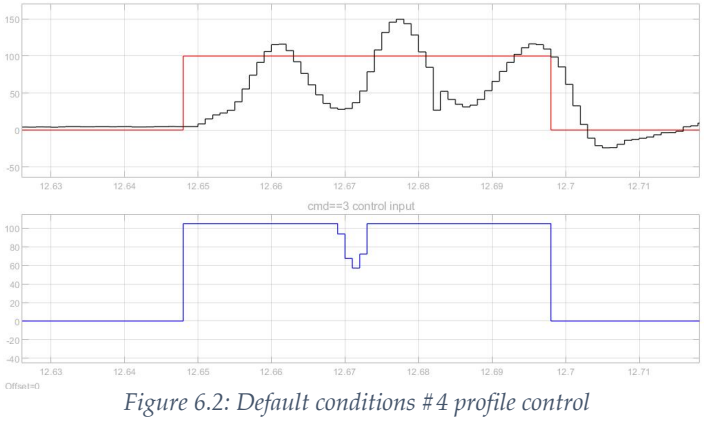

Figure 6.2: Default conditions #4 profile control

6.2.2 *Increased MVR weight control*

The following experiment was performed in order to reduce the control receptiveness, increasing the Manipulated Variable Rate (MVR) weight in the cost function as shown in Table 6.3. The control algorithm shows decent overall performance in both profiles, with poor raising time.

Parameter		Symbol	Value	Unit
Weights	MV	$R_u \epsilon \mathbb{R}^{1x1}$	0	
	MVR	$R_{\Delta u} \epsilon \mathbb{R}^{1x1}$	~~0,1~~ → 10	
	OV	$Q \epsilon \mathbb{R}^{1x1}$	1	

Table 6.3: MPC increased MVR weight conditions parameters

In detail:

- #1 Profile (Figure 6.3): the MPC algorithm prefers a lower control input speed with respect to the output performance. Nevertheless, noise receptiveness has been completely damped out;

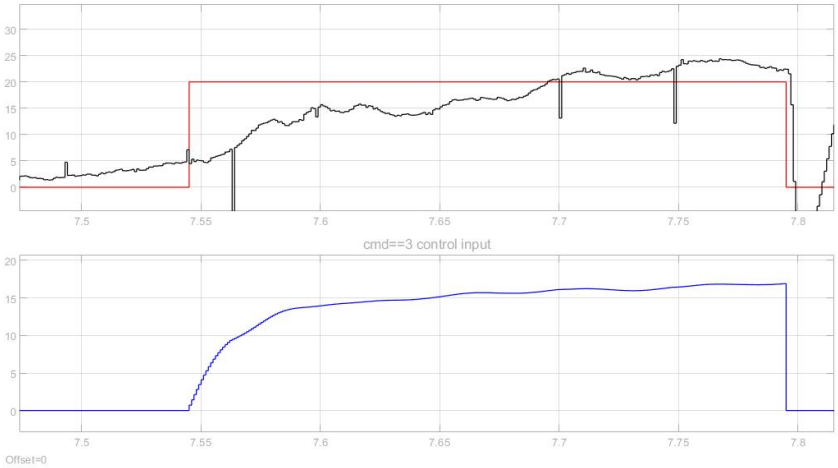

Figure 6.3: Increased MVR weight conditions #1 profile control

- #4 Profile (Figure 6.4): the MPC algorithm shows a behavior similar to Profile #1.

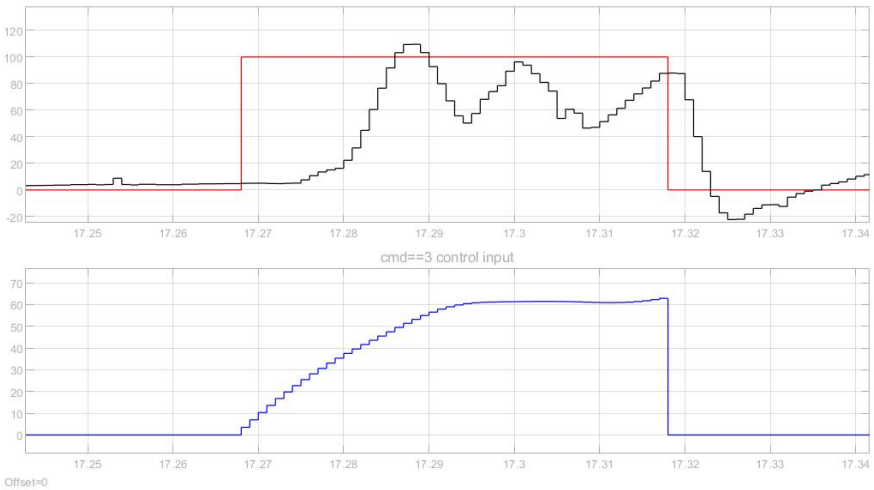

Figure 6.4: Increased MVR weight conditions #4 profile control

6.2.3 *Increased MVR and OV (MVR>OV) weights control*

The following experiment was performed in order to increase the output variable (OV) weight in the cost function and thus increase the performance. Due to the fact that the Default conditions experiment shown in Paragraph 6.2.1 gives a too instable control input, the MVR weight has been kept as in the previous experiment. As Table 6.4 shows, MVR still has an higher weight than OV.

Parameter		Symbol	Value	Unit
Weights	MV	$R_u \epsilon \mathbb{R}^{1x1}$	0	
	MVR	$R_{\Delta u} \epsilon \mathbb{R}^{1x1}$	10	
	OV	$Q \epsilon \mathbb{R}^{1x1}$	$1 \rightarrow 5$	

Table 6.4: MPC increased MVR and OV (MVR>OV) weight control parameters

In detail:

- #1 Profile (Figure 6.5): the MPC algorithm still shows the problems encountered in the Default experiment, as in Figure 6.1. The control algorithm cannot face the nonlinearities of the system and introduces excessive ringing, even if the control input does not saturate;

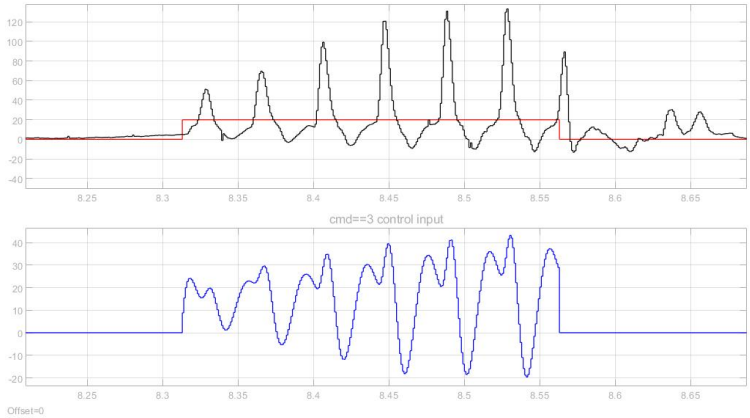

Figure 6.5: Increased MVR and OV (MVR>OV) weight conditions #1 profile control

- #4 Profile (Figure 6.6): the control algorithm provides good overshoot performance (near 30%) but it is unable to maintain the constant reference showing an excessive ringing.

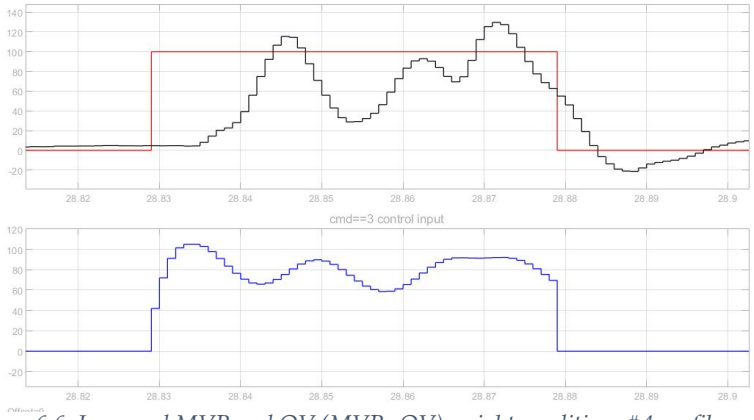

Figure 6.6: Increased MVR and OV (MVR>OV) weight conditions #4 profile control

6.2.4 *Optimized MVR and OV weights control*

After a due number of experiments, the tuning of MVR and OV weights outputted the optimized parameters in Table 6.5. The output signal still does not have decent values of raising time, but the algorithm is now receptive to output optimization.

Parameter		Symbol	Value	Unit
Weights	MV	$R_u \epsilon \mathbb{R}^{1x1}$	0	
	MVR	$R_{\Delta u} \epsilon \mathbb{R}^{1x1}$	10	
	OV	$Q \epsilon \mathbb{R}^{1x1}$	~~5~~ → 1,5	

Table 6.5: MPC increased MVR and OV (MVR<OV) weight control parameters

In detail:

- #1 Profile (Figure 6.7): the control algorithm, whose primary focus now is still input rate optimization, shows more decent raising times than the experiment in Paragraph 6.2.2 but not excessive ringing as in the previous one;

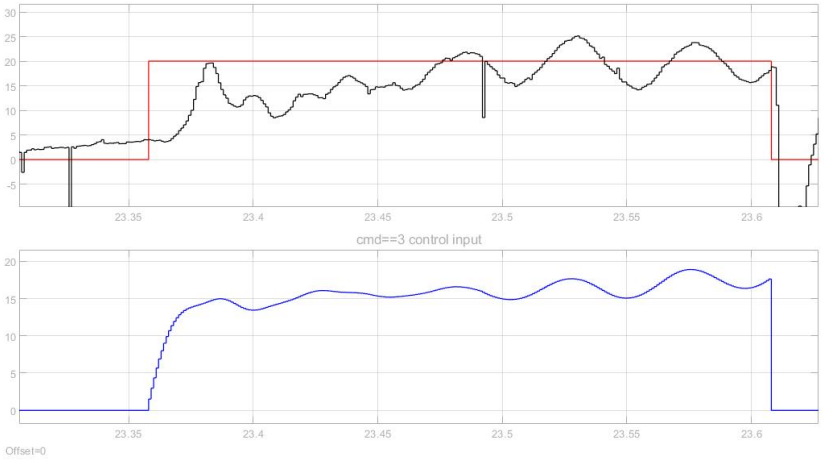

Figure 6.7: Optimized MVR and OV weights conditions #1 profile control

- #4 Profile (Figure 6.8): the MPC algorithm receptiveness has increased, but still cannot output a decent value of ringing.

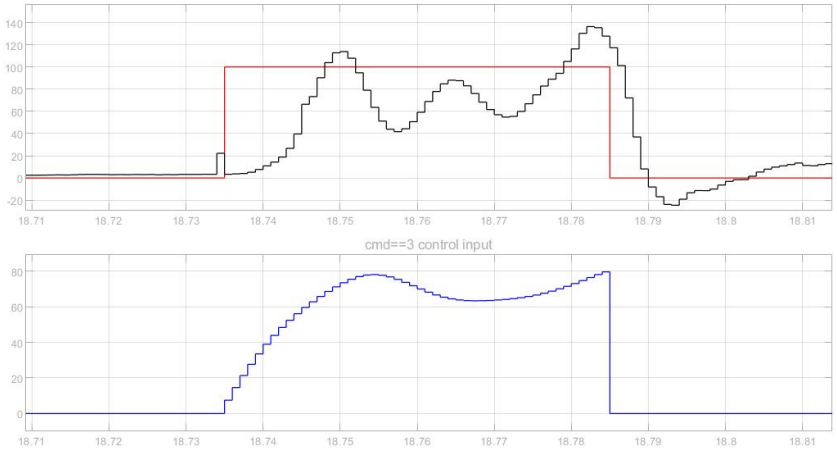

Figure 6.8: Optimized MVR and OV weights conditions #4 profile control

6.2.5 *Increased MV control*

Keeping the optimized MVR and OV weights found in the previous experiment, the Modified Variable (MV), i.e., control input, weight has been inserted according to Table 6.6.

Parameter		Symbol	Value	Unit
Weights	MV	$R_u \epsilon \mathbb{R}^{1x1}$	$0 \rightarrow 5$	
	MVR	$R_{\Delta u} \epsilon \mathbb{R}^{1x1}$	10	
	OV	$Q \epsilon \mathbb{R}^{1x1}$	1,5	

Table 6.6: MPC increased MV weight control parameters

The control input results to be too constrained to low values and cannot output decent force contact profiles, as shown in Figure 6.10 and Figure 6.9. The control is instead left with no significant modifications with lower MV weights.

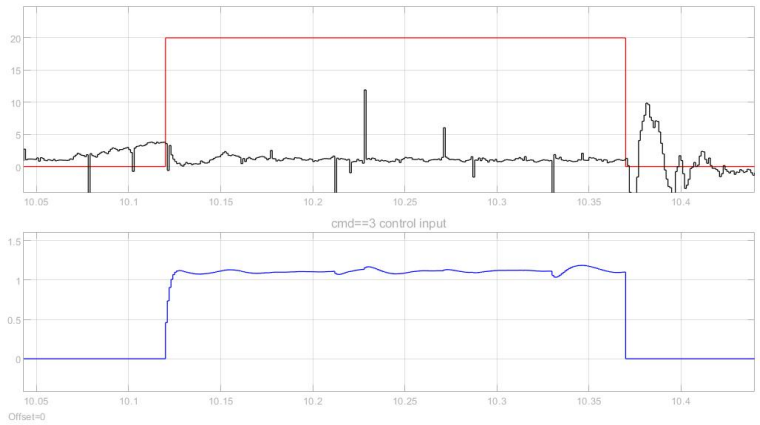

Figure 6.10: Increased MV weight conditions #1 profile control

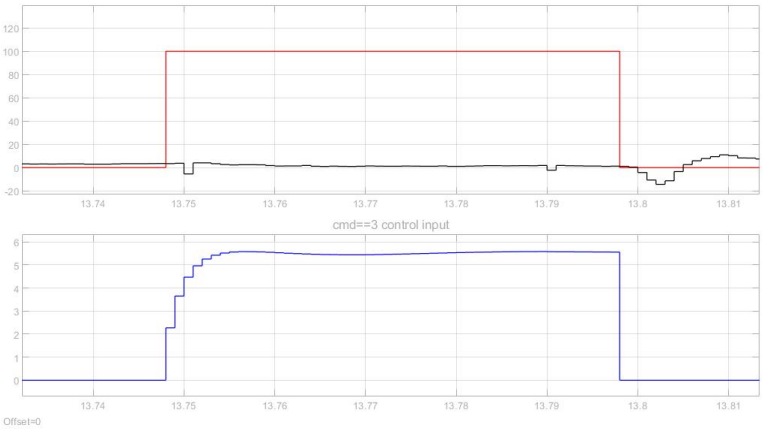

Figure 6.9: Increased MV weight conditions #4 profile control

6.2.6 *Increased Control and Prediction horizons control*

As the MPC algorithm's cost function weights have been optimized as in Paragraph 6.2.4, the following experiment was performed with longer Prediction and Control horizons as in Table 6.7. This increases the algorithm prognostication, but does not seem to increase its performance, mainly due to the gap between the modelled and real system.

Parameter	Symbol	Value	Unit
Prediction horizon	H_p	~~10~~ → 20	steps
Control horizon	H_c	~~10~~ → 20	steps
Control input constraint	$u_{max} = -u_{min}$	105	N

Table 6.7: MPC increased Control and Prediction horizons conditions parameters

In detail:

- #1 Profile (Figure 6.7): the MPC algorithm shows bad performance with unreached reference value;

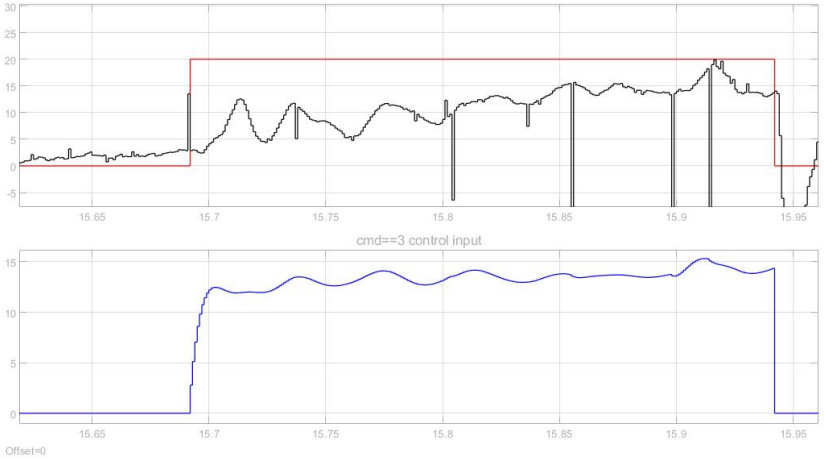

Figure 6.11: Increased Control and Prediction horizon conditions #1 profile control

- #4 Profile (Figure 6.8): control algorithm shows the behavior as in Paragraph 6.2.4, with no performance benefits.

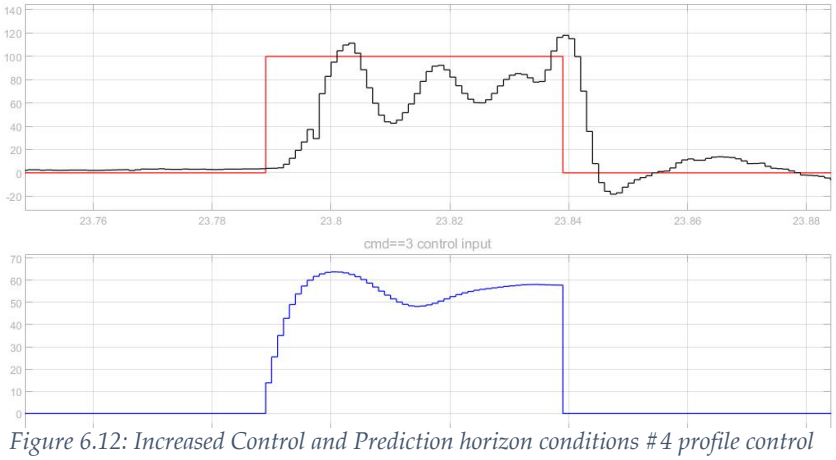

Figure 6.12: Increased Control and Prediction horizon conditions #4 profile control

6.2.7 *Decreased Control and Prediction horizons control*

The following experiment was performed with decreased Control and Prediction horizons values as in Table 6.8.

Parameter	Symbol	Value	Unit
Prediction horizon	H_p	~~10~~ → 8	steps
Control horizon	H_c	~~10~~ → 8	steps
Control input constraint	$u_{max} = -u_{min}$	105	N

Table 6.8: MPC decreased Control and Prediction horizons conditions parameters

The control algorithm results to be unaware of the system dynamics, outputting wrong contact force profiles as shown in Figure 6.10 and Figure 6.9.

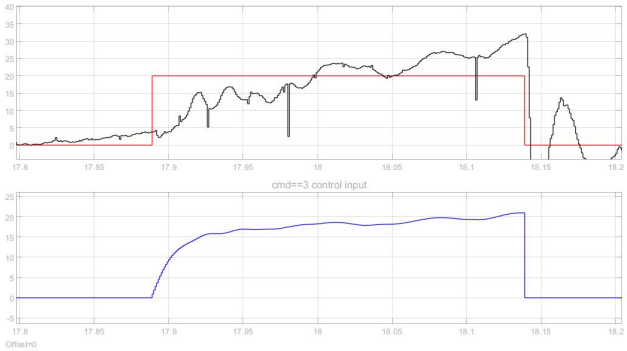

Figure 6.14: Decreased Control and Prediction horizon conditions #1 profile control

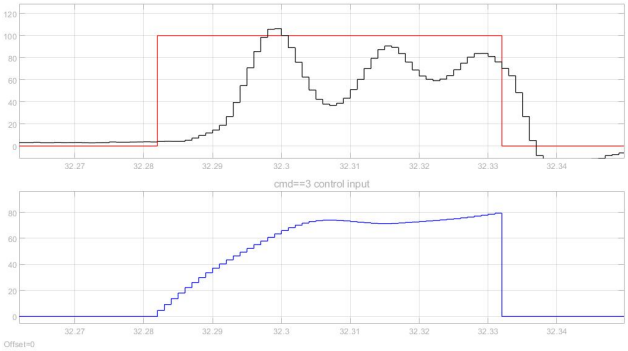

Figure 6.13: Decreased Control and Prediction horizon conditions #4 profile control

A similar behavior is shown also by decreasing only the Control horizon.

6.3 Final considerations

The solution found in Paragraph 6.2.4 is found as the most performative in terms of ringing, overshoot and raising time. Due to the fact that the introduction of a decreased Control horizon leads to worse performance, the parameter is removed. As one can note, the gap between Model-in-the-loop testing, described in Paragraph 3.3.3, and the optimized solution is large. This means that the non-linearities play a significant contribution in the phenomenon as expected. Nevertheless, the linear MPC algorithm outputs decent overall performance and can be assumed as a first valid control strategy.

Table 6.9 shows the following performance indices for the MPC algorithm tuning:

- Raising time, from profile reference issuing to 90% of the reference amplitude;
- Maximum force amplitude oscillation.

Experiment	Profile	Raising time [ms]	Maximum oscillation [N]
Default conditions	1	18	220
	4	10	120
Increased MVR weight	1	143	4
	4	17	60
Increased MVR and OV weights	1	10	145
	4	14	85
Optimized MVR and OV weights	1	22	10
	4	13	81
Increased MV weight	1	n. c.	2
	4	n. c.	n. c.
Increased Control and Prediction horizons	1	221	6
	4	11	68
Decreased Control and Prediction horizons	1	121	6
	4	16	67

Table 6.9: MPC tuning performance indices

Chapter 7: Conclusions and further developments

The PGAS project involving the fourth device led to significant design improvements in flexibility and control implementation. As the multi-domain actuation system was simplified by pneumatic circuit's removal, the uncertainties and nonlinearities involving pressured air dynamics were eliminated. The design choice to use a linear electric motor with respect to a rotational motor and rotational-to-linear mechanical coupling also reduced the nonlinearities and thermodynamic complications due to friction losses. Thus, the resulting plant system is subject to a vast simplification, which made it more suitable for the adoption of more refined control techniques.

As the main problem was the gap between the real system and the linear modeled one, the Model Predictive Control architecture, even if designed from a linear case, proved to be sufficiently reliable, at least in the test bench phase. As a matter of fact, the control algorithm outputted decent step responses in terms of transient, considering the magnitude of the nonlinearities involved and the fact that impulse duration must be no longer than 250 ms.

Another notable beneficial feature for the PGAS operation is the MPC algorithm's high flexibility through straightforward plant system data modification. As the MPC algorithm bases are computed at every *Simulink* build, and the control technique directly inputs the plant system, case-by-case data insertion does not require any effort and could be implemented in the user interface. In this way, focusing on the PGAS final task, not only the subject's but also the operator's parameters in the handled configuration could be inserted.

On the other hand, the algorithm still lacks an adequate uncertainty and noise damping structure, with ongoing problems in terms of control robustness. A future solution could be the entire redesign of the model with the introduction of nonlinear equations involving the actuator-body impact and the parallel design of Nonlinear Model Predictive Control (NMPC) strategies.

Control input saturation still is a central problem involving the control architecture. As prominent in the algorithm tuning various experiments, the actuation is directly aided by the mechanical inertia components generated during the Approach phase but still risks important saturations, especially during elevated force references issuing. According to this, serendipitous control approaches cannot be adequately implemented, while cautious approaches cannot output adequate responses due to the nature of the phenomenon and the reference profiles' typology.

Another source of disturbance is given by the elevated electromagnetic noise from the load cell device. A further development, which was discussed during the research as a viable solution, could be the implementation of the load cell directly on the target body. This could produce a significant noise removal enhancement, but at the expense of the system flexibility, involving the fact that the operator would be forced to strike on a particular target, fastened on the subject's study point through proper belts. Other than flexibility losses, this solution might lead to more biased results due to target points discreteness and patient strike unconscious preparation through muscle stiffening.

Future developments before operator-in-the-loop and patient-in-the-loop testing could also involve redesigning the test bench by making it much more similar to the final system by implementing a more fitting system for the body representation, like an inverse pendulum device. As a matter of fact, Taylor simplification could have led to the introduction of notable system uncertainties that cannot be appreciated during the bench testing due to its current morphology.

Assuming the bench testing implementation to be completed with all the previous hypotheses considered or implemented, handled configuration must be designed. In this regard, the PGAS fourth prototype overall dimensions and the double sided piston morphology represent a big design drawback with respect to its predecessors. In fact, handled configuration will require safety measures elements in order not to have any unwanted interaction between the operator and the back end of the rod during the retraction phase. A simple 3D printed back cap, covering the entire retracted position span, could fulfill the safety requirements, but overall compactness and weight must be first taken into account.

Ultimately, safety measures must be taken into account also in the control algorithm. Too big overshoot values must be promptly rejected, in order not to pose any potential hazard on both the operator and the patient to be studied. This could be implemented posing a output saturation in the MPC algorithm, with the drawback of burdening the calculus power, or in the Transition manager block using an upper contact force threshold value in order to switch from the Strike to the Retraction phase without reference profile accomplishing.

Appendix A: Charts

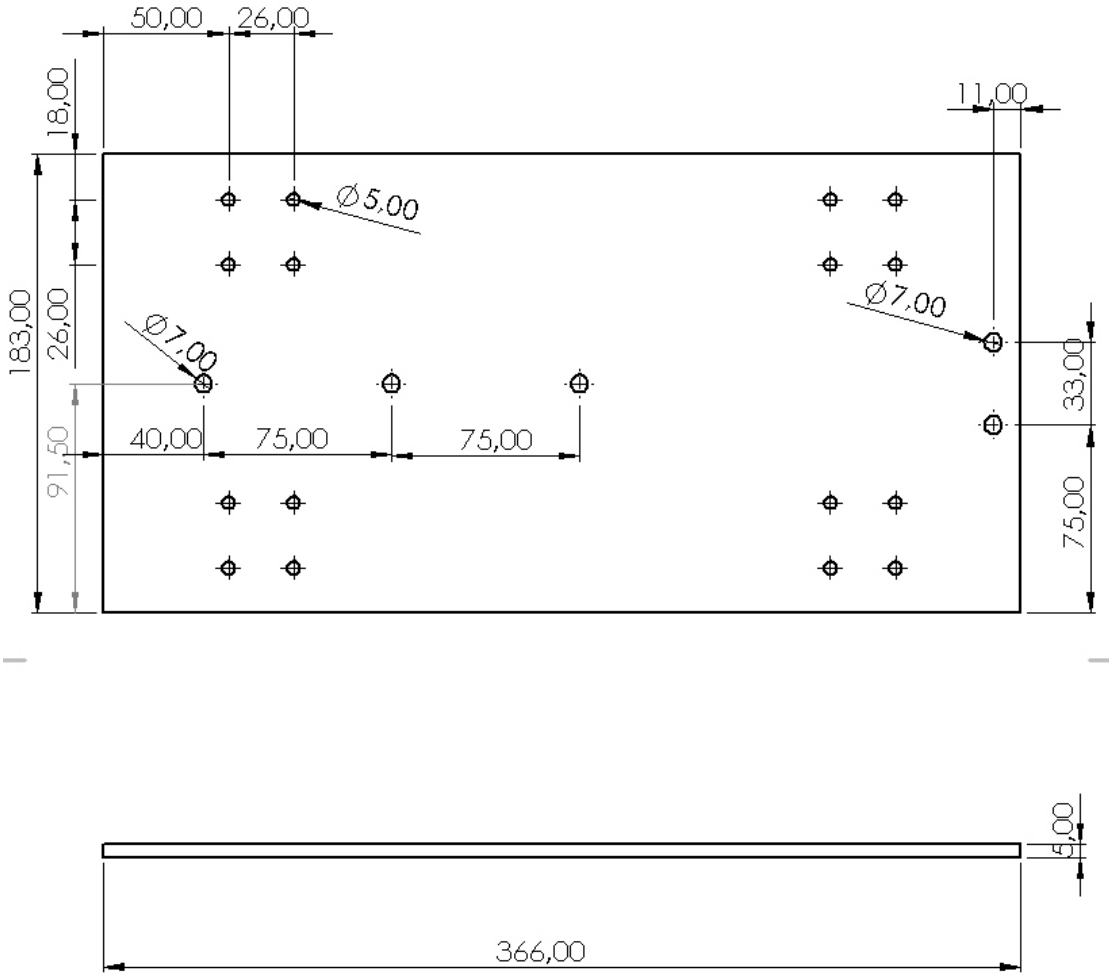

Chart 1: Actuator's sliding plate

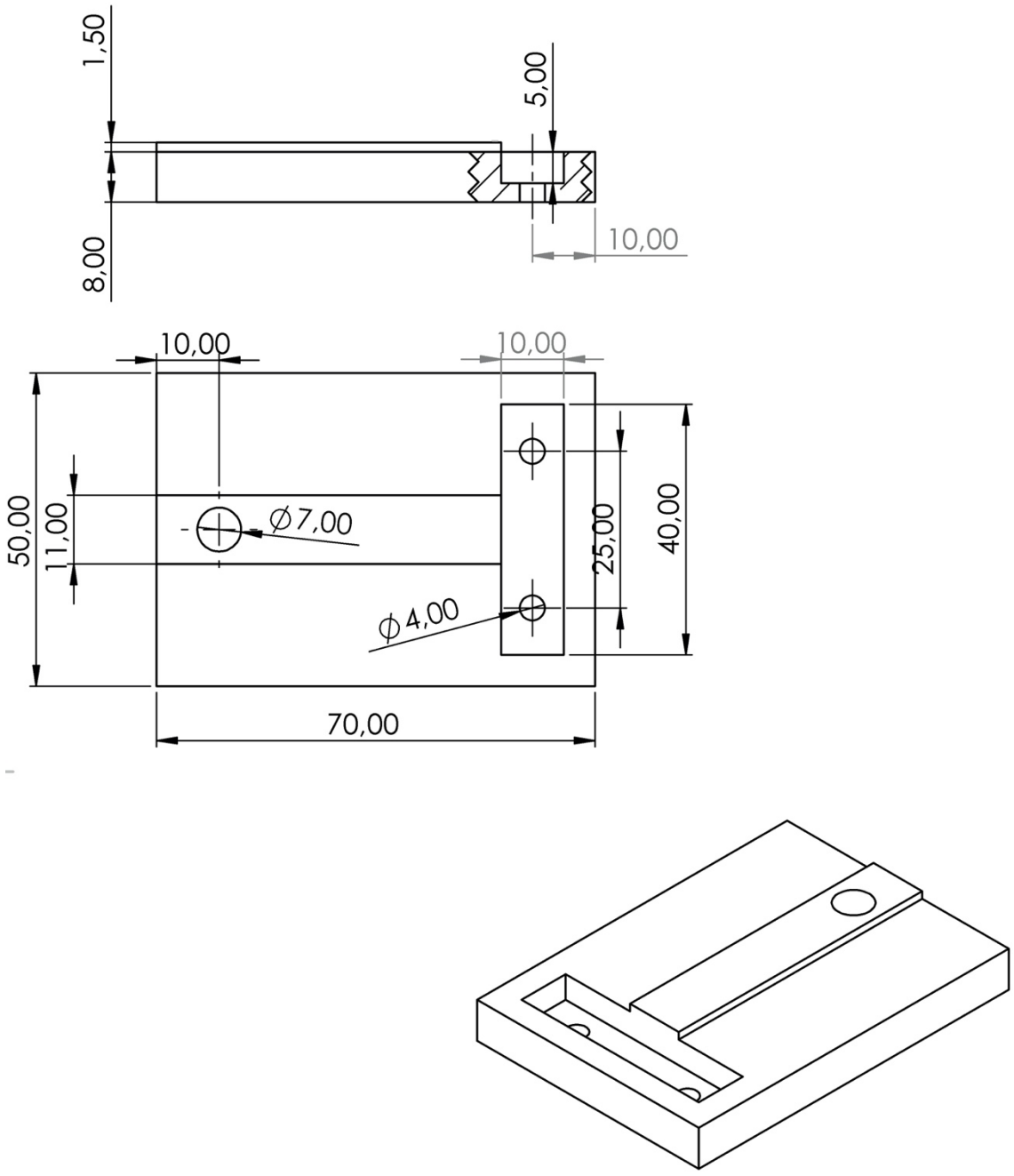

Chart 2: Optical Sensor support

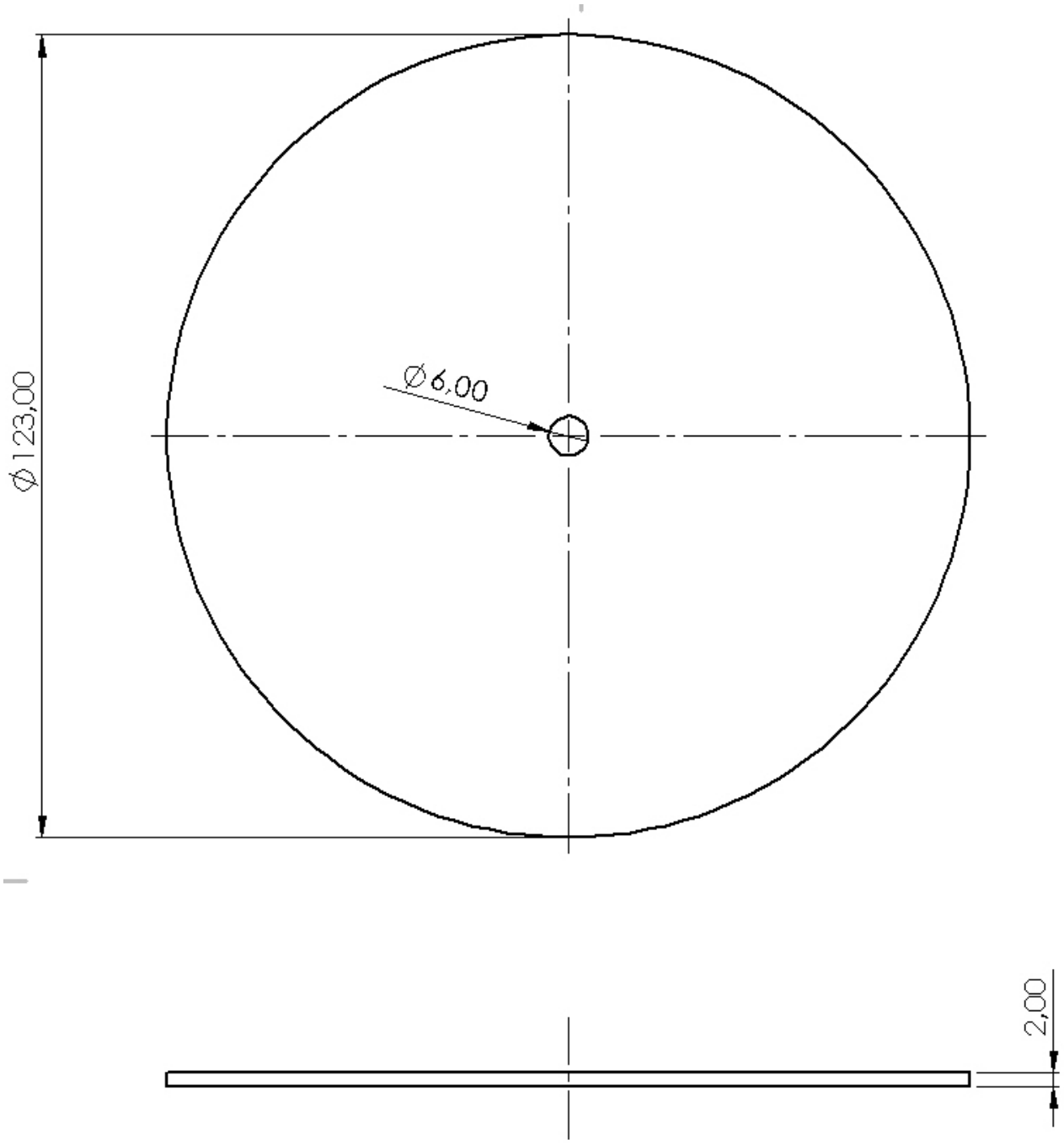

Chart 3: Optical sensor target

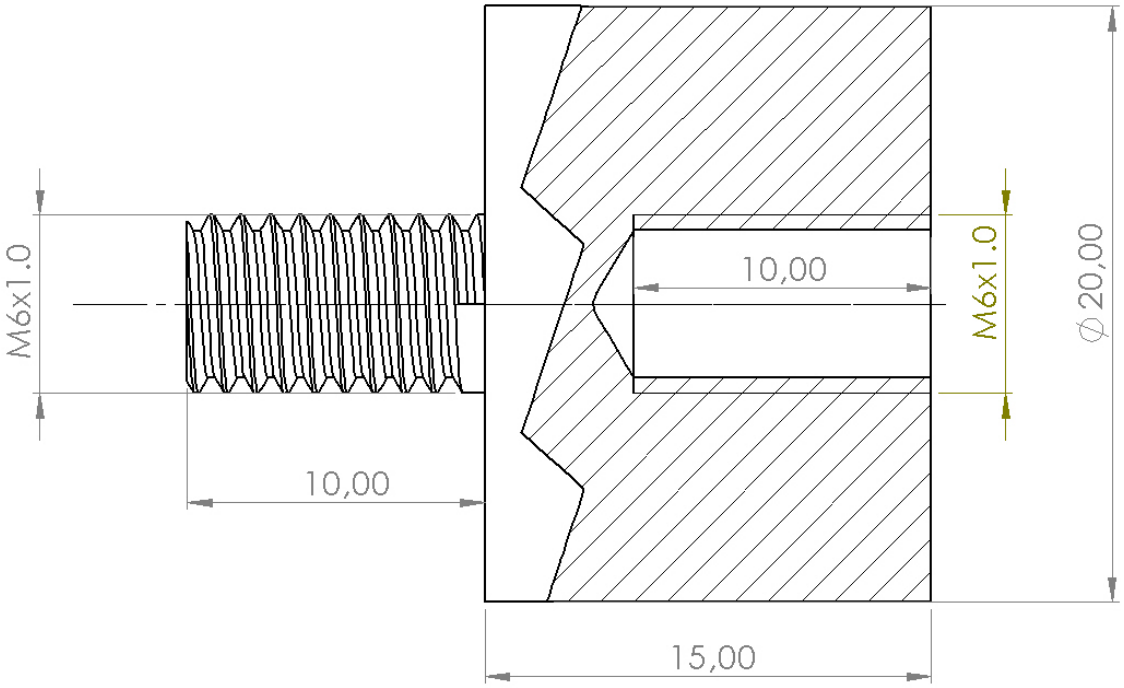

Chart 4: Load cell electrical insulation element

References

[1] M. Paterna, C. Ferraresi, S. Roatta, W. Franco and C. De Benedictis, "Caratterizzazione di un sistema di controllo posturale," *Tesi di Laurea Magistrale,* 2018.

[2] Z. Potocanac, J. de Bruine, S. van der Veenm, S. Verschueren, J. van Dieën, J. Duysens and M. Pijnapples, "Fast online corrections of tripping responses," *Experimental Brain Research,* vol. 232, pp. 3579 - 3590, 2014.

[3] A. Hof and J. Duysens, "Responses of human hip abductor muscles to lateral balance perturbations during walking," *Experimental Brain Research,* vol. 230, no. 3, pp. 301 - 310, 2013.

[4] A. Shahvarpour, A. Shirazi-Adl, H. Mecheri and C. Larivière, "Trunk response to sudden forward perturbations - Effects of preload and sudden load magnitudes, posture and abdominal antagonistic activation.," *Journal of Electromyography and Kinesiology,* vol. 24, pp. 394 - 403, 2014.

[5] A. Mansfield and B. E. Maki, "Are age-related impairments in change-in-support balance reactions dependent on the method of balance perturbation?," *Journal of Biomechanics,* vol. 42, no. 8, pp. 1023 - 1031, 2009.

[6] O. W. Pica, C. Ferraresi, D. Maffiodo, G. G. Muscolo and C. De Benedictis, "Sviluppo teorico e sperimentale di un perturbatore automatico per l'analisi del controllo posturale," *Tesi di Laurea Magistrale,* 2019.

[7] P. Flores and H. M. Lankarani, Contact Force Models for Multibody Dynamics, Springer International Publishing, 2016.

[8] D. A. Winter, Biomechanics and motor control of human movement, 2nd edition ed., New York: John Wiley & Sons Inc, 1990.

[9] R. J. Peterka, "Postural control model interpretation of stabilogram diffusion analysis," *Biological Cybernetics,* vol. 82, pp. 335 - 343, 2000.

[10] M. Canale, "Digital Control Technologies and Architectures slides," Politecnico di Torino, 2017.

Acknowledgments

The first big thank you goes to all my Supervisors and Eng. Carlo De Benedictis, for the generous, unwavering and precious support during all the research project, giving me the chance to improve professionally and learn from my own mistakes. I would like to extend my deepest gratitude to Eng. Lorenzo Chiavirano and Drss. Maria Paterna for the plentiful guidance and helpful advice on the system implementation and insight. I also wish to thank Eng. Maurizio Vallana, Mr. Luca Gambotto, and Mr. Giacomino Gandiglio, for the generous assistance and beneficial contributions.

A huge thanks to my friends that I met in Politecnico and in Turin: Giovanni, Dario, Sara, André, Matteo D. and Matteo G., Giada, Andrea P. and Andrea G., Enrico, Rosita, Valerio, Sole, Alexi, Tommaso, Genesis, Flavia, Davide, these three years wouldn't have been the same without your support and cherish. Many thanks to my old friends of Rome, that kept in touch with me and have been helping me from past to present: Giulio, Manuela, Stefano, Andrea, Adriano, Lorenzo, I will always think of you with sincere gratefulness.

Last but definitely not least, I'm extremely grateful to my family and Andrea, for standing to my side during this challenging phase. Your unparalleled care and love and profound belief in my abilities fueled me and proved a key role in carrying my research on, even during the quarantine. My heartfelt thanks.