UNIVERSITÀ DEGLI STUDI DI PADOVA

—

Dipartimento di Ingegneria dell'Informazione

—

MASTER'S DEGREE IN CONTROL SYSTEM ENGINEERING

CONTROL DESIGN OF A WEARABLE DEVICE FOR ARM REHABILITATION

Relatore: Prof. Ing. GIOVANNI BOSCHETTI

Laureando: GIOVANNI ANFODILLO

Matricola: 2091018

ANNO ACCADEMICO 2023-2024

ai miei genitori...

“ Al futuro o al passato, a un tempo in cui il pensiero è libero, quando gli uomini sono differenti l’uno dall’altro e non vivono soli. . . ”

George Orwell - 1984, 1949

Contents

Abstract

In this thesis it is explored the world of arm rehabilitation robots. In particular this project is specific on machines used for elbow rehabilitation for movements like flexion and extension.

Some of the reasons of why such machines exists are: better rehabilitation outcomes to lees workload of medical personnel.

This works also describes the state of the art of mechanisms used of elbow rehabilitation such as exoskeletons, which all have some specifics disadvantages such as lack of portability. Then it describes the current research on cable driven rehabilitation robots (CADEL), using cables solves the problems arisen with the exoskeleton variants.

This script also gives an in depth description of the hardware used and its mathematical models. Then this thesis gives a theoretical description of the control algorithms used for the project.

After the theory behind the controllers, some Simulink simulations are reported before tacking the actual tests using the real motors. All the results are the commented exploiting the data gathered during the tests.

Introduction

This works studies the robots used for elbow rehabilitation, focusing on the flexion-extension movement cycle.
With the growing interest of using robotics, various sectors ranging from industrial to domestic are experimenting with such technologies. One application which is seeing a lot of interest is the biomedical one, in which a lot of different robotic medical devices are researched and already used. An example of this are the cobots used for surgeries.
A particular approach to the medical use of robots is rehabilitation, which is a new field of application where this type of machine could have a great impact. This is due to the fact that often rehabilitation of joints requires a lot of repetitive movements for extended periods of time. Until now, this was carried out by physiotherapists, which requires a lot of time for each patient and with a high number of subjects this can become unbearable.
It is also important to note that a correct use of rehabilitation tools could make a difference between irreparable muscle damage and full joint motion recovery.

The first chapter describes the state of the art of mechanisms used of elbow rehabilitation. The fist examples of these machines are exoskeletons, which all have some specifics disadvantages like weight and lack of portability.
This chapter then describes the current research on cable driven rehabilitation robots (CADEL), showing how using cables solves some problems that are noticeable on the exoskeleton variants.
This chapter also confronts the different existing versions of the CADEL showing their performances and how the research advanced.

Chapter two explains the hardware choices for this robot. It describes in depth

all the major robots components and the sensors used for performance measurements.
This chapter then show the base for the control algorithm used for the machine, in particular the reference signal used by the feedback control.
Then in this part the previously mentioned hardware parts are described by the mathematical point of view and how their behaviour can be modelled during simulations.

Chapter three gives an in depth mathematical description of the control algorithms used in this project. In order it describes firstly PID control, then the eigenvalue allocation control lastly the LQR control. Each of these control design is studied in depth on how it is realized and its working principle.

Chapter four then gives an overview on the results, firstly on the simulation then the results achieved with the real motors. All the data from the experiment are then commented and studied to find the better control algorithm.

Chapter 1

Cable Driven Devices for upper limb motion assistance

In this thesis the main focus will be the rehabilitation of the elbow joint, one of the most important and complex joints which allows the flexion and extension of the forearm and make possible the pronation and supination of the hand and wrist. In fact, a limited range of motion of the elbow joint can severely hinder the independence of the patient in in his daily life and negatively effect his athletic abilities.

The use of this kind of robot would alleviate the workload on physiotherapists and it would also grant a reliable and customizable strategy to improve the accessibility of movement-based.

For such objective the idea was to design an extremely light and easily wearable robotic device fixed to the patient arm and based on a parallel mechanism driven by cables. The device designed is able to facilitate forearm and wrist movements thanks to the presence of an electro-mechanical support.

Studies show that immediate motion in pain-free regions of the movement can drastically reduce the negative effects of immobilization and stimulate the anti-inflammatory response[1]. This type of treatment is also beneficial to subjects who have experienced a long period of immobilization due to a major trauma. As a matter of fact low-load, long-duration treatments (**LLLD**) can help the patient by extending a limited range of motion (**ROM**) by producing a small deformation on the collagen tissue hence resulting in a muscle elongation[2].

As stated before these treatments require a lot of time and effort for both the patient and the caregivers so the main idea is to have a simple device to use that can be applied to the patient arm in order to allow him to perform joints movements for long sessions without the need of medical supervision.

This would allow a more prominent use of motion-based rehabilitation techniques with the result of yielding a better outcome for all types of patients[2]. In recent years, several medical devices for rehabilitation have been developed to favor the recovery of upper limbs.

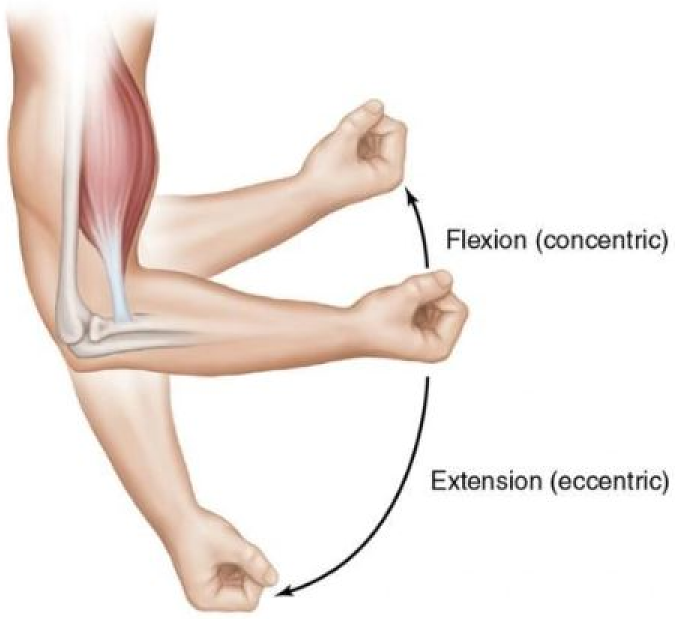

Figure 1.1: Elbow flexion movement pattern [3]

Figure 1.1 represents the movement pattern of the elbow flexion and extension. This movement is the focus of this project since it is fundamental for the patient's everyday life and its a key movement when considering patient's independence.

The first machines used for this goal have rigid links. An example of this is the chinese prototype called **MEDARM** [4], the **ExoRob**[5] or **NEUROexos**[6]. These robots work as a rigid exoskeleton, however they have three main problems: they are heavy, not portable and the moments could be limited by the rigid links.
Other possible realization of the same concept is an exoskeleton using an inflatable module to move the forearm [7].

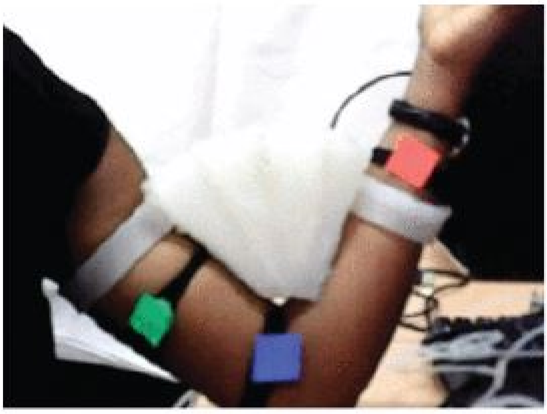

Figure 1.2: soft inflatable robot[7]

It is quickly noticeable that, while this soft inflatables realization solves the problem of portability it still is very cumbersome and not comfortable.
Using a cable based rehabilitation robot it is possible to solve the three problems mentioned before. Cables render the robot much lighter and smaller which in terms makes it more portable. Also cables being flexible, allow for larger range of motion.
To give a clear image of the work done in this project, it is important to have a brief discussion of the previous iterations of the prototypes used to tackle the problem of elbow rehabilitation:

1. `CADEL`

   The first prototype of a robot meant for elbow rehabilitation was developed at University of Tor Vergata.1.3 It consisted of a mechanism composed by two rings: the upper ring placed onto the patient's bicep, and the lower ring placed at the wrist.

These two platforms are connected by four cables: one in the front, one in the back and one on each side of the arm. The forearm platform also includes a plastic support where the patient can rest his hand.

This version however has several problems: first of all it is heavy for the patient, weighing about 20N(about 2kg)[8]; Secondly the platforms are difficult to secure to the arm making the solution more difficult to use. Eventually a badly fixed platform makes the motion non optimal.

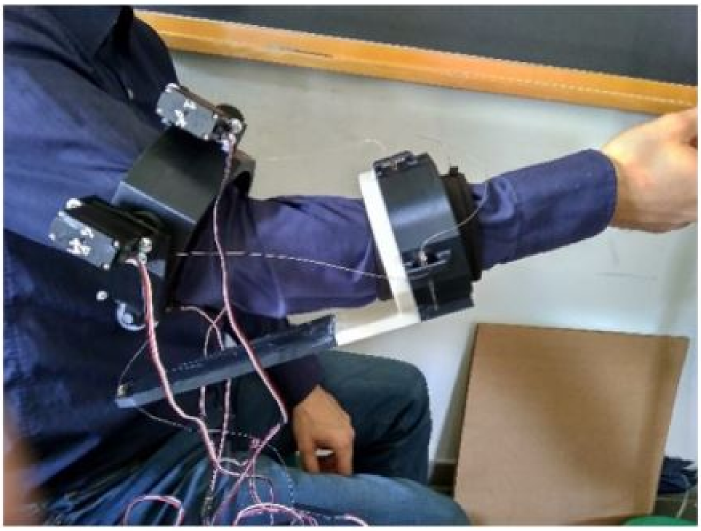

Figure 1.3: first prototype of CADEL[9]

2. `CADEL3`

   Starting from the `CADEL` prototype, a new prototype was designed at the University of Padova, called `CADEL3`.

   This new version of the robot aims to solve the two major problems previously highlighted. It is still a four-cable robot with the same configuration as before but in this iteration the platforms were re-designed.

   To fix the two rings to the arm a new approach was used: once the device is placed correctly onto the arm an inflatable padding would puff up wrapping the patient's arm and fixing the device. Then a sleeve-like cover was designed to help the patient correctly wear the device. An elbow guard with a pulley was used to keep the rear cable in the correct position1.4.

   This new version1.5 of the robot solves all the problems of the first version:

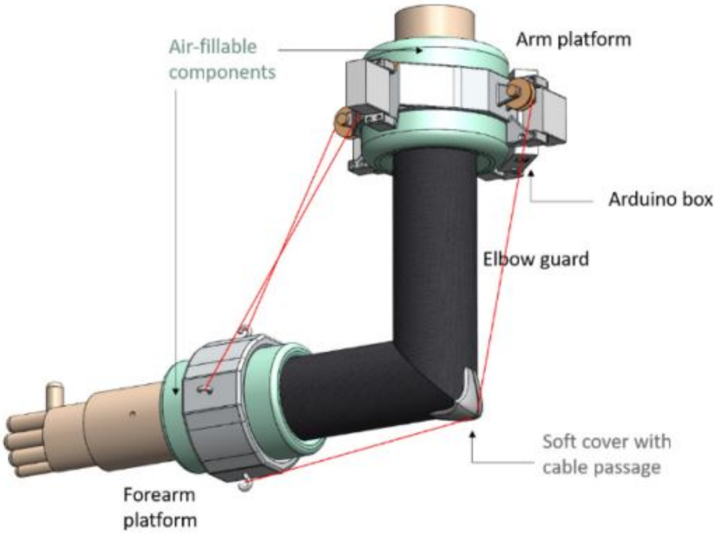

Figure 1.4: CAD model of CADEL3[8]

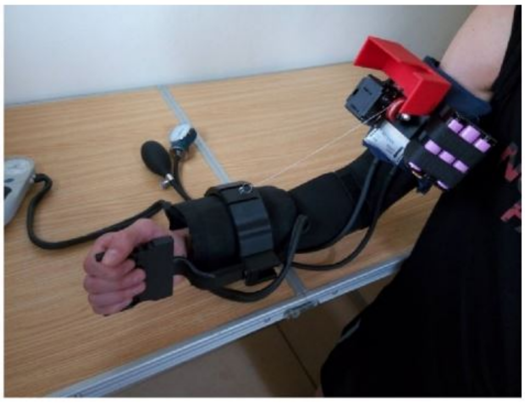

Figure 1.5: CADEL3[10]

it is lighter, more stable and easier to wear. However the elbow guard is still a concern since it could move during the exercise or the patient could offset it a bit making the rear cable working in a non-optimal way.
Moreover after measuring the torques on an extension/flexion cycle, the intervention of the two side cables has been found to be almost none except for the fact that they help keep the movement on the sagittal plane.

3. `l-CADEL v1`

   It was the first iteration of a different approach for a cable driven robot used for elbow joint rehabilitation. This model uses just two cables instead of the four cables used in the previous versions of the `CADEL` project1.6. This allows the new model to be lighter and cheaper. Moreover by using only the two lateral cables the problem of the wire passing behind the elbow is solved and hence it is not needed to design an elbow guard to keep the cable in the right position.

   This version still uses the two rings, one on the arm and one on the forearm fixed to the patient via inflatables1.6.

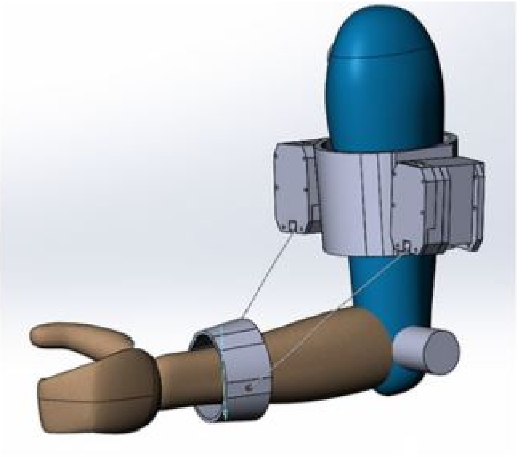

Figure 1.6: CAD model of `l-CADEL`[10]

This model does not need to have a sleeve to fit it on the patient arm and

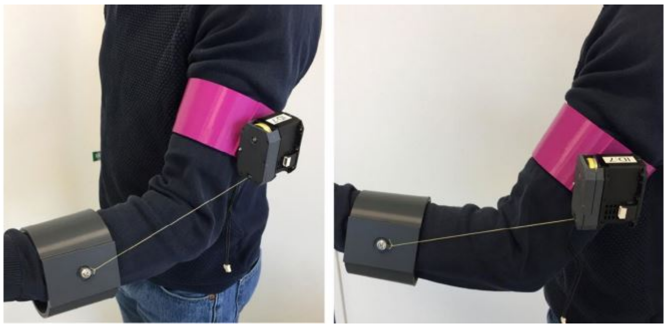

Figure 1.7: `l-CADEL`[10]

hence it is lighter and less clunky for the patient to wear. However, since the l-CADEL uses just two motors to perform the same movement it is necessary to have motors able to exert a bigger torque. Hence more powerful motors were needed for this version.

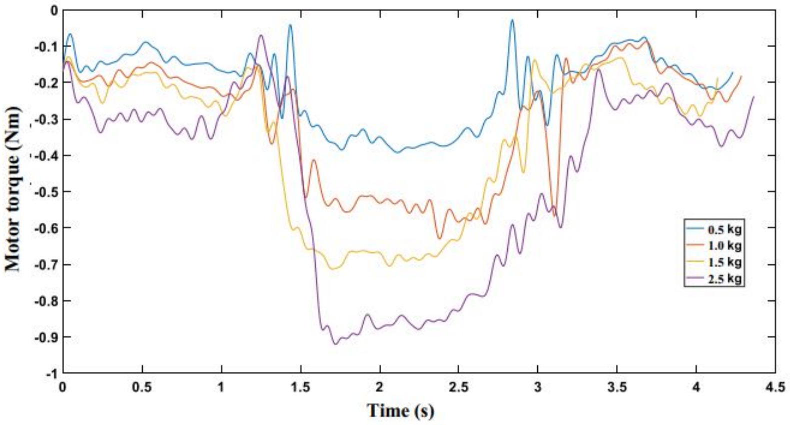

Figure 1.8: Torque graph of l-CADEL with different loads[10]

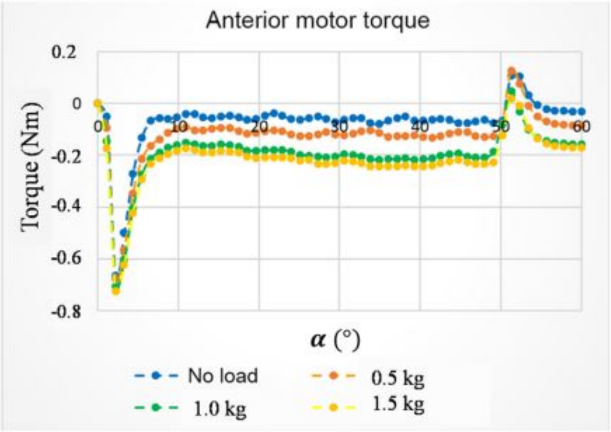

Figure 1.9: Torque graph of CADEL3[10]

In terms of power consumption, the four-cable version1.9 is slightly superior: by having four motors the load is more distributed with respect to the L-CADEL version1.8. We can also observe that the torques applied from the motors in this case1.8 are noisier which could represent an issue if the oscillations become too large. Otherwise in figure1.9 it is possible to observe that the torques have tighter oscillation, this is also due to both the structure of the robot and the presence of four motors.

4. `l-CADEL v2`

   This is the second version of the light `CADEL` robot1.10; this platform still uses two motors placed on the side of the arm and they are fixed to the patient via inflatables. A slight difference with respect to the previous version is the way the joint is moved. By using the glove and ring combination the device uses the wrist as second anchor point instead of the forearm. Indeed this device has a 3D printed U-shaped link fixed to a glove. The patient just needs to wear said glove to fix the device to his hand.

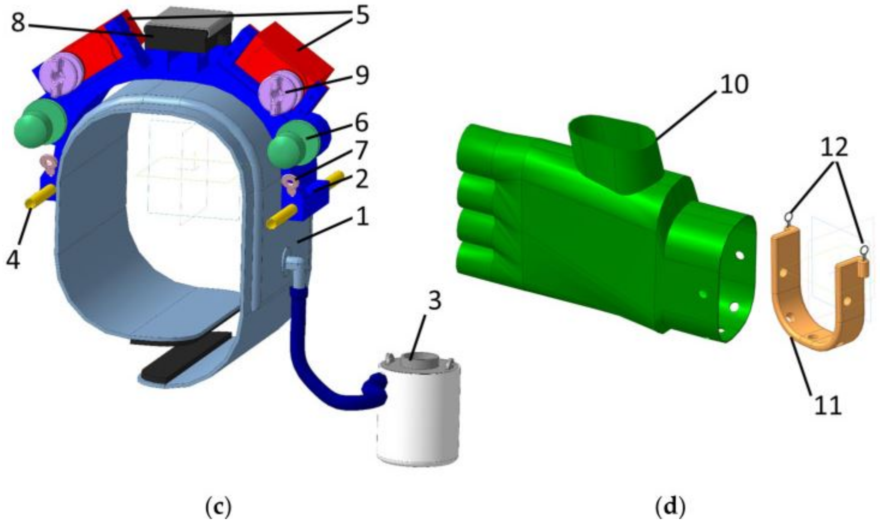

Figure 1.10: CAD model of the robot's platforms[11]

As shown in image 1.10 the arm platform (8c) is basically the same that it is used in the previous iteration of this project. However the second platform is very different: we can indeed observe the plastic ring (8d, the yellow coloured piece) and the glove used to fix the robot to the hand (8d, the green coloured piece). The lower ring provides for the two anchoring points (12 in Fig. 1.10 (c)) for the two cables responsible for the motion. The two exit points are placed instead in the upper ring (4 and 2 in figure 1.10 (d)).

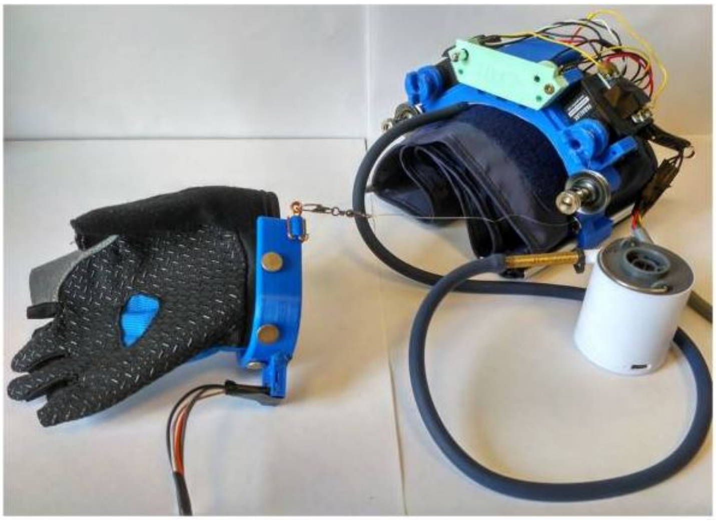

Figure 1.11: l-CADEL v2 real life prototype[11]

This model improves on various points from the first version: it is even lighter since the hand platform is composed only of a plastic ring and a fabric glove1.10.

This version still uses two motors so all the comparisons between the four-cable version and this one are the same as for the `L-CADEL v1.` Moreover, since the arm platform is conceptually identical to the one in the `L-CADEL v1` the motors used are the same.

This version is also equipped with a `IMU` sensor in order to read the angular flexion of the elbow during the movement1.12.

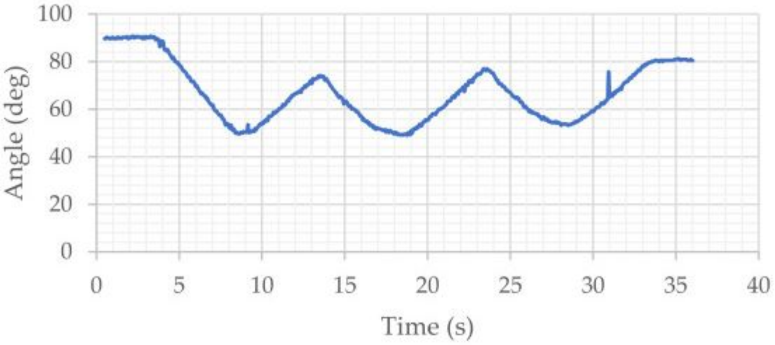

Figure 1.12: Elbow flexion profile[11]

In the study done on this version of the robot[11] the two cables are in a crossed configuration. This can help when performing pronation/supination movements of the wrist.

In this version the cable configuration could damage them with long repeated use. To solve this issue the prototype uses a particular set of cables called: dyneema fiber. This particular fiber has a much greater resistance to cut and abrasion(both in dry and wet condition).[12]

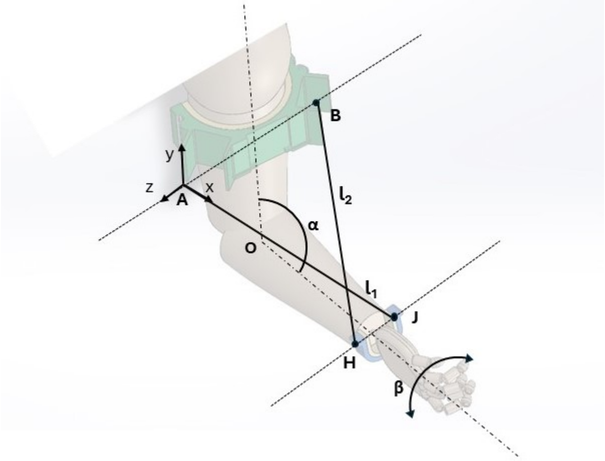

Figure 1.13: CAD scheme of the cable configuration

Since the focus of this thesis is on the flexion\extension, a symmetric cables version of this robot is designed.

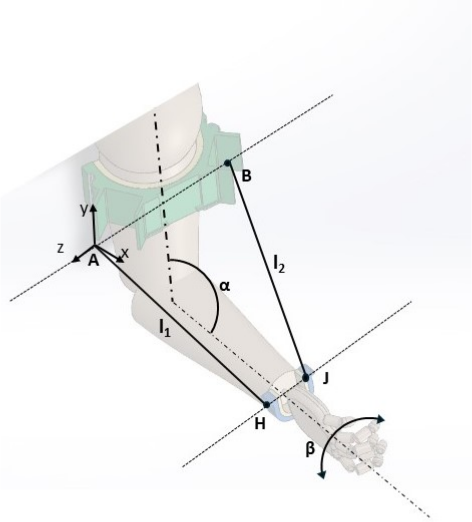

Figure 1.14: Kinematic scheme of the robot with symmetric cables

Chapter 2

Physical model

The first step to design a rehabilitation device for upper limbs is to define how the prototype has to fit the patient's arm. In this case it was decided to use a configuration similar to the l-CADEL v1 1.7 with a ring fixed on the bicep and one on the wrist of the user. In this configuration only two cables are needed, which allow the flexion/extension forearm movements and the pronation/supination of the wrist. In the upper ring there are the two exit points for the two cables. The lower ring provides for the two anchoring points at the wrist. Each cable is actuated by a DC motor placed onto the patient's shoulder by means of a third support 2.1. The batteries and Arduino board are also placed on the shoulder. This choice allows to put less strain on the patient's arm making the use of the robot more comfortable. Also having the components on the shoulder allows for greater range of motion.

2.1 Components

Before tackling the physical description of the problem, it is important to have a little detour on the major components used since the choices made have deep implications in the design:

1. DC motors

   The two actuators used for this project are a pair of Dynamixel XL430-

W250 motors that are supplied with 12V. This choice was done by following three main objectives:

- provide a sufficiently high torque needed to move the arm;
- lightness of the entire device both for patient's sake and also for easier structural design. Indeed each motor weight $0.057kg$.
- it is important for the motors to be small enough to not hinder motion during the exercises.

2. Microcontroller

   To implement and execute the control routine, it can be used an Arduino mega 2560 board which can be easily interconnected with other software interfaces such as the Matlab and Simulink developing environment. Furthermore, it is also a cheap yet powerful option to deploy the control algorithm directly on the medical device.

3. Motor driver

   To drive the two DC motors it is used a Dynamixel motor shield XL430-W250 which is compatible with Arduino and can drive up to 16 DC motors.

4. Sensors

   In terms of sensors both motors are equipped with a contactless absolute encoder. These encoders are equipped with a sophisticated slotted pattern on the rotating part, which allows the sensor to associate a specific code to each position. The encoders output the angular position of the motors, based on the signal read by the photo-detector. In order to exploit these position information, a position control algorithm is used to control the motors.

   A 9 DoF IMU tracks the position of the patient's hand. This sensor is composed by three parts: an accelerometer used to measure acceleration on the three axis, a gyroscope to measure orientation and a magnetometer to measure gravitational force. By using these sensors the IMU is able to measure velocity, orientation and gravitation force on the hand.

   It is also present a BITalino EMG sensor to measure muscle activation during the execution of the movements. These sensors measure the electric potential difference generated by muscle cells when stimulated. In this case it is used a surface EMG. This type of measure is easier to perform and less invasive for the patient, since it uses sticky pads to perform the measurements.

2.2 Device design

To better understand how this device is structured and how it operates in a flexion/extension task, it is considered Fig.2.1:

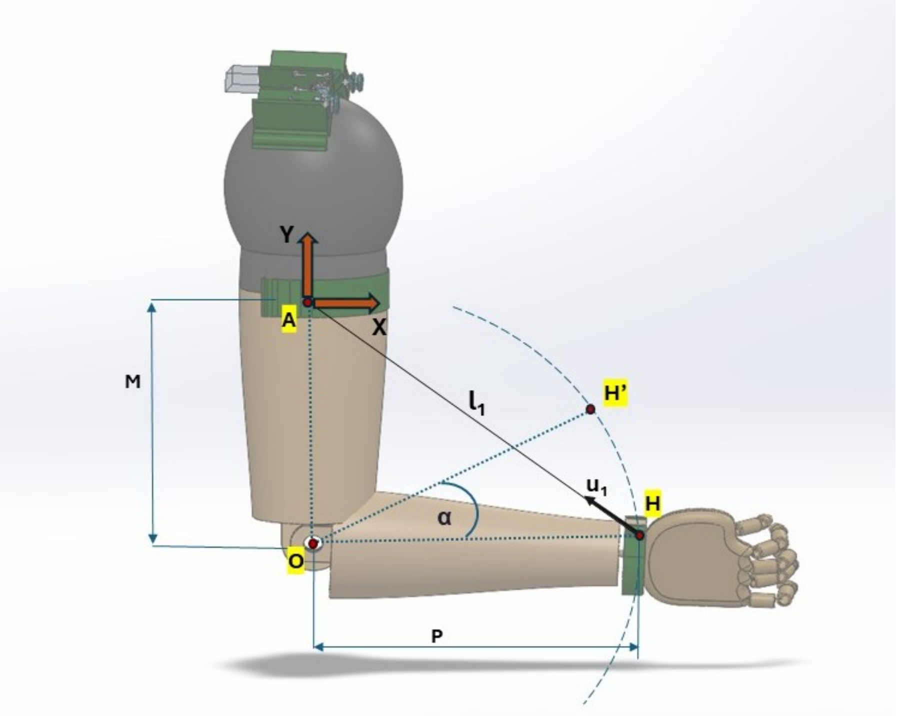

Figure 2.1: A kinematic scheme of the proposed design

During the motion point H describes a circumference with radius P and center O. Where L is the forearm length. The cable l1 is wound around the pulley at point A which constitutes the exit point of the wire. It is then anchored at point H on the wrist. The pulley is actuated by a DC motor which is necessary to vary the length of the cable. For the flexion movement the motor winds the cable, hence

pulling the hand. To execute the extension motion instead, the arm exploits the gravitational force and hence the motor unwinds the cable.
An important parameter to keep in mind is the cable tension. To correctly perform the movement the cables must be pulled enough not to be slack but not too much to avoid braking them[12].

2.3 Trajectory planning

It is planned a three-phase trajectory for the flexion/extension movement of the forearm as shown in Fig.2.2.

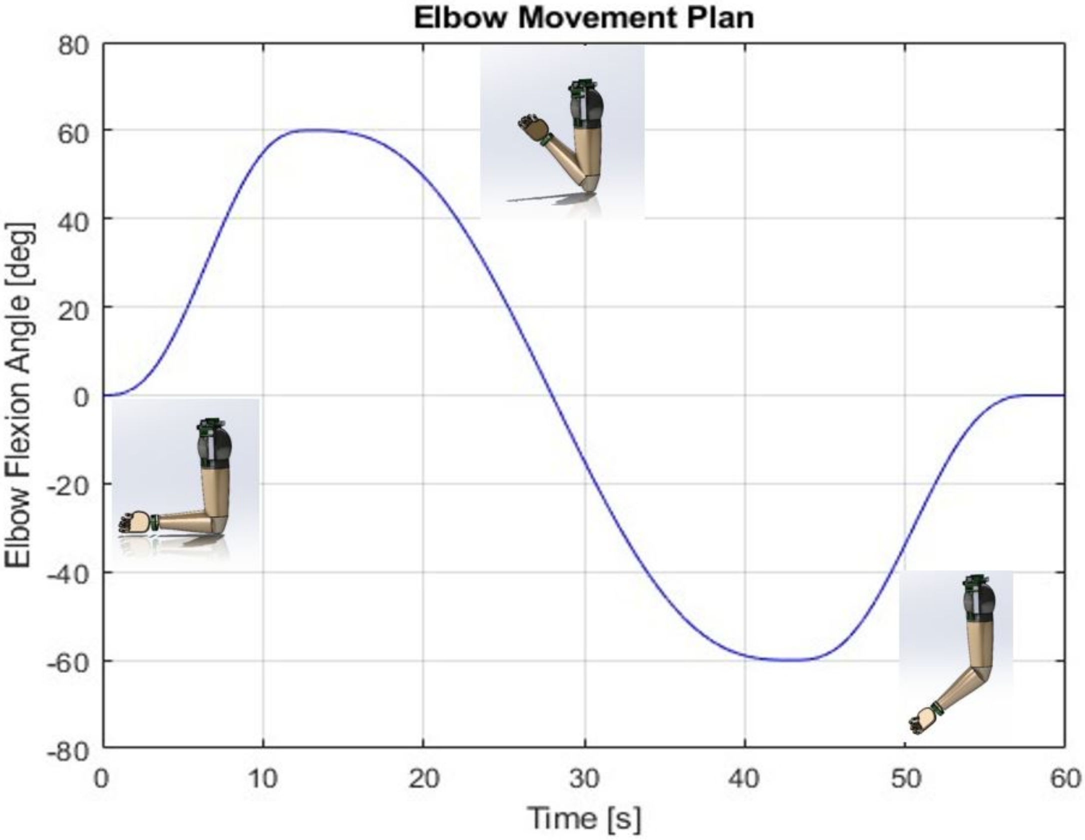

Figure 2.2: alpha trajectory during elbow flexion\extension

The motion is associated with a periodic winding and unwinding of the cables. The motor shaft angular position reference signal can be obtained by using a fifth-grade polynomial which ensures a smooth trajectory. Considering Fig.2.1, the origin of the reference system is placed on point A. We can define α as the angle between the forearm and the x-axis, which is computed using a fifth order polynomial function. This path is obtained using a custom Matlab script, which computes the coefficients for a fifth grade polynomial using the starting point, the ending one and the movement duration. Then the position of H is obtained with the following equation(2.1):

$$H = [L * \cos\alpha; -(M - L * \sin\alpha)] \tag{2.1}$$

Now that H is computed by using the trigonometric equations(2.2),(2.4) is possible to obtain the motor shaft position θ at every instant.

$$l = \sqrt{(x_H - x_A)^2 + (y_H - y_A)^2} \tag{2.2}$$

$$\theta = \left(\frac{l}{R}\right) - \theta_{in} \tag{2.3}$$

The resulting trajectory plan for the motor shaft is represented by in the folowing plot:

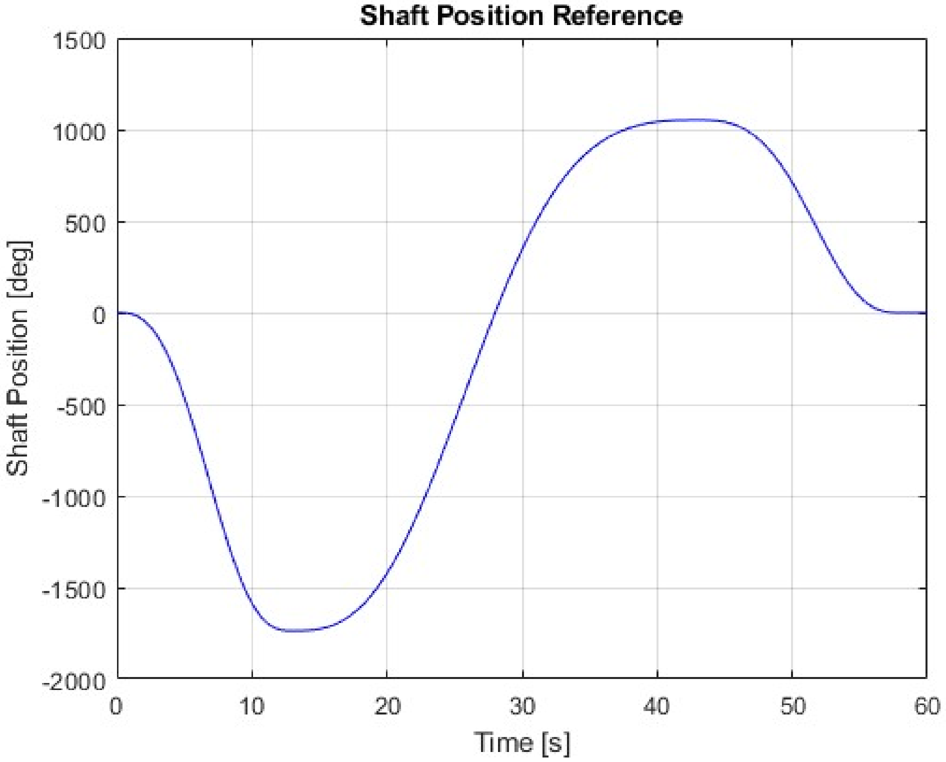

Figure 2.3: Motor shaft trajectory plan (θ) during elbow flexion\extension

2.4 DC motor model

Now that the angular position of the motors 2.3 is well defined, the second step is to correctly model the plant, which in this case is composed mainly form the DC motors and their driver. During the simulations it is possible to consider just one motor since in practice the two motors always work together and hence they have the same control input.

2.4.1 Motor Driver

The first element to model is the motor driver, this component is mainly a non-inverting input OP-amp 2.4.

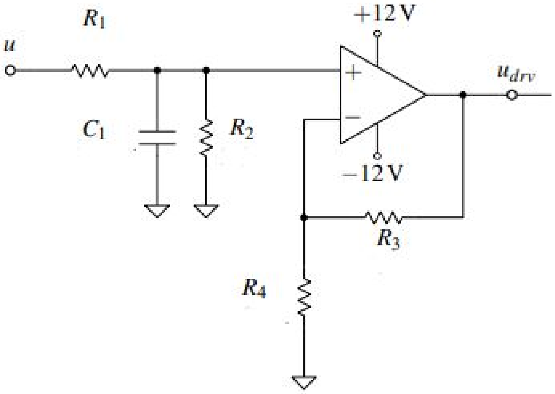

Figure 2.4: Circuit scheme of motor driver[13]

Based on the driver schematics 2.4 it is possible to obtain the following equations:

$$i_1 = C_1 \frac{du_+}{dt} \tag{2.4}$$

$$i_2 = \frac{u_+}{R_2} \tag{2.5}$$

$$u_+ = u - R_1(i_1 + i_2) \tag{2.6}$$

Where u_+ in equation (2.4) is the input voltage of the OP amp. We can define i_1 and i_2 as the currents flowing in R_1 and R_2.

By combining equations (2.4), (2.5), (2.6) the result, after some algebra, is:

$$\left(\frac{R_1 R_2 C_1}{R_1 + R_2}\right)\frac{du_+}{dt} + u_+ = \left(\frac{R_2}{R_1 + R_2}\right)u \tag{2.7}$$

Where u_+ is the input voltage at the OP amp. Equation (**??**) is the first step to link the input voltage u to the output voltage u_{drv} Also, since the OP-amp is in non-inverting configuration, the following relation stands:

$$u_{drv} = (1 + \frac{R_3}{R_4})u_+ \tag{2.8}$$

After defining k_{drv} and T_{drv} as follows:

$$k_{drv} = \left(1 + \frac{R_3}{R_4}\right)\frac{R_2}{R_1 + R_2} \tag{2.9}$$

$$T_{drv} = \frac{R_1 R_2 C_1}{R_1 + R_2} \tag{2.10}$$

Combining equation (2.8) and (2.7) the result is:

$$T_{drv}\frac{du_{drv}}{dt} + u_{drv} = k_{drv}u \tag{2.11}$$

When moving into the frequency domain equation (2.11) becomes the transfer function of the voltage driver from u to u_{drv}:

$$u_{drv} = \frac{k_{drv}}{T_{drv}s + 1}u \tag{2.12}$$

2.4.2 DC motor model

To better understand the motor model it is possible to divide it into its mechanical part and its electrical part.

The first part is described by the following equation:

$$I_m\ddot{\theta}(t) + F_m\dot{\theta}(t) = C_m(t) - C_r(t) \tag{2.13}$$

Where:θ is the motor shaft position, I_m is the motor inertia, F_m is the viscous friction coefficient, C_m is the motor torque and C_r is the resistant torque of the load.

By moving equation 2.13 to the Laplace domain it is possible to obtain the resulting transfer function:

$$\Theta(s) = \frac{C_m(s) - C_r(s)}{I_m s^2 + F_m s} \tag{2.14}$$

This transfer function links the motor torque to the shaft position and it is the first step to obtain the whole model of the DC motor.

To compute the motor torque it is necessary to analyze the electrical part. It is important to remember that a DC motor works by having a rotor, which is fed by the armature current i_a, which is the current flowing in the rotor windings. These windings are subject to a magnetic field generated by permanent magnet in the stator.

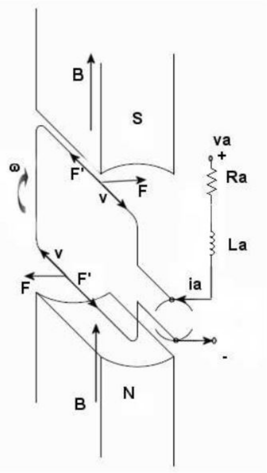

Figure 2.5: Electric scheme of a DC motor[14]

The motor torque is generated by the Lorentz law and, by inverting the current i_a it keeps moving when the winding and the magnetic field are perpendicular. Knowing the working principle it is possible to obtain the following equations:

$$u_a - e = R_a i_a + L_a \frac{di_a}{dt} \tag{2.15}$$

$$e = k_e \dot{\theta} \tag{2.16}$$

In equation 2.15 R_a is the armature resistance and L_a is the motor inductance, k_e is the back electromotive force constant. While e in equations 2.15 and 2.16 is the back electromotive force which is a voltage generated by the variation of the magnetic flux through the windings. The electromotive force is opposite with respect to the supply voltage and generates in the winding a current opposite of i_a, this is the reason why it has to be subtracted from the supply voltage u.

By moving equation 2.16 in the Laplace domain it is possible to derive the following transfer function:

$$I_a = \frac{U_a(s) - k_e \dot{\Theta}(s)}{R_a + sL_a} \tag{2.17}$$

This transfer function links the armature current, which is linked with the supply voltage by (2.15), and the motor torque C_m.
Finally by combining 2.14 and 2.17 it is possible to represent the full block scheme for a DC motor:

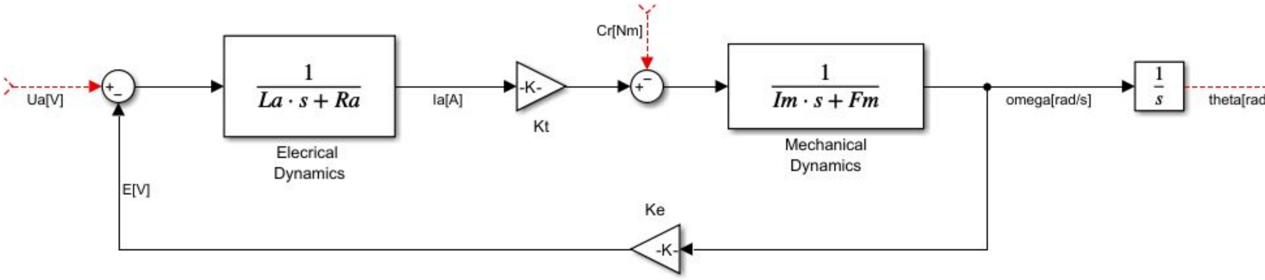

Figure 2.6: full block scheme [14]

In figure 2.6 it is possible to observe that there are two gains not present in the transfer functions: k_e which is present in (2.16). k_t is the torque constant of the motor, which links armature current and motor torque as follows: $C_m = i_a k_t$.
To simplify the notation of the plant transfer functions it is possible to define these two terms:

$$T_m = \frac{R_a I_m}{R_a F_m + k_t k_e}, \qquad k_m = \frac{k_{drv} k_t}{N R_a I_m} \tag{2.18}$$

Hence the plant behaviour can be described by the following equations:

$$\begin{cases} \dot{\theta} = \omega \\ \dot{\omega} = -\frac{1}{T_m}\omega + \frac{k_m}{NT_m}u \end{cases} \tag{2.19}$$

Chapter 3

Control design

This robot has the main goal of making repeated and controlled movements. The best way to achieve this objective is to implement a control algorithm. The use of a controller allows the robot not only to follow a precise desired behaviour, but also to work in an autonomous manner.

Control algorithms work by having a reference signal to follow, in this case the possible reference signal could be on the shaft position θ, on the shaft speed ω or on the supply voltage of the motor u_a. In this thesis it was chose to develop a position feedback control using θ as a reference signal as anticipated in 2.

To determine what represents a good closed-loop response it is necessary to introduce some specifications for the controller. Starting by the most important, the stability specification: the controller must ensure that the system is asymptotically stable. Or at least BIBO stable, where BIBO stands for Bouded Input Bounded Output, meaning that a bounded input must trigger a bounded response from the system.

A possible way to describe the system behavior is to use its transfer function. This function describes the behaviour of a system in the frequency domain by linking the input and the output.

In linear time-invariant (LTI) systems the transfer function is defined as the Laplace transform of the impulse response.

Note that these functions must be at least proper, this because if the frequency tends to infinity a non proper transfer function grows unbounded. In figure 2.6 it

is possible to see the transfer function of the DC motor.
To study the system stability, starting from its transfer function, it is possible to exploit the following theorem:

Theorem 1. *A continuous time linear system is asymptotically stable if and only if all the poles of the closed-loop transfer function have negative real part, is BIBO stable if all the poles have negative or null real part[15].*

By using theorem 1 it is possible to determine if a controller stabilizes the system, however stability is not the only specification required from controllers. The second group of specifications is the time specifications, which can be divided into two subgroups: transient-state specifications and steady-state specifications. For what regards the transient-state requirements the three main specifications are:

- **Rising time** t_r: this parameter is an indication of the promptness of the system. It represents the time in which the system response varies form 5% of the reference to 95% of the reference value. This case is also known as rise time at 5%.

- **The settling time** t_s: is the time needed for the output response to reach and stay within a certain percentage (usually 5%) of the reference signal value.

- **Overshoot** M_p: is the second transient requirement and it is the percentage of which the system response exceeds the reference signal, this is also measured as a percentage of the reference value at steady state.

In terms of steady-state requirements the most important is the steady-state error which is the difference between the reference signal and the system output.

3.1 PID control

The most popular solution to solve this control problem is through the **PID** controller. The `PID` acronym stands for *Proportional Integral Derivative*, which relates to the control action being composed of three separate terms:

1. The proportional term which is represented by the gain K_p.This part of the controller is the one mostly responsible to lower the reference error during the transient. This is due to the fact that The proportional term mainly impact the cut-off frequency of the closed-loop system[16].

2. The integral term is represented by K_i. The most noticeable effect of this part is regulating the steady state error. However it may slow down the response. This because the integral term adds a pole in the origin[16].

3. The derivative term, represented by the gain K_d, has the effect of controlling the speed of the response: the higher the gain the faster the response, which could impact negatively on the overshoot. This effect is caused by the fact that this term impact the phase margin of the closed-loop system[16].

In the time domain the PID control algorithm is the following:

$$u(t) = K_P e(t) + K_i \int_0^T e(t)dt + K_d \frac{de(t)}{dt} \tag{3.1}$$

Where $e(t)$ is the reference error and T is the duration of the reference signal.
These three gains are the ones used to tune the controller in order to obtain a satisfactory closed-loop response.
Using the previously mentioned specifics it is possible to design a PID controller using the frequency response method.
The first step to incorporate the previously mentioned requirements in the control design is to check if the closed-loop system can be approximated to an easier one. When considering the model 2.6 with the voltage driver model (2.12), it is possible to note that, via dominant pole approximation, the closed-loop system can be approximated by a second order system of the type:

$$T(s) = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \tag{3.2}$$

Where ω_n represents the natural frequency of the system which coincides with the frequency of the dominant poles. This fact holds because, in time domain, the dominant poles represents the slower part of the response. Since the faster parts become zero quickly they can be neglected when considering asymptotic behaviour. While δ is the damping factor of the system: this factor directly impacts the response of the system in terms of oscillation and stability. Hence the PID control will also impact this term.

Exploiting the approximation of equation (3.2) and by having clear control specification it is possible to write the following expressions:

$$t_{s,5\%} = \frac{3}{\delta \omega_n} \tag{3.3}$$

$$M_p = e^{-\frac{\pi\delta}{\sqrt{1-\delta^2}}} \tag{3.4}$$

For a closed-loop system which admits a second order approximation the natural frequency ω_n is approximately equal to the gain crossover frequency ω_{gc} which is the frequency at which the absolute value of the open-loop system is equal to one.

Using this fact is possible to rewrite equation (3.3) as follows:

$$\omega_{gc} = \frac{3}{\delta t_{s,5\%}} \tag{3.5}$$

After some algebraic manipulations it is possible to modify (3.5) as such:

$$\delta = \frac{\ln M_p}{\sqrt{\pi^2 + \ln^2 M_p}} \tag{3.6}$$

In second order systems the phase margin ϕ_m represents how much the phase of the system can vary before the system becomes unstable. It is computed using the following formula:

$$\phi_m = \arctan\left(\frac{2\delta}{\sqrt{\sqrt{1+4\delta^4} - 2\delta^2}}\right) \tag{3.7}$$

Now that all the necessary parameters are well defined, it is considered the definition of a general PID controller:

$$C(s) = K_p + K_i \frac{1}{s} + K_d s \tag{3.8}$$

From equation (3.8) it is possible to see the three terms composing the controller.

It is important to note that the derivative term can be problematic since it is not strictly proper. To solve this problem the derivative term will be approximated by a second order filter of the type:

$$H(s) = \frac{\omega_c^2 s}{s^2 + 2\delta\omega_c s + \omega_c^2} \tag{3.9}$$

Where ω_c is the cut-off frequency of the filter, which represent the frequency at which the gain falls of -3dB from the passband response. While δ still represents the damping factor of the system.

By denoting the full plant 2.6 with $P(s)$ and the controller as $C(s)$ it is possible to write the open-loop transfer function as:

$$L(s) = C(s)P(s) \tag{3.10}$$

The three gains of the controller are computed to respect the specifics on the gain cross-over frequency (3.5) and on the phase margin (**??**) in order to have:

$$L(j\omega_{gc}) = C(j\omega_{gc})P(j\omega_{gc}) = e^{j(-\pi+\phi_m)} \tag{3.11}$$

From (3.11) by some algebraic manipulations it is possible to obtain the following formulation:

$$C(j\omega_{gc}) = \Delta K e^{\Delta\phi} \tag{3.12}$$

Where $\Delta K = |P(j\omega_{gc})|^{-1}$ and $\Delta\phi = -\pi + \phi_m - \angle(P(j\omega_{gc})$ Now combining (3.12) and (3.8) it is possible to obtain an identity, and by posing the equality between the real and imaginary part of the identity the following formulas are obtained:

$$\begin{cases} K_p = \Delta K \cos\Delta\phi \\ K_i = -\omega_{gc}\Delta K \sin\Delta\phi \\ K_d = \frac{1}{\omega_{gc}}\Delta K \sin\Delta\phi \end{cases} \tag{3.13}$$

By using (3.13), (3.5) and (3.7) it is possible to obtain a PID controller with desired specifications, if these are still not satisfactory these gains can be slightly adjusted via trial and error.

3.1.1 Anti-windup

Another simple solution to improve the PID control performance is to design an anti-windup mechanism. This slight modification to the PID control is useful in case the control input saturates the actuator. If saturation happens it is seen that the integrator value keeps increasing but the plant output is locked due to saturation of the system, this is called "windup". If this happens then the control input will continue to grow, without having an effect on the system since the actuator is saturated. The control error then has to switch sign for a long time to bring back the integration to its steady state. The result is a slower response and a higher overshoot[17]. In order to understand how the anti-windup mechanism works consider the following block scheme:

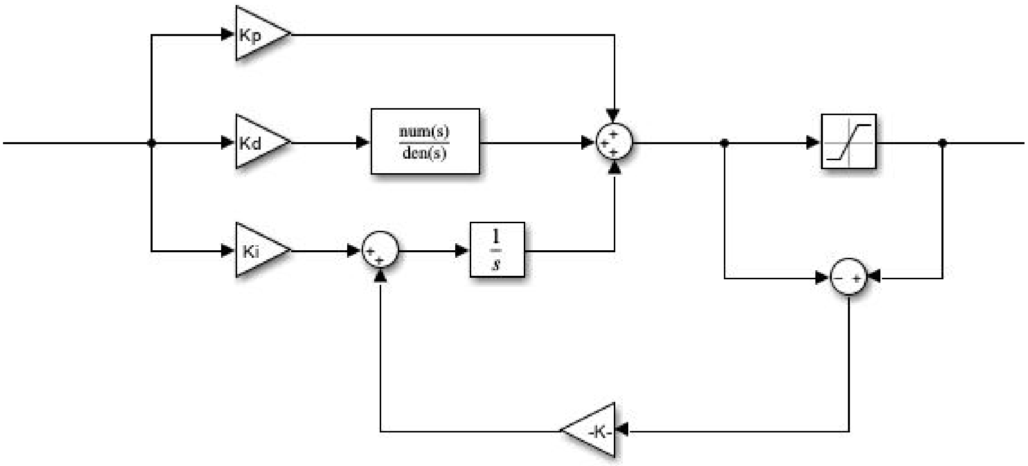

Figure 3.1: anti-windup Simulink scheme

In figure 3.1 after the PID scheme it is possible to note a saturation block followed by a gain block, which represents the anti-windup gain. The saturation block after the PID output shares the same saturation of the actuator. It is possible to see that if the control input is greater than the saturation value then the anti-windup output produces a negative value which is then summed to the integral part. This will reduce the control input preventing the integrator windup.

3.2 State-Space Control via eigenvalue allocation

3.2.1 State-space description of a system

This section introduces the state-space description of a system, useful for analyzing the system properties. This way of studying the system makes use of matrices to analyze the plant.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{3.14}$$

Where:

- $x(t)$ is the system state, this is a vector having some measurements of the system as coordinates. An example could be the state vector having position and speed as coordinates:

$$x(t) = \begin{bmatrix} \theta(t) \\ \omega(t) \end{bmatrix}$$

- $y(t)$ is the output of the system
- $u(t)$ is the system input
- **A** is the state matrix. This is a $n \times n$ matrix where n is the dimension of the state vector. n is also known as the system dimension.
- **B** is the input matrix. This is an $n \times m$ matrix where m is the input vector.
- **C** is the output matrix. this is a $n \times p$ matrix where p is the dimension of the output vector.
- **D** is the matrix linking the output and the input of the system. A system is said to be strictly proper if $D = 0$.

When in state space form a system can be written as $\Sigma = (A, B, C, D)$.
As it was done for transfer functions there exists a theorem to ensure system stability:

Theorem 2. *A system $\Sigma = (A, B, C, D)$ is asymptotically stable if and only if the eigenvalues of A have negative real part, and it is only simply stable if the eigenvalues of A have negative or null real part.*

To determine the reachability (see A.2) of a system, it is important to define the **reachablity matrix**.

Consider the discrete time system $\Sigma = (F, G, H)$, the state after k steps is defined by:

$$x(k) = \sum_{i=0}^{k-1} F^{k-1-i} G u(i) \tag{3.15}$$

Equation (3.15) can also be written in matrix form as follows:

$$x(k) = \begin{bmatrix} G & FG & ... & F^{k-1}G \end{bmatrix} \begin{bmatrix} u(k-1) \\ u(k-2) \\ ... \\ u(0) \end{bmatrix} \tag{3.16}$$

By varying the input vector $u(k)$ it is possible to obtain the set of the reachable states. This set corresponds with the image of $\begin{bmatrix} G & FG & ... & F^{k-1}G \end{bmatrix}$ which takes the name of reachability matrix $\mathcal{R}$.

3.2.2 Reachability in continuous time systems

The reachability problem is solved differently between continuous time and discrete time systems. Since the system analyzed in this thesis is a continuous time one this section explains only the reachability conditions for continuous time ones. Given a system described as (3.14), a state is said to be reachable at time t if it exists an input $u(t)$ such that:

$$x(t) = \int_0^t e^{A(t-\sigma)} B u(\sigma) d\sigma \tag{3.17}$$

The previous definition can be rewritten in the following way: said $\mathcal{U}$ the space of the input functions and R_t the linear operator defined as:

$$R_t : u(.) \longrightarrow \int_0^t e^{A(t-\sigma)} B u(\sigma) d\sigma \tag{3.18}$$

$x(t)$ is said to be reachable at time t if and only if it belongs to the image of R_t. From (3.18) it is possible to define the reachable subspace at time $t > 0$, as the image of the linear operator R_t.

By exploiting a property of linear operators, it is possible to obtain a matrix description of the reachable subspace. To do so, consider the adjoint of the linear operator R_t: $R_t^* : X \longrightarrow \mathcal{U} : x \longrightarrow u(.)$, with $u(\sigma) = B^T e^{A^T(t-\sigma)}x$.

Since the dimension of ImR_t is finite then $ImR_t = Im(R_t R_t^*)$. $R_t R_t^*$ is defined as follows:

$$R_t R_t^* : x \longrightarrow \int_0^t e^{A(t-\sigma)} B B^T e^{A^T(t-\sigma)} d\sigma x \tag{3.19}$$

From (3.19) it is possible to note that the reachable subspace is the image of $R_t R_t^*$. Moreover $R_t R_t^*$ is a $n \times n$ matrix that can be defined as:

$$\mathcal{W}_t = \int_0^t e^{A(t-\sigma)} B B^T e^{A^T(t-\sigma)} d\sigma x \tag{3.20}$$

$\mathcal{W}_t$ is called reachablility gramian.

An easier way to compute the reachablility subspace is to exploit the following theorem:

Theorem 3. *The reachable subspace at time $t > 0$ is the image of the reachability matrix: $\mathcal{R} = \begin{bmatrix} B & AB & ... & A^{n-1}B \end{bmatrix}$ [15]*

Note that using theorem 3 it is possible to state that, in continuous time systems, the reachable subspace does not depend on the length of the interval $[0; t]$.

To state that a system is reachable is equal to say that, with a series of inputs, every state is possible to reach. This fact is expressed by this following relation: $Im\mathcal{R} = \mathcal{X}(t)$, where $\mathcal{X}(t)$ is the space of all the system states.

Due to its independence from time the reachable subspace is noted as $\mathcal{X}^R$. Also from the previous observation it is possible to derive a reachability condition: A system is said to be reachable if and only if $\mathcal{R}$ is full rank(a.k.a $rank(\mathcal{R}) = n$).

If a system does not respect the reachability condition it is said to be unreachable. If a system is not reachable this means that not all states are accessible using a series of inputs. If a state is unreachable this could represent a problem since, even using controllers, it is impossible to force certain states on the system.

If a system $\Sigma = (A, B, C, D)$ of dimension n is not reachable, the reachable subspace $\mathcal{X}^R$ has dimension $\rho < n$.
Assume now that there is a base of $\mathcal{X}$ in which the first ρ vectors are also a base for $\mathcal{X}^R$ and let T be the change of base matrix.
Recalling that $\mathcal{X}^R$ is A-invariant (see A.2) and $ImB \subseteq \mathcal{X}^R$, is obtained a system $\Sigma^e = (A^e, B^e, C^e)$ algebraically equivalent to Σ.
Σ^e is described by:

$$A^e = T^{-1}AT = \begin{bmatrix} A^e_{11} & A^e_{12} \\ A^e_{21} & A^e_{22} \end{bmatrix}; B^e = T^{-1}B = \begin{bmatrix} B^e_1 \\ 0 \end{bmatrix}; C^e = CT = \begin{bmatrix} C^e_1 & C^e_2 \end{bmatrix} \tag{3.21}$$

Where $A^e_{11} \in \mathcal{R}^{\rho\times\rho}$ and $B^e_1 \in \mathcal{R}^{\rho\times m}$ is the reachable subsystem of Σ. Equation (3.21) is called standard reachability form.

3.2.3 Controllability in continuous time systems

A state is said to be controllable to zero at time $t > 0$ if there exists an input $u(.)$ such that:

$$0 = x(t) = e^{At}x + \int_0^t e^{A(t-\sigma)}Bu(\sigma)d\sigma \tag{3.22}$$

Equation (3.22) can be rewritten in the following way:

$$-x = \int_0^t e^{-A\sigma}Bu(\sigma)d\sigma \tag{3.23}$$

As it was done for the reachability problem it is possible to define a controllable subspace $\mathcal{X}^C_t$. This vector subspace is defined by the set of initial states $x(0)$ that, with a series of inputs $u(t)$, can be brought to the final state $x(t) = 0$.
Then from (3.23) it is possible to deduce that $e^{At}\mathcal{X}^C_t \subseteq ImR_t = \mathcal{X}^R$. It also hold the reverse inclusion since every element of $\mathcal{X}^R$ can be written as:

$$\int_0^t e^{A(t-\sigma)}Bu(\sigma)d\sigma = e^{At}\int_0^t e^{-A\sigma}Bu(\sigma)d\sigma \tag{3.24}$$

Equation (3.24) implies also:

$$e^{At}\mathcal{X}^C_t = \mathcal{X}^R \tag{3.25}$$

By multiplying both members of (3.25) by e^{-At} and recalling that $\mathcal{X}^R$ is A-invariant and so also e^{-At} invariant, this implies that $\mathcal{X}_t^C = e^{-At}\mathcal{X}^R \subseteq \mathcal{X}^R$ [15]. Due to the invertibility of e^{-At} it is possible to state that $\mathcal{X}_t^C = \mathcal{X}^R$. This fact shows that, for continuous time system, controllability(see A.2) and reachability are equal.

3.3 Eigenvalue allocation with state feedback

Consider a system $\Sigma = (A, B, C)$, D is neglected since it is assumed D=0. Assume also that all the components of the state $x(t)$ are accessible at any instant t.

Consider a control law in which the components of $u(t)$ are a linear combination of the system states at instant t $x_i(t)$, and the values of the new inputs at time t $v_i(t)$.

$$u(t) = k_{l1}x_1(t) + k_{l2}x_2(t) + ... + k_{ln}x_n(t) + v_i(t) \tag{3.26}$$

Where $l = 1, 2, ..., m$ are the number of inputs.

By exploiting (3.26) it is possible to write the system input in the following form:

$$u(t) = Kx(t) + v(t) \tag{3.27}$$

With K being a $m \times n$ matrix.

By combining (3.14), (3.27) and recalling that it was assumed D=0, it is possible to write the following equation to describe the system:

$$\begin{cases} \dot{x}(t) = (A + BK)x(t) + Bv(t) \\ y(t) = Cx(t) \end{cases} \tag{3.28}$$

In the single input case ($m = 1$) the matrix K is a $1 \times n$ vector.

State feedback has one major advantage over output feedback. It in fact uses n parameters to modify the dynamic of the system Σ instead of the single one used by output feedback.

Consider now the closed loop system $\Sigma_k = (A + BK, B, C)$ and the original system $\Sigma = (A, B, C)$, it hold the following theorem:

Theorem 4. *The reachable subspaces of Σ and Σ_k are the same for any matrix K. Moreover Σ_k is reachable if and only if Σ is reachable.[15]*

Another useful theorem when studying state feedback is the following:

Theorem 5. *Let $\Sigma = (A, B, C)$ in standard reachability form with*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$$

Then also Σ_k is in standard reachability form with the same unreachable subsystem A_{22}.[15]

As consequence of theorem 5 to analyze the effects of state feedback is only necessary to study the reachable part (A_{11}, B_1).
Consider now a reachable system $\Sigma = (A, B, C)$ with one input (meaning m=1), it is possible to represent Σ in the following way:

$$A_c = \begin{bmatrix} 0 & 1 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ -\alpha_0 & -\alpha_1 & \cdots & \cdots & -\alpha_{n-1} \end{bmatrix} ; B_c = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} \tag{3.29}$$

Where α_i are the coefficients of the characteristic polynomial of matrix A, which is defined as:

$$\Delta_A(s) = det(A - sId_n)$$

This form is called controllable canonical form.

Theorem 6. *The system $\Sigma = (A, B)$ is reachable if and only if it is algebraically equivalent to a system $\Sigma_c = (A_c, B_c)$ with A_c and B_c in controllable canonical form*(3.29)*.[15]*

Note that to obtain A_c and B_c it is not necessary to use a change of base but it is only necessary to have A,B and $\Delta_A(s)$.
Consider now a control law such as $u(t) = K_c x_c(t) + v(t)$, with $K_c = [k_0, k_1, ..., k_{n-1}]$ the state equation become:

$$\dot{x}_c(t) = (A_c + B_c K_c)x_c(t) + B_c v(t) \tag{3.30}$$

From (3.30) it is possible to derive the form of $A_c + B_cK_c$ which is:

$$A_c + B_cK_c = \begin{bmatrix} 0 & 1 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ -(\alpha_0 + k_0) & -(\alpha_1 + k_1) & \cdots & \cdots & -(\alpha_{n-1} + k_{n-1}) \end{bmatrix}; \quad (3.31)$$

The coefficients of the characteristic polynomial of (3.31) can be red on the last row of $A_c + B_cK_c$. So the characteristic polynomial of $A_c + B_cK_c$ is:

$$\Delta_{A_c+B_cK_c}(s) = s^n + (\alpha_{n-1} - k_{n-1})s^{n-1} + ... + (\alpha_0 - k_0) \quad (3.32)$$

From (3.32) it is possible to note that the roots of the characteristic polynomial are dependant on the components of K_c. This fact shows how it is possible to compute K_c in order to place the eigenvalues of $A_c + B_CK_c$ arbitrarly since the eigenvalues of $A_c + B_CK_c$ are also the roots of $\Delta_{A_c+B_cK_c}(s)$.
Since the spectrum of a matrix is invariant with respect to a change of bases, the eigenvalues of $A_c + B_cK_c$ are equal to the ones of $A + B(K_cT^{-1})$. Hence to chose the eigenvalues for $A + BK$ it is possible to choose $K = K_cT^{-1}$.
All these reasoning are well summed by the following theorem:

Theorem 7. *Let $\Sigma = (A, B, C)$ be a reachable system of dimension n. For any monic polynomial $p(s)$ of degree n, there exists a matrix $K \in \mathcal{R}^{1\times n}$ such that the characteristic polynomial of $A + BK$ is $p(s)$.*

Theorem 7 is also known as **allocation of eigenvalues,m=1**.[15]
Consider now a non reachable system, which can be always expressed in standard reachability form (3.21). In this case when considering the reachable subsystem theorem 7 still stands. For what regards the non-reachable part as the name suggests cannot be controlled, hence the system can be stabilized only if the eigenvalues of the non-reachable subsystem are stable.

3.4 State-space model of a DC motor and control design

Starting from the DC motor transfer function obtained in 2 and described by the block scheme 2.6,which are:

$$\begin{cases} \dot{\theta} = \omega \\ \dot{\omega} = -\frac{1}{T_m}\omega + \frac{k_m}{NT_m}u \end{cases} \tag{3.33}$$

it is possible to obtain a state space realization for the system.

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T_m} \end{bmatrix}; B = \begin{bmatrix} 0 \\ \frac{k_m}{NT_m} \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \end{bmatrix} D = 0 \tag{3.34}$$

Where $T_m = \frac{R_a I_m}{R_a F_m + k_t k_e}$, $k_m = \frac{k_{drv} k_t}{N R_a I_m}$ (all these parameters are mentioned in 2).
It can be noticed that A is already in controllable canonical form. Also note that $x(t) = \begin{bmatrix} \theta(t) & \omega(t) \end{bmatrix}^T$ is the state vector of $\Sigma = (A, B, C)$.
Since A is in controllable canonical form it is possible to apply theorem 6 deriving that A is also reachable. Since A is reachable theorem 7 holds, hence full eigenvalue allocation of the system is possible.
The goal of the state space control in this thesis is to achieve robust tracking of the reference signal, to do so it is possible to use eigenvalue allocation. The concept of robust tracking means that the controller is able to track the reference signal even with modelling uncertainties and disturbances.
Since in this project the robot has to track a complex signal it is necessary to exploit the **internal model principle**. The latter is defined as follows: the controller needs to incorporate a model of the dynamics that generate the signal which the control system is intended to track.[18]
As mentioned in the previous definition, the controller needs to have in its structure a model of the reference dynamics. This is called exosystem, these models are used to describe the behavior of the reference signals and are often expressed via homogeneous linear ordinary differential equations (ODEs).
These differential equations are of the type:

$$r^{(q)}(t) + \beta_{q-1} r^{(q-1)}(t) + ... + \beta_1 r^{(1)} + \beta_0 r(t) = 0 \tag{3.35}$$

Where $r^{(i)}$ denote the time derivatives of order i. Since r is a solution of (3.35) for some non-zero initial condition, it follows that the signal is a linear combination of the following elementary modes.

$$\frac{t^l}{l!}e^{\lambda_i t} \tag{3.36}$$

$$\frac{t^l}{l!}e^{\sigma_i t}\cos\left(\omega_i t\right) \tag{3.37}$$

$$\frac{t^l}{l!}e^{\sigma_i t}\sin\left(\omega_i t\right) \tag{3.38}$$

with λ_i, ω_i, $\sigma_i \in \mathcal{R}$.

λ_i and $\sigma_i \pm j\omega_i$ represent the generic, both real and complex, roots of the characteristic polynomial associated to (3.35). i.e:

$$p(s) = s^q + \beta_{q-1}s^{q-1} + ... + \beta_1 s + \beta_0 \tag{3.39}$$

Equation (3.39) can also be written in state-space form as:

$$\begin{cases} \dot{x_r} = A_r x_r \\ r = C_r x_r \end{cases} \tag{3.40}$$

Equation (3.40) represents the previously mentioned exosystem and it is a matrix description of the dynamics of the reference signal.

Some of the most common exosystems are reported in the following list as examples:

- constant signal: $z(t) = z_0\delta_{-1}(t)$
  its ODE is: $\dot{z} = 0$ and its characteristic polynomial is: $p_z(s) = s$
  the exosystem $\Sigma_z = (A_z, C_z)$ is:

$$A_z = 0, \quad C_z = 1 \tag{3.41}$$

- sinusoidal signal: $z(t) = z_0\sin\left(\omega_0 t + \phi_0\right)\delta_{-1}(t)$
  its ODE is: $\ddot{z} + \omega_0^2 z = 0$ and its characteristic polynomial is: $p_z(s) = s^2 + \omega_0^2$
  the exosystem $\Sigma_z = (A_z, C_z)$ is:

$$A_z = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix}; \quad C_z = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{3.42}$$

- constant plus sinusoidal signal: $z(t) = [z_1 + z_0 \sin(\omega_0 t + \phi_0)]\delta_{-1}(t)$
  its ODE is: $z^{(3)} + \omega_0^2 z^{(1)} = 0$ and its characteristic polynomial is: $p_z(s) = s^3 + \omega_0^2 s$
  the exosystem $\Sigma_z = (A_z, C_z)$ is:

$$A_z = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_0^2 & 0 \end{bmatrix} ; \quad C_z = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{3.43}$$

The problem of tracking a reference signal $r(t)$ can be compared to regulating the tracking error $e(t)$ to zero in a robust sense.

Assume that $r(t)$ follows a dynamic of the type described by (3.35). That equation can be rewritten as follows:

$$r(t) = \sum_{i=0}^{q} \beta_1 r^{(i)} = 0 \tag{3.44}$$

Consider now a System: $\Sigma = (A, B, C, 0)$, which can be written as:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \tag{3.45}$$

The reference error e is defined as follows:

$$e = y - r = Cx - r \tag{3.46}$$

By expressing $r = y - e$ and by substituting it in (3.44), the following equation is obtained:

$$\sum_{i=0}^{q} \beta_i r^{(i)} = -\sum_{i=0}^{q} \beta_i e^{(i)} + \sum_{i=0}^{q} \beta_i C x^{(i)} = -\sum_{i=0}^{q} \beta_i e^{(i)} + C \sum_{i=0}^{q} \beta_i x^{(i)} = 0 \tag{3.47}$$

Exploiting (3.47) it is possible to define the error dynamic as follows:

$$\xi := \sum_{i=0}^{q} \beta_i x^{(i)} \tag{3.48}$$

$$\dot{\xi} = \sum_{i=0}^{q} \beta_i x^{(i+1)} = \sum_{i=0}^{q} \beta_i (Ax^{(i)} + Bu^{(i)}) = A \sum_{i=0}^{q} \beta_i x^{(i)} + B \sum_{i=0}^{q} \beta_i u^{(i)} = \tag{3.49}$$

$$= A\xi + Bu_\xi \tag{3.50}$$

By combining (3.47) and (3.48) it is possible to obtain the model $\Sigma_z = (A_z, B_z)$ in error space:

$$\Sigma_z = \begin{bmatrix} e^{(1)} \\ e^{(2)} \\ e^{(3)} \\ \vdots \\ e^{(m)} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 & \mathbf{0} \\ 0 & 0 & 1 & 0 & \cdots & 0 & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 1 & \mathbf{0} \\ -\beta_0 & -\beta_1 & \cdots & \cdots & \cdots & -\beta_{q-1} & C \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & A \end{bmatrix} \begin{bmatrix} e \\ e^{(1)} \\ e^{(2)} \\ \vdots \\ e^{(q-1)} \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ B \end{bmatrix} u_\xi \tag{3.51}$$

Where $\mathbf{0}$ indicates a vector of zeros. In fact the last row of A_z is composed by A which is an $n \times n$ matrix and a $n \times q$ zero matrix. The same reasoning holds for the last column of A_z.

The system does not depend on the value of r. This means that if Σ_z is asymptotically stable then its state $z = [e, e^{(1)}, e^{(2)}, ..., e^{(q-1)}, \xi]^T$ tends to zero as t tends to infinity. The same holds for the error e.

As it was explained in the previous section if the system Σ_z is reachable it is possible to design a state feedback control law, that places the eigenvalues of the systems in any desired locations.

To study the reachability of Σ_z consider the matrix A_z of equation (3.51). A_z can be divided into four blocks in a structure of this type:

$$A_z = \begin{bmatrix} A_{z_{11}} & A_{z_{12}} \\ A_{z_{21}} & A_{z_{22}} \end{bmatrix}$$

Where A_{z22} corresponds to A, A_{z21} is a $n \times m$ zero matrix, A_{z12} is a $q \times n$ matrix with C as last row and A_{z11} is the remaining $q \times q$ block.

A_{z11} reachablity is granted by theorem 6 since it is possible to note that it is in controllable canonical form. Due to this observation it is possible to state that: Σ_z is reachable if A is reachable.

Consider now the following control law:

$$u_\xi = -K_z z = -\begin{bmatrix} k_0 & k_1 & \cdots & k_{q-1} & \mathbf{K}_\xi \end{bmatrix} \begin{bmatrix} e \\ e^{(1)} \\ e^{(2)} \\ \vdots \\ e^{(q-1)} \\ \xi \end{bmatrix} \tag{3.52}$$

Assume u_ξ to be a stabilizing control law for Σ_z. By using the definition of ξ in (3.48) it is possible to rewrite (3.52) in terms of the original state x and input u.

$$\sum_{i=0}^{q} \beta_i u^{(i)} = -\sum_{i=0}^{q-1} k_i e^{(i)} - \sum_{i=0}^{q} \beta_i \mathbf{K}_\xi x^{(i)} \Longrightarrow \sum_{i=0}^{q} \beta_i (u + \mathbf{K}_\xi x)^{(i)} = -\sum_{i=0}^{q-1} k_i e^{(i)} \tag{3.53}$$

Equation (3.53) defines a controller with input e and output $\tilde{u} := u + \mathbf{K}_\xi x$ and its transfer function is the following:

$$\tilde{U}(s) = -\frac{\sum_{i=0}^{q-1} k_i s^i}{\sum_{i=0}^{q} \beta_i s^i} E(s) = -\frac{k_{q-1}s^{q-1} + \cdots + k_1 s + k_0}{s^q + \beta_{q-1}s^{q-1} + \cdots + \beta_1 s + \beta_0} E(s) = H(s)E(s) \tag{3.54}$$

From (3.54) it is possible to obtain a controller in the starting state x and input u:

$$u = -\mathbf{K}_\xi x - H(s)e \tag{3.55}$$

With H being the transfer function from $\tilde{u}$ to e introduced in equation (3.54). The equations (3.55),(3.54) are then summed up into the following Simulink block scheme:

As mentioned previously, under the reachability condition, it is possible to allocate the eigenvalues of the system using state feedback. To decide where to actually place the system eigenvalues it is important to take note of a few factors. First of all the stability of the closed-loop system, as stated in theorem 2, it is necessary that the eigenvalues of $(A + BK)$ have negative real part. Secondly, a controller needs to respect the given specifics such as rise time, settling time or overshoot. To integrate the requirements previously mentioned in the system dynamic it is still possible to exploit the dominant pole approximation as it was done for the PID control.

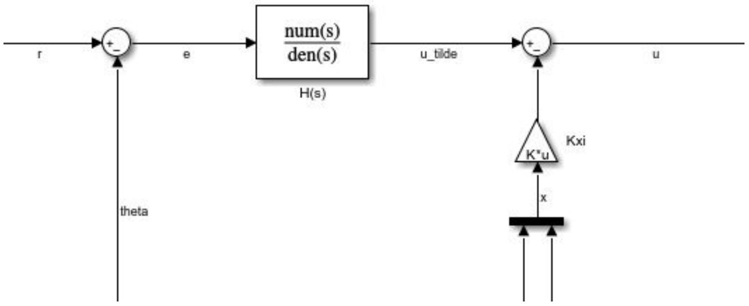

Figure 3.2: State-space feedback controller Simulink scheme

By applying the approximation described in (3.2) it is possible to describe the position of the eigenvalues via ω_n and or δ. An example of this process is computing the eigenvalues in the following way:

$$\lambda_{1,2} = -\delta\omega_n \pm j\omega_n\sqrt{1-\delta^2} \tag{3.56}$$

Once the eigenvalues are selected it is possible to apply theorem 7. In doing so the components of K are chosen to modify the roots of the characteristic polynomial of $A+BK$. The act of changing the roots is equivalent to placing the eigenvalues of the closed-loop system on the position defined by (3.56).

Note that (3.56) is an approximation, so it is used as starting position for the eigenvalues. Via empirical observation on the system, the eigenvalues may be changed to obtain a satisfactory response.

3.5 Linear Quadratic Regulator (LQR)

Before introducing the linear quadratic regulator it is necessary to define optimal control.

The previous control strategy is based on eigenvalue allocation, however the choice of the eigenvalues is often based on empirical observations. Optimal control strategies define parameters that are used to compute a more efficient state feedback matrix K.

The first step for the optimization is to define a cost function, the goal is to minimize the said cost function. For continuous time systems, defined as (3.14), it is possible to define such function as follows:

$$J(u, x_0) = \int_0^T [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \tag{3.57}$$

Where Q is an $n \times n$ positive semi-definite matrix, R is an $m \times m$ positive semi-definite matrix $(Q, R > 0)$, u is the system input and x_0 are the initial conditions. The cost function assigns to every subsequent input $u(.)$ a non-negative number, this is ensured by the fact that both Q and R are positive semi-definite. Hence problem of finding the best series of input becomes the minimization of (3.57). Note that the interval of integration of (3.57) goes to T. This shows that the cost function has a finite time horizon. This means that the function J has to be minimized over the interval $[0; T]$.
Notice that in equation (3.57) there are two recognizable members: $x^T(t)Qx(t)$ and $u^T(t)Ru(t)$. These two addends represents the two aspects that compose the evaluation criteria for the system behaviour in the interval $[0; T]$.
$x^T(t)Qx(t)$ represents the cost for each time the intermediate states are different form zero, $u^T(t)Ru(t)$ represents the energy used for the control.
To compute the best series of input $u(.)$ which minimizes (3.57) consider the following theorem:

Theorem 8. *If $x(.)$ and $u(.)$ satisfy the following differential equation: $\dot{x}(t) = Ax(t) + Bu(t)$ and if $M(.)$ is an $n \times n$ matrix differentiable on the interval $[0; T]$, the the following identity holds:*

$$\int_0^T \begin{bmatrix} u^t(t) & x^T(t) \end{bmatrix} \begin{bmatrix} 0 & B^T M(t) \\ M(t)G & \dot{M}(t) + A^T M(t) + M(t)A \end{bmatrix} \begin{bmatrix} u(t) \\ x(t) \end{bmatrix} dt = x^T(t)M(t)x(t)|_0^T \tag{3.58}$$

[15]

By exploiting this lemma it is possible to rewrite the cost function (3.57) as follows:

$$J(u, x_0) = \int_0^T \begin{bmatrix} u^T(t) & x^T(t) \end{bmatrix} \begin{bmatrix} R & B^T M(t) \\ M(t)B & A^T M(t) + M(t)A + \dot{M}(t) \end{bmatrix} dt \tag{3.59}$$

Since M can be chosen freely it is possible to compute it by exploiting the Differential Riccati Equation (DRE):

$$-\dot{M}(t) = Q + A^T M(t) + M(t)A - M(t)BR^{-1}B^T M(t) \tag{3.60}$$

With the previous hypothesis done on the system and matrices it is possible to state that a solution for (3.60) exists finite in $(-\infty; T]$. Such solution is symmetric positive semi-definite in any point of $(-\infty; T]$.

Theorem 9. *Let $M(.)$ be the solution of equation* (3.60) *on the interval* $[0; T]$*, and by posing $M(T) = S$ $(S \geq 0)$ and $x(0) = x_0$, the input $u(.)$ expressed by the following control law:*

$$u(t) = -R^{-1}B^T M(t)x(t) \tag{3.61}$$

minimizes (3.57) *and the corresponding minimum value is:*

$$J_{min} = x_0^T M(0)x_0 \tag{3.62}$$

It is important to note that the following control law yields a *regulator*. A regulator is a controller which will bring the state of the system $x(t)$ to the final state $x(T) = 0$.

To modify the controller's behaviour consider the infinite time horizon optimization problem. To do so it is needed to introduce the following notation:

$$x_\infty = \lim_{t \to \infty} x(t) \tag{3.63}$$

$$u_\infty = \lim_{t \to \infty} u(t) \tag{3.64}$$

$$J(u, x_0) = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \tag{3.65}$$

It is also necessary to modify the cost function in order to achieve the tracking task.

$$J' = \int_0^\infty (x(t) - x_\infty)^T Q(x(t) - x_\infty) + (u(t) - u_\infty)^T R(u(t) - u_\infty)dt \tag{3.66}$$

With this notation it is possible to define a new optimization problem which uses a new state and input defined as:

$$x_\delta = x - x_\infty \tag{3.67}$$

$$u_\delta = u - u_\infty \tag{3.68}$$

Hence the cost function for the new LQR task becomes:

$$J^{'} = \int_0^{\infty} x_\delta^T Q x_\delta + u_\delta^T R u_\delta dt \tag{3.69}$$

Using the notation in (3.67) it is also possible to modify the plant dynamics as follows:

$$\dot{x_\delta} = Ax_\delta + Bu_\delta + Ax_\infty + Bu_\infty \tag{3.70}$$

Note that in equation (3.70) the term: $Ax_\infty + Bu_\infty$ is equal to zero. This statement holds since by using a regulator $\dot{x}_\infty$ is equal to zero.
To obtain an LQR controller which is able to track a complex reference signal not only it is necessary to use the cost function described by (3.69). It is also necessary to include a model of the reference signal via the internal model principle, mimicking what it was done for the control via eigenvalue allocation. This means that the system state matrix has the structure described by (3.51).
In order to design an LQR controller it is possible to use the following theorem:

Theorem 10. *Let $\sqrt{Q}$ be a matrix such that $Q = \sqrt{Q}^T \sqrt{Q}$ it holds that:*

- *the Algebraic Riccati Equation (ARE)*

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{3.71}$$

has a unique positive semi-definite solution P_∞

- *the state feedback control law:*

$$u(t) = -Kx(t) \quad with \quad K = R^{-1}B^T P_\infty \tag{3.72}$$

minimizes the infinite horizon cost function (3.63) *and makes the closed-loop system asymptotically stable.*

Theorem 10 grants the existence of an optimal state feedback matrix under certain conditions.
The existence of such controller however does not grant the fact that it achieves satisfactory results. In fact these controller needs to be tuned and to to do it is

possible to act on the two matrices Q and R.

First of all it is important to remember that the matrices can be modified to tune the controller, however it is important that they maintain certain characteristic for example being positive semi-definite. This because if some traits of the structure of Q and R are lost then theorem 10 may not hold, meaning the existence of K is not ensured.

Some guidelines used to tune the LQR controller are the following:

- If the input or state of the system are decoupled then it is useful to select Q and R as diagonal matrices.

  The state or input is said to be decoupled if each component is independent from the others.

  This because a diagonal matrix keeps the components decoupled and simplify calculations.

- For each component of the input or state that should be near zero choose a large value for the associated matrix component.

  This because if said components has a large associated value it will negatively impact the cost function and the controller will try to correct it quickly.

- Keep in mind that the behaviour of the controlled is regulated by the value of Q relative to R and vice versa. Changing both by the same amount will not change the controller's output. In particular choosing Q and R is a trade-off between control energy $\int_0^\infty u^T R u dt$ and control accuracy $\int_0^\infty x^T Q x dt$. Increasing Q will make the controller more accurate but slower since it will respond with weaker inputs. Increasing R will make the control faster but less accurate since stronger inputs may cause overshoot.

- Note with $\bar{x}_i$ and $\bar{u}_i$ the maximum acceptable deviations of the i^{th} component of the state x and input u vector. It is possible to apply the **Bryson's**

rule to obtain a starting Q and R.

$$Q = diag\{q_{ii}\}_{i=1,\dots,n} \quad with \quad q_{ii} = \frac{1}{\bar{x}_i^2} \tag{3.73}$$

$$R = diag\{r_{ii}\}_{i=0,\dots,m} \quad with \quad r_{ii} = \frac{1}{\bar{u}_i^2} \tag{3.74}$$

The matrices computed via the Bryson's rule (3.73) are a starting indication for the tuning process. Usually these two matrix are then modified via trial and error to obtain a satisfactory controller response.

The block scheme for the LQR controller is identical to the one used for the control via eigenvalue allocation 3.2. This is expected since these to design methods are conceptually very similar. The aim of both is to compute a state feedback matrix K such that the resulting closed-loop system respects some specifics in terms of stability, settling time and overshoot. However the major difference is in the way the K matrix is computed. The first method is more based on empirical observation while the second uses an optimization procedure.

Chapter 4

Empirical Trials

All the theory work that was described in chapter 3 needs to be accompanied by coherent tests.

The first step in testing is: developing a Simulink model of the system in order to perform a set of simulations. These trials are used to observe the controllers behaviour and to tune them without risking to damage the actuators.

As example consider the PID control block scheme:

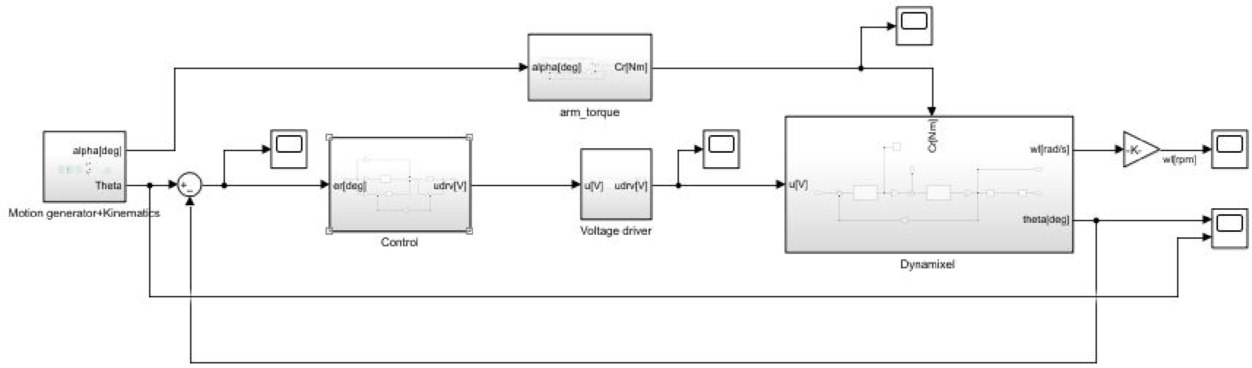

Figure 4.1: Block representation of the PID control strategy

Starting from the left it is possible to see the following blocks:

- **Reference Generator**: it is the first block and it outputs the reference signal described in chapter 2 figure 2.3.

  It is also visible that said block outputs the trajectory of α which is the flexion angle of the elbow. α is then used by another block to compute the resistant torque made by the arm's weight.

- **Control Block**: this block represents the controller used during the simulation. In figure 4.1 the controller used is a PID, when using a state feedback control strategy the block input changes since the control uses the reference and the system state. However the control output is always a voltage.
- **Voltage Driver**: it represents the transfer function described in (2.12). It is the digital equivalent of the motor shield and the Dynamixel motor XL430-W250 in series connection.
- **Torque Computations**: This block in figure 4.1 is above the voltage driver. Its function is to compute the opposing torque due to the forearm mass. It represents the following equation:

$$C_r = m_{for} g l_{for} \cos\alpha \tag{4.1}$$

Where C_r is the resistant torque, g is the gravity coefficient, l_{for} and m_{for} are respectively the length and mass of the forearm. To compute the arm torque the angle α is used, this because the angle between the gravitational force and the forearm is the same as the flexion angle.
The arm torque is then halved since two motors move the arm.

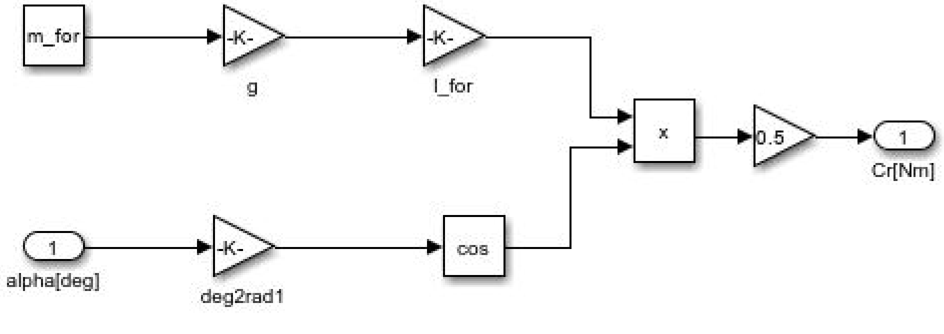

Figure 4.2: Block scheme computation for the resistant torque

Figure 4.2 is the block scheme of the torque computations.

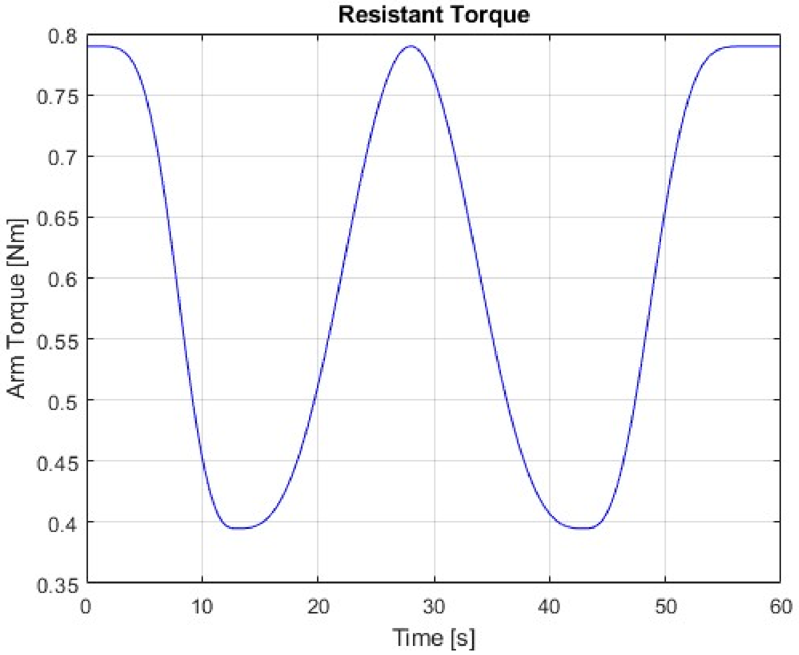

Figure 4.3: Resistant torque (C_r) profile during elbow flexion\extension

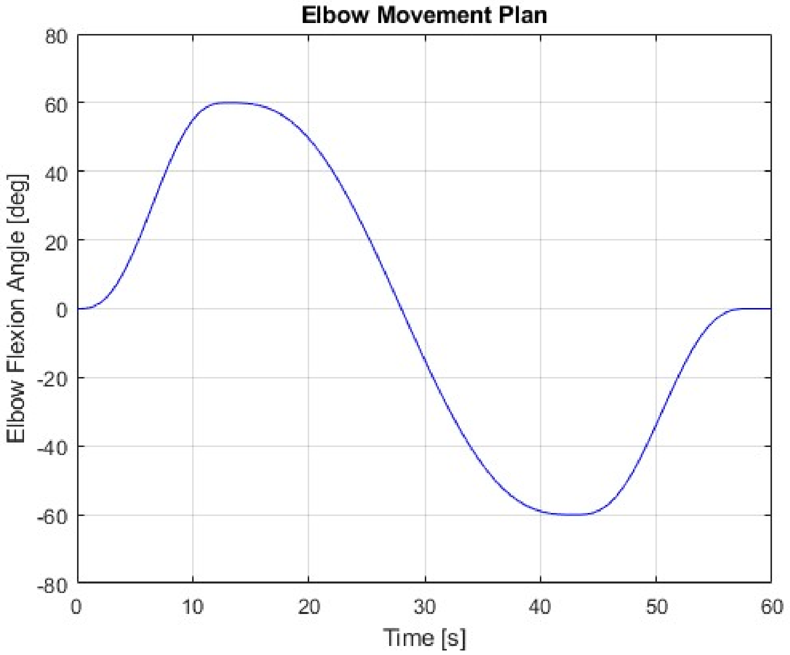

Figure 4.4: Elbow flexion angle (α) profile during elbow flexion\extension

From image 4.3 it is possible to note how the resistant torque reaches its peak at the starting α position. This is to be expected since the resistant torque (4.1) is directly proportional to the cosine of α, hence is maximum when α is equal to zero. This resoning is even more clear when confronting the resistant torque behaviour with the elbow flexion angle profile 4.4.

- **Motor Block**: the last block is the Dc motor model. It is the block scheme described by 2.6.

4.1 Parameter Estimation

The first step to build an accurate model for any system is to have a clear view on the system dynamics. In this thesis the behavior of the DC motors used is described in 2. However not all the system parameters were immediately available; hence some parameters need to be estimated.

The motor datasheet already gives some useful information such as:

- The stall torque C_s, which is the torque expressed by the motor with zero rotational speed on the shaft.
- The stall current $i_{a,s}$, which is the current drawn by the motor at zero rotational speed.
- The no load speed ω_0, which is the speed of the motor at nominal voltage and without external loads.
- The internal gearbox ration N.
- The motor inertia I_m: the datasheet gives the complete inertia matrix, in this study however the only useful information is the inertia along the z axis. This because in the application the only rotations happen around the z axis.

The first estimated parameter is the torque constant k_t. To do so consider the following equation:

$$C_m = i_a k_t \tag{4.2}$$

Where i_a is the armature current and C_m the motor torque. By replacing in (4.2) the motor torque with the stall torque and doing the same with the stall current it si possible to compute k_t as:

$$k_t = \frac{C_s}{i_{a,s}} \tag{4.3}$$

After computing k_t it is possible to estimate the armature resistance R_a using the following equation:

$$R_a = \frac{V_{dc}}{C_s} k_t \tag{4.4}$$

Where V_{dc} is the supply voltage.
Once k_t is computed it is also obtained the electromotive constant k_e since it is equal to the torque constant k_t. It is important to note that the previous fact is only valid for DC motors. It is also only true when using the International Measurement System
Using the result of equation (4.3) and the previous fact, it is possible to compute the back electromotive force E:

$$E = k_e w_0 \tag{4.5}$$

By using the results of (4.5) and (4.4) it is possible to compute the no load current i_{a0}

$$E = V_{dc} - R_a i_a \quad \longrightarrow \quad i_{a0} = \frac{V_{dc} - E}{R_a} \tag{4.6}$$

At last the results of (4.6) and (4.3) are needed to compute the friction coefficient F_m

$$F_m = \frac{k_t i_{a0}}{\omega_0} \tag{4.7}$$

4.2 Control Simulations Results

In this section are discussed the results obtained by the Simulink model 4.1.
The output data analyzed in this section are four for each control strategy for an elbow flexion of $[0°, 60°]$ and an extension of $[0°, -60°]$:

- **motor shaft position** θ: It is the most important result, since it shows the behavior of the motor shaft with respect to the position reference signal.

- **motor speed** ω: The rotational speed of the shaft is a notable measurement. It has to be a smooth enough curve to avoid jerking the patient's arm.
- **motor voltage** u: It represents the supply voltage of the motor. It is important to have a smooth curve since that translates into a smooth functioning cycle of the motor.
- **reference error** e: It shows how far the motor position is from the reference. This data coupled with θ give a clear image on how well the controllers behave.

For every controller, other than the stability requirement, it was also needed to have a settling time t_s equal or less then 0.15s and a maximum overshoot M_p of about 5%.

4.2.1 PID Simulations

The first control strategy taken in exam is the PID control. This controller takes as input the reference error as explained in 3.

The control design consists in finding the values of K_p, K_i and K_d.

As mentioned previously the starting requirements, $t_s \leq 0.15s$ and $M_p \leq 5\%$ are used in (3.5) and (3.6), the results of these equations are then used in (3.13) obtaining the three characteristic gains. K_p, K_i and K_d are then finely tuned using a trial and error procedure, in particular the first controller response was too fast causing oscillations. This problem was solved by lowering the three values. The resulting controller has:

Table 4.1: PID gains

$K_p = 11.778$	$K_i = 199.72$	$K_d = 0.17364$

This controller yields the following results:

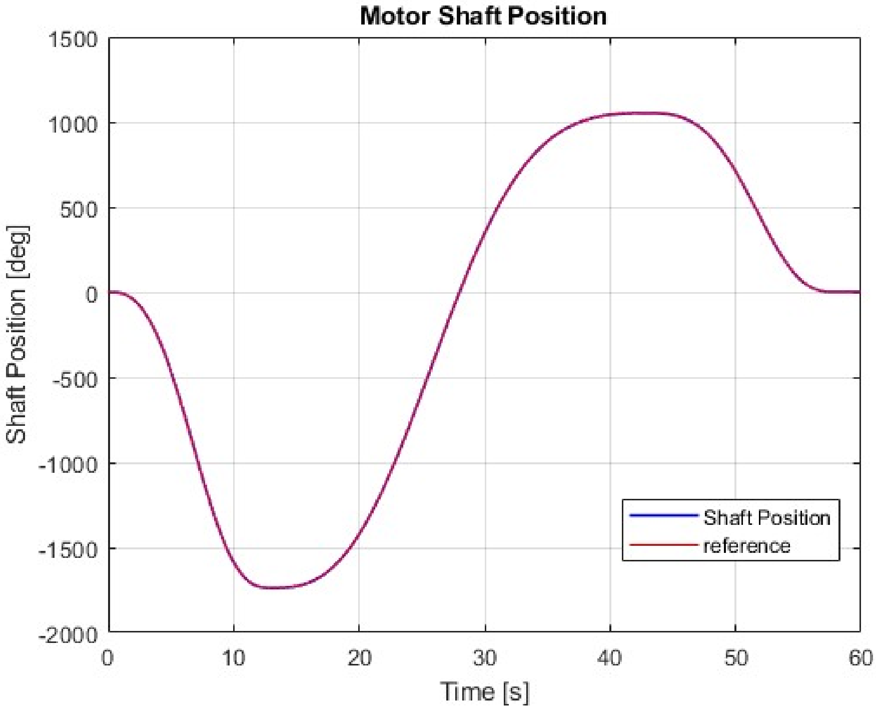

Figure 4.5: Motor shaft position (θ) profile during elbow flexion\extension and position reference using PID control

It is immediately seen by 4.5 how the control allows for almost perfect tracking of the reference signals. To better appreciate the precision of this controller consider the following image:

This image is a zoomed version of the previous one. It is clear how close the behavior of the motor is almost equal to the reference. However still there is a difference between the desired trajectory and the actual motor behaviour. To analyze the differences in behaviours consider the following image:

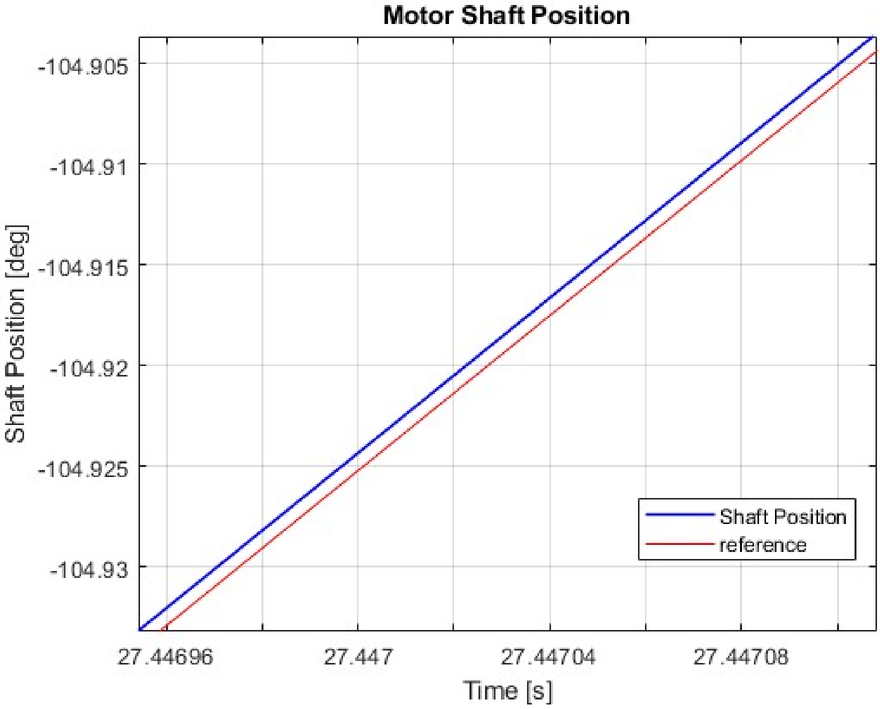

Figure 4.6: Motor shaft position curve using PID control zoomed

Image 4.7 represents the shaft reference error, which is also the controller input. As expected the first part shows the largest values of the error, there are also present some oscillations.This is to be expected since there is a slight delay between the beginning of the reference signal and the action of the controller. From the error curve it is possible to see that the controller behaves as expected, having a very low peak error and reaching steady state very fast.
Another clear indicator of the correct functioning of the controller is that,after a short transient, the error tends to be zero showing an accurate tracking. In figure 4.7 there are also noticeable two small bumps. These variations happen around second 5 and second 37. By looking at 4.5 it is possible to see that these bumps in error corresponds with the changing of rotation of the motor.

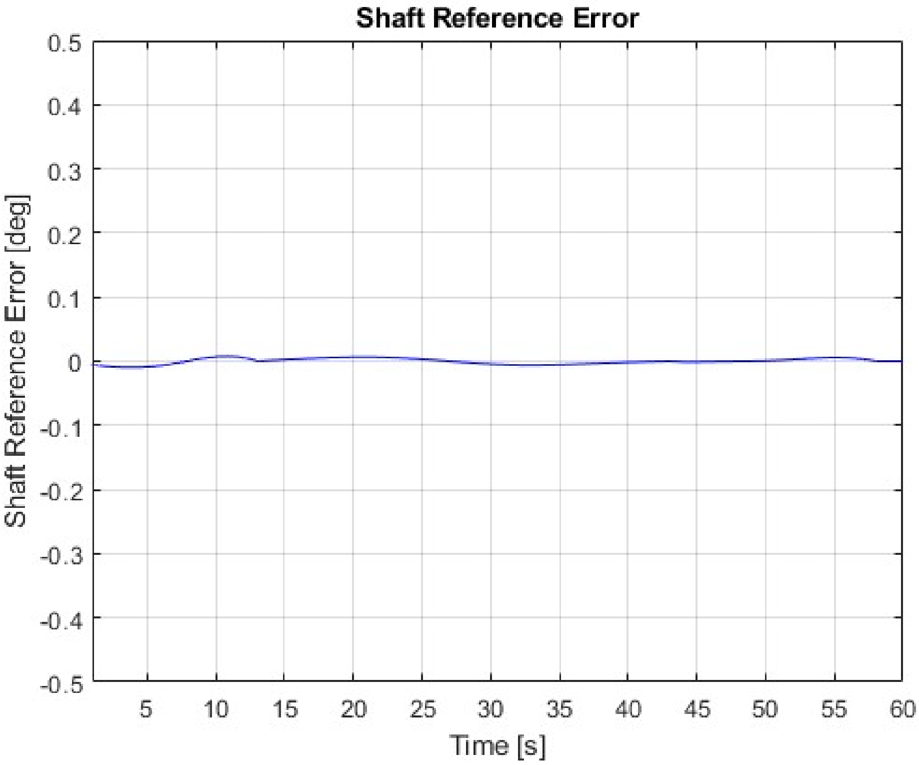

Figure 4.7: Shaft position error during elbow flexion\extension using PID control

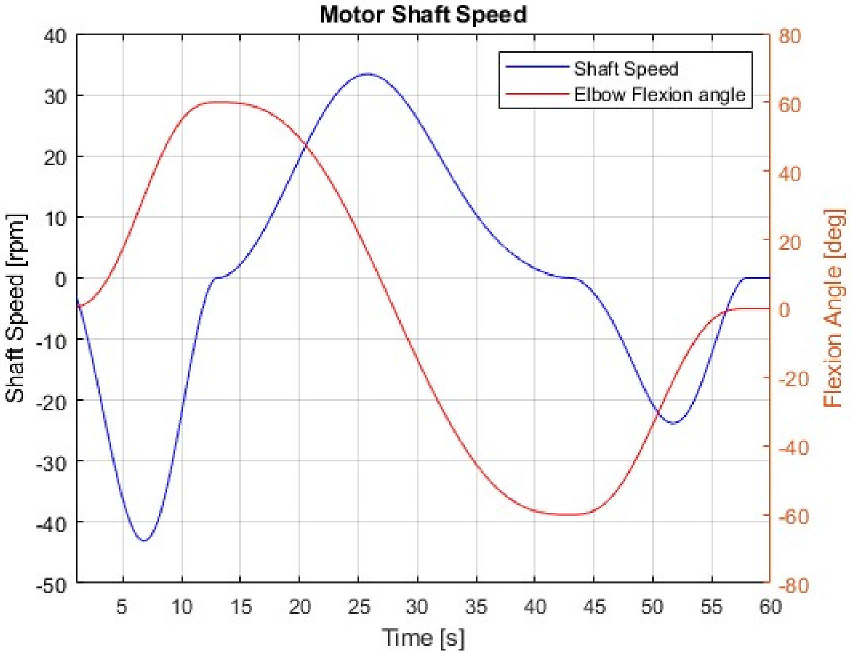

Figure 4.8: Shaft speed (ω) during elbow flexion\extension using PID control

Figure 4.8 shows the speed curve of the motor. The speed shows again another wanted trait of the control: smoothness. The curve is in fact pretty smooth showing a gradual variation of the speed, which implies that the patient's arm is not being jerked. Also as expected the speed is null at the peak of the movements, this reflect the pauses planned in the theta trajectory. Another important fact is that the speed never reaches high values. This is an important note since it shows how the movement happens slowly. The low velocity of the motor has the objective of moving slowly the arm so to not cause pain or discomfort.

Another important set of data that can give useful information is the input voltage of the motor. This data is strictly connected to the shaft speed, so some of the reasonings done on the speed curve can also be applied for the voltage curve. This can be seen by 2.19

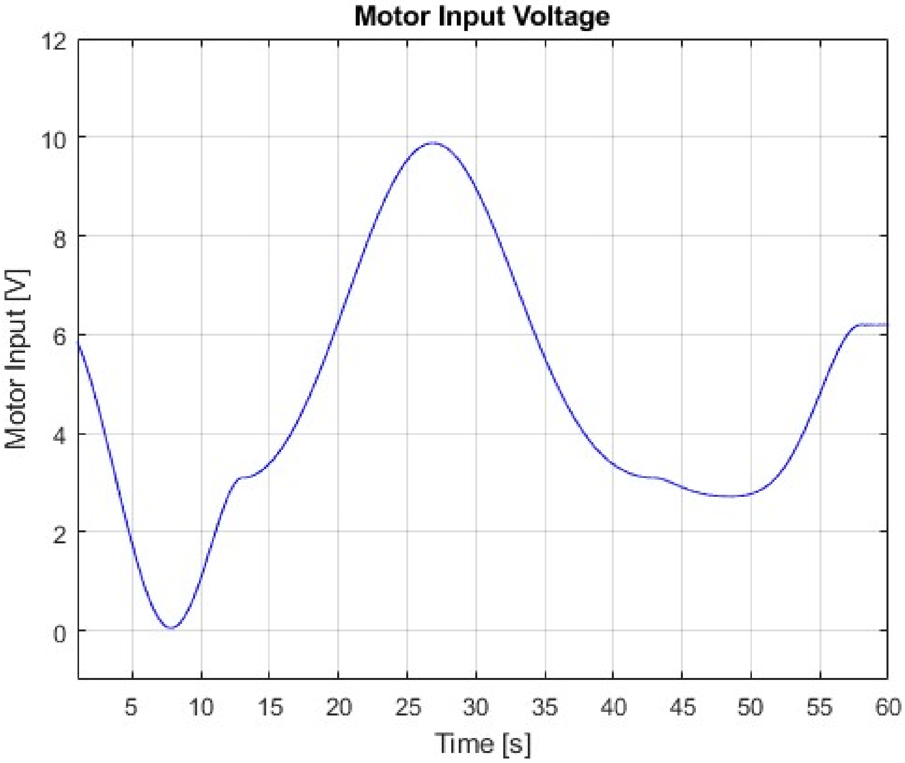

Figure 4.9: Motor voltage input curve during elbow flexion\extension using PID control

Figure 4.9 shows the voltage curve feeding the motor.

The motor is a twelve volts DC motor hence the maximum accepted value is 12V. It is important to keep in mind that the motor is rated for such voltage, however it is not safe to have it work at maximum value for prolonged periods of time. This fact is clear when looking at 4.9, in fact the peak voltage is reached reached after the first pause4.4. This is explained because changes in rotation direction require the motor to slow down and then accelerate.

Note also that in the pauses the voltage is different from zero. This happens because of the resistant torque associated with the arm weight.

As for the speed curve the voltage one is also smooth ensuring gradual movement of the motor shaft.

4.2.2 Eigenvalue Allocation Simulations

The second control design used for this application is the controller based on eigenvalue allocation. As described in the respective section in chapter 3, this is a state feedback controller. This control strategy requires the controller to have the full state, defined as $x(t) = \begin{bmatrix} \theta(t) & \omega(t) \end{bmatrix}^T$ and the reference signal 2.3.

The reference signal used cannot be described by ordinary differential equations. This fact is justified because it is the combination of three different fifth degree polynomial curves with pauses at each end. Hence when choosing the appropriate exo-system it is important to use the one the reflect the reference behaviour better. For this reason this project exploits exo-system (3.43).

Consider now the system $\Sigma = (A, B)$ defined as (3.34), when combined with the previously mentioned exo-system a 5×5 system is obtained.

The controller's performances are strictly tied to the choice of eigenvalues, in this case five eigenvalue need to be chosen. As for the PID control the same requirements on overshoot and settling time are required.

For this control strategy the following eigenvalues are used:

- $\lambda_1 = -10000$
- $\lambda_2 = -5.301$
- $\lambda_3 = -0.3870 + 0.8392i$

- $\lambda_4 = -0.3870 - 0.8392i$
- $\lambda_5 = -0.7888$

Note how all the eigenvalues have negative real part in order to ensure asymptotic stability. Also the two complex eigenvalues appear as expected in a conjugate couple.

Once the eigenvalues are fixed using the Matlab function **acker**, which computes the matrix K starting from the couple (A, B) and the selected eigenvalues. Consider now the trajectory tracking capabilities of this controller:

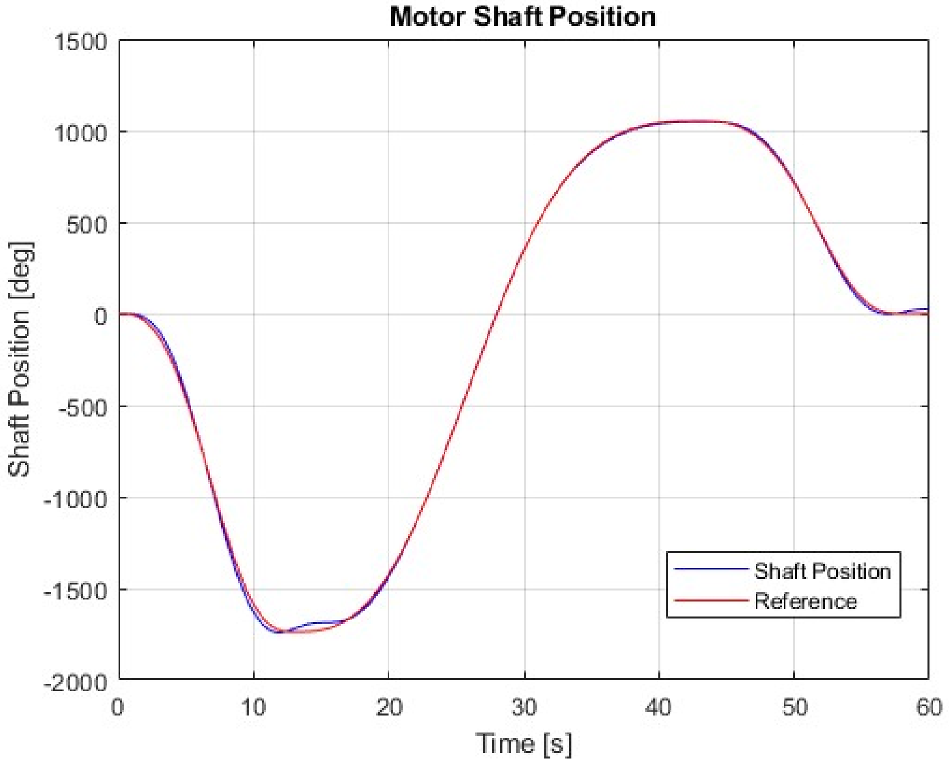

Figure 4.10: Motor shaft position (θ) during elbow flexion\extension using eigenvalue allocation control

It is immediately apparent from image 4.10 that this controller has worse tracking performances compared to the PID control.

It is in fact possible to note overshoot peaks near the changes of direction. This is explained by the fact that changing the direction, especially after the first part, tends to quickly change the resistant torque.

However even considering the presence of overshoot the controller proves to be able to track a complex signal. This is not trivial since this control uses a sinusoidal plus constant exo-system to tack a different type of reference signal. This is another confirmation that the approximation done when choosing the exo-system is accurate enough.
To better understand the entity of the overshoot consider the following image:

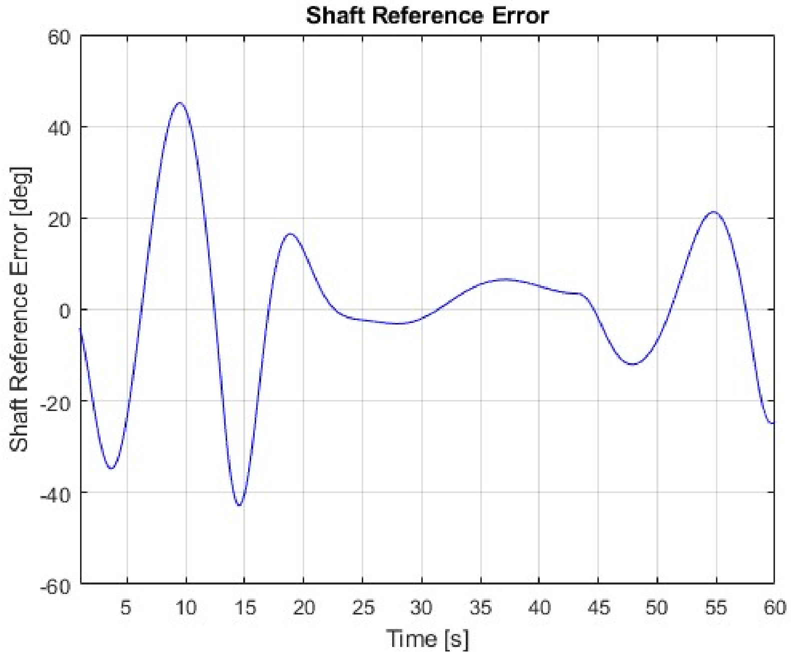

Figure 4.11: Shaft position error profile during elbow flexion\extension using eigenvalue allocation control

It is easily noticeable how the shaft position error is contained inside the boundary of 10%.
Yet this controller achieves a much worst tracking with respect to the PID. In fact the error never stabilizes around zero. It is possible to notice that after the first peak the error tends lower and then oscillate around zero. However these oscillations are too impactful to be considered acceptable for the application.
This controller reaches steady state faster but with worse results on both overshoot and overall reference error. This is not a suitable compromise since, especially when working with a human arm, overshoot needs to be controlled as much as possible.

After studying the position behavior of this controller, observe now its speed behavior.

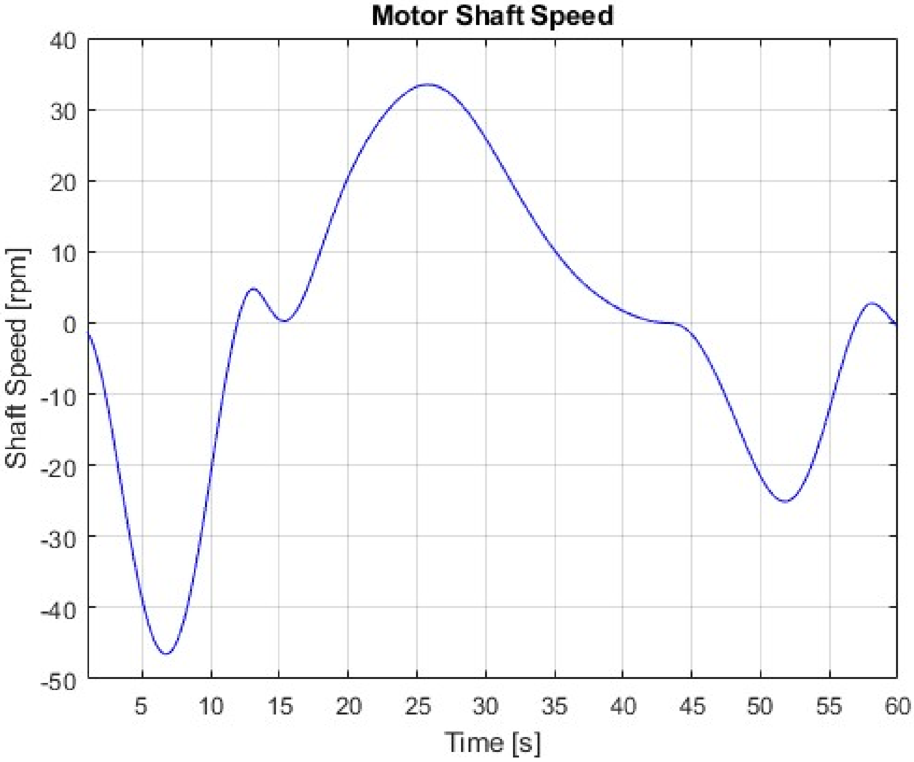

Figure 4.12: Shaft speed (ω) curve during elbow flexion\extension using eigenvalue allocation control

The speed curve 4.12 shows greater general oscillations caused by the presence of overshoot and by the response of the controller to said overshoot. However greater peak values will require higher peak voltage input.

This could represent another obstacle to this control strategy especially in the context of long working cycles, since it is not optimal for the motor to work for long times near maximum supply voltage value.

Reaching higher speed values could also cause another problem. In fact moving the patient's arm faster could cause discomfort since it would result in a jerking motion.

To better see this phenomenon consider the voltage curve shown in the following picture:

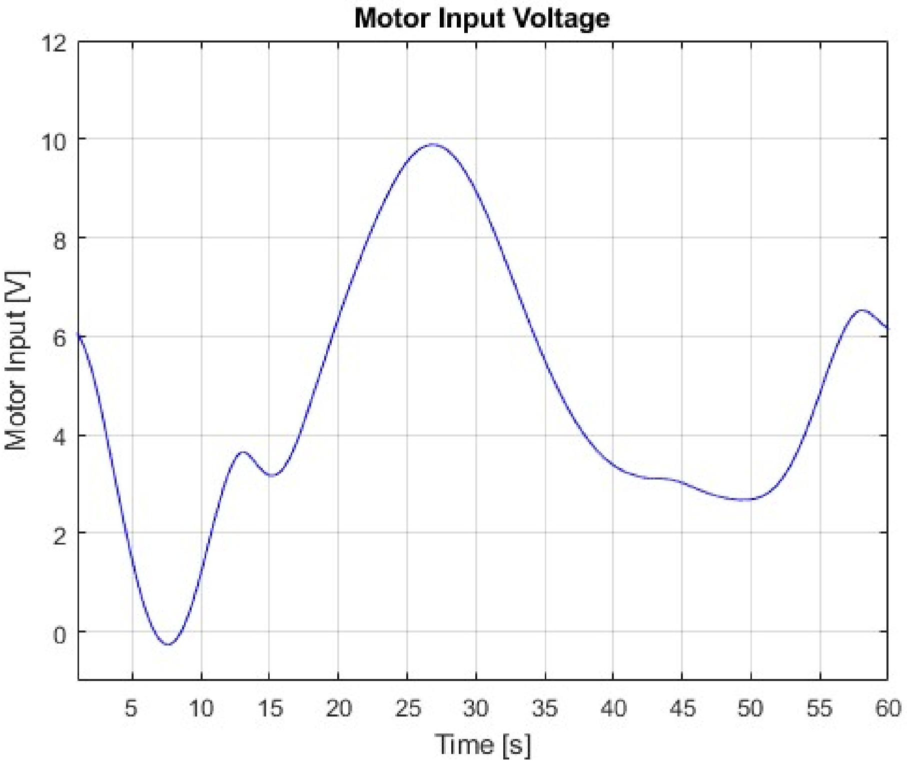

Figure 4.13: Motor voltage input profile during elbow flexion\extension using eigenvalue allocation control

As expected the voltage curve shows some initial oscillations, these are expected since they are also present in the speed curve.
Consider now image 4.9, as the peak values are higher,in both cases the input voltage reaches near motor saturation value only during the changes of direction. Image 4.13 also shows, as expected, how the voltage peaks are higher in correspondence to the overshoot peaks. This is to be expected since an overshoot also implies a rapid change of direction which impacts on the voltage absorption.

4.2.3 LQR Simulations

The last control design tested on this project is the LQR. It is still a state feedback control strategy, which uses the same state $x(t)$ as the previous controller.
As it was done for the eigenvalue allocation case, a proper exo-system is chosen to obtain the wanted results. Also for the LQR control the sinusoidal plus constant exo-system is used to approximate the reference signal.
The resulting 5×5 system is exactly equal to the eigenvalue allocation case. This is to be expected since this control strategy differs from the previous one only in the way the feedback matrix K is computed.
The feedback matrix used by this controller is computed using the **lqr** Matlab function. This routine requires as inputs the couple (A, B) and the matrices Q and R. The last two matrices Q and R are computed using the Bryson's rule (3.73) as a starting point and then are fine-tuned using trial and error. The resulting matrices are the following:

$$Q = \begin{bmatrix} 50000 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix} \qquad R = 0.003 \tag{4.8}$$

Note that the matrix Q is a 5×5 matrix. This is to be expected since the extended system is of dimension 5, R is a scalar and this is due to the fact that B is a 5×1 matrix.
Q is a positive definite matrix since all its eigenvalues are positive, the same can be said for R. This fact grants the existence of a solution for the LQR optimization problem.
This control strategy has its foundation on solving the minimization of the cost function (3.66). This means that, from a performance point of view, it is expected to be better then the eigenvalue allocation design since it computes the optimal feedback matrix K.

In order to study the control performances start by analyzing its tracking capabilities:

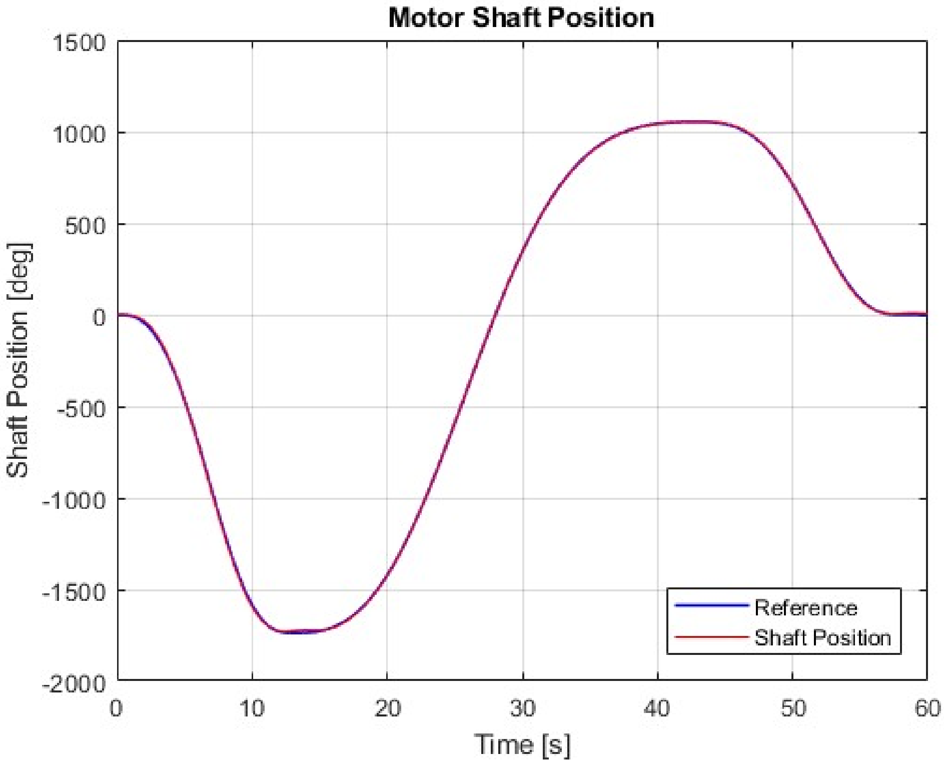

Figure 4.14: Motor shaft position (θ) during elbow flexion\extension using LQR control

Image 4.14 shows a better tracking then 4.10, having also lower overshoot. This is expected and it is in accordance with the reasoning done before.
However this control also shows a worse performance than the PID (see 4.5). This could be motivated by the approximation done when selecting the exo-system.
This control strategy nonetheless reduces the overshoot peaks, which was one of the most critical problem of the eigenvalue allocation control.
The requirements for both overshoot and settling time are still the same also for this control. This is important to note because, when considering the eigenvalue allocation control, they are not acceptable.
So to better evaluate the entity of the overshoot peaks, and to ensure it respects all the requirements. It is useful to study the reference error curve for this controller.

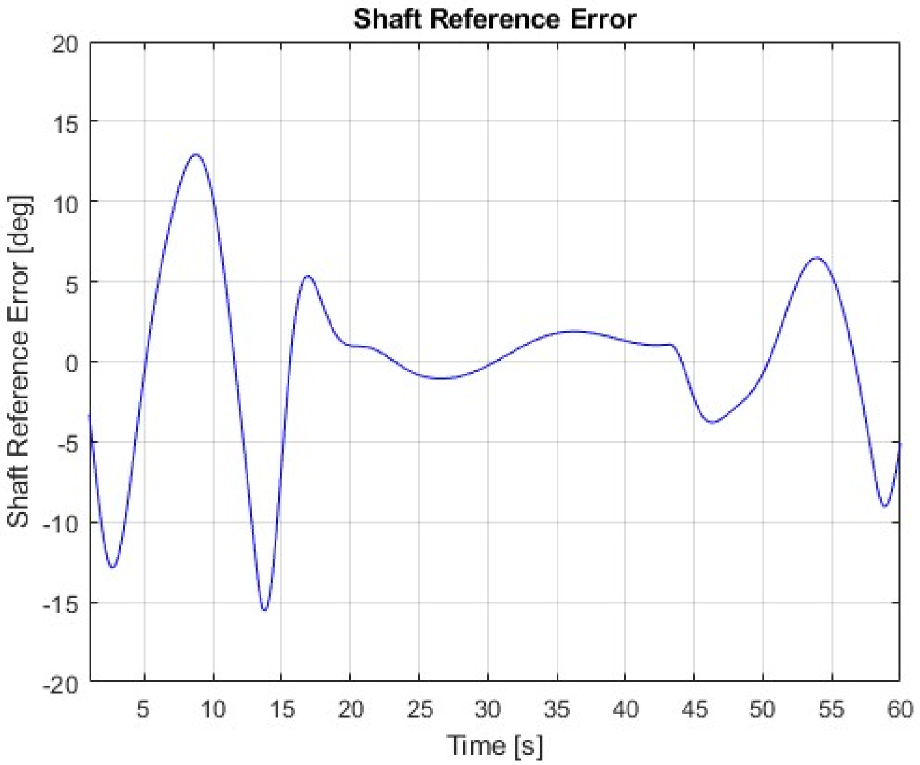

Figure 4.15: Shaft position error profile during elbow flexion\extension using LQR control

Image 4.15 clearly shows how this control strategy has a lower shaft reference error. It is also clear how this controller respects the specific on the system overshoot.

The error curve in this case not only shows a much lower overshoot but it also tends to be much closer to zero then 4.11. This directly translates in better tracking performances of the controller.

The next comparison is with the error curve of the PID 4.7. Also in this case the PID shows to be superior having a much lower tracking error. This is still due to the approximation done when choosing the exo-system. This phenomenon has to be expected since eigenvalue allocation and LQR control are very similar.

After observing the tracking capabilities of this controller via the position and reference error curves, it is important to study its behaviour in terms of smoothness of the movement. To do so consider the speed profile of the movement

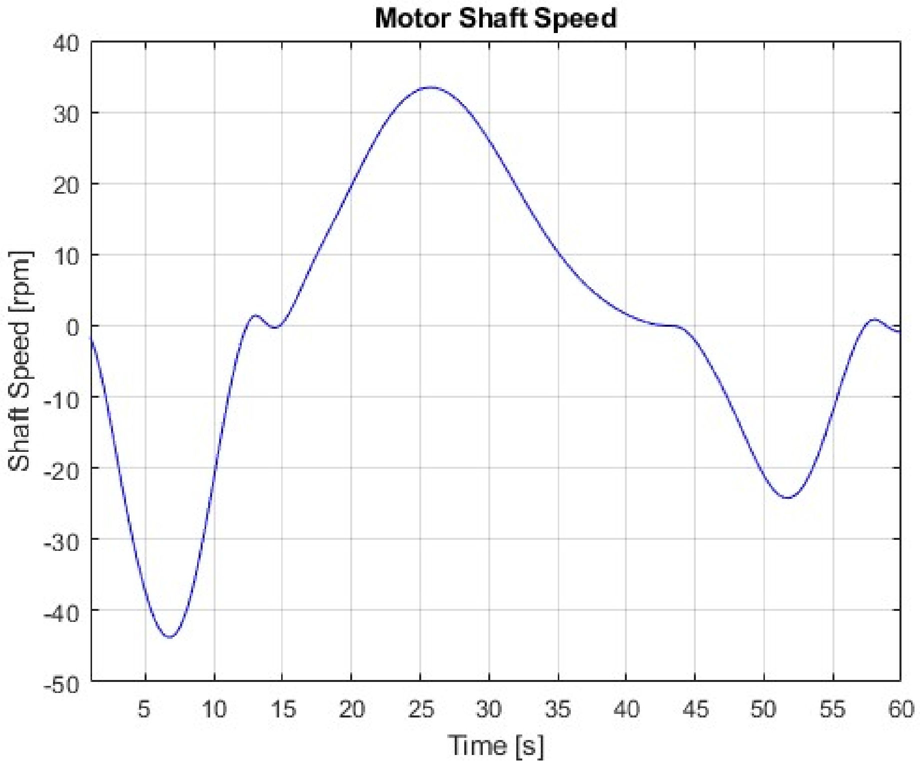

Figure 4.16: Motor shaft speed (ω) curve during elbow flexion\extension using LQR control

Image 4.16 shows a smooth profile for the motor speed, resulting in a gradual movement of the arm.

When comparing this result with the ones obtained previously 4.8 and 4.12 two things are notable: Firstly the speed profile of the LQR control is generally lower and more smooth with respect to the ones relative to the eigenvalue allocation strategy. This is to be expected since the basis for the LQR control is very similar. However the LQR offers an optimal solution and hence better results.

Secondly when confronted with the PID output data the results shown in 4.16 are pretty similar. However the PID shows to be more stable in the changes of direction. In this area the LQR performs worst since during the changes of direction is still present a slight overshoot. The presence of overshoot can be attributed to the exo-system approximation.

For what regards the motor input voltage this is its curve:

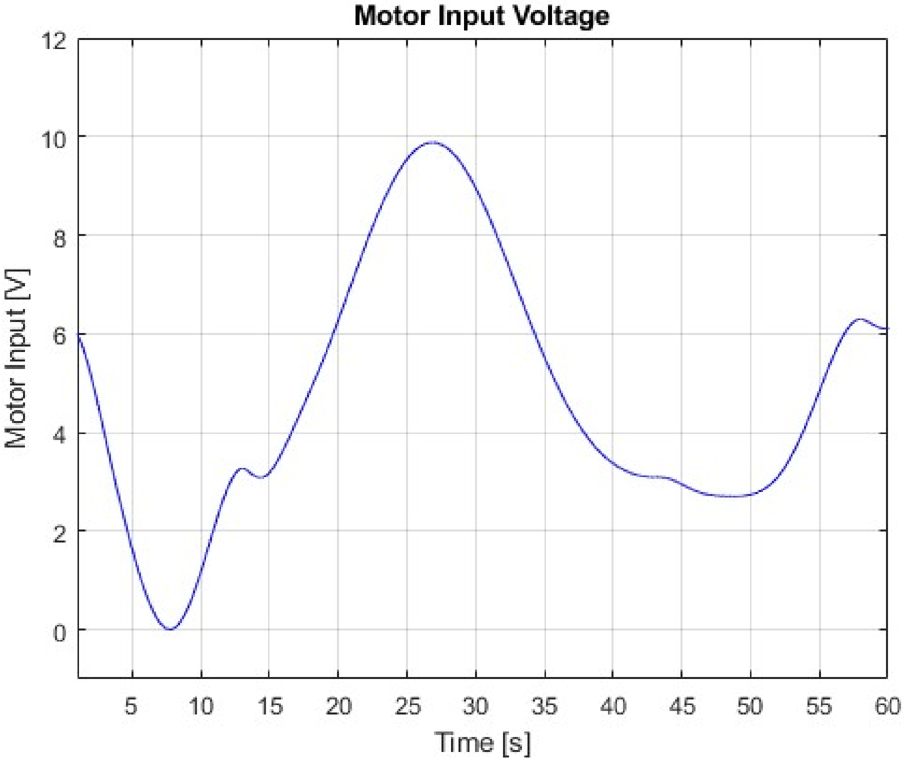

Figure 4.17: Motor voltage input profile during elbow flexion\extension using LQR control

Image 4.17 shows exactly what was expected. In fact this graph shows a smooth curve which never reaches saturation like 4.9 and 4.13.
Similar to the speed analysis the LQR shows better input voltage data. When confronted with the other results it shows a smoother curve then 4.13. This still was expected due to the results obtained during the speed analysis.
The smoothness of the voltage curve still confirms the fact that the shaft moves in a gradual way. This grants the absence of jerking on the patient's arm which could result in pain or discomfort.

4.2.4 Results Comparison

Now that all the simulation results are clear it is possible to draw the first conclusions.

Starting from the tracking capabilities the PID shows to be the best control. The second best is the LQR followed by the eigenvalue allocation control.

This is the most important parameter since the presence of overshoot translates into the motor moving more then intended. This could result in patient's discomfort and if too high could represent a risk.

This fact is also reflected in the reference errors. Images 4.7, 4.11, 4.15 clearly shows how the PID superiority in terms of tracking performances. In fact it has a peak tracking error of about 1.3 degrees and it also stabilizes around zero the fastest and nearest.

The second fact to keep in mind is the presence of the speed oscillations. In this area the PID control has the strongest and longest fluctuations. Yet the effects of these oscillations are very limited and almost none when considering the shaft behaviour.

However if needed the LQR control has much lower oscillation but at the cost of having a much higher overshoot.

Another non-negligible argument regarding the controller choice is its complexity. This fact is very important especially when the controller need to be applied to the real robot. Also in this field the PID shows to be the superior one, being much less complex then both the eigenvalue allocation and the LQR.

Another important fact is that the PID control resulted to be the most robust to changes in the reference signal, making it the most adaptable controller. In fact all the controllers have better performances when increasing the duration of the functioning cycle. Yet the state feedback controllers present worst results when using a faster reference signal. This does not happen with the same extent when using the PID.

All the simulations done via Simulink exploiting an accurate model of the motor. This means that the results obtained during simulations will be similar to the behaviour of the real system. Starting from this fact the PID seems to be the best choice for the application.

This is motivated by its tracking capabilities and overall satisfactory performances. Also due to its complexity the PID is the best choice for this application.

Another set of data analyzed during simulations is the error on the elbow flexion angle trajectory.

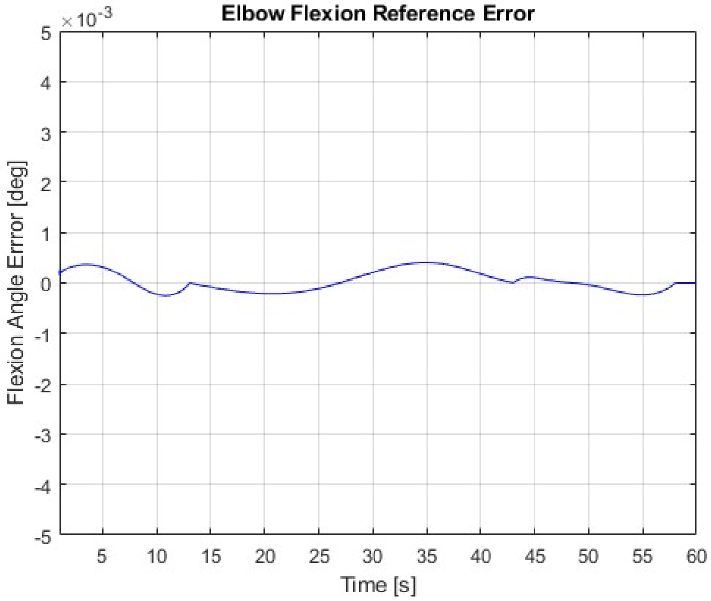

Figure 4.18: Elbow flexion angle error during flexion\extension movement with PID control

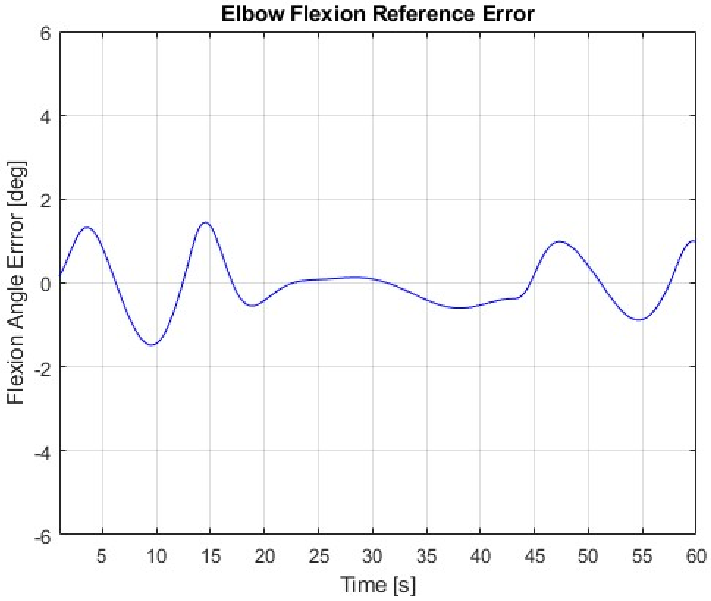

Figure 4.19: Elbow flexion angle error during flexion\extension movement with eigenvalue allocation control

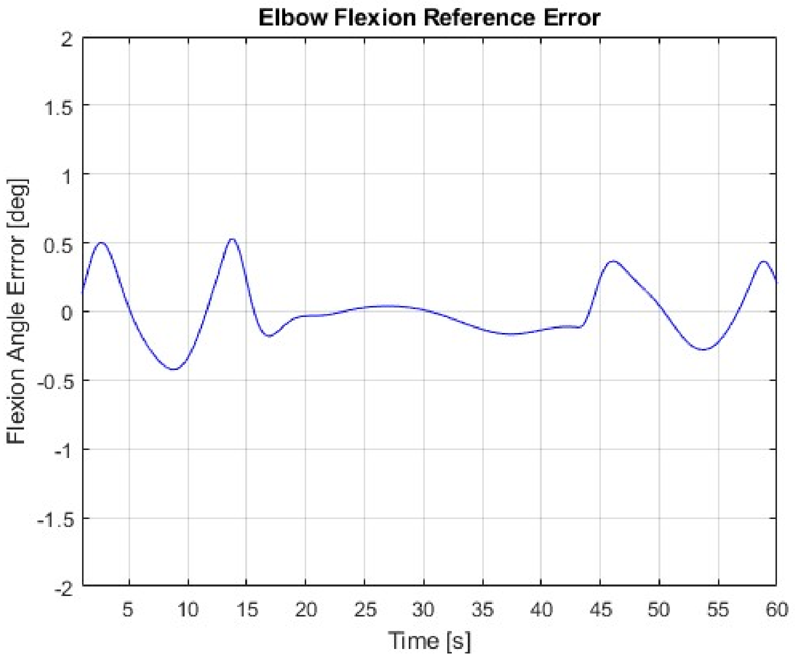

Figure 4.20: Elbow flexion angle error during flexion\extension movement with LQR control

From the previous figures (??,4.19, 4.20) it is possible to note that even when considering the error on the flexion angle the PID has the best tracking. This is to be expected since the PID shows to have the best performances in all the data gathered.

The comparison of these three figures shows two important facts: first of all it shows how the overshoot peaks presents in 4.10 and 4.14 translate into small errors in the elbow movement. This is the most important fact since ensures that patients arm is moved correctly.

The second fact is another confirmation that, when moving into the real motor test, the PID shows to be the best controller having the lowest tracking error also in the elbow trajectory.

Before tacking the real motor tests there is another data that needs to be checked from the simulations: the cable length.

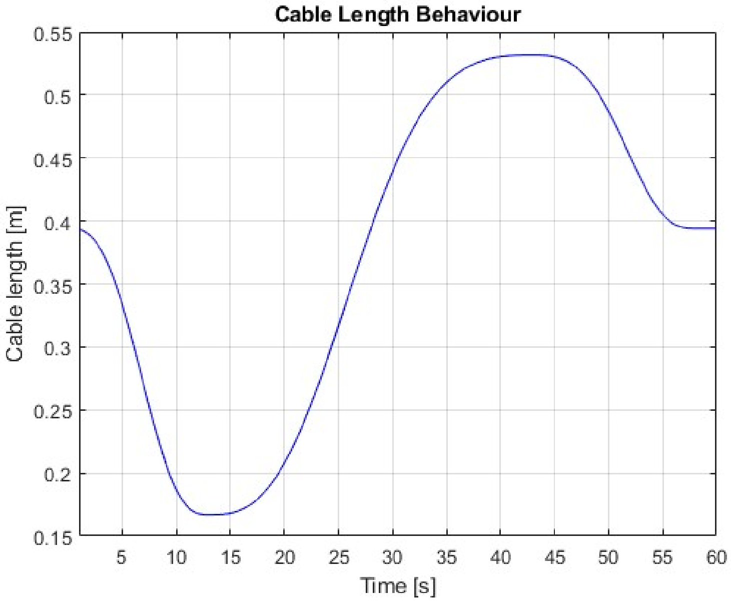

Figure 4.21: Cable length (l) during elbow flexion\extension with PID control

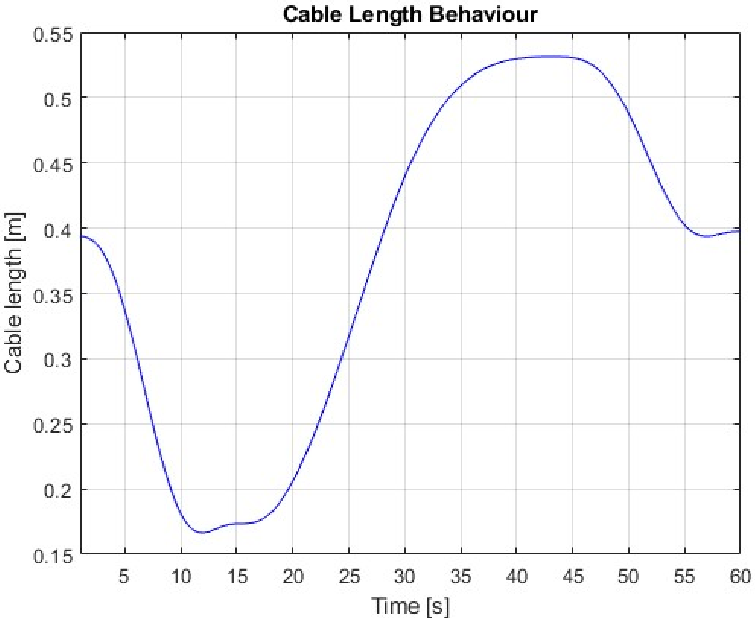

Figure 4.22: Cable length (l) during elbow flexion\extension with eigenvalue allocation control

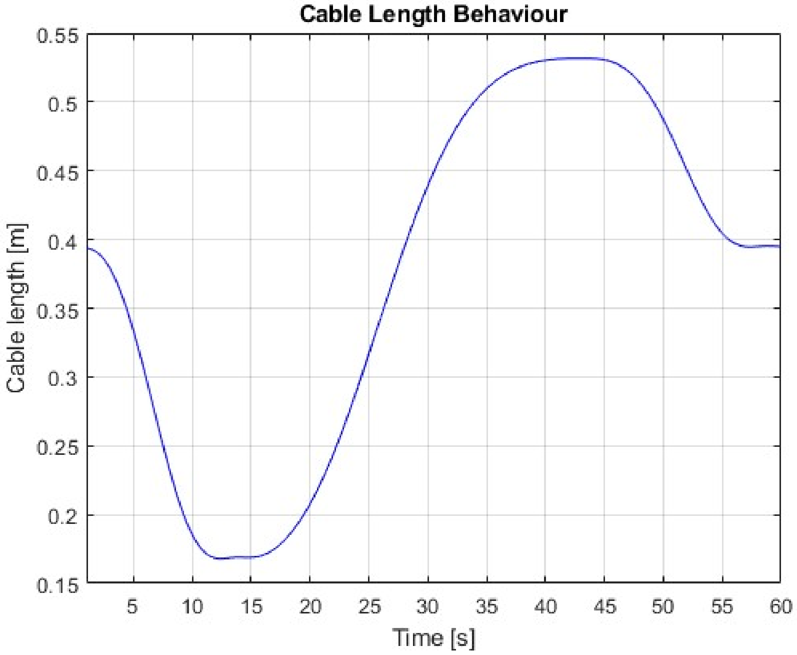

Figure 4.23: Cable length (l) during elbow flexion\extension with LQR control

From Images 4.21, 4.22, 4.23 it is possible to observe how the length of the cable varies during movement.

The curve described by these three plots is similar to both the motor shaft trajectory 2.3 and the flexion angle trajectory 4.4.

This fact is due because during the movement the cable is initially wound up on the pulley, explaining the shortening of the length. Then the cable gets unwound to allow the full extension of the arm, hence the increase of this parameter.

During the pauses the cable does not vary in length hence why the constant segments during the movement.

It can also be noted that the cable length final value is the same as its initial one. This is another confirmation of the correct functioning cycle of the device.

4.3 Experimental Results

The results achieved during testing require a brief preamble. When using the experimental setup Matlab is only used for the data acquisition.

Matlab results to be extremely complex to used when applied to the Dynamixel motors. This means that the control algorithm needs to be developed on the Arduino IDE, as a consequence the most complex controller cannot be implemented. This however does not constitute a problem since, as it was said before, the PID yields the best results. Another problem arisen when used the Arduino IDE is the computation of the reference signal. To solve this as a reference it is used a trapezoidal speed law. This can be done using a particular setting of the motor accessible via the Dynamixel library for Arduino. This library offers various function to setup the motor functions.

The first setup option used is the so called **time based drive mode**. The motor has two parameters called: **profile acceleration** and **profile velocity**. When in time based mode the first option allows to choose the duration of the acceleration phase, while the second determines the duration of the movement. By tuning these two parameter, coupled with the amplitude values obtained in the simulations, is possible to obtain a polynomial curve very similar to the one used during the simulations.

In terms of control the motors come with an internal PID structure, which values can be set by the user, for the trials they are set equal to the one obtained using the simulations.

Since the reference signal is different from the simulations the plot of the error is less significant. However there still is the need for a way to evaluate the plant behaviour. The methods used for evaluation of tacking capabilities are purely qualitative. They are based on three main factors: the motors do not have to present sudden changes in direction, this ensures that the motors do not perform jerking motions.

The peak values of θ needs to be the same reached in the simulations, this assures that each motor turns the correct amount of times.

The movement must have a long enough duration, this ensures that the movement respects the decided duration and that it is smooth enough.

After the shaft position, as it was done for the simulations, the next analyzed measurements are the shaft speed ω and the input voltage. Observe now the shaft behaviour during one full operation cycle:

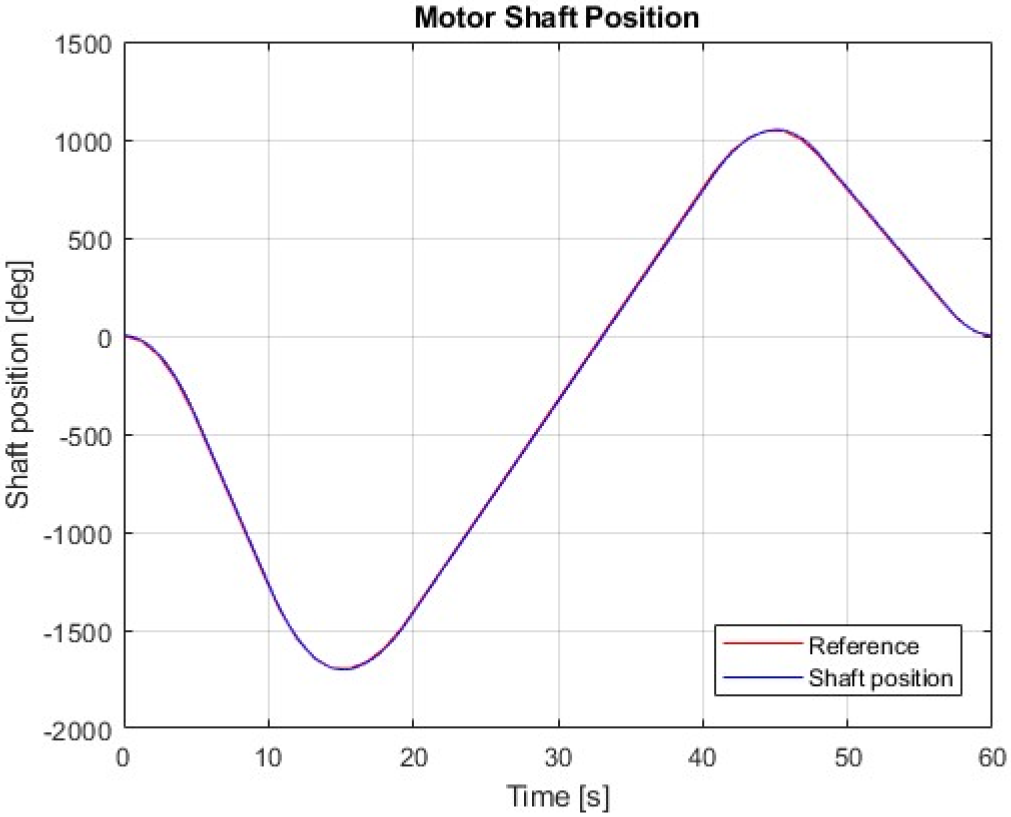

Figure 4.24: Motor shaft position (θ) profile during elbow flexion\extension with reference signal

As it expected the reference signal and the actual behaviour of the motor shaft are pretty different. This is due to the difference in calculation for the reference signal.

Figure 4.24 however shows how the motor still follows a polynomial behaviour. Not only this but also it reaches the same peaks without overshoot, also it starts and finishes at zero as wanted.

Another important feature to note is the duration, in fact the full cycle takes one minute to complete. This is wanted since it grants smoothness of the movements. Figure 4.24 also shows that the motor does not perform any jerking or oscillation.

This fact coupled with the smoothness and the absence of overshoot should avoid any pain or discomfort of the patient during the movements.

Lastly the fact that the final point corresponds with the initial one allows for multiple subsequent working cycle. This feature not only was wanted but is also fundamental when considering the physiotherapy application of the machine.

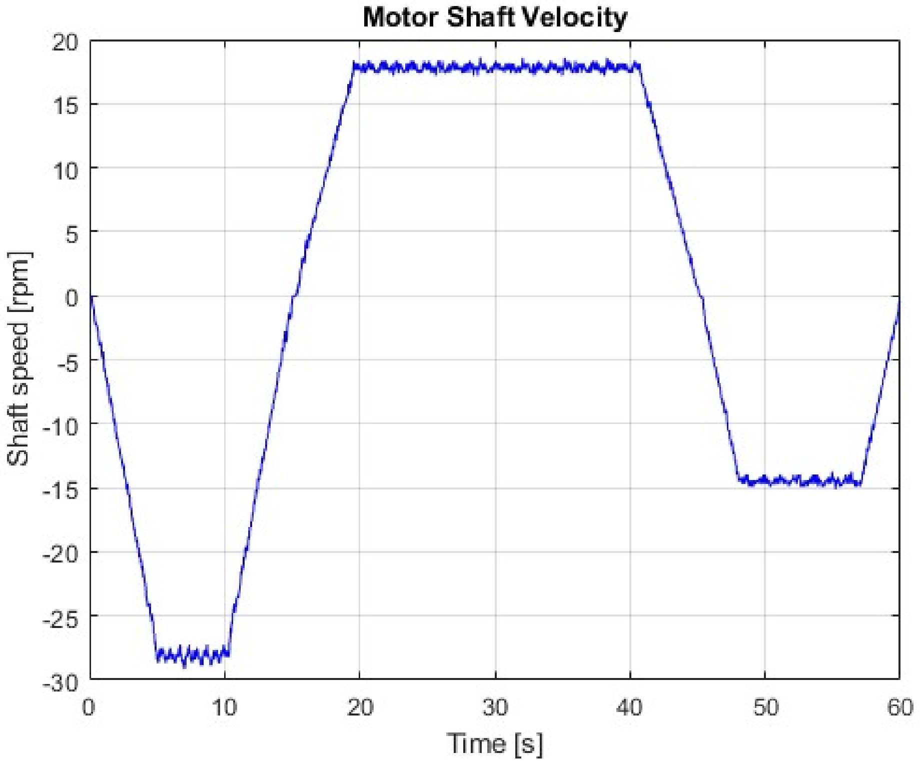

Figure 4.25: Motor shaft velocity (ω)profile during elbow flexion\extension

Figure 4.25 represents the speed behaviour of the motor. The speed follows a trapezoidal behaviour, this was expected due to how the reference was implemented via the Arduino IDE.

Compared to the results of simulations 4.8, the data obtained are much different. It is important however to notice that there are no oscillation at the start of the movement differently to what was observed during simulations.

This is due to due to the different implementation of the reference signal which exploits proprietary setup options of the motors.

Moreover form the speed graph it is possible to deduce two important points:

firstly the velocity never reaches very high values. This not only was wanted during the design process but also grants a gradual movement of the patient.
Secondly the speed profile is smooth enough in the changes of direction to avoid jerking in terms preventing pain or discomfort during use.

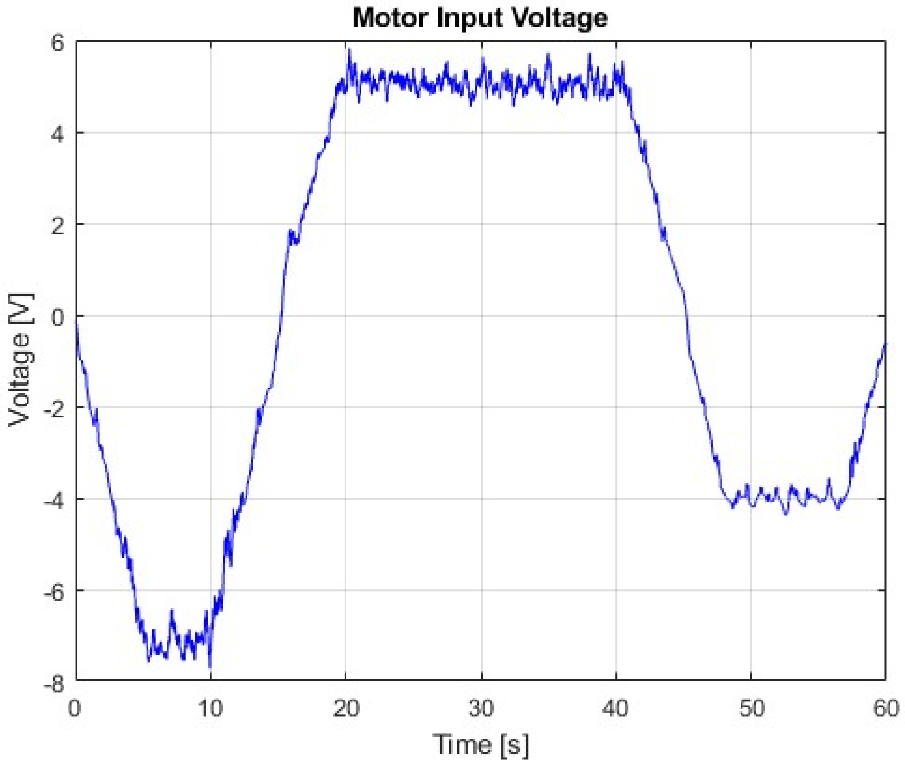

Figure 4.26: Motor input voltage curve during elbow flexion\extension

Image 4.26 displays the input voltage of the motor. Differently from the simulations results (4.9) the values obtained a lower. This is still due to the difference in implementation from the simulations.
However having a lower input voltage not only ensures that the motors run distant from the maximum value but also a lower impact on the battery allowing for longer working cycles.
Since the robot works with two motors always coupled it is important to check for the synchronization of them.
As it was done previously even when checking for the correct synchronization in position, velocity and input voltage.

Position and speed represent the most important data since even a slight desynchronization could result in a wrong movement pattern or could cause pain for the patient.

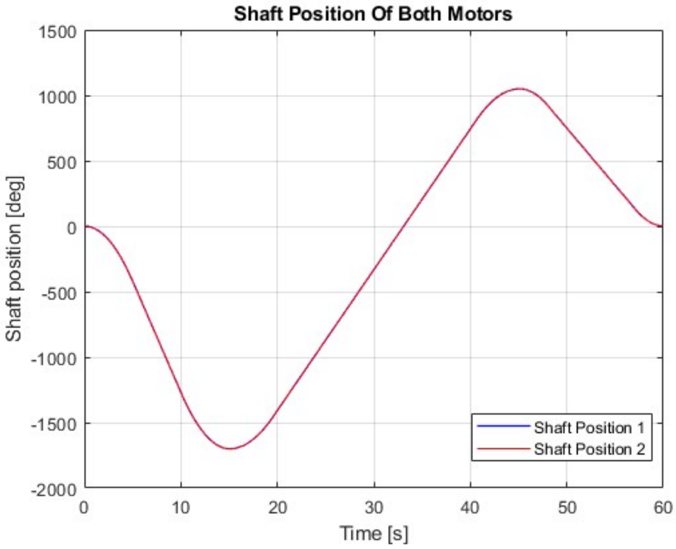

Figure 4.27: Shaft position (θ) profile of both motors during elbow flexion\extension

Figure 4.27 represent the movement profile of both motors. It is clear that both are synchronized during the whole length of the cycle.

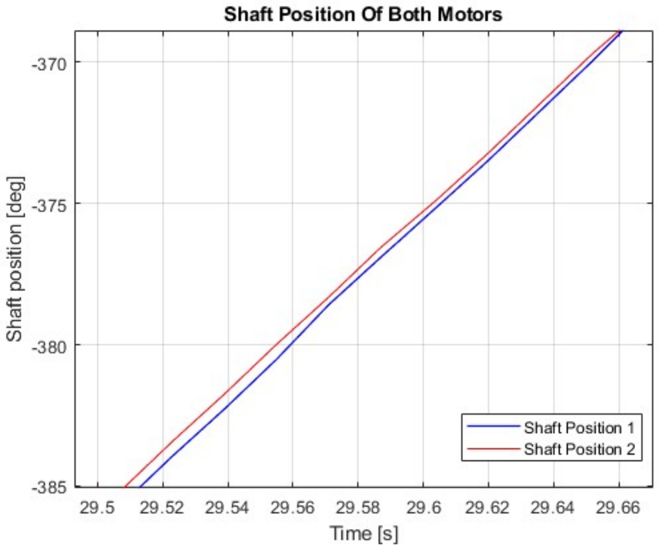

Figure 4.28: Zoomed shaft position of both motors during elbow flexion\extension

Image 4.28 represents a zoomed version of image 4.27. From 4.28 it is possible to note that there is a slight delay on the motors movement. However it is so small to be considered negligible during the working cycle.

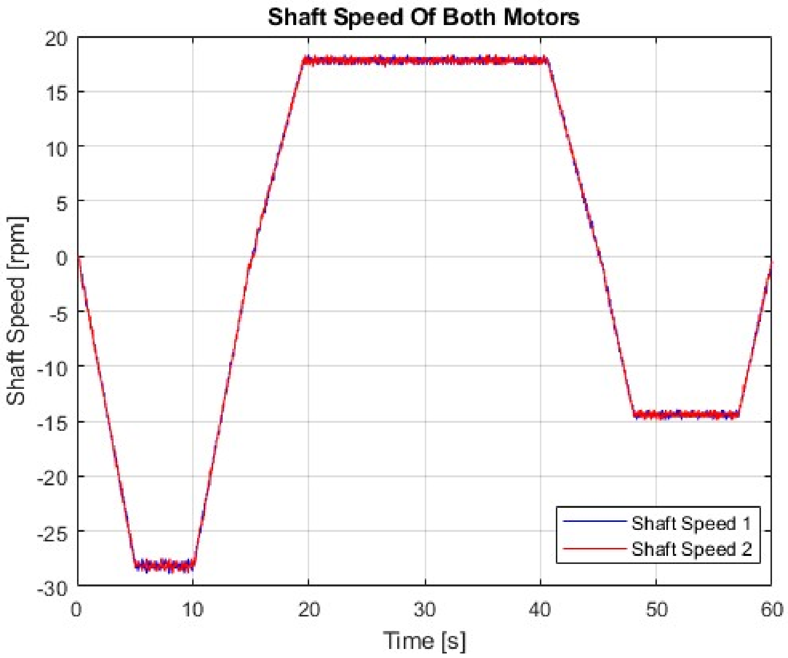

Figure 4.29: Shaft speed (ω) profile for both motors during elbow flexion\extension

Figure 4.29 shows the velocity profile for the motors during movement.

This image also show that even in terms of speed the motors are synchronized as wanted, not only in terms of time but also in terms of value the speed profile are basically overlapped.

The last value to analyze is the input voltage. After the consideration done on 4.27 and on 4.29 it is reasonable to assume that very similar reasoning could apply even for this case.

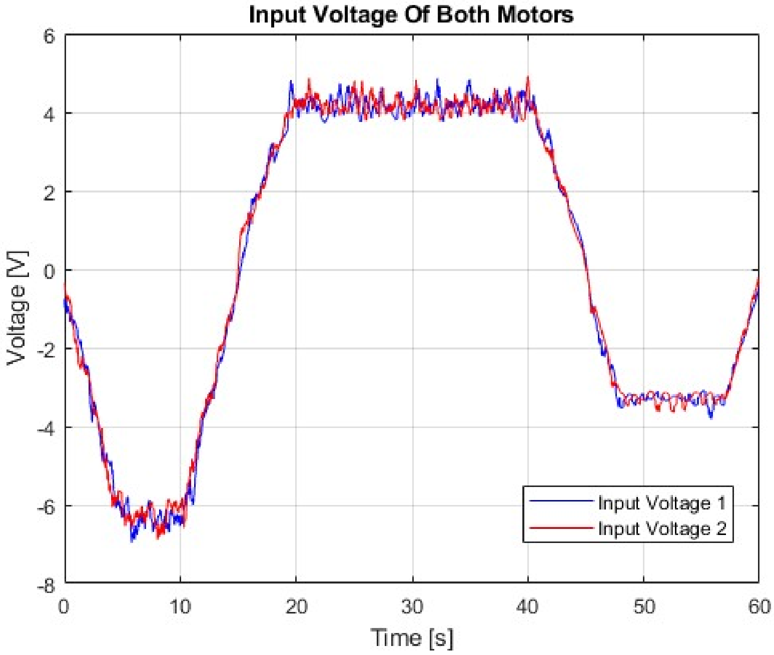

Figure 4.30: Input voltage behaviour for both motors during elbow flexion\extension

Image 4.30 shows the input voltage of the motors, the voltage curve are also almost overlapped. This not only is wanted but also expected due to all the results obtained previously.
Image 4.30 is the last confirmation that both motors work correctly both in terms of tracking capabilities and synchronization.

Conclusions

This thesis work started with the idea of developing a robot for arm rehabilitation purpose.

The first point to tackle was the design of the prototype. It was decided that best design choices were: using two motors and cables to move the arm.

The choices in the design were dictated by one main guiding principle: portability. This meant that the robot must be light and flexible hence why the choice of two motors and cables instead of an exoskeleton.

The next step was the choice of motors that needed to be light and strong enough to move the arm, these problems was solved using the Dynamixel XL430-w250-t.

To control the motor an Arduino mega was used with the Dynamixel shield to interface it with the motors.

After the hardware choice it was necessary to design the movement profile of the motor. It was decided to use a fifth grade polynomial curve due to its adaptability and smoothness.

To control the motor it was decided to implement three different algorithm: PID, eigenvalue allocation and LQR. The first one is only based on the reference error and is also the least complex, the last two exploit the state space description of the system.

All three controllers were then modelled via Matlab and Simulink. Then a series of simulations was performed to better understand the performances of the controllers in term of tracking capabilities, reference error, velocity and input voltage.

For all the controller's simulations the aim was to replicate an elbow flexion of $[-60°, 60°]$.

When performing the tests the PID showed the best tacking results having a flexion angle peak error of about $0.5 * 10^{-3}$ degrees. The eigenvalue allocation strategy showed the worst tracking results having a flexion angle peak error of about 1.5 degrees. The LQR control showed better results then the eigenvalue allocation strategy, having a flexion angle peak error of about 0.5 degrees. Yet the PID showed to be the best option since, for safety reasons, the movemnet needs to be very tightly controlled.

After the simulations the PID resulted to be the best performing controller. This coupled with the easy implementation characteristic of the PID control were the driving factors of the final controller choice.

After the simulations a series of tests on the motor were performed to check that the control satisfied all the requirements in terms of tacking capabilities, time and smoothness.

The results of the empirical trials were satisfactory and encouraging in further developing of this machine. Possible further experimentation may involve new material or mew mechanical configurations. In terms of control new algorithm exploiting artificial intelligence and different sensors. Also a more advance human machine interface could grant a more personalized experience for the final user.

Appendix A

Glossary

A.1 Glossary

- **ABDUCTION** - *abduzione*

  Movement responsible of straightening the arm.

- **ADDUCTION** - *adduzione*

  Movement responsible of flexing the arm at the elbow.

A.2 Mathematical Definitions

- **Reachability** problem: Which is defined in the following way: what is the set of states which can bring the system starting from the origin (a.k.a. $x(0) = 0$) to a final state $x(t) = \mathbf{x}$.

- **controllability** problem. It is defined as: what is the set of states which can bring the system, starting from an initial state (different from 0) to a final state $x(t) = 0$.

- **matrix invariant** A vector-subspace is said to be F-invariant if said $x \in \mathcal{X}$ then also $Fx \in \mathcal{X}$.

A.3 Keywords

Inglese	**Italiano**
elbow rehabilitation	riabilitazione del gomito
wire-driven robot	robot a cavi
position control	controllo in posizione
robot-aided rehabilitation	riabilitazione assistita da robot
feedback control	controllo in retroazione
state-space control	controllo in spazio di stato

Bibliography

[1] [Online]. Available: https://www.hss.edu/conditions_elbow-arthritis-rehabilitation-physical-therapy.asp

[2] [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3435939/

[3] [Online]. Available: https://bodysmart.healthcare/arm/

[4] C. Xiaohong, W. Binrui, L. Han, and C. Jianu, "Design and passive training control of elbow rehabilitation robot," *electronics*, 2021.

[5] M. Rahman, M. Saad, J. Kenné, and P. Archambault, "Exoskeleton robot for rehabilitation of elbow and forearm movements," in *18th Mediterranean Conference on Control and Automation, MED'10*, 2010, pp. 1567–1572.

[6] N. Vitiello, T. Lenzi, S. Roccella, S. M. M. De Rossi, E. Cattin, F. Giovacchini, F. Vecchi, and M. C. Carrozza, "Neuroexos: A powered elbow exoskeleton for physical rehabilitation," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 220–235, 2013.

[7] V. Oguntosin, W. S. Harwin, S. Kawamura, S. J. Nasuto, and Y. Hayashi, "Development of a wearable assistive soft robotic device for elbow rehabilitation," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2015, pp. 747–752.

[8] B. Matteo, C. Marco, M. C. Cuauhtemoc, and R. Giulio, *Advances in Italian Mechanism Science.* Italy: Springer Nature Switzerland, 2021.

[9] Z. Giacomo, B. Matteo, C. Marco, and R. Giulio, “Design and performance of an elbow assisting mechanism,” *Machines*, vol. 8, pp. 404–420, 2020.

[10] L. Med Amine, C. Marco, S. Juan, B. Matteo, and R. Giulio, “Experimental validation of light cable-driven elbow-assisting device l-cadel design,” *Journal of Bionic Engineering*, 2022.

[11] C. Marco, R. Mykhailo, F. Axel, R. Matteo, L. Med Amine, and U. Monica, “Design and experimental characterization of l-cadel v2, an assistive device for elbow motion,” *Sensors*, 2021.

[12] L. Hoppe, “Performance improvement of dyneema(r) in ropes,” in *Oceans ’97. MTS/IEEE Conference Proceedings*, vol. 1, 1997, pp. 314–318 vol.1.

[13] T. Francesco and A. Riccardo, *Position PID–control of a DC servomotor*, March 2021.

[14] G. Rosati, *MODELLO DEL MOTORE IN CORRENTE CONTINUA*, October 2020.

[15] F. E. and M. G., *Appunti di Teoria dei Sistemi*, edizioni libreria progetto padova ed. Padova: Libreria Progetto, 1988.

[16] M. B. Maria Elena Valcher, *Controlli Automatici.* libreria progetto, 2015.

[17] C. Bohn and D. Atherton, “An analysis package comparing pid anti-windup strategies,” *IEEE Control Systems Magazine*, vol. 15, no. 2, pp. 34–40, 1995.

[18] W.M.Wonham, “The internal model principle of control theory,” 2018.

Ringraziamenti

In quest'ultima parte della tesi volevo ritalgiare lo spazio per alcuni ringraziamenti:
A cominciare dalla mia famiglia, mia mamma, mio papà e mia sorella i quali mi hanno sempre sostenuto in ogni momento di difficoltà e ascoltato ogni mio problema. Se non fosse per loro probabilmente non sarei arrivato dove sono oggi.

Ai miei amici Giacomo Z. Lorenzo M. e Gabriele R. Amici fedeli ancora da prima di inizare questo percorso universitario. Con loro ho avuto la fortuna di condividere molto e non potrei desiderare amici migliori.

Ai miei amici Irene A. Michele P. Giacomo C. Valentina Z. e Emanuele S. Con voi ho avuto la fortuna di condividere questo percorso universitario. Abbiamo affrontato insime molti scogli e mi ritengo onorato ad avere avuto la possibilità di condividerer questo percorso con voi. Ognuno di voi ha suo modo mi ha ispirato.

Alla mia fidanzata Aurora M. Dulcis in fundo sei stata la persona che più mi ha capito, in breve tempo hai campiato la mia vita in un modo che non mi sarei mai aspettato e non tornerei mai indietro. Con te ho condiviso tanto e non vedo l'ora di condividere ancora molto altro.