CHALMERS
UNIVERSITY OF TECHNOLOGY

Intelligent Elbow Orthosis

The construction of a mechatronic physiotherapy device

Gustav Bardun, David Silberberg, Oliver Silfveroxel

DEPARTMENT OF ELECTRICAL ENGINERING
MAY 10, 2024

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

1

Acknowledgements

We want to extend our gratitude to our supervisors, Emmanuel Dean and, Rita Laezza. This project would not have been possible without your guidance and enthusiasm. We also want to thank Adam, Filip, and many others active at the C.A.S.E lab. The experiences and tips shared with us have been invaluable and has aided our design effort tremendously.

An appropriate thanks to the many coffee shops around Chalmers. Although work would have been done without you, we would have been slightly more tired while doing it.

2

Abstract

Physiotherapy is crucial for patients with pain or reduced function of muscles and joints. While mechatronic solutions exist to aid care, they are costly, complex, and lack portability. The goal of this project was to construct a smart elbow orthosis to enhance physiotherapy, allowing more accessible and personalized care. This project does not make any claims regarding the clinical efficacy or potential use of the system. The project has consisted of developing multiple different systems from the ground up, including an exoskeleton, tactile sensors, data communication system, and control system. Most of these systems were merged successfully and their functions evaluated.

The primary issues encountered concern the communication between devices, and the integration of the controlling algorithms, both of which seem solvable. This limits the current use of the system in following pre-determined trajectories. Data from the tactile sensors can still be collected, though at its current stage, cannot be used to adapt the motion of the orthosis. Although the implementation is not complete, the core idea shows promise, meaning mechatronic solutions might be a valuable step in reducing barriers of entry, and improving care in physiotherapy.

Contents

3

Introduction

3.1 Background

Physiotherapy is a key tool in primary healthcare for treatment of discomfort and pain among patients. It can help patients maintain their physical functionality and reduce musculoskeletal pain [1]. Access to this treatment is however limited by, for example, the availability of human resources which is in short supply [2]. Further, as stated in [3], a significant portion of patients do not complete their assigned exercise regiments. This is due to a multitude of factors, most being related to accessibility. Patients may need to cover large distances to meet with physiotherapists, and, in some instances, pay for the treatment themselves. Thus, physiotherapy can be a large economical burden for patients. Some exoskeletons have been developed for rehabilitation purposes which could reduce the demand on manpower in the intermediate phase of rehabilitation, while also benefiting the recovery by providing data to the physiotherapist [4]. However, since these exoskeletons are expensive and lack portability, the economical obstacle remains and patients still need to transport themselves to the clinic. Since the accessibility issues remain, the benefits these robots can bring to the rehabilitation process are still limited.

3.2 Purpose

The purpose of this project is to build an elbow orthosis to be used for supporting physiotherapy treatments. The system includes a motor to control the motion of the elbow and tactile sensors to provide feedback from the user's motion.

Inspired by similar attempts [2] and with the aforementioned accessibility issues in mind, this project will strive to reduce certain barriers that faces physiotherapy today. By developing a portable system that can be used by the patient at home, high-precision exercises could be achieved without the presence of a physiotherapist, thus removing barriers regarding geographical distance and transportation time. Another target is for the system to be easy to use both for the patient and the physiotherapist. Therefore, it can be utilized with as little effort as possible while maintaining the benefits of the system. Moreover, the economic demands of the system will be challenged by keeping the expenses as low as possible. Since the behaviour of the orthosis will be based on data gathered by applied sensors, an intelligent interpretation of this information is to be investigated such that this could benefit the patients during their exercises. Hence, making the rehabilitation process more efficient.

3.3 Scope

The project aims to implement a system for physical rehabilitation through the continuation of existing research and development on intelligent elbow orthoses [2]. By simplifying existing designs, this system should provide an easier-to-use system that gives qualitative data regarding mobility and torque in the elbow. The scope of the project is built upon certain requirements based on the feasibility of the system to be used with a clinical purpose along with limitations to make exploration of such a system feasible.

Attachment of the orthosis to an arm is limited to fit a specific individual. This mechanical design should provide an efficient transfer of torque from the actuator to the forearm. It should be lightweight enough so that it does not limit the patient's ability to perform the motion while also, for safety reasons, be limited to the natural range of motion of the elbow joint. Modularity of the skeleton design is preferred in order to allow for multiple choices of actuators and sensor types, this will however not be prioritized within the scope of this project.

The actuator follows similar limitations as the skeleton. The motor needs to provide an output torque strong enough to lift an arm unassisted, as well as being small enough to easily fit on the skeleton. However, for safety reasons, it should also have a maximal output torque small enough to be overcome by healthy muscular strength. Safety barriers limiting the range of motion and torque provided in the actuator are also to be implemented into the software.

Concerning the dynamical modeling of the system, the shoulder joint will be considered to have a fixed position where the upper arm is in a vertical orientation. However, deviations from this position, which could impact the functionality of the system, are to be presented in a simulation. This should present the

positioning of the entire arm, including orientation of shoulder and elbow joint in a graphical user interface (GUI).

The project will not make any claims regarding the clinical efficacy of the system, and the apparatus may not be tested on humans. The main goal will be to control the actuator with the graphical user interface, GUI, combined with the sensory feedback data. This at a fast enough rate to maintain a stable system. The result should hopefully aid further research on the use of intelligent orthosis systems in rehabilitative treatments by providing a simple framework with a high degree of expandability.

4

Theory

4.1 Modelling in robotics

In robotic applications, the execution of any specific motion requires an accurate analysis of the characteristics of the mechanical structure, actuators, and sensors [5]. Such analysis aims to derive mathematical models to provide the control system with an input-output relationship that encapsulates the robotic system as a whole. This section will provide a theoretical framework used in finding the mathematical models required for motion and control of the intelligent elbow orthosis, taken from the collective work of Bruno Siciliano et al in [5].

4.1.1 Kinematics

Kinematic modelling describes motion with respect to any fixed Cartesian coordinate frame, ignoring forces and moments that give rise to such movements [5]. This relationship allows the study of two fundamental problems in robotics: the direct- and inverse kinematics problem. The forward kinematics aims to determine the relation of the motion of the robot's joint with the motion of the end-effector. This relation is usually defined in the form of algebraic equations. The inverse problem transforms a desired motion of the end-effector into corresponding joint motions.

A rigid body can be completely described in 3D space by its pose, where pose is defined as the position and orientation of a body with respect to a reference frame. Using an orthonormal coordinate frame attached to the rigid body, we can describe it's position by the vector pointing from the reference frame origin

O to the body frame origin O', and it's orientation with the body-frame axis (x', y', z'). Figure 4.1 shows the pose of a rigid body in 3D space.

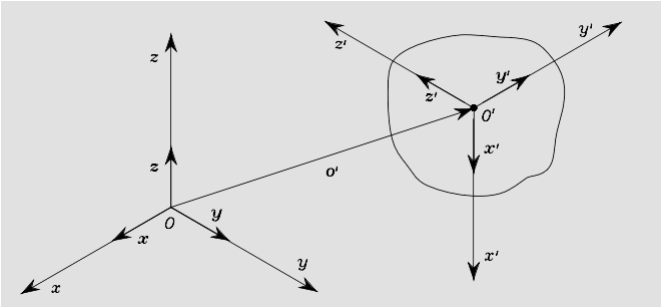


Figure 4.1: Position and orientation of a rigid body. Adapted from [5]

A robotic manipulator can be represented as a kinematic chain of links, connected to one another by either revolute or prismatic joints. One end of the kinematic chain is connected to the base, and one end is connected to the end-effector, as illustrated in Figure 4.2.

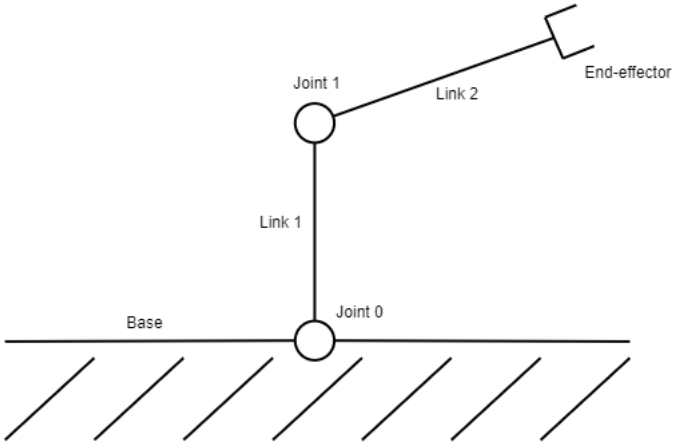


Figure 4.2: Robotic manipulator arm

The total motion of the structure can be described by composing the elementary motion of all bodies relative to the coordinate frame of the previous body in the chain. Thus, the end-effector pose can be represented with respect to the base frame as a function of joint position variables.

The pose of any coordinate frame relative to another frame can be represented by a series of *rotation* and *translation* operations. Rotation of frame $\{i\}$ to form frame $\{i'\}$ can be described by the matrix $\mathbf{R} \in SO(3)$, given by

$$\mathbf{R}^{i}_{i'} = \begin{bmatrix} r_{xx'} & r_{xy'} & r_{xz'} \\ r_{yx'} & r_{yy'} & r_{yz'} \\ r_{zx'} & r_{zy'} & r_{zz'} \end{bmatrix}, \tag{4.1}$$

where each element is the scalar product of the corresponding unit vectors. The translation property is described by the position vector $\mathbf{O'} \in R^{3\times1}$, as shown in Figure 4.1, given by

$$\mathbf{O'} = \begin{bmatrix} p_{x'} \\ p_{y'} \\ p_{z'} \end{bmatrix}. \tag{4.2}$$

These elementary manipulations can be represented as a linear transformation by expanding their dimensions. This linear transformation is a homogeneous transformation matrix $\mathbf{T}^{i}_{i'} \in R^{4\times4}$ that completely describes the position and orientation of a rigid body in frame $\{i\}$ with respect to frame $\{i'\}$, given by

$$\mathbf{T}^{i}_{i'} = \begin{bmatrix} r_{xx'} & r_{xy'} & r_{xz'} & p_{x'} \\ r_{yx'} & r_{yy'} & r_{yz'} & p_{y'} \\ r_{zx'} & r_{zy'} & r_{zz'} & p_{z'} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.3}$$

Such matrices that describe motion between sequential links can be combined to give a matrix $\mathbf{T}^{0}_{n}$ that describes the end-effector with respect to the base frame for any amount of connected bodies. This matrix is given by:

$$\mathbf{T}^{0}_{n} = \mathbf{T}^{0}_{1} \cdot \mathbf{T}^{1}_{2} \ldots \cdot \mathbf{T}^{n-1}_{n}. \tag{4.4}$$

The Denavit-Hartenberg (DH)convention simplifies the definition of the direct kinematics function through a systematic, general method to define the position and orientation of two consecutive link coordinate frames as functions of their joint variables [5]. The DH method provides a set of rules for defining each link frame relative to the previous. Once established, the configurations of each link are completely specified by four parameters: link length a, link offset d, joint angle θ, and link twist φ.

a_i : distance between $O_{i'}$ and O_i along x_i

d_i : coordinate of $O_{i'}$ along z_{i-1}, variable for prismatic joints

θ_i : angle between axis x_{i-1} and x_i about axis z_{i-1}, variable for revolute joints

φ_i : angle between axis z_{i-1} and z_i about axis x_{i-1}

The resulting coordinate transformation matrix in Eq. (4.5) shows the rotational and translational relationships between two consecutive link frames interconnected by a revolute joint variable q_i.

$$\mathbf{T}^{i}_{i'}(q_i = \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\varphi_i) & \sin(\theta_i)\sin(\varphi_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\varphi_i) & -\cos(\theta_i)\sin(\varphi_i) & a_i\sin(\theta_i) \\ 0 & \sin(\varphi_i) & \cos(\varphi_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.5}$$

Differential kinematics

Differential kinematics relates the joint velocities to the corresponding linear and angular velocities of the end-effector, characterized by the *geometric Jacobian*. Considering an $n - DOF$ manipulator, the Jacobian matrix is written in the form

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix}, \tag{4.6}$$

where $\mathbf{J}_P \in \mathbb{R}^{3\times n}$ relates the joint velocities to the end-effector *linear* velocity, and $\mathbf{J}_O \in \mathbb{R}^{3\times n}$ relates the joint velocities to the end-effector *angular* velocity; the resulting jacobian is $\mathbf{J} \in \mathbb{R}^{6\times 1}$. Assuming the joint is revolute, deriving the linear and angular velocities from the position vectors and corresponding rotational axes gives the contribution of the velocity from the single joint i to the end-effector e as

$$\begin{bmatrix} \mathbf{J}_{P_i} \\ \mathbf{J}_{O_i} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix}, \tag{4.7}$$

where $\mathbf{z}_{i-1}$ is given by the first three elements of the third column in the rotation matrix $\mathbf{R}^0_{i-1}$, $\mathbf{p}_e$ and $\mathbf{p}_{i-1}$ are given by the first three elements of the fourth column in the transformation matrices $\mathbf{T}^0_e$ and $\mathbf{T}^0_{i-1}$ respectively. The linear part of the Jacobian can be used to calculate the torque applied in the base frame by an external applied force by

$$\tau_f = \mathbf{J}^0_P \mathbf{F}^0_i, \tag{4.8}$$

where the external force $\mathbf{F}_i$ in frame $\{i\}$ is referenced to the base frame $\{0\}$ using the corresponding rotation matrix in Eq. (4.1) through

$$\mathbf{F}_i^0 = \mathbf{R}_i^0 \mathbf{F}_i. \tag{4.9}$$
$$\mathbf{F}_i = [F_x, F_y, F_z]^T, \{F_x, F_y, F_z \in \mathbb{R}\}$$

4.1.2 Dynamics

In the development of the dynamic model for the robotic manipulator, the Lagrange-Euler formulation is applied; a method esteemed for its efficacy in capturing the dynamics of complex mechanical structures [5]. The *Lagrangian* of the mechanical system can be defined as

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}) \tag{4.10}$$

where $\mathcal{T}$ and $\mathcal{U}$ denote the total *kinetic* and *potential* energy, respectively. The Lagrange equations are expressed by

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \tag{4.11}$$

where ξ_i is the corresponding force associated with the coordinate q_i.

Computing the total kinetic and potential energy of the system, and taking the derivative as required by the Lagrange equation in Eq. (4.11) gives the general dynamic model as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \tag{4.12}$$

Where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the viscous friction matrix, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ is the gravitational torque, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is the Coriolis matrix, $\boldsymbol{\tau} \in \mathbb{R}^{n \times 1}$ is the input torque, $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^{n \times 1}$ are the joint position, velocity, and acceleration, and n is the number of degrees of freedom (DOFs) for the system.

4.2 Control theory

Control systems play a pivotal role in dictating the behaviour and performance of any robotic system, with a fundamental goal of achieving desired actions through precise commands and regulations of the mechanical components. The

following sectopm presents a theoretical foundation upon which the control system for the robotic exoskeleton will be designed. Subsequent content is based upon the concepts and definitions provided by Bengt Lennartson in [6].

4.2.1 Open-loop, closed-loop control systems

Control systems can be classified based on their feedback path. For an open-loop system, the output signal is not fed back to the input. Thus the controller acts independently from the output. Whereas for a closed-loop system, also called a feedback control system, the output is continuously fed back to the input. This can be done through positive feedback, where the output is added to a reference signal input, or negative feedback, where the output is subtracted from the reference signal. The following Figures illustrate the concepts of open loop and negative closed loop configurations.

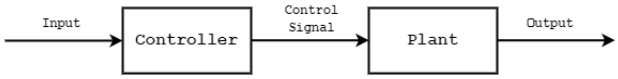


Figure 4.3: Open loop control, taken from [6]

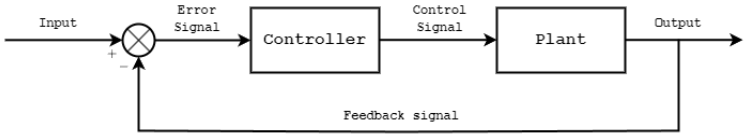


Figure 4.4: Negative feedback control, , taken from [6]

4.2.2 Output regulation

The choice of controller hinges on the specific requirements set by the system it is meant to regulate. Classical controllers, such as P, PD, and PID offer a straightforward way to regulate an activity based on the error signal they receive. The proportional component (P) produces an output proportional to the current error value through a gain coefficient Kp; this is a simple control approach that in the presence of external perturbations will show steady-state errors [6]. The derivative component (D) is prediction-based component that considers the rate of change of the error signal, applying a control action proportional to how quickly the error is changing; a counteracting measure that improves stability and dampens the response. The integral term (I) aims to eliminate residual steady-state errors; if the present error has been present for some time, the integral component increases over time thereby increasing the control effort. This effort can also introduce overshoot and increase oscillations.

A thorough system analysis is often required to tune these individual components to achieve desirable regulation of the system's behaviour. The method of tuning effectively requires balancing responsiveness (how quickly the controller reacts to changes [P]), stability (the ability of the system to reach and maintain a steady state without excessive oscillations [D]), and accuracy (how closely the system reaches the desired setpoint [I]) [6]. Other factors, such as the controller's ability to react and compensate for an external load, also play a big role in the tuning of said parameters. It is therefore crucial to investigate the conditions for which the system will be used, to choose a suitable control method.

4.2.3 Nyquist frequency

In order to maintain stable data transmission and control, sampling of a continuous signal into discrete elements has to be done with at least the rate of the Nyquist frequency [7]. That is, twice as fast as the frequency of our system, described as a pendulum around the elbow joint. For small amplitudes [8], the motion of a pendulum with one degree of freedom approximates a harmonic motion with natural frequency, f_n, given by

$$f_n = \frac{1}{2\pi}\sqrt{\frac{g}{L}}, \tag{4.13}$$

where L is the pendulums' length and g is the gravitational acceleration. Thus, to promote stable data transmission, a condition for the sampling rate, f_s, can be expressed as

$$f_s \geq 2\, f_n = \frac{1}{\pi}\sqrt{\frac{g}{L}}. \tag{4.14}$$

4.3 Trajectory generation

Some exercises, defined by a physiotherapist, can be described as one-dimensional functions of a single degree of freedom, i.e., the angle of the elbow joint, as a time function value. The trajectories generated can subsequently serve as reference signals for system controllers. A selection of such trajectories are described below along with adjustable parameters allowing for variations of the different types of trajectories. Common trajectory parameters for the different types of trajectories are start time, t_s, and duration of the exercise, t_d, where the starting time is indirectly adjustable since the user can decide when to begin the

exercise in the GUI. Thus, the general domain of trajectories is described in Eq. (4.15), where t is the time variable.

$$S_t := \left\{ t : \; t_s \leq t \leq t_s + t_d \right\} \tag{4.15}$$

4.3.1 Smooth transition

Starting from the initial useful function in Figure Eq. (4.16), shown in Figure 4.5, it is possible to derive smooth transition functions as follows.

$$\alpha_1(t) = 3\,t^{\,2} - 2\,t^{\,3} \tag{4.16}$$

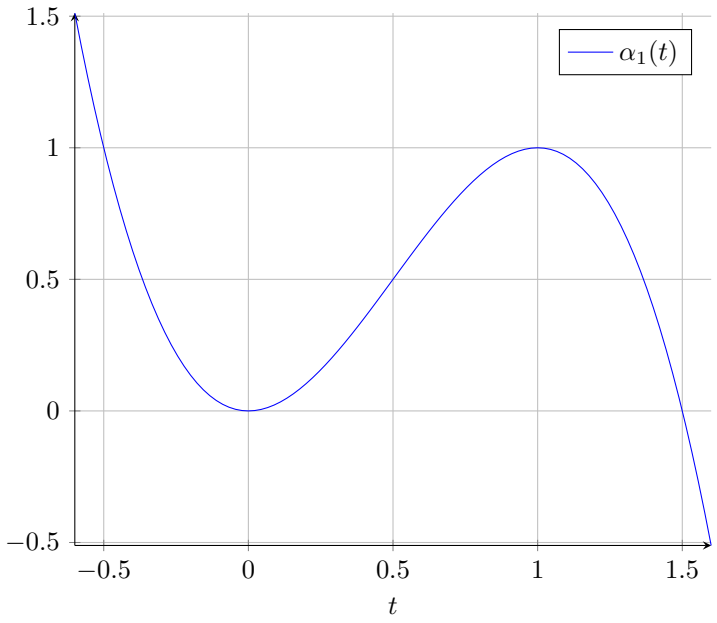


Figure 4.5: Smooth transition basis

The interval containing the smooth transition part, $t \in [0, 1]$, of Eq. (4.16) is the domain of interest for deriving a smooth transition functions. Thus, $t \notin [0, 1]$ are to be disregarded from this function. Scaling Eq. (4.16) by the inverse of t_d gives

$$\alpha_2(t) = \alpha_1\left(\frac{t}{t_d}\right) = 3\left(\frac{t}{t_d}\right)^2 - 2\left(\frac{t}{t_d}\right)^3. \tag{4.17}$$

Scaling Eq. (4.17) by amplitude, b, gives

$$\alpha_3(t) = b\,\alpha_2(t) = b\left(3\left(\frac{t}{t_d}\right)^2 - 2\left(\frac{t}{t_d}\right)^3\right). \tag{4.18}$$

Offsetting Eq. (4.18) with regards to the y-axis, using parameter c, gives the final function, $\alpha(t)$, as in Eq. (4.19) with velocity in Eq. (4.20).

$$\alpha(t) = \alpha_3(t) + c = b\left(3\left(\frac{t}{t_d}\right)^2 - 2\left(\frac{t}{t_d}\right)^3\right) + c \tag{4.19}$$

$$\frac{d\alpha}{dt}(t) = \frac{6b}{t_d^2}\,t\left(1 - \frac{t}{t_d}\right) \tag{4.20}$$

$$\frac{d^2\alpha}{dt^2}(t) = \frac{6b}{t_d^2}\left(1 - \frac{2\,t}{t_d}\right) \tag{4.21}$$

Utilizing Eq. (4.19), arbitrary smooth transition trajectories can be chosen by altering the following adjustable parameters in the GUI, along with t_d.

Adjustable parameters besides t_d:

$from$: Start position, with the option to use current position.
to : End position.

Using parameters $from$ and to, b and c can be computed.

Some examples are displayed in Figure 4.6 with parameter values as in Table 4.1.

Table 4.1: Parameter values for Figure 4.6

t_s	t_d	b	c
0	3	-2	3

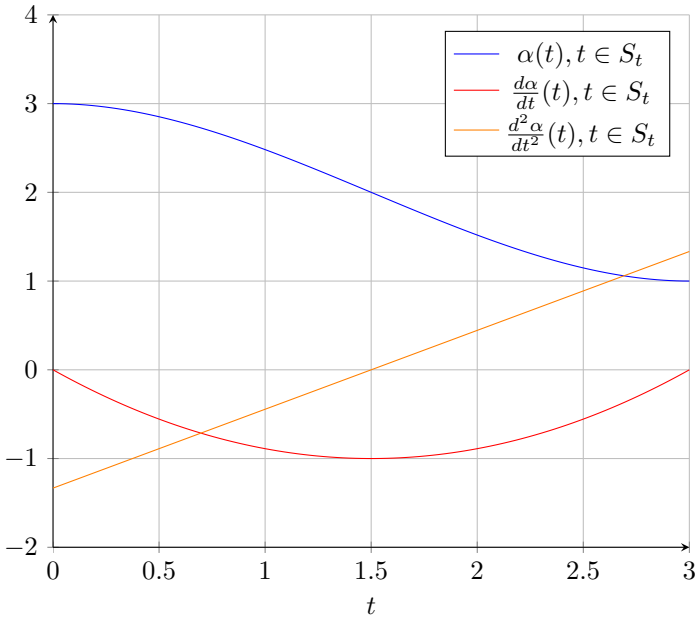


Figure 4.6: Example of smooth transition trajectory

4.3.2 Periodical steps

A step function alternating between two constant values can be defined with t_d along with the adjustable parameters:

l : Low position, value of position with the lesser value
h : High position, value of position with the greater value
t_l : Duration to have lesser position as reference
t_h : Duration to have greater position as reference

Periodic step trajectories initializing with lesser value, can be described as

$$\beta(t) = \begin{cases} l, & t \mod (t_l + t_h) < t_l \\ h, & t \mod (t_l + t_h) > t_l \end{cases}, \forall t \in S_t, \tag{4.22}$$

with velocity as

$$\frac{d\beta}{dt}(t) = \begin{cases} (h-l)\,\delta(t), & t \mod (t_l + t_h) = t_l \\ (l-h)\,\delta(t), & t \mod (t_l + t_h) = 0 \\ 0, & \text{otherwise} \end{cases}, \tag{4.23}$$

$\forall t \in S_t$, where $\delta(t)$ is the impulse distribution [9], making its derivative a task for further exploration. An example of this trajectory type, with parameter values as in Table 4.2, is shown in Figure 4.7.

Table 4.2: Parameter values for Figure 4.6

t_s	t_d	l	h	t_l	t_h
0	12	$-\frac{\pi}{4}$	$\frac{3\pi}{4}$	4	2

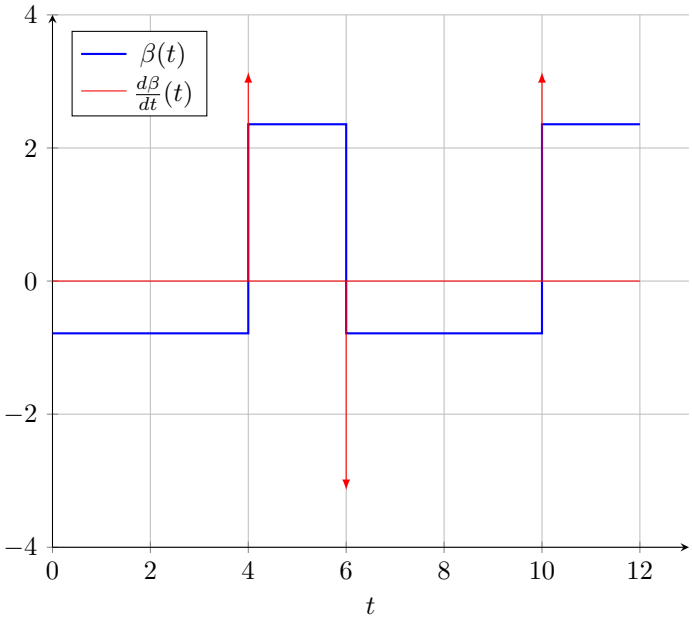


Figure 4.7: Example of periodic step trajectory

4.3.3 Sinusoidal

Similar to smooth step trajectories, sinusoidal trajectories uses adjustable parameters b and c, for scaling amplitude and vertically offsetting the trajectory. Additional parameters for sinusoidal trajectories are period time, T, and phase shift, ϕ. Using these, a sinusoidal trajectory, $\gamma(t)$, can be described as

$$\gamma(t) = b \sin\left(\frac{2\pi}{T}\, t + \phi\right) + c, \tag{4.24}$$

with corresponding velocity

$$\frac{d\gamma}{dt}(t) = b\,\frac{2\pi}{T}\,\cos\left(\frac{2\pi}{T}\, t + \phi\right), \tag{4.25}$$

and acceleration,

$$\frac{d^2\gamma}{dt^2}(t) = -b\left(\frac{2\pi}{T}\right)^2 \sin\left(\frac{2\pi}{T}\, t + \phi\right). \tag{4.26}$$

An example with parameter values as in Table 4.3 results in an oscillating trajectory within the interval $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$, shown in Figure 4.8.

Table 4.3: Parameter values for Figure 4.8

t_s	t_d	b	T	ϕ	c
0	8π	$\frac{\pi}{4}$	2π	0	$\frac{\pi}{2}$

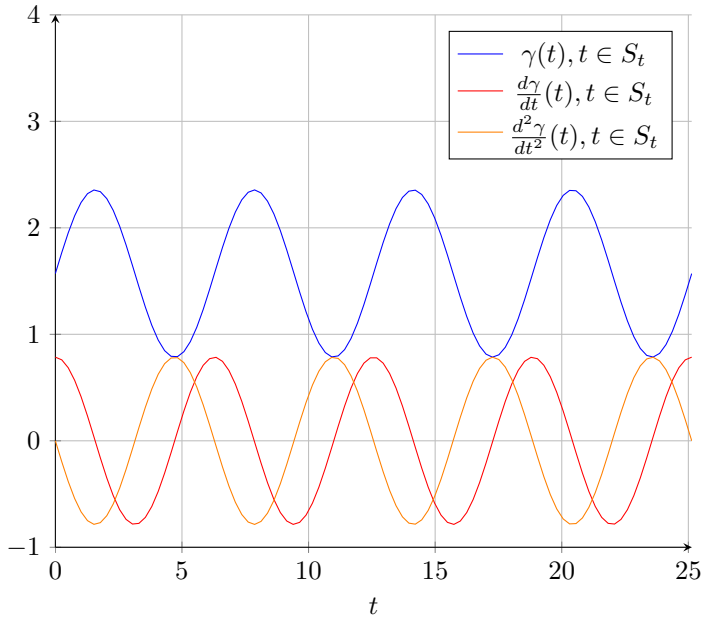


Figure 4.8: Example of sinusoidal trajectory

5

Methodology

5.1 Overview

The mechatronic system is constituted by several subsystems that will need to work together in order to construct a functional robot that can be used for rehabilitation. The project will follow a parallel workflow, where each subsystem is independently developed and tested. After proper validation, each subsystem will be integrated into the architecture and further development will be done to ensure that each component works as expected when combined with the remaining parts of the system. The development procedure will follow an iterative process, where recurrent validations may lead to reconsideration of previous processes to ensure they meet the additional criteria set by further development.

An overview of the workflow is presented in Figure 5.1. The following chapter will provide insight into each of these subsystems, the considerations made regarding its components, and the validation required before integrating them into the system.

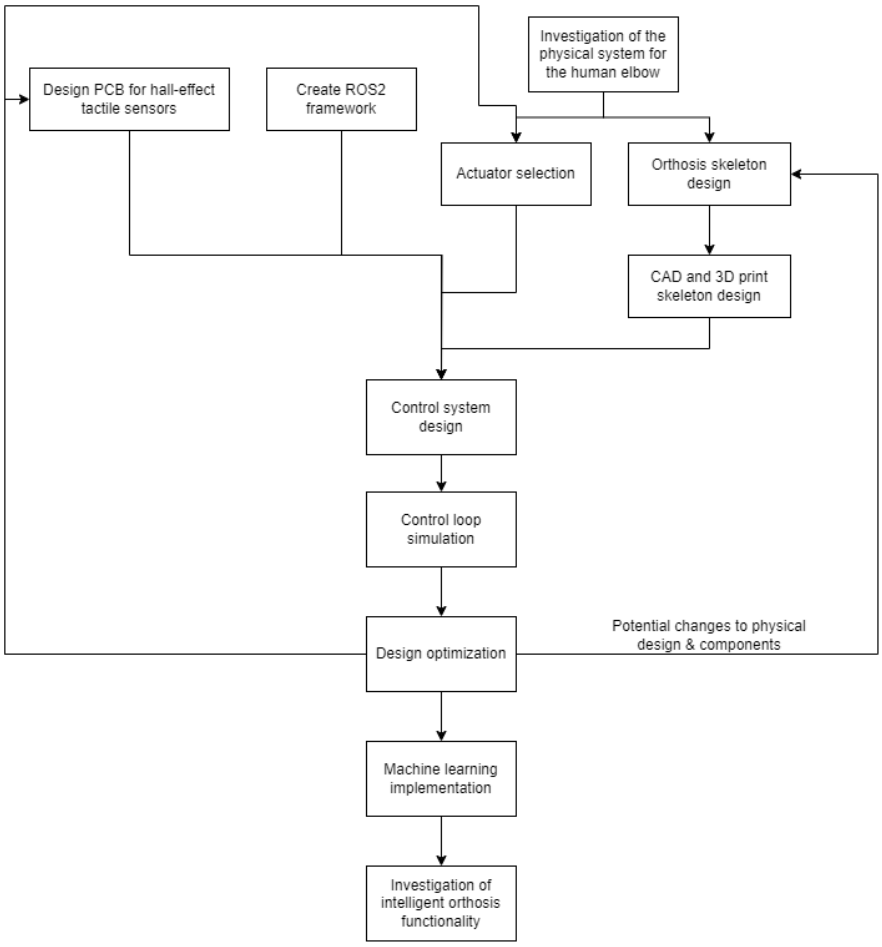


Figure 5.1: Block diagram of adopted workflow

5.2 Investigation of the physical system for the human elbow

A thorough investigation of the human arm will be made to identify important parameters that will dictate how the design and implementation should be adopted in order for the exoskeleton to be used in a rehabilitative environment. Many of the subsystems will be influenced by the limitations set by these parameters, including the design, control and electronics. It is therefore optimal for such factors to be realized early on when conceptualizing the project, in

order to build a foundational framework that will aid future decision-making and not hinder development in later phases.

5.3 Design of hall-effect tactile sensors

The tactile sensors designed during this project function based on a simple principle. Hall effect sensors return values based on the strength of magnetic fields around them. By embedding a magnet inside a malleable material above such a sensor we can create a basic force sensor. Applying pressure on the material deforms it, moving the magnet closer to the sensor, giving a higher reading. A simple illustration is provided in Figure 5.2. This reading can be converted to an approximate force through calibration and calculation.

This project utilizes off-the-shelf breakout boards for the hall-effect sensors, to make development easier and faster. Using breakout printed circuit boards (PCBs) does make the individual sensor components larger, but since no more than 2 sensors will be used, this is not an issue.

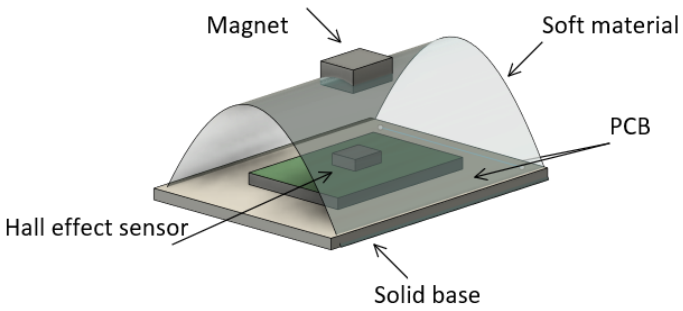


Figure 5.2: The central idea behind a hall-effect based tactile sensor

5.4 Create ROS2 framework

In order to make the system modular as well as extendable, ROS2 was chosen as middleware to handle data and creating a structure over the software components on the PC side. By having nodes interact with topics and services, different modules in the software architecture could access the appropriate data in a structured and modular way. This also allows for visualization of the available data using associated tools with ability to plot and simulate transmitted

data. However, a software called Foxglove with ability to interact with data within a ROS2 environment was discovered as a tool to monitor the system in a convenient manner. Thus, monitoring data in Foxglove served as the main way of validating the processes in the ROS2 framework. The choice of using ROS2 motivated the system to be built upon Ubuntu on the PC side.

5.5 Actuator selection

The selection of actucator depends on multiple factors, the most important, for this report being price. The actuator should be able to lift, or almost lift, a relaxed arm on its own. This may be important when using the "resistive" mode of operation. A micromotors E192 DC-motor with appropriate torque was chosen. Servo motors within the given price range lack sufficient power to lift an arm, or even resist its motion. Most adults, and likely many patients in physiotherapy, can exert force far greater than that of such a servo motor. This would impact the usability of the "resistive" mode of operation, since the resistance offered would be almost negligible.

Linear actuators, although powerful, are usually quite slow and would complicate the mechanical design. Questions regarding safety come up in regards to linear actuators, since many of them are powerful enough to cause significant injury.

5.6 Orthosis skeleton design

The orthosis consists of two distinct parts. The fixed system fastened with straps on the upper arm, and the moving forearm structure. The electric motor is fixed to the upper arm, and connected to the forearm piece, which contains the sensor arrays.

Safety is the most pressing issue in regards to the design. The orthosis has to be designed in a way that prevents injury. Developing gearboxes and similar systems with specific upper thresholds for delivered torque requires precision, expertise, time, and funding, resources currently lacking. Instead, a more fundamental approach is required. Some inherent safety is achieved by choosing materials and designing parts in a way that promotes structural failure when forces become too strong. This approach may not scale well since every patient's needs are different, so maximum torque applied to the arm will be different as well. Considering the scale of the project, however, such an approach may be necessary due to the constraints mentioned.

5.7 CAD and 3D print skeleton design

The elbow will be manufactured using 3D-printed plastic components and laser cut acrylic, since these are the tools available for the project. These tools have great versatility and are not predicted to pose any major hindrance during development.

Certain 3D printed parts may be under stress during operation. The nature of 3D printed components allow for breakage along layer lines, and designs may have to be compromised in order to avoid such failures. This inherent fragility may also be utilized as a safety feature. Parts can be designed to fail under any sizable load, and therefore protect the user.

The "axles" of the elbow, that is, the elongated singular pieces keeping other parts connected, will be made out of acrylic. Acrylic, depending on thickness, is very fragile. It will bend to a point where it snaps immediately. The skeleton will be designed to break if forces become to great, which is an invaluable safeguard.

5.8 Control system design

The control system will be built on underlying theory presented in Chapter 4.1. The middle level controller will dictate how the robotic arm moves given both a trajectory input set through the GUI, as well as the sensor input given by the tactile sensors. Extensive work will be put into the modelling; ensuring that the kinematic and dynamic models are done in compliance with well-established conventions and integrated into the control architecture with a high degree of modularity to aid further development.

Inspired by control method iterations in previous projects [2], some simplifications will be made given the limited knowledge in dynamic robotics and model-based control theory. Our hope is that said simplifications will not severely hinder performance of the exoskeleton and will still provide a functional proof of concept that behaves as expected. Given the rigorous amount testing required for human trials, testing of the control system and the exoskeleton as a whole will most likely be done with a prop that can replicate the dynamics of the human arm.

5.9 Control loop simulation

To perceive the state of the system, at any given point in time during usage, a simulation environment is to be implemented. This could could allow for further development, without the need of hardware, and will likely ease troubleshooting if the system is not behaving in an expected manner. One scenario where this may be useful is when the orientation of the shoulder joint may influence the control of the system. This since the controller only recognizes one degree of freedom, around the elbow joint, the gravitational acceleration may be interpreted to be affecting the system in the wrong direction. That is because the simulation would show this unexpected state of the arm, via the intertial measurment unit (IMU) placed on the upper arm, while the middle level controller considers the shoulder joint to be in a fixed orientation with the upper arm in a vertical position. As stated, the simulation would also monitor the calculated signals used to control the system to allow for further monitoring and understanding of the system's behaviour.

5.10 Design optimization

As further development progresses and all subsystems are integrated, the initial design may require optimization in its mechanical structure, choice of components, or their implementation. This stage serves as a point of reflection to ensure that the design can meet the demands set by the control system and that the robotic arm is functional for the required tasks. Following a modular design philosophy, we hope that any optimizations made to the design will not disrupt the overall workflow or render elements of previous subsystems obsolete. It is therefore crucial that all development is undertaken with a unified goal among group members to avoid bottlenecks in the process.

5.11 Machine learning implementation

Machine learning could be implemented to personalize the use of the exoskeleton, enabling the robot to detect improvements in arm movement during use and to adapt its support accordingly. However, this area verges on physiological claims that are beyond the scope of this project, and may therefore be excluded. While other applications of machine learning to enhance robotic arm functionality could be explored, other areas of development are likely to benefit more from reevaluation and fine-tuning. Given the limited time frame, machine learning will not be prioritized.

5.12 Investigation of intelligent orthosis functionality

The final phase of development will involve extensive data collection and analysis. The system will be tested under conditions that simulate human arm motion to evaluate the controller's performance, force-data resolution, and hardware communication. Despite limited knowledge of the rehabilitative process, the functionality of the robotic arm can be assessed under various conditions to evaluate its ability to support a human arm. These conditions may include different orientations of the arm and varied motion trajectories.

6

Implementation

This chapter presents how the theory in chapter 4 was implemented, the simplifications made, and the areas that suffered as a result. Initially, an overview of the system is presented in Figures 6.1, 6.2, which outlines the entire system architecture, divided into ROS2-USB, and USB-micro controller communication. What follows is a deep-dive into each subsystem presented in the figures, including the choice of electrocics, design considerations, kinematic and dynamic models used, as well as control methodology implemented.

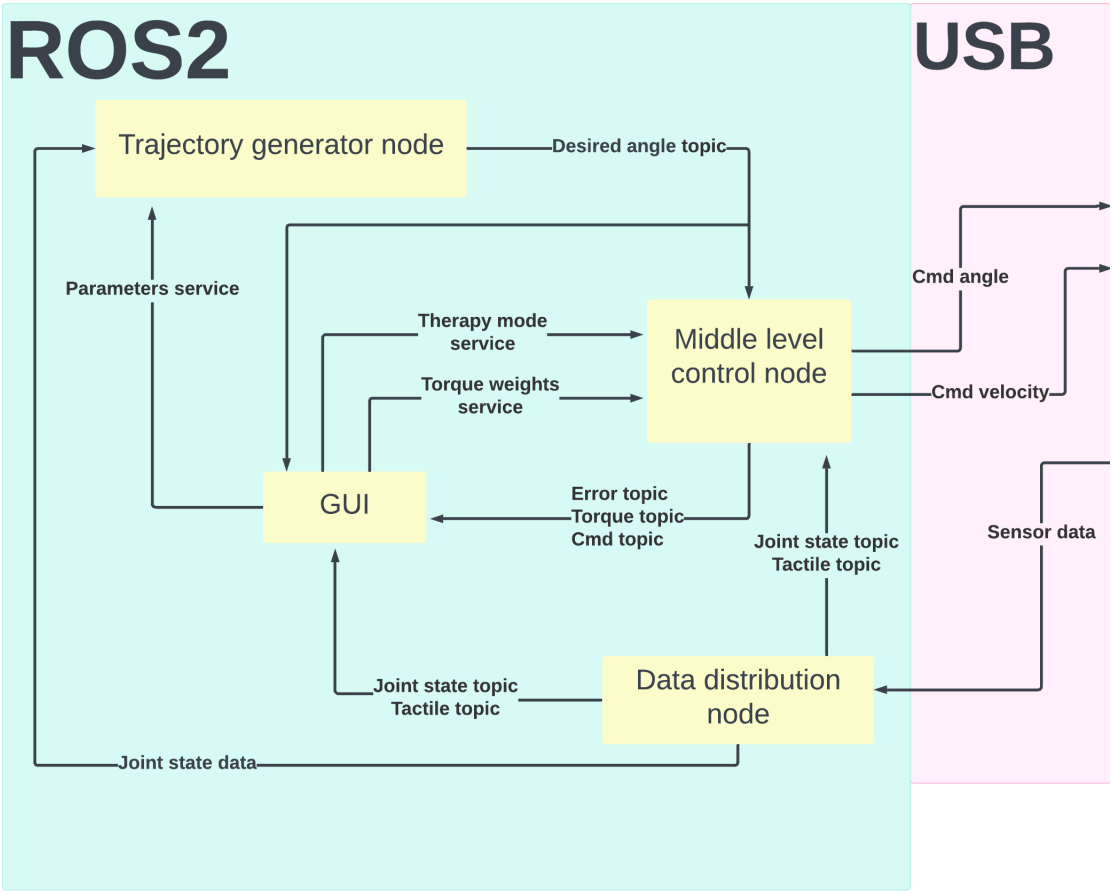


Figure 6.1: The system architecture. Overview over ROS2 to USB communication.

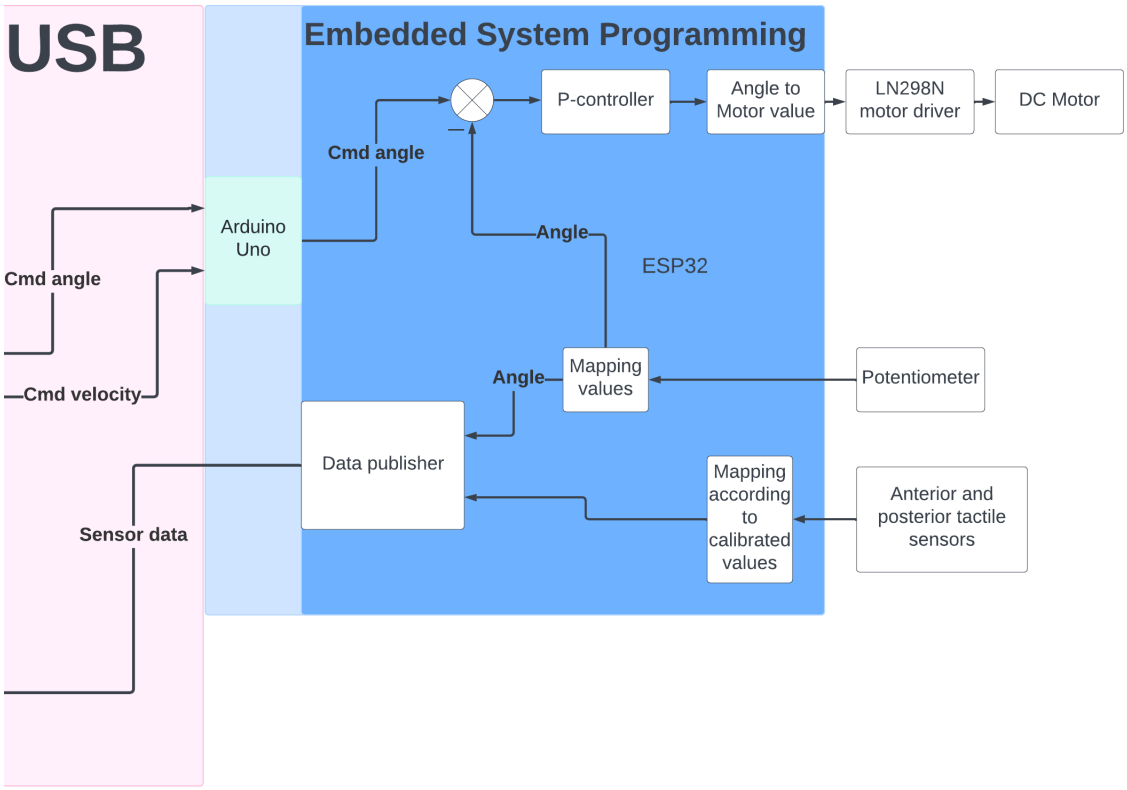


Figure 6.2: The system architecture. Overview over USB to micro controller communication.

6.1 Mechanical design

The mechanical design had to be adapted in ways hampering usability due to the actuator chosen. A noteworthy compromise is that the axis of rotation is offset from the natural center of rotation for the elbow. This affects how comfortably the full range of motion can be utilized. As can be seen in Figure 6.3, the motor mount will end up below the elbow when the upper arm is strapped in, with the white supports cupping the triceps. The full orthosis with sensors mounted is seen in Figure 6.4.

Figure 6.3: The elbow orthosis without sensors mounted. Note how the motor ends up below the elbow.

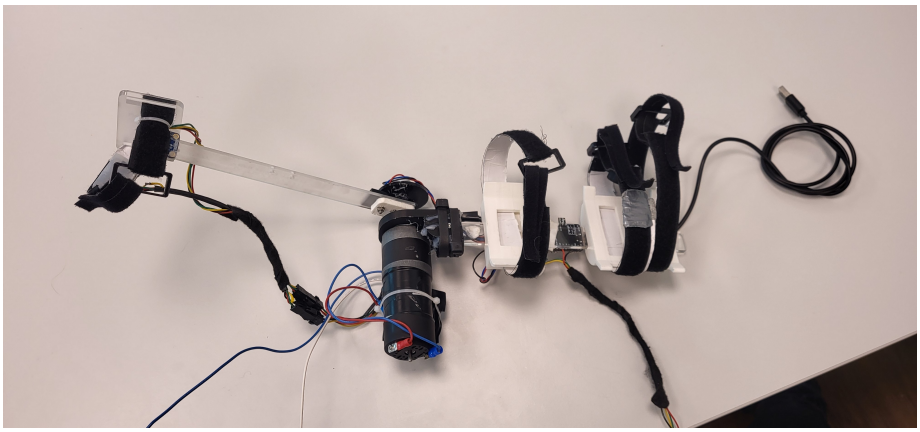


Figure 6.4: The elbow orthosis

The orthosis needed a method of knowing the angle of the elbow joint. Since the motor does not have an encoder, and IMUs were too noisy, a potentiometer was mounted as can be seen in Figure 6.5. To keep the potentiometer steady, a screw is attached to the motor. The screw, which can be seen in Figure 6.6, also doubles as a safeguard against hyper extension of the elbow.

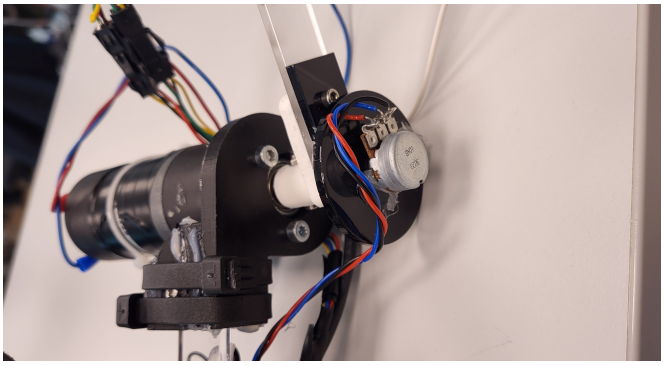


Figure 6.5: Potentiometer used to calculate the angle of the elbow

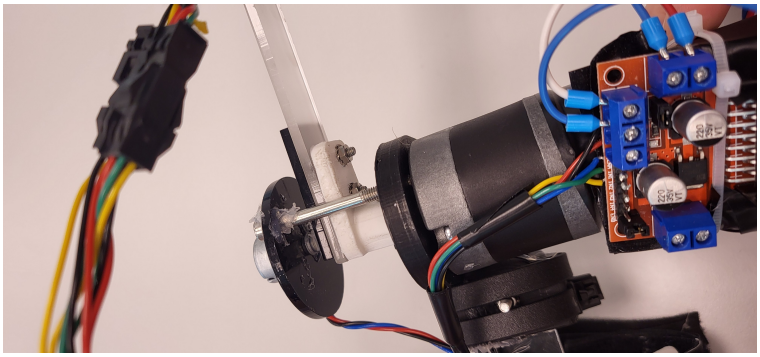


Figure 6.6: A screw functioning both as a fastener and as a way to prevent over extension

6.2 Electronics

The general layout of the electronics can be seen in Figure 6.7.

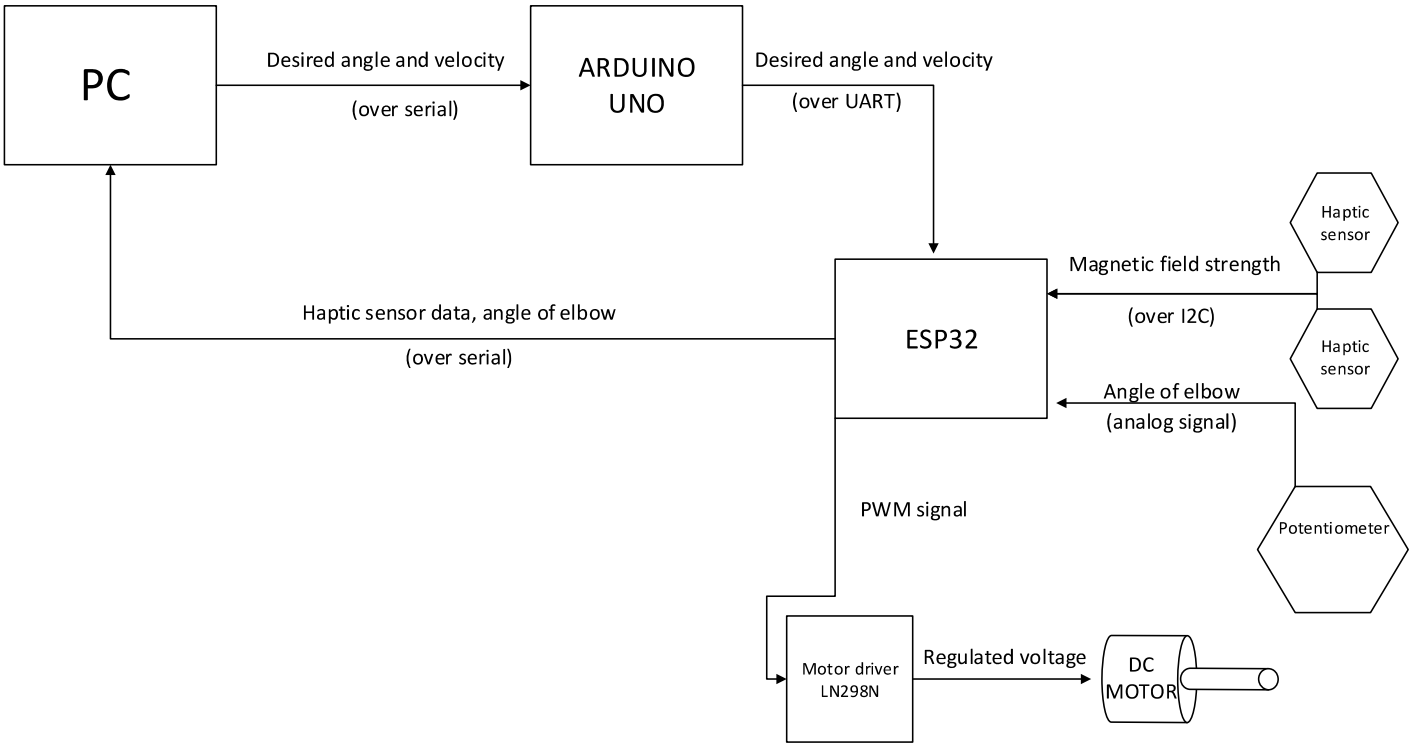


Figure 6.7: Schematic over the implemented on-device electronics

Micro controllers

The micro controller chosen for gathering sensor data, sending data, and controlling the motor is an ESP32. The ESP32 was chosen thanks to its speed, price, and most importantly its dual-core functionality. The ESP32 allows each processor core to be addressed separately, which means that two completely

separate code loops can run simultaneously on one device. This allows for the same functionality as having a separate micro controller purely dedicated to collecting sensor data, but all on a single device.

There are some changes to the system implementation caused by issues in PC - ESP32 communication. Data could be sent from, but not received by the ESP32. A restrictive time frame limited efforts to reconcile this issue. Data could, however, be sent reliably using the same code to an Arduino Uno. Data is therefore sent from a PC to an Arduino Uno, which in turn sends that data over the UART protocol to the ESP32.

Sensors

The tactile sensors are constructed using off-the-shelf breakout boards for the MLX90393 hall-effect sensor, made by Adafruit. The PCBs were placed in 3D printed plastic mold, and skin-safe silicone was poured on top. After the silicone had stiffened somewhat, a magnet was placed in the silicone above the hall-effect sensor. IMUs were tested, but not used in the final product.

6.3 Exoskeleton modelling

6.3.1 Kinematics

Coordinate frames for the robotic manipulator were set up to allow for positional- and orientational description of all its mechanical components. The base frame is located at the shoulder point, frame {0} at the elbow joint and the tactile sensor frames were placed at the posterior- and anterior forearm. Figure 6.8 shows the implemented model, along with corresponding kinematic parameters.

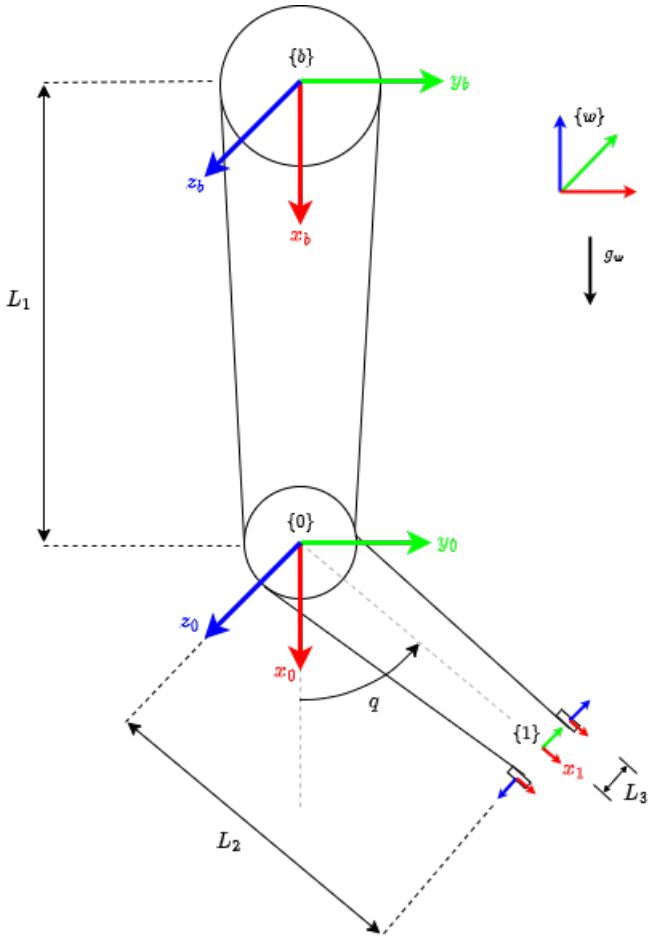


Figure 6.8: Modelling diagram

Given the above configuration, DH parameters were set up and used with Eq. (4.5) to compute the transformation matrix of frame {1} with respect to frame {0}. Table 6.1 shows the DH parameters, and Eq. (6.1) shows the corresponding linear transformation matrix.

Table 6.1: DH parameters

a_1	d_1	θ_1	φ_1
L_2	0	q	0

These DH parameter values give the pose of frame {1} as

$$\mathbf{T}_1^0(q) = \begin{bmatrix} \cos(q) & -\sin(q) & 0 & L_2\cos(q) \\ \sin(q) & \cos(q) & 0 & L_2\sin(q) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.1}$$

The transformation matrices for the sensor frames were obtained by rotation around x_1, translation along y_1 and referenced to the joint frame {0} using Eq. (4.4). The position vector P_1 is extracted from this matrix and used in combination with $z_0 = [0,0,1]^T$ and $P_0 = [0,0,0]^T$ to compute the linear component of the geometric jacobian, as per Eq. (4.7).

When a force is applied by the arm on either sensor, the torque exerted on the elbow joint can be calculated using Eq. (4.8) and Eq. (4.9). This parameter is pivotal as it allows for the exoskeleton to follow the movements of the user, further explained in Section 6.4.

6.3.2 Dynamics

The dynamic model in Eq. (4.12) was simplified for a system with 1 degree of freedom. The dimensional reduction of the model results in the contributions from each dynamic torque components shown in Eq. (6.2).

$$\begin{aligned} \boldsymbol{M} &= ml^2 \\ \boldsymbol{B} &= B \\ \boldsymbol{G}(q) &= mgl sin(q) \end{aligned} \tag{6.2}$$

Where m is the mass of the forearm, l is the distance from the elbow joint to the center of mass, and g is the gravitational constant. This gives the final dynamic model as

$$ml^2\ddot{q} + B\dot{q} + mgl\sin(q) = \tau. \tag{6.3}$$

The initial values of the dynamic parameters were based on the average human elbow, taken from [2]. These values were meant to serve as a starting point for implementing the dynamics of the arm in the control architecture; where further adjustments can be made depending on it's performance. Table 6.2 shows the kinematic and dynamic parameters used for integrating the exoskeleton model into the system.

Table 6.2: Kinematic and dynamic parameters

Description	Name	Value	Unit
Upper arm length	L_1	0.50	m
Forearm length	L_2	0.31	m
Distance to center of mass	l	0.25	m
Mass of forearm	m	1.0	kg
Elbow damping	B	0.1	Nms/rad
Distance to sensors	L_3	0.04	m

6.4 System control

The middle-level controller uses a weighted sum of the two control parameters τ_k and τ_f to allow the robotic manipulator to follow any desired movement. Adjusting the weight values allows the user to switch between a trajectory-dictated path and a sensor-dictated path, where the robot will follow the movement of the human arm through the tactile sensor feedback. The torque is used as an input in the dynamic model, Eq. (6.3), to determine the resulting angular acceleration $\ddot{q}$ of the arm. Upon integration, an angular command q can be sent to the servo motor. This enables the control system to dictate precise adjustments, ensuring that the arm follows the desired trajectory, while taking into account the physical dynamics of the structure. An overview of the middle-level control architecture is shown in Figure 6.9.

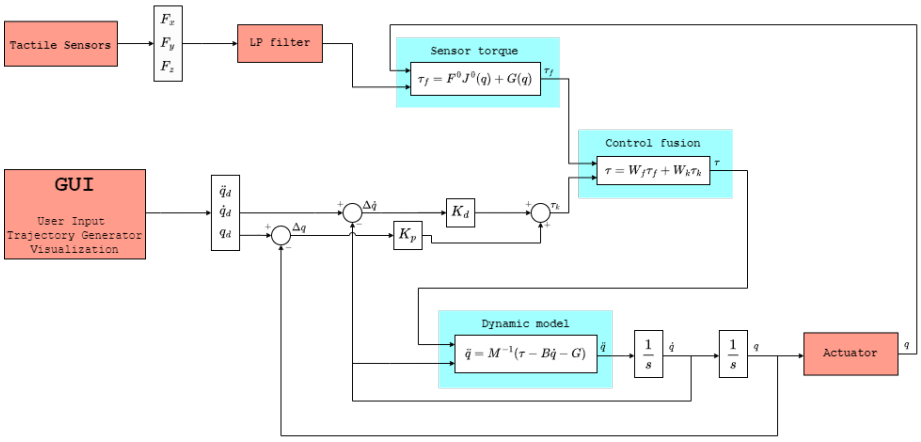


Figure 6.9: Middle level control architecture

6.5 ROS2

The ROS2 environment is of the distribution Iron Irwini and developed in Python on Ubuntu 22.04. It receives sensory data from the hardware and sends back commands for movement. Data are transmitted within this environment as shown in the Figure A.1, in the appendix, where the data distribution node receives data from a micro controller over serial communication. Outgoing data from the ROS2 environment is transmitted from the middle level control node over serial communication back to the micro controller.

6.5.1 Nodes

Data distribution

This node receives sensory data from a micro controller using serial communication. The data is processed, distributed and published into joint state topic and tactile topic. This node also sets the update time interval for all ROS2 topics. For verification of the ROS2 environment this node is also able to produce arbitrary sensor values in order to verify the system behaviour in the GUI.

Trajectory generator

This node generates trajectories, serving as reference positions of desired exercises, including velocities and acceleration, specified in Section 4.3. These are published to the desired angle topic for the high level control node to calculate more sophisticated reference signals.

A flaw of the system is that the first and second order derivatives of the periodical steps trajectory, described in Section 4.3.2, were naively put to zero over the entire domain before realizing the property of the Unit step having a Dirac delta function as derivative at the actual step. By subscribing to the joint state topic, trajectories can be generated based on the systems current state, i.e., a smooth transition from current position to an arbitrary position.

Due to a limited timeframe, the controller could not be properly integrated into the system why the architecture was altered in order to test the controller with the hardware, as seen in Figure A.2 in the appendix. That lead to functionality which were supposed to be in the control node was moved into this node in order to reduce sources of errors. In these test cases this node was also responsible for calculating commanded angle and velocity. Additionally, the error and torque was also calculated here before sending the commanded angle and velocity to the micro controller.

Simulation

Using data from its subscription to the joint state topic, this collection of nodes creates a hierarchy of transformed coordinate frames. This allows for having a visual representation over the state of the arm to monitor in the GUI. Inability to pass absolute position of the orientation of the IMU placed on the lower arm caused this to visualize its orientation falsely as a transformation of its parent frame. Which, however, is of minor importance. This was noticeable when verifying the functionality using arbitrary sensor values. Even though usage of IMUs was eventually abandoned, the simulation allows for a complete visualisation of the positioning of the arm.

Middle level control

This node is supposed to calculate commanded position, velocity and acceleration based on trajectories from the trajectory node and the state of the arm with functionally explained in Section 6.4. Additionally, the error between the commanded position and velocity compared to the current values as well as torques are to be calculated in this node. For cases when the intended ROS2 architecture was used along with the middle level controller's output as reference signals, as seen in Figure A.1 in the appendix, verification of its functionality using the GUI proved unsuccessful why the tasks of this node was moved into the trajectory node. This to simplify the system to the cost of sacrificing some modularity.

Foxglove bridge and Foxglove bridge component manager

The Foxglove bridge node allows ROS2 data to be visualized in Foxglove Studio, an interactive GUI [10]. However, the specific functionalities of the Foxglove bridge component manager node remains unsure but it is likely serving a supplementary purpose. It launches along with the foxglove bridge node when the task of the launching script is to open the Foxglove bridge in order to connect ROS2 data with the GUI. Foxglove also allows for simple storing of data which is used to store test results for the system.

6.5.2 Topics

Joint state

Over this topic sensory data about the positioning of the arm is transmitted. Due to data transmission issues with IMUs and limited timeframe, usage of

IMUs was overlooked in favour of a potentiometer measuring the elbow angle. Consequently, this custom message involves some redundancy which allows for transmission of IMU data and angular velocity around the elbow joint since the only useful information from this topic eventually was the elbow position. However, leaving this possibility to transmit IMU data facilitates further development. IMU messages include additional redundancy, stretching beyond the positioning of the arm since these also include angular velocity and acceleration. Reasons being expandability, convenience and stability. Since this uses an IMU message provided by ROS2 [11], it is convenient to use and also less prone to errors than if a custom message was to be implemented.

Tactile

This topic transmits sensory data from tactile sensors localized on the anterior and posterior parts of the arm respectively. This topic carries force data for both of them in three dimensions each.

Desired angle

This topic is uses the same message as for the commanded position and velocity described below. This because specifics of the commanded values was detailed prior to the ones regarding desired values and they send the same type of data. The desired angle is specified by the user in the GUI and calculated in the trajectory generator node, see Section 6.5.1, where also desired angular velocity and acceleration are calculated and transmitted over this topic.

Commanded angle

This topic transmits commanded values from the middle level controller regarding position, velocity and acceleration of the elbow angle. These values are published for monitoring and verification in the GUI. Thus, this topic has no subscribers apart from Foxglove, the GUI.

Error

Error is in this context referring to the error between the current state and the reference signal. The reference signal was supposed to be the commanded angle and the commanded angular velocity, thus the corresponding errors is the difference between these and the current position and angular velocity respectively. However, suffering from lack of time, the controller was not integrated

properly why the reference angle and velocity was, for most test cases, not the ones calculated by the controller but the raw trajectories from the trajectory node. However, when testing the functionality of the control integration the commanded position and velocity was used to calculate the error.

6.5.3 Services

The following services are called by the user in the GUI as specified in Section 6.6.1.

Parameters

This service specifies the trajectory intended for the system. The available ones are specified in Section 4.3 and their parameters are set in the GUI where the service is also called, which starts the trajectory.

Therapy mode

This service allow for different controllers to be used in the high level control node. The purpose is to allow for the user to choose between different types of support or resistance applied by the actuator elaborated in [2] by inputting "passive", "assistive" or "resistive". However, different controllers were not implemented leading to the default one being the only one used.

Torque weights

The controller includes parameters which, if altered, will vary the system behaviour. This service allows the user to set some of its values, namely τ_k, τ_f, K_p, K_d, displayed in the middle level control architecture in Figure 6.9.

6.5.4 Verification

Verification of whether transmitted data within the ROS2 environment looks reasonable was done using the GUI called Foxglove which can display ROS2 data in a condensed yet informative manner, see Figure 6.10. Additional images of the individual plots can be found in the appendix, Section B.

Plots and the 3D view of a visualized arm facilitate monitoring and analysis of whether the system is behaving as expected. The plots displays data passed

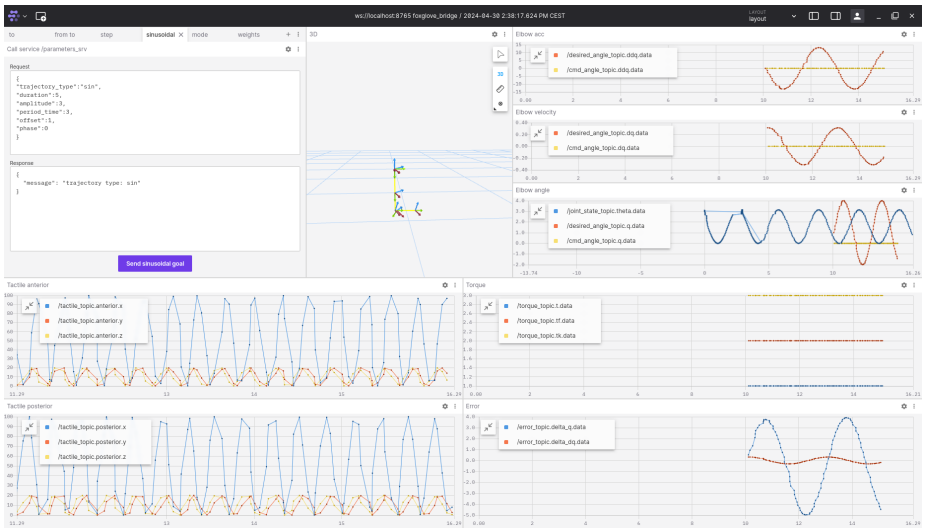


Figure 6.10: GUI overview. Image generated using Foxglove Studio [12]. Reproduced with permission.

through the topics explained in Section 6.5.2. The 3D view of the arm presents a visual representation of the current position of the arm.

6.5.5 Control integration

Due to a limited timeframe the middle level controller was not integrated properly into the system, causing disturbances in the results presented later in Section 7.5. To ease the integration such that these results could be collected, the sources of error in the ROS2 environment was reduced to what is shown in Figure A.2 where the data processing previously handled in the control node was moved to the trajectory generator node. Here, the middle level control node only serves the purpose of calculating the error between the current state and reference signal and publishing it to the error topic. Hence, this node is misnamed.

6.6 Interfaces

6.6.1 User interaction with PC

To create exercises as trajectories explained in Section 4.3, the user can input their respective parameters in the GUI as shown in Figures 6.11, 6.12, 6.13, 6.14 respectively, followed by an execution by pressing the blue button at the bottom. Further, the torque weights service and therapy mode service can also be called using the GUI as shown in Figures B.5, B.6 in the appendix.

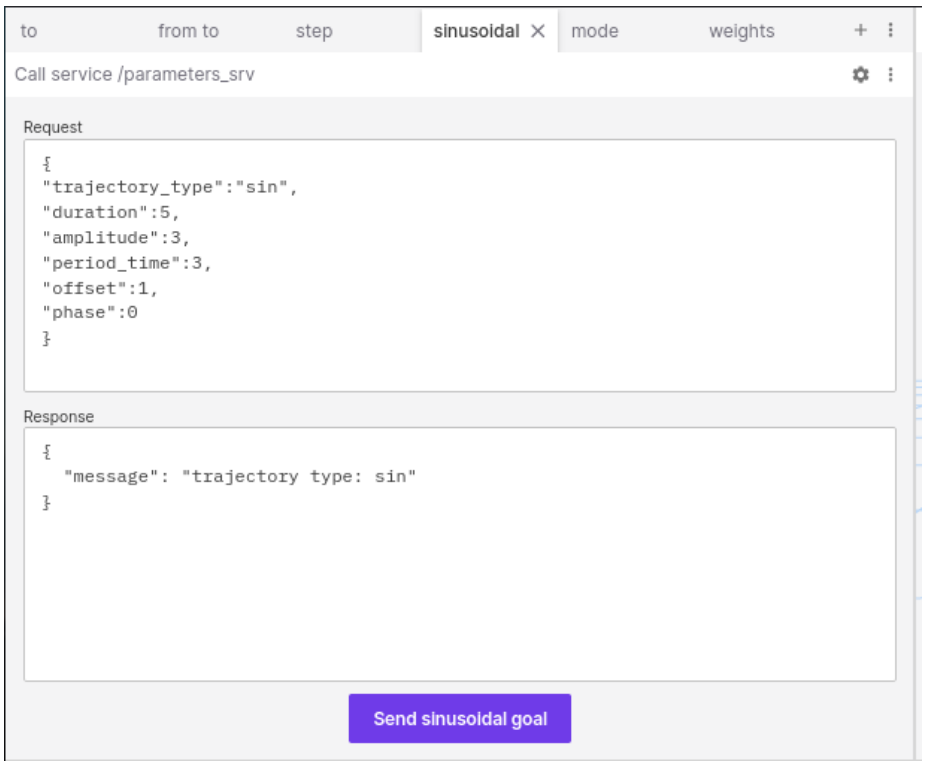


Figure 6.11: Setting sinusoidal trajectory. Image generated using Foxglove Studio [12]. Reproduced with permission.

6.6.2 Serial communication between PC and micro controller

Serial communication was used to transmit data between micro controller and the PC running ROS2. To allow for sufficient data transmission speed, a baud

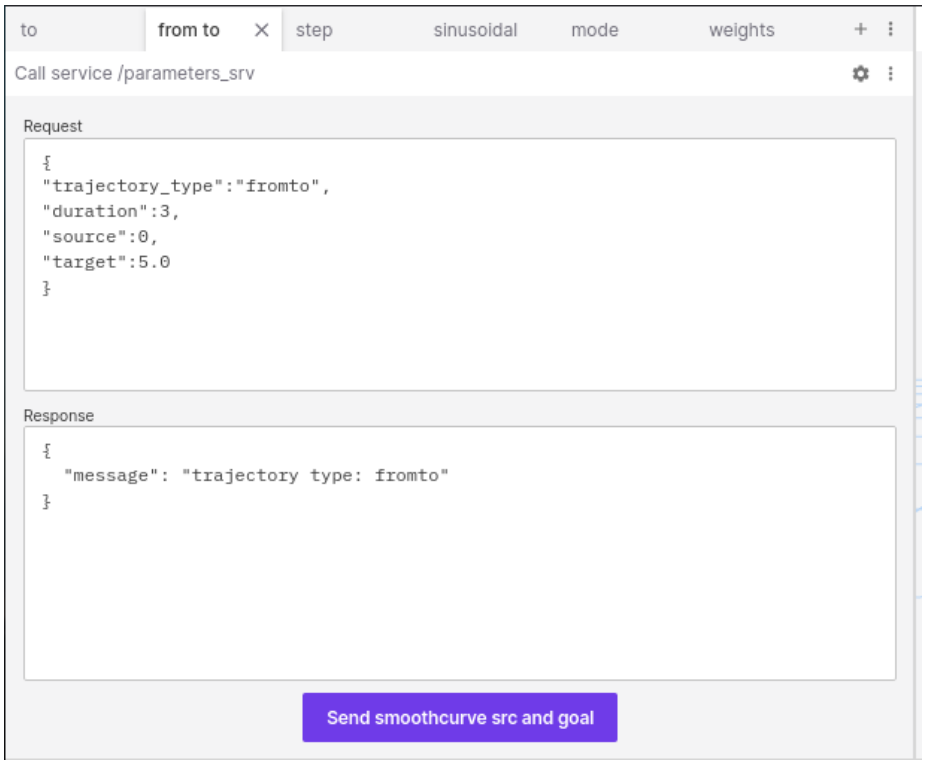


Figure 6.12: Set smooth transition trajectory from any position. Image generated using Foxglove Studio [12]. Reproduced with permission.

rate of 115 200 bps was used in both directions. The micro controller sends data from sensors to the PC and receives an angle and angular velocity as reference signals. The sensory data transferred from the micro controller consists of information from the developed tactile sensors along with the current position which is the angle of the elbow joint.

6.6.3 Verification and monitoring

See Section 6.5.4 where verification of the transferred data is elaborated which, to a major extent, include data transferred over serial communication.

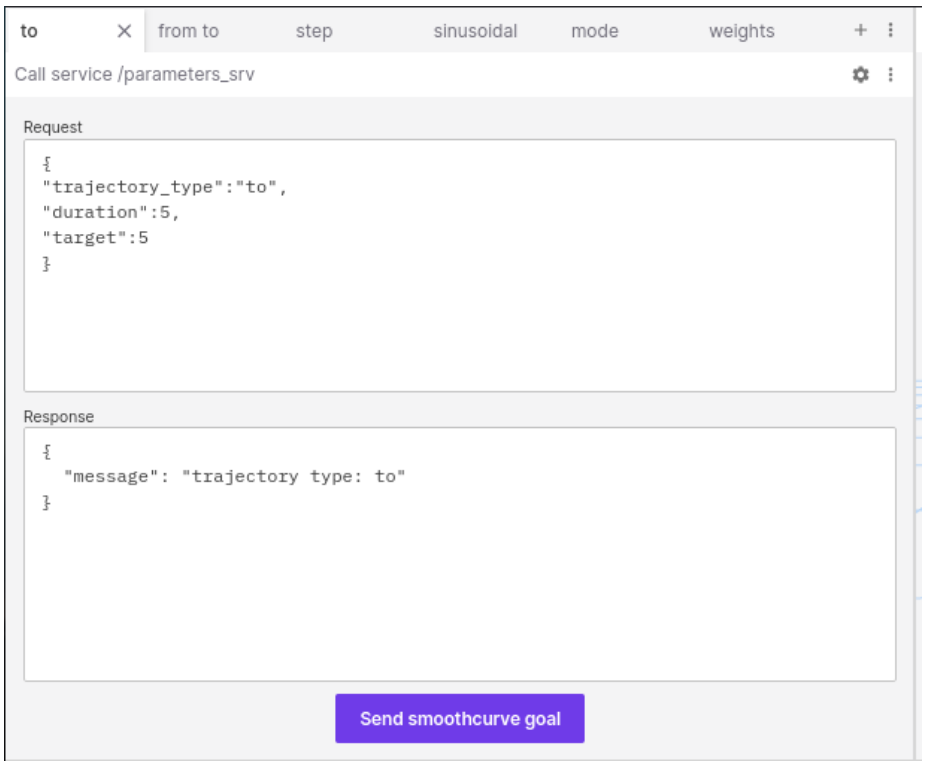


Figure 6.13: Set smooth transition trajectory from current position. Image generated using Foxglove Studio [12]. Reproduced with permission.

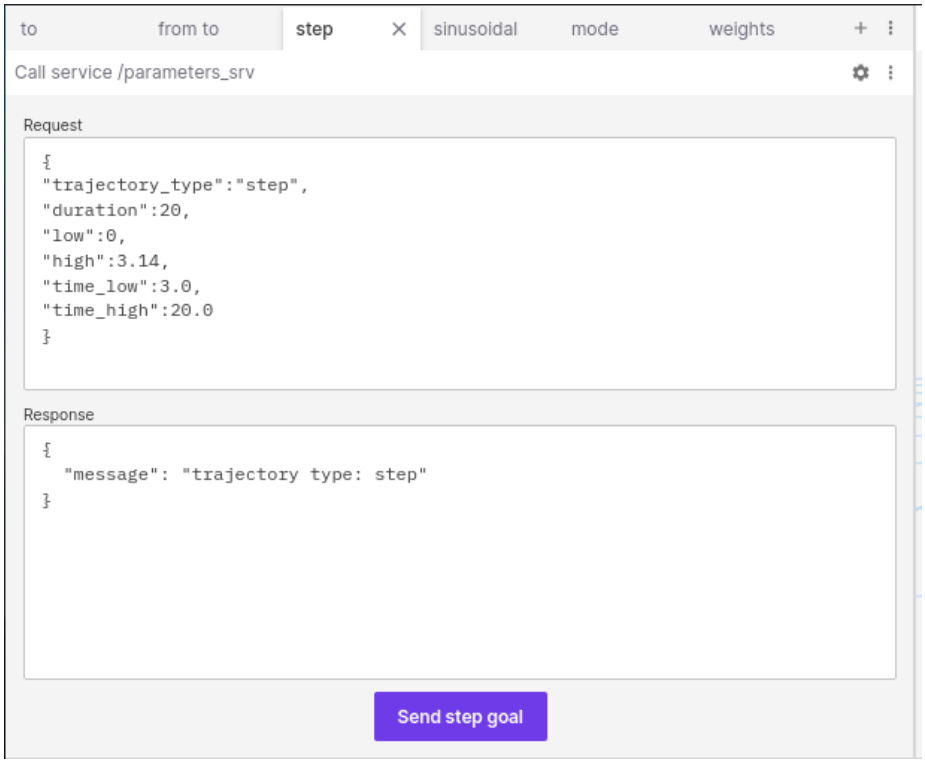


Figure 6.14: Set periodic step trajectory. Image generated using Foxglove Studio [12]. Reproduced with permission.

7

Results

Data in Sections 7.1, 7.2, 7.3 was collected using a sock filled with gravel to simulate a human arm, constructed to mimic the dimensions and mass used in the controller dynamics. See Figures C.1, C.2 in the appendix for photos. The trajectory types employed were sinusoidal, step, and smooth transition as reference signals. These trajectories were executed across various upper arm orientations: neutral, horizontal, and upside down as specified below. With a forearm length of 0.31 m as in Table 6.2, the required sampling rate of the elbow position, defined by the Nyquist criterion from Eq. (4.14) becomes at least 1.8 Hz. The average sampling frequency whilst following these trajectories varied between 23.7 Hz and 26.7 Hz, all fulfilling the Nyquist criterion. The position of the elbow is defined by θ as in Figure 6.8. Since a sock was used instead of an actual arm, the calibration cannot be completed in the intended manner. Values returned by the tactile sensors may therefore differ between tests.

Due to the aforementioned limitations, these trajectories and the data collection were conducted without the middle level controller. Instead, controller results will be presented in a separate simulated environment, demonstrating the expected behavior of the actuator angle commands. Subsequently follows also physical test results where the controlled signal serves as the reference signal for the orthosis, without a sock attached.

Neutral position means the orthosis is hanging freely, as if a human were standing with their arms down.

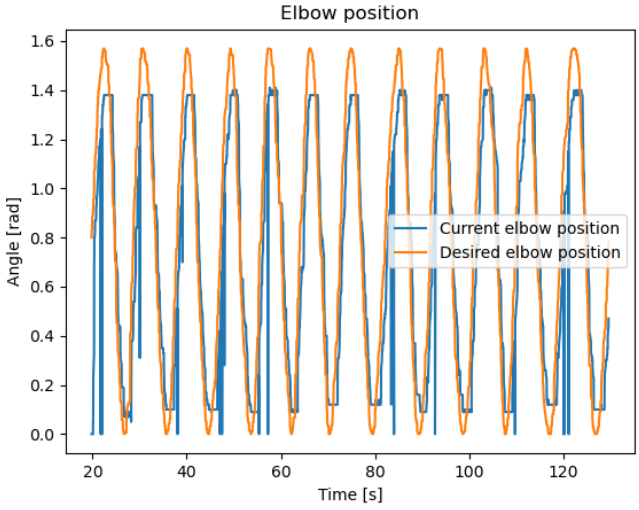
Horizontal position means the orthosis is laying flat on a table or is pointed in the air, as if a person extended their arms straight out.

Upside down position means the orthosis is held upside down, in a similar position as if a person were to curl their arm and move their elbow as high as possible, as if they were trying to scratch their back.

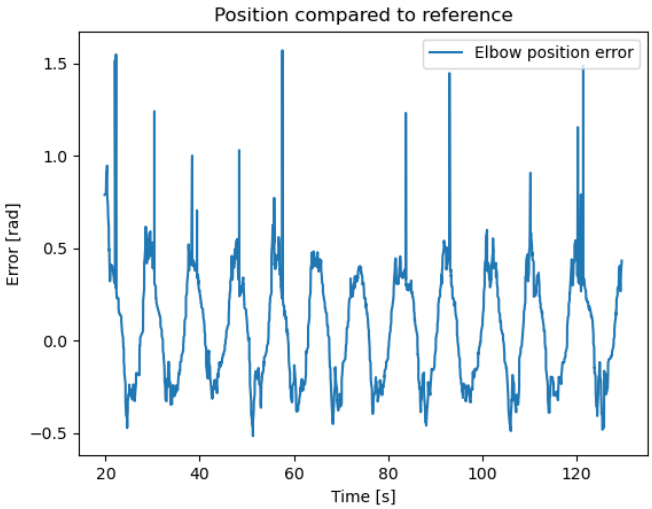
7.1 Sinusoidal trajectory

7.1.1 Neutral orientation

Below, in Figures 7.1, we see the data collected when moving the orthosis in a sinusoidal trajectory between $0\,^\circ$ and $90\,^\circ$. It can be seen in Figure 7.1a that the orthosis falls behind the commanded angle, and never quite reaches the peak values.



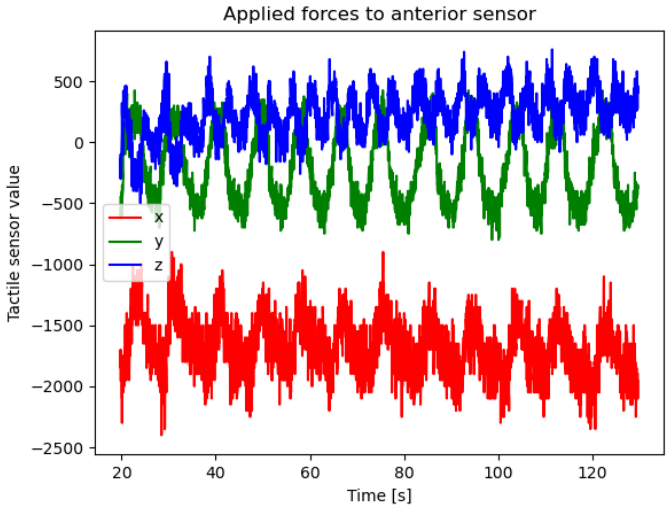
(a) Angle position compared to reference



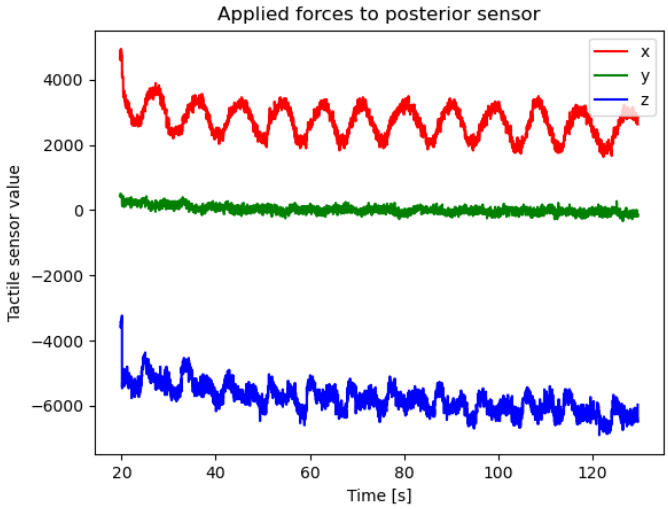
(b) Angle position error

Figure 7.1: Angular position and error

The data gathered from tactile sensors, visible in Fig.7.2 also show a clear pattern, indicating predictable behaviour.



(a) Anterior forces

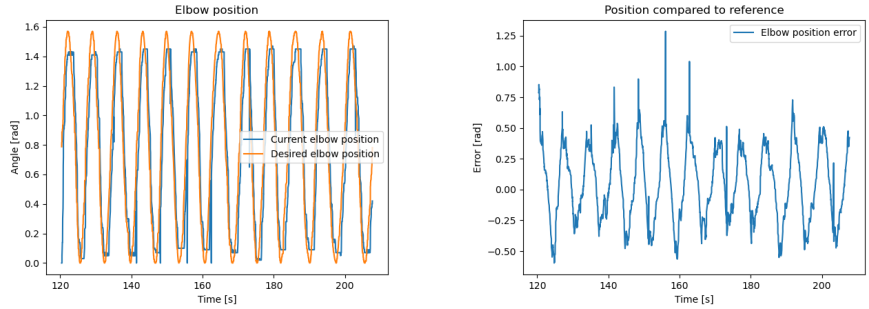


(b) Posterior forces

Figure 7.2: Tactile data

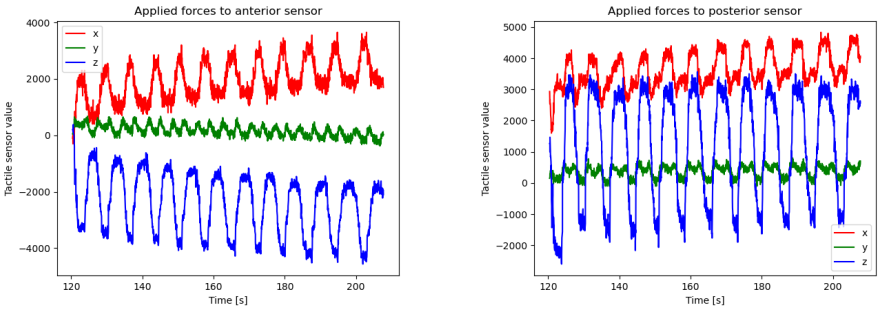
7.1.2 Horizontal orientation

The results from the test in the horizontal position differ in a predictable way from the neutral-position. The main thing of note is how the data from the tactile sensors is less noisy, seen in Figure 7.4. In Figure 7.3, We can observe the same pattern of the orthosis lagging behind the commanded angle and never quite reaching the peak value.



(a) Angle position compared to reference (b) Angle position error

Figure 7.3: Angular position and error

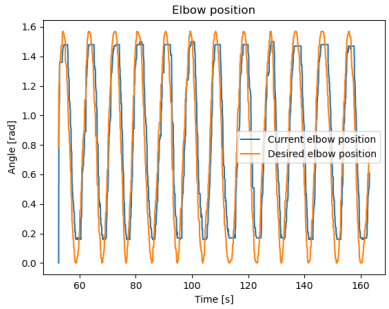


(a) Anterior forces (b) Posterior forces

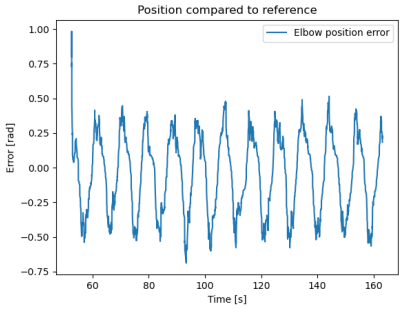
Figure 7.4: Tactile data

7.1.3 Upside down position

In the upside down position, we would expect the feedback from the tactile sensors to be reversed in terms of anterior and posterior tactile sensors. In figure 7.6, we see that this holds true. The output from the posterior sensor is less noisy and has a clearer sinusoidal pattern.

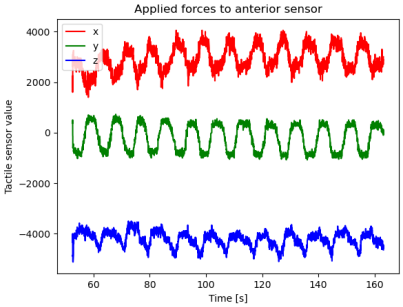


(a) Angle position compared to reference

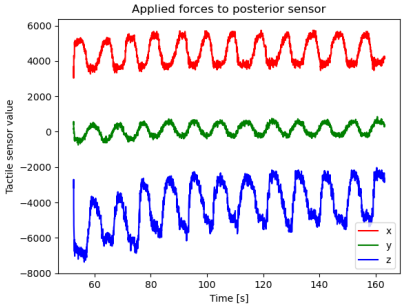


(b) Angle position error

Figure 7.5: Angular position and error



(a) Anterior forces

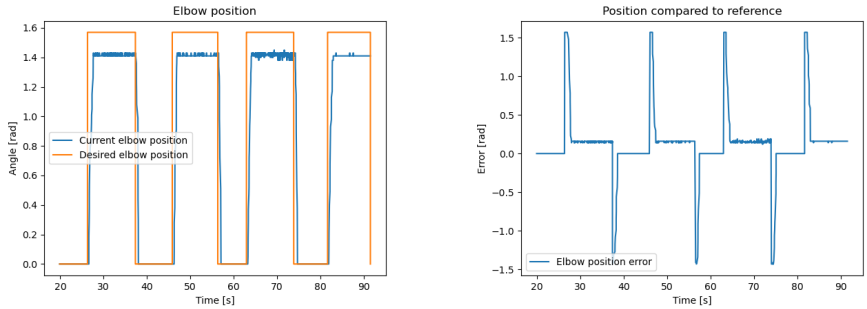


(b) Posterior forces

Figure 7.6: Tactile data

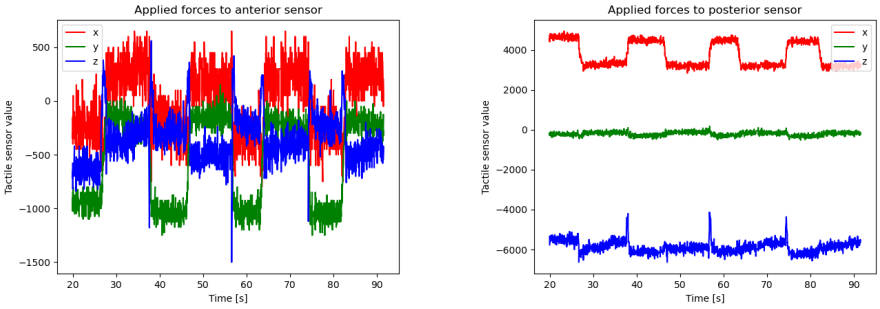
7.2 Periodical steps trajectory

7.2.1 Neutral orientation



(a) Angle position compared to reference (b) Angle position error

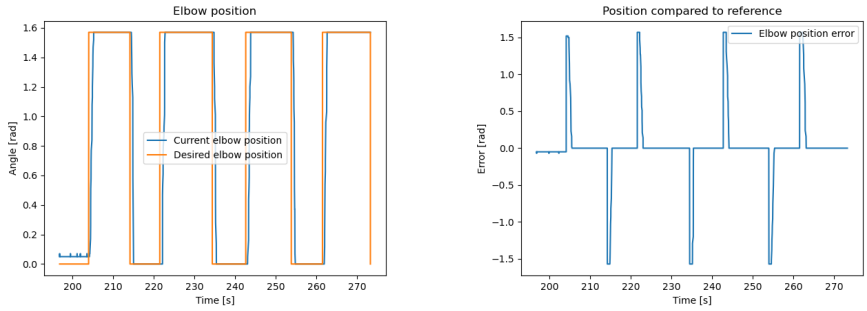
Figure 7.7: Angular position and error



(a) Anterior forces (b) Posterior forces

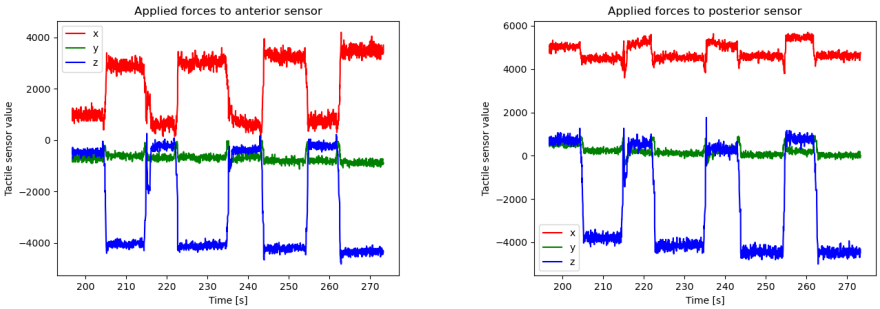
Figure 7.8: Tactile data

7.2.2 Horizontal orientation



(a) Angle position compared to reference (b) Angle position error

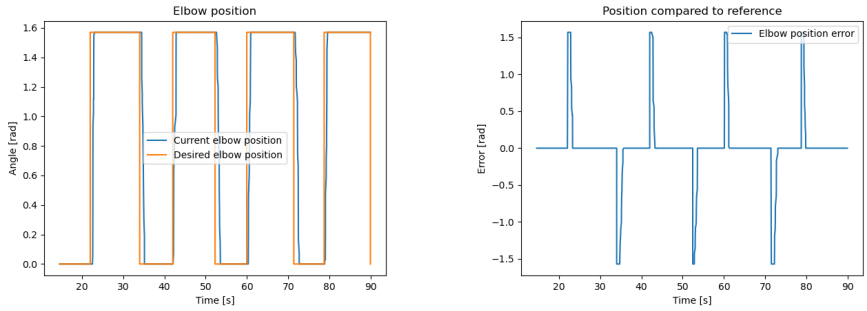
Figure 7.9: Angular position and error



(a) Anterior forces (b) Posterior forces

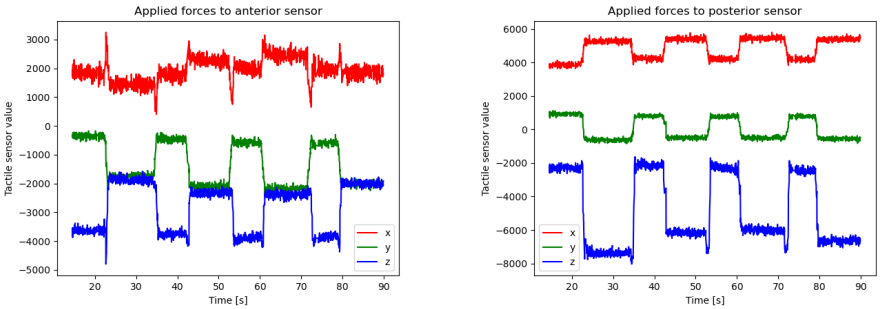
Figure 7.10: Tactile data

7.2.3 Upside down position



(a) Angle position compared to reference (b) Angle position error

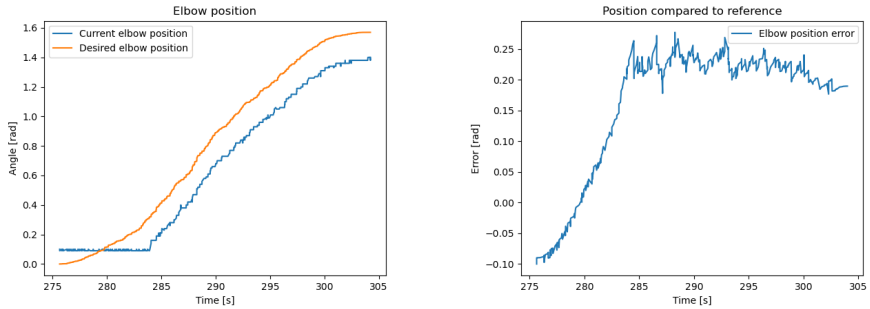
Figure 7.11: Angular position and error



(a) Anterior forces (b) Posterior forces

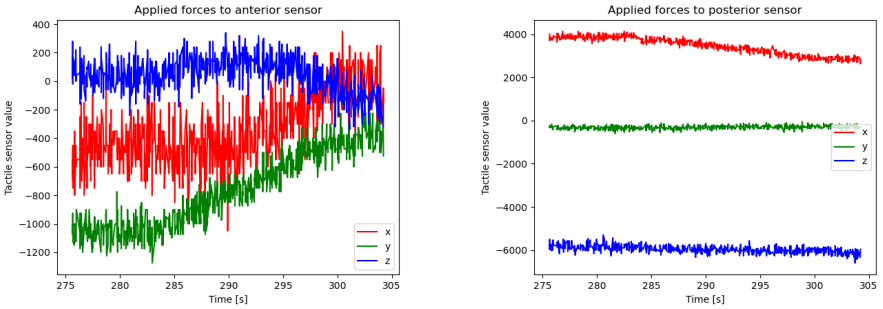
Figure 7.12: Tactile data

7.3 Smooth transition trajectory in the neutral position



(a) Angle position compared to reference (b) Angle position error

Figure 7.13: Angular position and error



(a) Anterior forces (b) Posterior forces

Figure 7.14: Tactile data

7.4 Simulated middle level controller in the neutral position

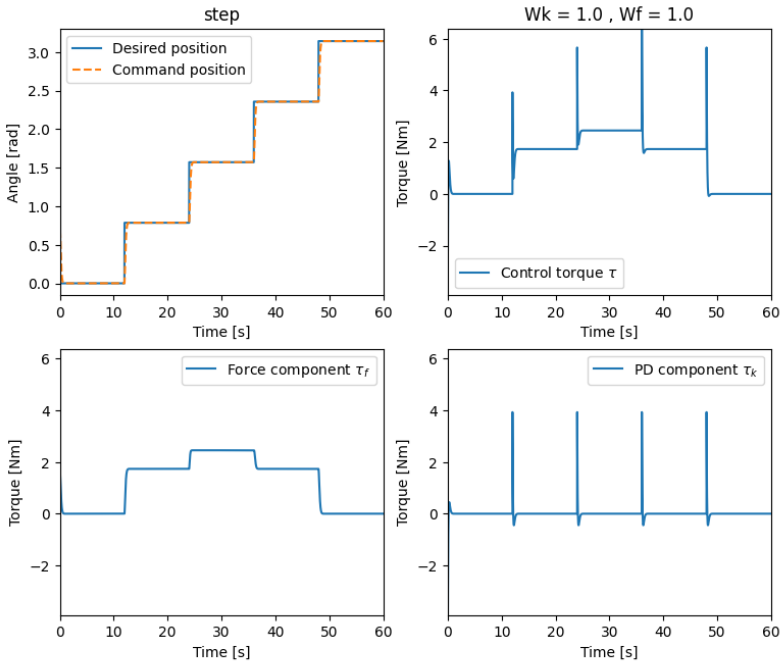


Figure 7.15: Step trajectory

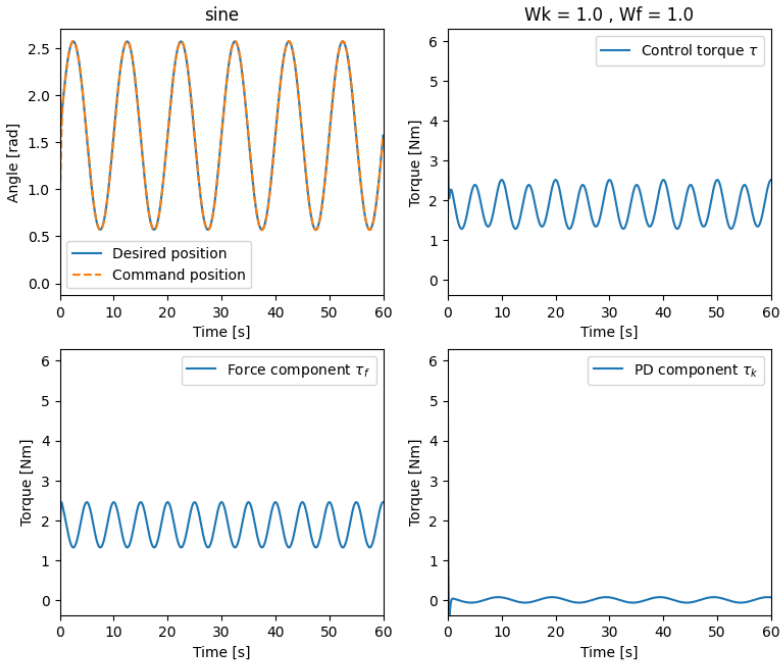


Figure 7.16: Sinusoid trajectory

7.5 Controlled sinusoidal trajectory

These results were gathered without anything attached to the orthosis in a horizontal orientation.

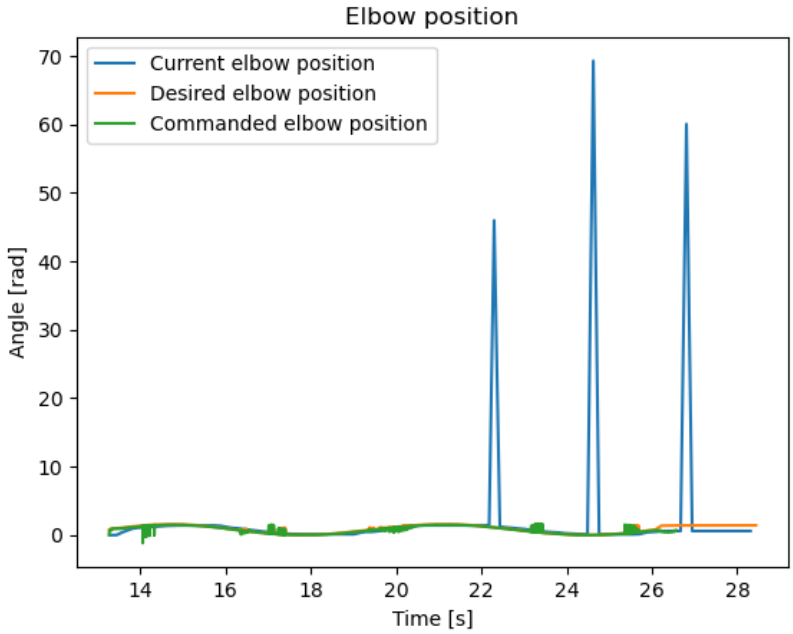


Figure 7.17: Angle position compared to reference

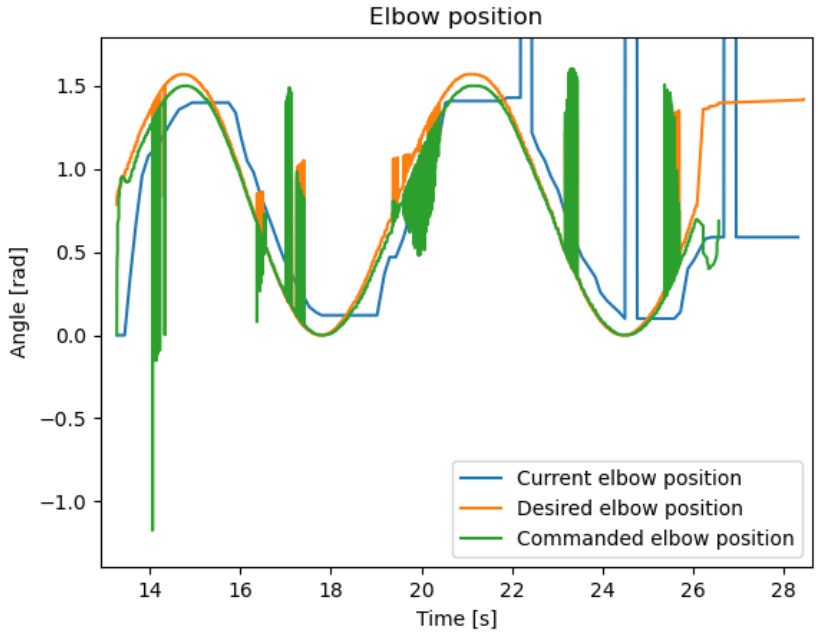
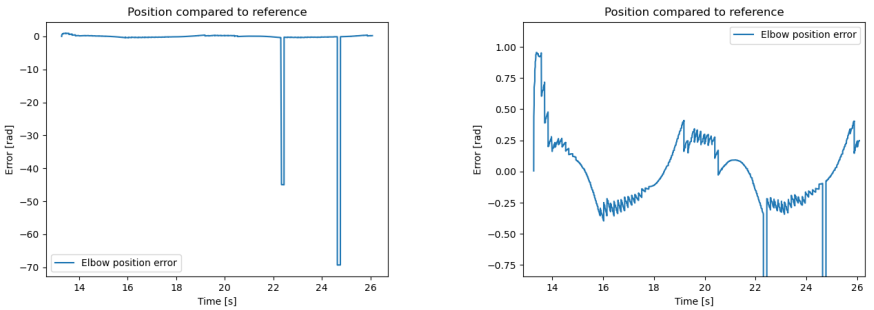


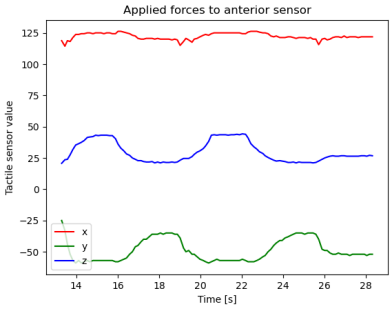
Figure 7.18: Angle position compared to reference zoomed in



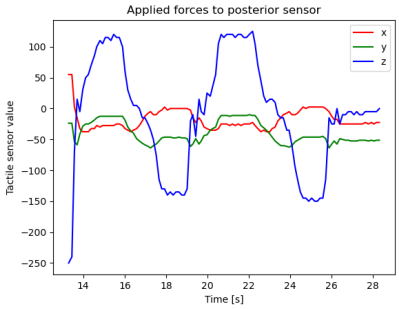
(a) Angle position error

(b) Angle position error zoomed in

Figure 7.19: Elbow position error

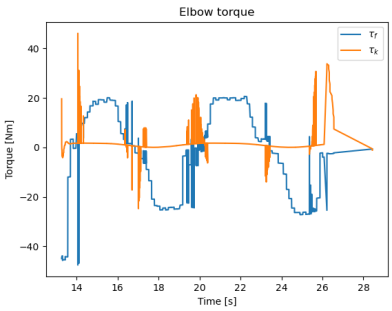


(a) Anterior forces

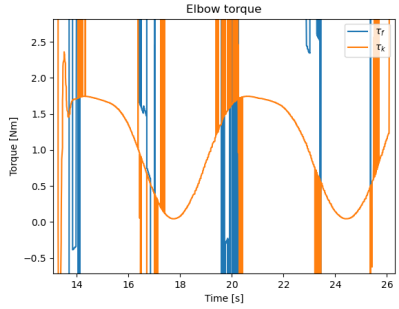


(b) Posterior forces

Figure 7.20: Tactile data



(a) Torque from middle level controller



(b) Torque from middle level controller zoomed in

Figure 7.21: Torque data

8

Discussion

8.1 Interpretation of results

8.1.1 Trajectories and movement

In all tests involving a sinusoidal trajectory, seen in Figures 7.1, 7.3, 7.5, and one with a step reference signal, Figure 7.7a, it can be seen that the orthosis never quite reaches the peak values. There are multiple possible explanations for this. One possible reason is the P-controller implemented. The controller causes a constant delay between the commanded angle and the true angle, and when the orthosis would reach the peak angle, it is already being commanded in the opposite direction. This is also seen in Figure 7.1b, where the spikes indicate that the elbow is not following the commanded angle.

In the case of the step signal seen in Figure 7.7a, another possible reason becomes apparent. When differences between the desired angle and the true angle become small, so does the output from the P-controller. At low speeds, the motor is not able to overcome friction without assistance, leading to it getting stuck. This is supported by the fact that one could hear the motor still trying to move, even though it was still. We can also see that the motor successfully reaches the bottom position, which makes sense since the weight of the sock/arm helps weigh the motor down. This is further supported by the "Smooth transition trajectory" test, seen in Figure 7.13a. It's seen that the motor does not start moving before the error has become large enough.

8.1.2 Tactile sensor data

We can observe a pattern when collecting data from the tactile sensors, exemplified in Figure 7.2a. It can bee seen than the x-axis is quite noisy. It is reasonable that the x-axis would be the most noisy, since this axis, going across the arm transversely (see Fig.6.8), is not going to experience the same predictable change in force as y- and z-axis. Another observation is that the least noisy data point is consistently the y-axis. A likely reason for the y-axis being the least noisy is the high friction of the sock against the sensor, which pulls on the silicone of the sensors. From the data, it seems the silicone is more malleable side to side than from compression only. In some tests, seen in Figure 7.4a, for example, we observe some drifting of values as the tests progresses. This is likely due to the sock shifting position, which would not happen to the same extent with a real arm.

8.1.3 Controller behaviour

The controller seems responsive and accurate in the simulated environment, as shown in Figures 7.15, and 7.16. The total control effort τ can be partially explained by the current position of the lower arm, given by the angle. The gravitational torque is at is maximum when the elbow angle $\theta = \pi/2$, which explains the offset in the controller output required to follow the trajectory. Sudden spikes in the output are due to the instantaneous transitions of the trajectory input.

8.2 Electronics design and operation

8.2.1 PC-to-ESP32 communication

A major hindrance has been communication between a PC and the ESP32 micro controller. The ESP32 could not successfully receive data from the PC, but it could send it. Instead of sending data directly, the data is routed through an Arduino Uno. This is a sub-optimal solution since it increases our transmission time, but it worked well enough that focus could be shifted to more pressing matters. The cause for this issue is still unknown, but it is with all likelihood a solvable one.

Switching the ESP32 for another micro controller was discussed, but ultimately avoided. The dual-core functionality is vital, since the tactile sensors are very slow at sending data. One core is dedicated to fetching data from the sensors.

8.2.2 Lack of IMUs

Originally, the plan was to use IMUs to track the position of the arm in space. This would not only allow tracking of the angle of the elbow, but also of the shoulder. Since the angle of the shoulder affects the dynamics of the system, this would have been a valuable inclusion. However, the IMUs available at short notice were too slow at returning data, and the data was too noisy. A potentiometer was used to calculate the angle instead.

8.2.3 Slow data retrieval from tactile sensors

The hall effect sensors are quite slow at sending data. By reducing the amount of filtering and oversampling, the time can be reduced to around $25ms$. Although a significant improvement over the original $127ms$, it is still too slow for accurate control.

Attempts to convert the values returned by the sensors to an approximate force in Newtons failed since the range of values the sensors return vary dramatically depending on the placement of the sensor on the arm. The forces calculated become so far separated from the real force applied that a completely arbitrary range of values is easier to implement and easier to handle during calculations since it can be scaled in any way appropriate. Currently, the sensors re-calibrate every time the orthosis is switched on. If one were to calculate the actual forces applied to the sensors instead of an arbitrary range of values, a way more advanced calibration stage would be required. Future developments should involve a custom PCB and optimized code, to enable easier manufacturing of sensors and faster retrieval of data.

8.2.4 DC motor as an actuator

The choice of actuator has brought with it both benefits and drawbacks. DC motors are simple to control, and achieving smooth motion is trivial. The DC motor chosen has gearbox mounted, giving it enough torque to lift the forearm by itself. Due to budget constraints, the motor chosen has no encoder. A potentiometer is instead used to retrieve the angle of the elbow. IMUs were considered instead of a potentiometer, but they were too slow and the data was too noisy. The motor is also quite heavy and can therefore not be placed alongside the elbow, since it caused too much torque on the orthosis, causing it to slide on the arm of the person wearing it.

A future development could be to use either a smaller DC-motor with adequate power, or a servo motor. This may allow for more comfortable use of the orthosis, greater range of motion, and more precise control.

8.2.5 Limitations in low-level control

The on-board control of the motor had to be adapted due to the slow transmission of data. The original plan was to use the desired angular velocity as the guideline for motor movement. This was not possible to implement effectively since the update-rate of the system is slow enough that a significant difference in desired velocity can occur between each refresh. This leads to jerky and inaccurate motion. Instead, the desired angle is used and a proportional controller is implemented. The reason for using a P-controller instead of a PID-controller is the very small range of acceptable speeds the motor can operate at. Below certain speeds, the motor will not move without assistance due to friction, and above certain speeds it is too fast for precise and safe operation. A PID-controller is also more heavily affected by delays in data transmission, as large shifts in received values will lead to an even larger response from the controller. The inconsistencies in data transmission also make a PID-controller difficult to tune.

8.3 Middle level control

The controller's effectiveness was occasionally compromised by inherent limitations in the hardware configuration and certain theoretical constraints in the implementation approach. These limitations, ranging from hardware inadequacies to simplifications in the control strategy, not only influenced the accuracy and reliability of the controller but also shed light on critical areas for future development.

8.3.1 Hardware limitations and their effect on the control system

IMUs

The absence of IMUs in the system means it cannot detect the orientation of the upper arm. Consequently, arm orientations must be hard coded (vertical, horizontal, etc.), which significantly impacts the accuracy of calculated gravitational torques in the dynamics and the computation of the Jacobian matrix. This limitation restricts the system's adaptability to varying user postures, potentially reducing its effectiveness in real-world application.

Angle discrepancy and actuator placement

The angle measurement derived from the potentiometer does not accurately represent the actual angle of the human arm. This discrepancy can introduce errors in both the kinematics and dynamics models, where this parameter is crucial for accurately representing the human arm position and adapting a control effort accordingly.

Placing the actuator at an offset below the elbow would most likely effect the dynamics of the system as well. With significant mass, this placement alters the effective center of mass of the system, affecting the inertial, gravitational, and frictional components. With our simplification of assuming the center of mass aligned with the lower arm at half its distance; the modelled dynamics does not accurately simulate the real-world environment and most likely cause skewed results.

Tactile sensor mapping

The Hall effect tactile sensors used in our system were not calibrated to map magnetic field strength to force values accurately. This lack of accurate force mapping leads to erroneous torque calculations, which in turn compromise the control accuracy when the sensor data is serves as primary control for the arm movement.

Theoretical simplifications effect on controller

The decision to implement a Proportional-Derivative (PD) controller, instead of a model-based approach such as using a regressor, introduces certain limitations in the control strategy. While PD controllers are simple and easier to implement, they are prone to steady-state errors and may not compensate for non-linear dynamics as effectively as model-based methods. Implementing a regressor would involve data collection to find the relationship of the angle position and measured torque. Using trajectories that emphasize acceleration, velocity, and position independently would provide a quantitative basis for establishing the inertial, frictional, and gravitational components of the dynamics. This method would provide a more robust controller that uses the complete dynamical model of the human arm together with the exoskeleton to calculate the internal forces and adjust its control effort accordingly.

Further simplification was done on the dynamic model, assuming linear relationships for the inertia and viscous friction and excluding the elasticity of the human arm. These simplifications mean that the model may not truly capture the complex, non-linear behaviors of the physical system, leading to inaccuracies

in the simulated responses.

Since the middle level controller was not implemented into the system to where data could be measured, it can only be theorized the extent of which said simplifications would affect the overall performance and stability of the robotic arm.

8.4 Integration of the middle level controller

The results presented in Section 7.5 show significant disturbances in elbow position, desired trajectory and commanded trajectory, where the latter originates from the controller. However, Figure 7.18 displays major disturbances from the desired trajectory propagating to the commanded trajectory which seems to follow even these disturbances to some extent. An objective for further development of the system can be to integrate the middle level control properly into the ROS2 system such that nodes are limited to specific tasks and disturbances in the resulting signals are avoided.

These results also show spikes in the elbow position spanning far beyond the limited range of motion of the elbow joint. These values did not appear in reality and was later filtered out for the measurements done without the controller. To facilitate analysis of the system behaviour in this context, zoomed in plots of the results are presented in Figures 7.21b and 7.19b which disregard these false values.

9

Conclusion

The purpose of this report was to investigate the possibility of constructing an elbow orthosis that, with affordable and maintainable hardware, can aid efforts in physiotherapy. None of the issues encountered during this project seem to be fundamental to the idea of an intelligent elbow orthosis. The problems stem from either a lack of time, funding, or expertise. The more pressing issues, such as integration of the mid-level controller, and slow response times, are most certainly solvable ones.

It is apparent that mechatronic solutions can help solve some of the problems that beset physiotherapy and its accessibility. With more resources and expertise, a more stable solution similar to the one presented in this report could be constructed. Such an orthosis would likely have a legitimate opportunity to allow for more accessible and personalized care than the status quo.

References

[1] Thimal Kempitiya, Daswin De Silva, Ebonie Rio, Richard Skarbez, and Damminda Alahakoon. Personalised physiotherapy rehabilitation using artificial intelligence and virtual reality gaming. In *2022 15th International Conference on Human System Interaction (HSI)*, pages 1–6, 2022.

[2] Natalia Paredes-Acuña, Nicolas Berberich, Emmanuel Dean-León, and Gordon Cheng. Tactile-based assistive method to support physical therapy routines in a lightweight upper-limb exoskeleton. *IEEE Transactions on Medical Robotics and Bionics*, 4(3):541–549, 2022.

[3] Carma Ayala, Jing Fang, Cecily Luncheon, Spencer C King, Terry Chang, Matthew Ritchey, and Fleetwood Loustalot. Use of outpatient rehabilitation among adult stroke survivors - 20 states and the district of columbia, 2013, and four states, 2015. *MMWR Morb Mortal Wkly Rep*, 67(20):575–578, May 2018.

[4] Manuel Cardona, Marie Destarac, and Cecilia García Cena. *Robotics for Rehabilitation: A State of the Art*, pages 1–11. Springer Singapore, Singapore, 2020.

[5] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics*. Advanced Textbooks in Control and Signal Processing. Springer London, London, 2009.

[6] Bengt Lennartson. *Reglerteknikens grunder*. Studentlitteratur, Lund, 3., [omarb.] uppl. edition, 2001. OCLC: 186570597.

[7] H.J. Landau. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706, 1967.

[8] L. P. Pook. *Simple Pendulums*, pages 7–25. Springer Netherlands, Dordrecht, 2011.

[9] Luis F. Chaparro and Aydin Akan. *Section 1.4: Signal Generation*. Elsevier, 2019.

[10] Foxglove Technologies Inc. Foxglove Studio. https://docs.foxglove.dev/docs/connecting-to-data/ros-foxglove-bridge/, 2024. [Online; accessed May 6, 2024].

[11] ROS2. common interfaces. https://github.com/ros2/common_interfaces, 2024. [Online; accessed May 3, 2024].

[12] Foxglove Technologies Inc. Foxglove Studio, 2024.

A

ROS2 graphs

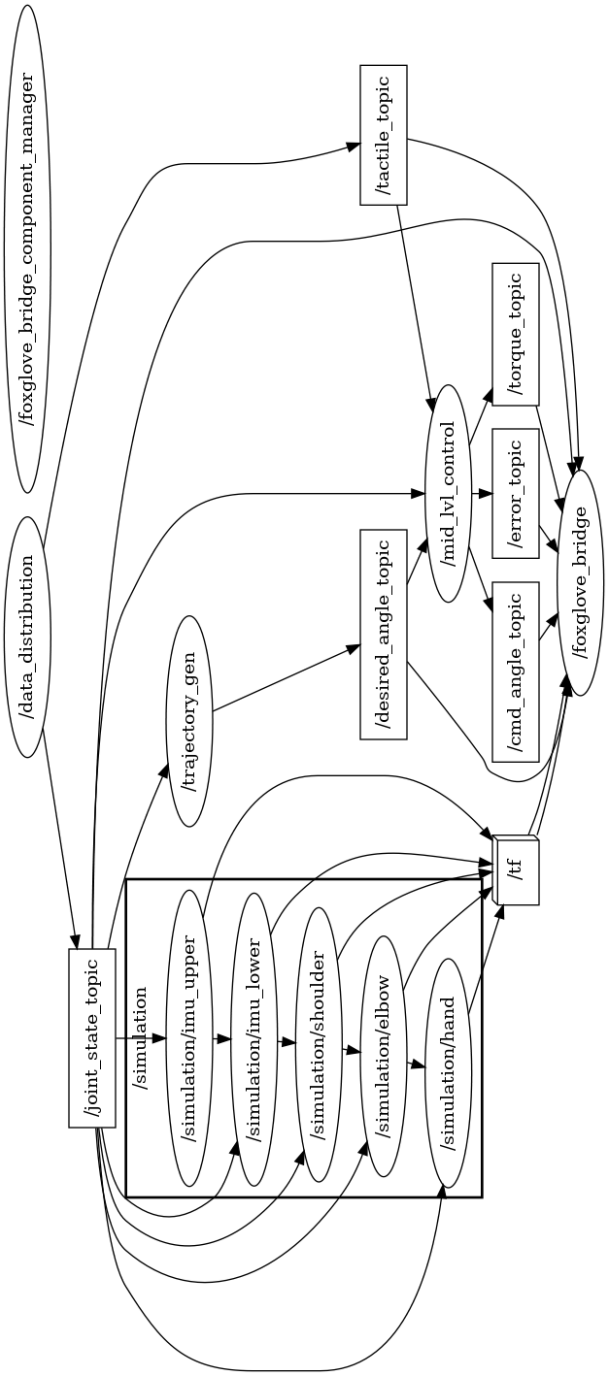


Figure A.1: ROS2 architecture

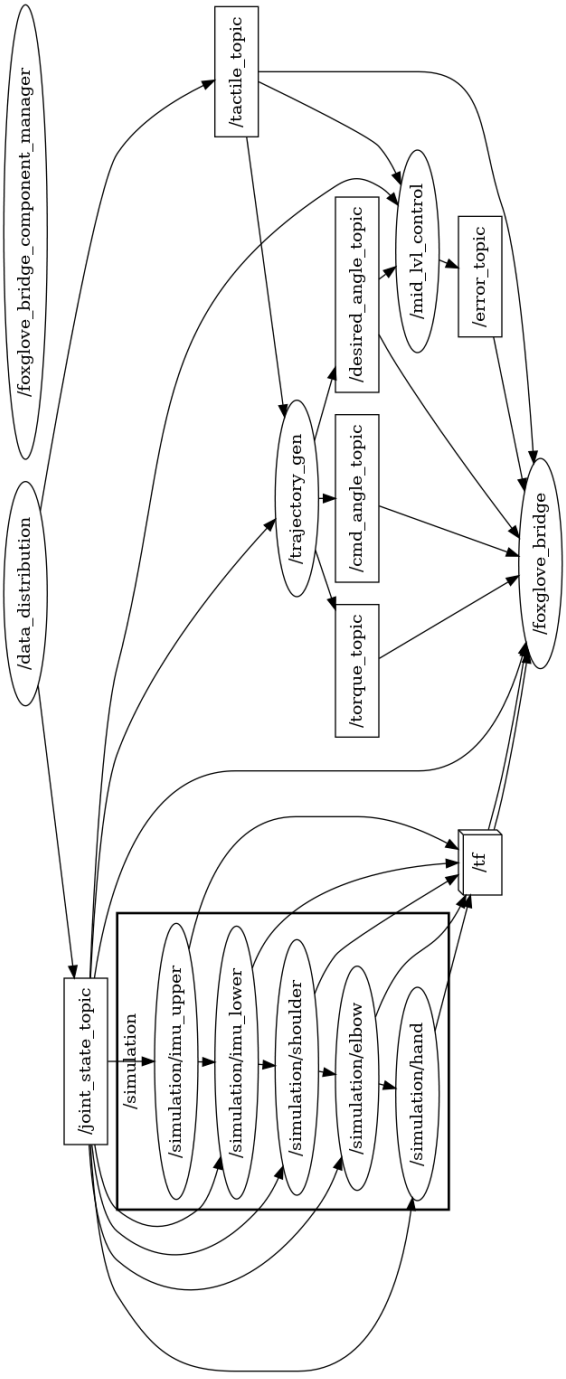


Figure A.2: Reduced ROS2 architecture

B

GUI screenshots

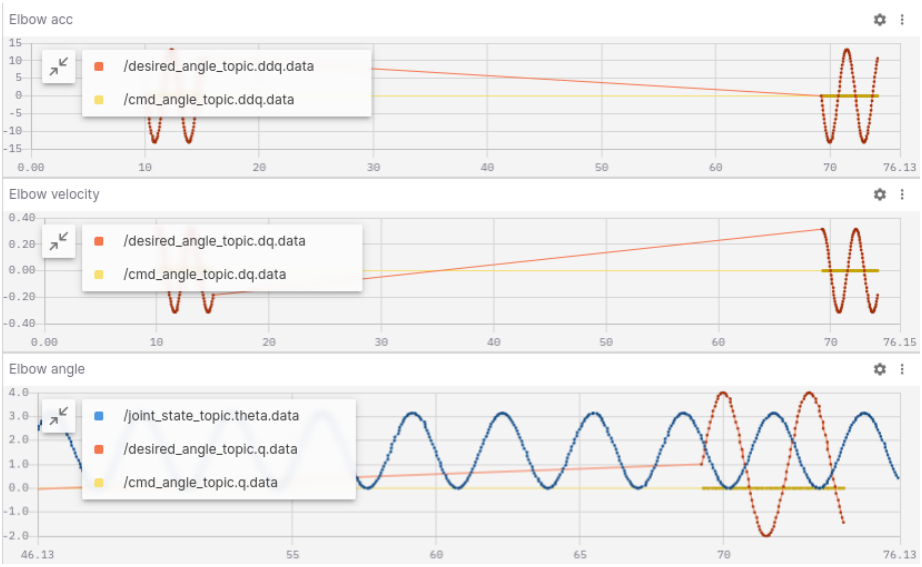


Figure B.1: Monitoring of current state. Image generated using Foxglove Studio [12]. Reproduced with permission.

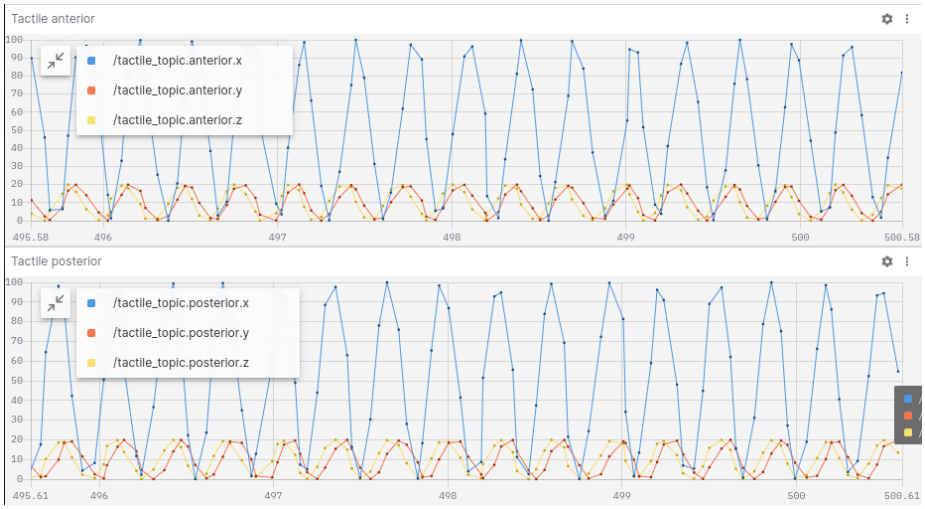


Figure B.2: Monitoring of tactile data. Image generated using Foxglove Studio [12]. Reproduced with permission.

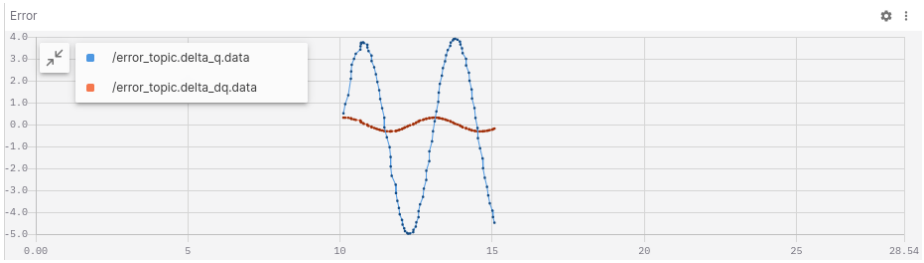


Figure B.3: Monitoring of error, reference signal compared to state. Image generated using Foxglove Studio [12]. Reproduced with permission.

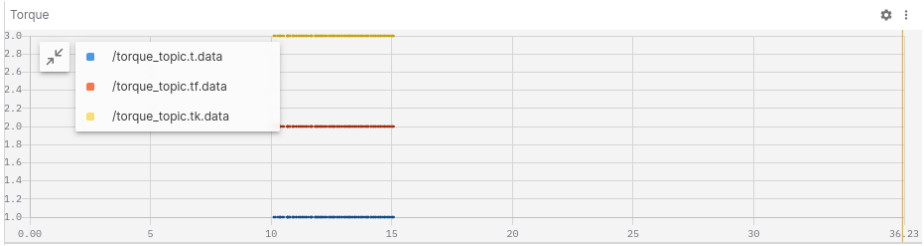


Figure B.4: Monitoring of torque. Image generated using Foxglove Studio [12]. Reproduced with permission.

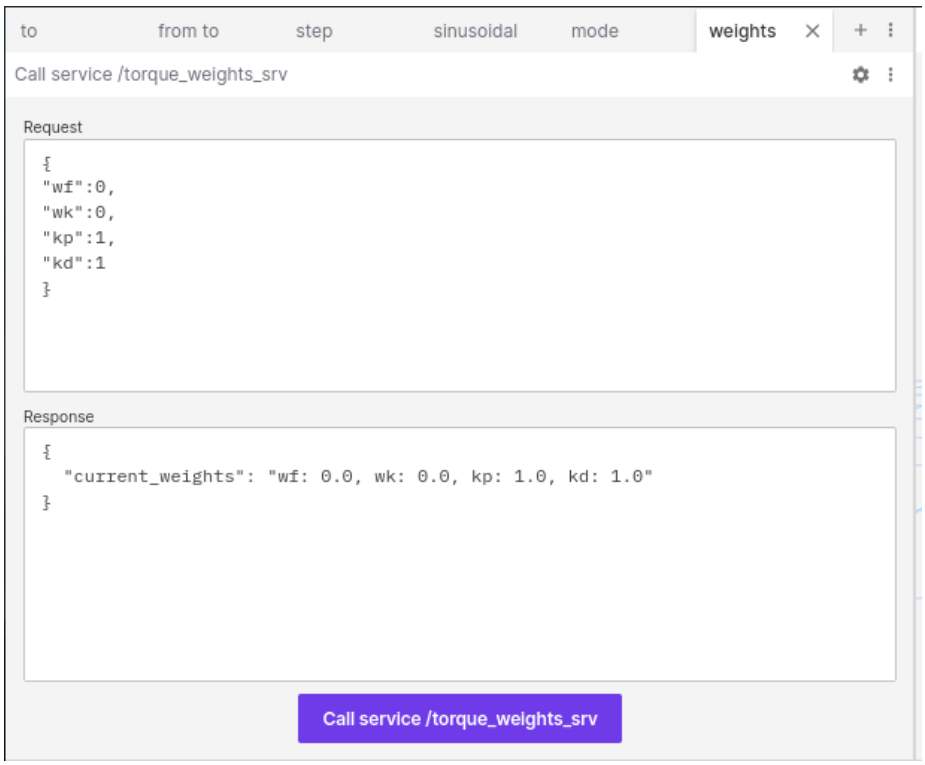


Figure B.5: Set torque weights. Image generated using Foxglove Studio [12]. Reproduced with permission.

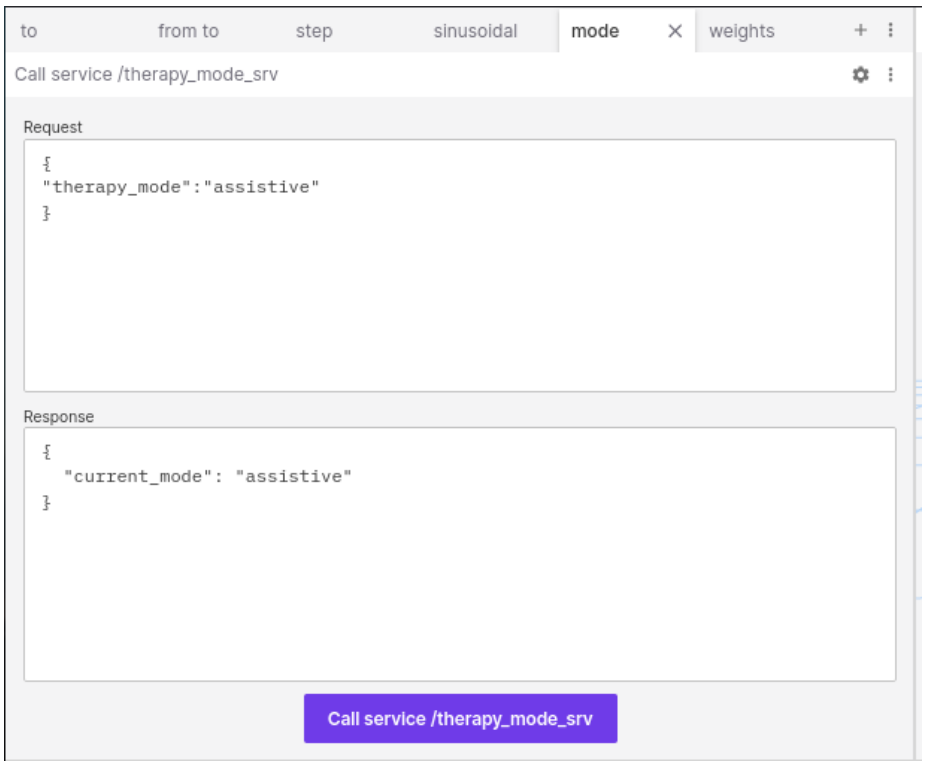


Figure B.6: Set control mode. Image generated using Foxglove Studio [12]. Reproduced with permission.

C

Test object photos

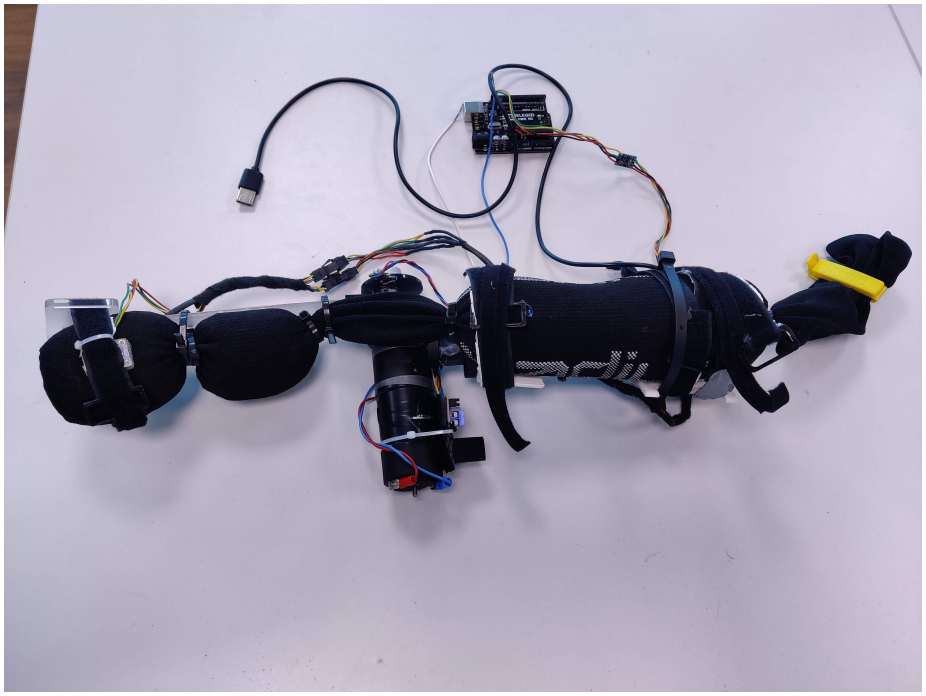


Figure C.1: Medial view of test sock

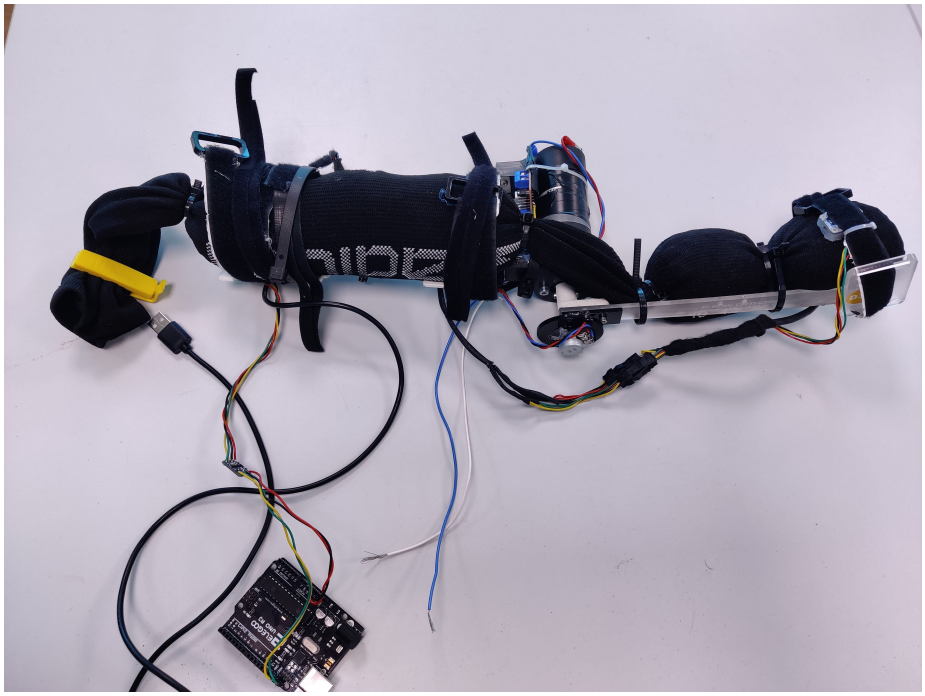


Figure C.2: Lateral view of test sock

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 20xx
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY