PDI Report

Submitted by

David Kirov
Final year student
at
Ecole Centrale de Lille

Under the guidance of

Alexandre Kruszewski
Professor at Ecole Centrale de Lille

2023-2024

centralelille
ÉCOLE CENTRALE DE LILLE

Acknowledgements

First I would like to thank Alexandre Kruszewski for his guidance during this project. Many thanks to Antoine Dequidt for his insights on wearable robotics. Thanks to Fabien Verbrugghe and Gilles Marguerite for lending me their technical expertise. Last but not least, big thanks to the DEFROST team for accepting my participation in this project.

Introduction

In the context of Centrale Lille's last year project, Alyssia Colas and I tackled the design of a soft wearable robotic orthosis for the elbow. This topic was proposed by Prof. Kruszewski, who is affiliated with the DEFROST team at Inria Lille. The team has secured funding for the development of this system. The goal is to provide an efficient low-cost and low-maintenance solution for elbow rehabilitation.

The loss of motor function is commonly caused by neurological disorders such as stroke, cerebral palsy or Parkinson's disease. Rehabilitation therapy is essential to regain functional capacity in the impacted areas through neuropasticity. Traditionally, physical therapists would conduct one-on-one manual treatment to recovering patients. In recent years, researchers have proposed numerous robot-aided rehabilitation devices as a prospective solution. The most recent research explores soft wearable devices because they present a number of advantages compared to the currently more prevalent rigid exoskeletons. They present a more compact structure, lighter weight, lower inertia, safer operation, more comfortable interaction, and they are easier to adapt to anatomical variations.

To create a novel solution to the problem, the project was split into two separate parts: the design of the mechanical system responsible for the kinematic properties of the solution, and the design of the control system responsible for the transformation of user intention into movement. The latter is the topic of this report paper and the subject of my work.

Contents

State of the art

3.1 Introduction

Multiple research papers have covered the topic of soft wearable orthoses for upper as well as lower limbs. Lu et al. [1] have developed a joint torque estimation control strategy for a soft elbow exoskeleton to provide effective power assistance. Later, Wu at al. have expanded on this work by optimizing anchor points in the mechanical design and modifying the control strategy [2], then by updating the estimation of joint torque to determine motion intention [3].

3.2 Sensors

When it comes to the control system itself, multiple sensors can be used to acquire relevant data. This data is required for the generation of input as well as the control feedback. In the case of an upper limb orthotic device, the system input is defined by the user's motion intention and feedback can include position, velocity, acceleration, torque and force.

According to a systematic review of the signals, sensors and methods for controlling active upper limb orthotic devices [4]:

> The results indicate that electromyography (EMG) was the most used signal for controlling these devices, described in 36 articles, followed by electroencephalography (EEG), used by 15 authors. Among the other signals used were force myography (FMG), force-sensing resistors (FSR), inertial measurement unit (IMU) sensors and external torques.

3.2.1 Motion intention

Multiple signals can indicate motion intention in a patient. As previously stated, EMG and EEG are most often used for this application. Alternatives such as FMG may also be used.

- EMG measures muscle response or electrical activity in response to a nerve's stimulation of the muscle
- EEG measures electrical activity in the brain
- FMG measures the volumetric changes of the underlying musculotendinous complex during muscle activities [5]

One advantage of EMG and EEG signals over FMG is that they manifest before the corresponding muscle response. Begovic et al. show that on average, the electromechanical delay between EMG and force is 50ms [6]. Furthermore, EEG precede their EMG response by around 23ms [7]. Therefore, EMG and EEG signals can allow for a quicker reaction to the user's motion intention compared to FMG.

3.2.2 Kinematics

Various sensors can serve the purpose of gathering information about the system kinematics. The most

3.2.3 Dynamics

3.3 Actuators

3.4 Processing

Materials and methods

4.1 Initial decisions and simplifications

From the research that was conducted up to this point, a couple of key decisions could already be made.

First, as EMG signals seemed to be the most promising for the control of upper limb active orthotic devices [4], and given the fact that a non-invasive system was required, surface EMG (sEMG) sensors were chosen. These sensors measure the electrical activity from nerve stimulation at the surface of the skin.

Second, the research revealed that the actuator system best adapted to the project requirements would be a system of cables pulled by DC motors. Initially, a three-cable system was imagined in order to mimic the three muscles involved in elbow flexion: the biceps brachii, the brachioradialis, and the brachialis. After consideration of the force produced by each muscle in a single flexion, it was concluded that the system can be functional with a single cable. Therefore, a single cable-motor system was chosen for simplicity.

4.2 SysML Models

4.2.1 Context

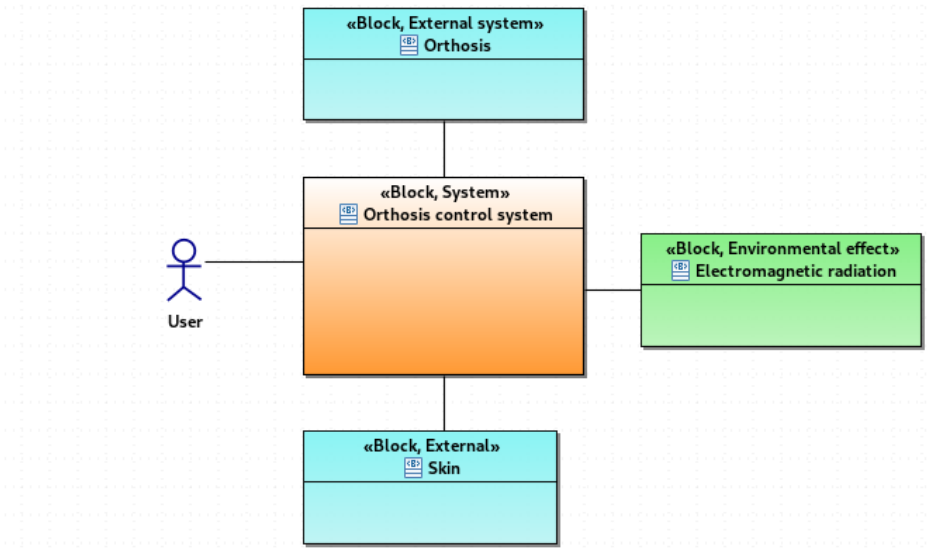


Figure 4.1: Context diagram

4.2.2 Use case

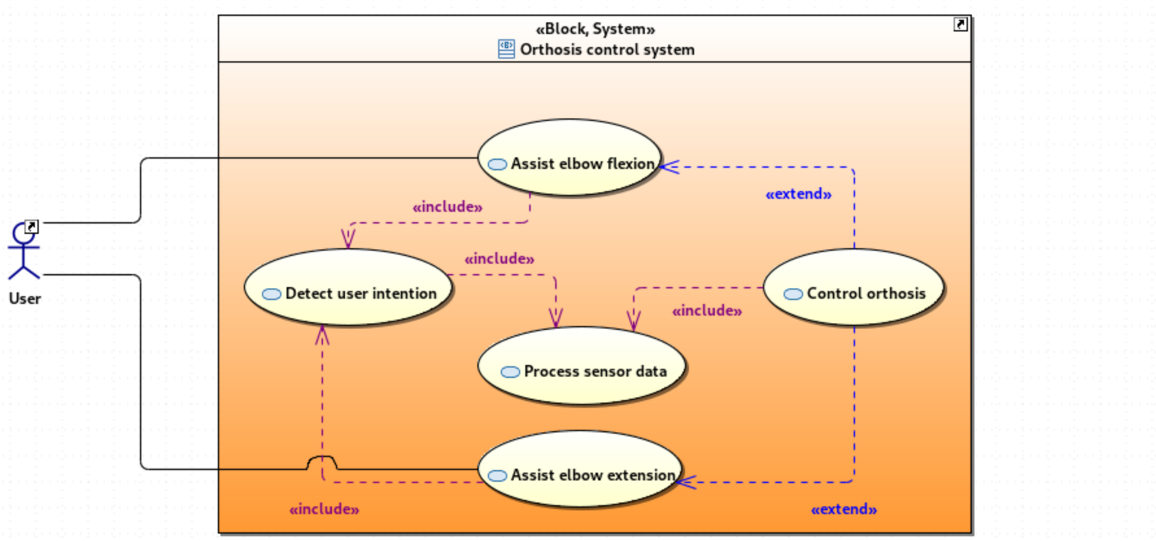


Figure 4.2: Use case diagram

4.2.3 Requirements

	getAppliedStereotypes()
Allow the control of an elbow orthosis	Requirement, functionalRequirement
Quantify user motion intention	Requirement, functionalRequirement
Receive user sEMG data	Requirement, extendedRequirement
Process user sEMG data	Requirement, extendedRequirement
Control actuators	Requirement, functionalRequirement
Receive system state feedback	Requirement, extendedRequirement
Calculate input from motion intention and system state	Requirement, extendedRequirement
React quickly enough to provide assistance	Requirement, performanceRequirement
Deliver enough force to provide assistance	Requirement, performanceRequirement
Be light enough to be carried daily	Requirement, performanceRequirement

Figure 4.3: Requirement table

4.3 Implementation

A few research papers [1, 2, 3] served as guidelines for the project. The goal was to reproduce as much as possible from these papers within the project time frame.

4.3.1 Hardware

The main computer system used to run all programs is a 64bit Linux machine with 8GB of RAM, a 4-core Intel M-5Y10c @ 2.000GHz and Intel HD Graphics 5300, running Fedora 39 (Workstation Edition).

A FireBeetle 2 ESP32-E was used. Power was supplied by a 3.7V and 400mAh battery due to the sensitivity of the analog sEMG sensor.

Two Adafruit BNO055 IMUs were used to measure joint angle, rotation speed and acceleration.

The sEMG sensor initially used was the DataLITE Wireless EMG sensor from Biometrics Ltd. However, there was difficulty integrating the sensor with a Linux environment as the Biometrics application was only available for Windows machines. Emulation through Wine was tested but library dependencies could not be resolved. Therefore, another sensor was required. The Myoware Musce Sensor 2 stood out among other options as it was used in multiple prior works [1, 2, 3, 8]. As shown in Figure 4.4, this sensor also had the advantage of having built-in signal processing.

All sensors were connected as shown in Figure 4.5. Cables were carefully twisted to minimize electromagnetic noise in the sEMG signal.

The Actuator system was composed of a 150W brushless DC motor from Maxon (BLDC-Motor - EC60fl. BL Y 150W 1WE A K) and an EPOS4 Maxon motor driver (EPOS4 70/15). Power was supplied by a 3A-30V power supply found at Centrale.

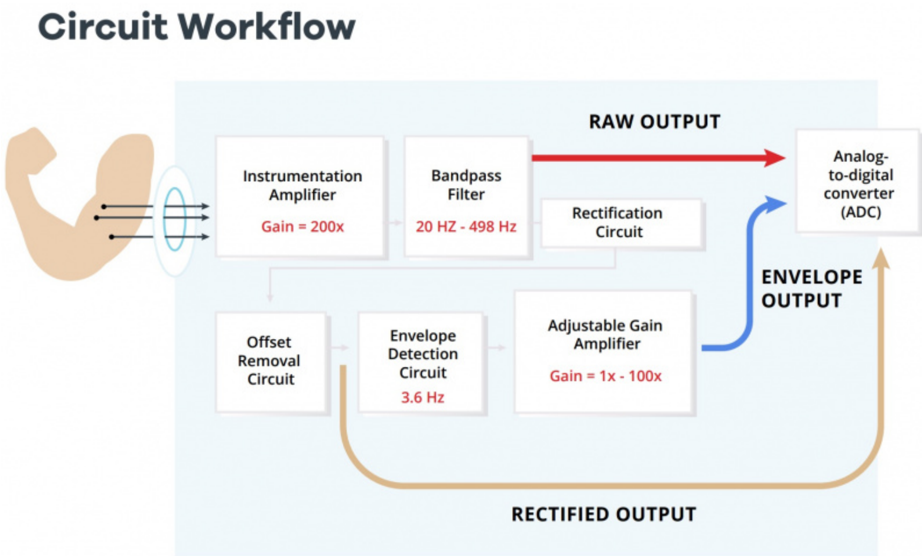


Figure 4.4: Myoware muscle sensor preprocessing diagram

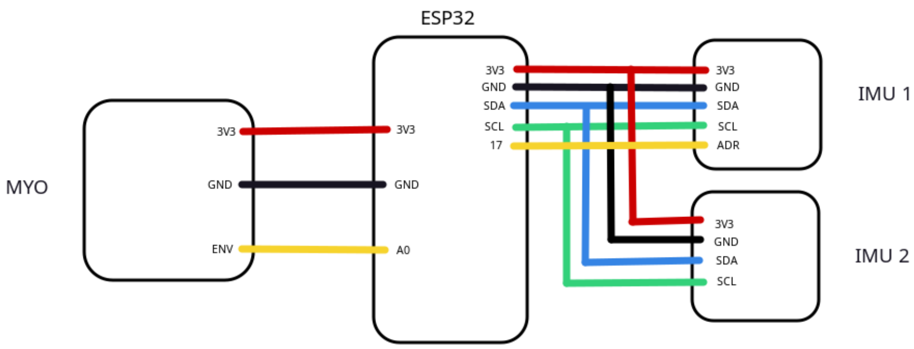


Figure 4.5: Sensor system circuit diagram

4.3.2 Control strategy

A sEMG torque estimation control strategy following the work of Lu et al. [1] was put in place. The key difference between the control strategy presented in this report and the one presented by Lu et al. is that the system output torque is controlled instead of the system angle.

A version of the control strategy is illustrated in Figure 4.6. The sEMG signal gathered from the Myoware muscle sensor (Myo) is fed into a torque estimation algorithm that determines user intended torque with a pre-trained neural network (NN). Depending on the NN inputs, the control strategy could either be open or closed loop.

In the case of an open loop system, only the user sEMG signal is used to determine motion intention and intended torque. That torque is then transformed into required motor torque by the orthosis inverse model which is, in turn, transformed into a current requirement for motion assistance.

A closed-loop version could also be considered. In that case, user elbow joint angle is also considered in the torque estimation algorithm.

No force feedback is used because a high degree of accuracy in the assistance torque was not considered necessary.

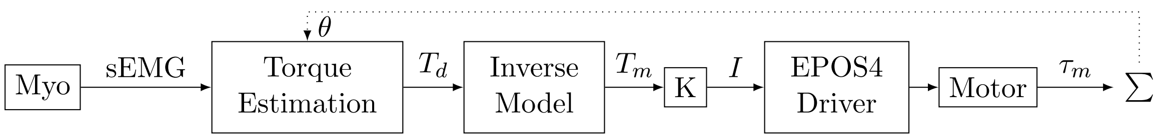


Figure 4.6: Control Diagram - T_d designates user desired torque (the torque necessary to perform the movement that the user intends), T_m is the corresponding motor torque required as input to the orthosis system, θ is the angle of the elbow joint, K is a gain that sets the assistance level of the system and turns T_m into a current requirement I

4.3.3 Communication strategy

Establishing a robust communication was essential to ensure reliable data transmission between the sensors and processing units. Most of the communication was pre-determined by the sensors and equipment. The IMUs communicate via I2C, the Myoware sensor outputs an analog value, so is connected to an analog pin on the ESP32, and finally the ESP4 driver communicates with the host PC via USB.

The sensor subsystem, comprised of the two IMUs, the sEMG sensor and the ESP32, communicates with the host PC via BLE. This decision was made due to the sensitivity of the analog sEMG sensor to outside electromagnetic radiation. Indeed, using the sensor while powering the ESP32 by USB produces sub-optimal results. In that case, if the PC is charging while using the system, the sensor readings are over 3600 without any muscle activity. If not charging, the sensor reads correct values most of the time but experiences value spikes in the order of 1000, which renders the signal unusable. Figure 4.8 illustrates this behavior. It was suspected that imperfections in the PC power supply are the cause of these abnormalities, so the choice was made to use a battery. BLE was chosen over WiFi or Bluetooth for its advantage on battery life and because the data that need to be transferred consist of two numbers every 10-40ms, which corresponds to a data rate low enough for BLE.

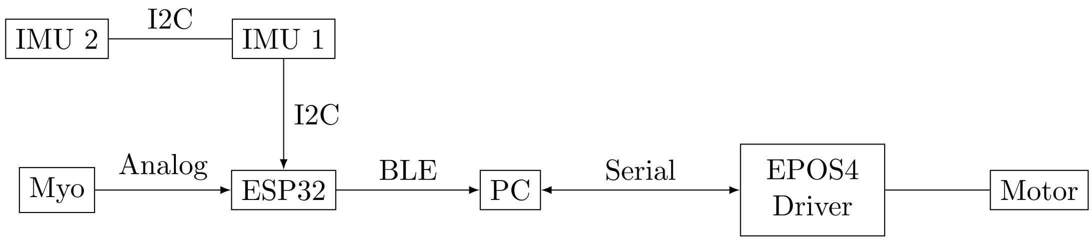
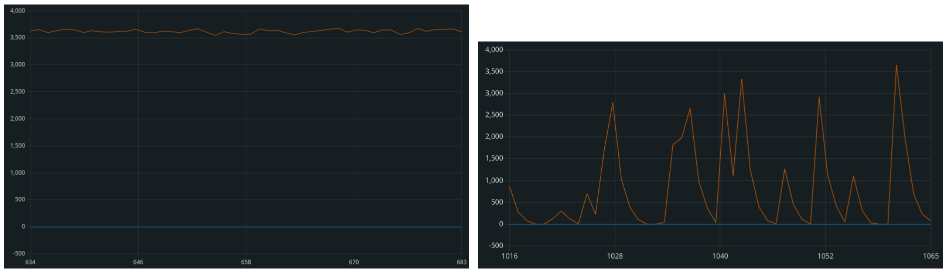


Figure 4.7: System communication diagram



(a) While PC is charging (b) While PC is not charging

Figure 4.8: Signal read from the Myoware sensor (orange) while the ESP32 is powered by USB and the biceps muscle is completely relaxed

4.3.4 Software

Implementing the control strategy required multiple software components to be developed. Effort was made to render said software easily comprehensible, reusable and upgradeable. A Github repository was created to share all code and provide a guide for anyone interested in replicating the methods described in this report.

Embedded software

A program that reads sensor data, transforms the IMUs' quaternions into a joint angle and sends processed data over BLE was developed for the ESP32.

The $Adafruit\_BNO055$ library was used to acquire data from both IMUs. The ADR pin of the first IMU on the I2C bus was set to high to allow for both IMUs to be used simultaneously. At each measurement, two quaternions, q_1 and q_2, are obtained. To infer the joint angle, first the relative quaternion q_r representing the rotation needed to transform q_1 into q_2 is calculated, then Euler angles (ψ, θ, ϕ) are extracted from q_r. The joint angle is then identified to one of the obtained Euler angles by experimentation.

For the sEMG data, the ESP32's A0 pin was simply read.

Two ways of sending the sensor data were implemented: USB and BLE. The USB approach is straightforward but produces undesirable effects on the sEMG sensor as discussed in previous sections. The BLE approach involves using the ESP32 as a BLE server, appropriately named $ESP32\_BLE\_server$, and advertising a service with a single characteristic containing both the joint angle and sEMG value.

Torque estimation

For torque estimation, a data-driven NN approach was taken.

Initial data was gathered at INRIA with the help of Yiru Guo and Alyssia Colas. Said data consisted of joint angles and sEMG from the biceps brachii muscle. Joint angle measurements were made with the Biometrics Ltd. DataLITE goniometer and sEMG was measured with the DataLITE sEMG sensor. The data acquisition protocol involved performing multiple controlled elbow flexions, covering the entire range of motion (ROM) of the joint. Multiple experiments were performed, differing in carried load, flexion speed and the presence of pauses in flexion at intermediate elbow angles.

The acquired sEMG data was processed according to the method described by Xu et al. [3]: a second order Butterworth filter with a pass-band of 10-490Hz was used to remove undesirable noise, a 50Hz notch filter was then applied to remove power frequency disturbance from nearby equipment and a first order Butterworth filter with a high-pass cutoff frequency of 410Hz was applied. Finally, the sEMG envelope was obtained with a full-wave rectifier followed by a first order Butterworth filter with a low-pass cutoff frequency of 1Hz. The resulting signal was then normalized to serve as training data for the NN.

A sequential NN with a single hidden layer composed of 16 neurons was developed. The training and validation data were created from the initial experiments. Reference torque was obtained through calculation with the following formula, gotten from Lu et al. [1]:

$$\tau = ((m+\omega)l_{cm}^2 + J)\ddot{\theta} + \beta\dot{\theta} + (ml_{cm} + \omega l)g\sin\theta$$

Where m is the mass of the subject forearm and arm, ω is the mass of the carried load, l is the distance from the elbow to the center of the hand, l_{cm} is the distance from the elbow to the forearm center of mass, J is the moment of inertia of the forearm, g is gravitational acceleration and β is joint friction.

Four neural network models with identical one hidden layer architecture were trained. The initial two models were trained on a singular dataset. One model was trained solely on sEMG signals to predict torque, while the other model incorporated angle information in addition. These models were labeled as *open_model* and *closed_model* respectively. Subsequently, a second set of models were trained utilizing all accessible test data. These models also varied in their inputs akin to the first pair. They were denoted as *multi_open_model* and *multi_closed_model* respectively.

Inverse model

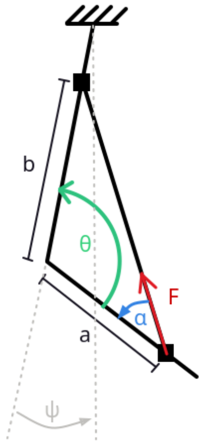
To calculate the required motor torque T_m from the desired torque T_d, an inverse model of the orthosis force transmission must be applied. Unfortunately, such a model was not available so ideal conditions were considered. The model illustrated in Figure 4.9 was used and perfect force transmission was hypothesised. The upper arm segment was considered to be along the vertical axis for simplicity, i.e. $\psi = 0$.

Let

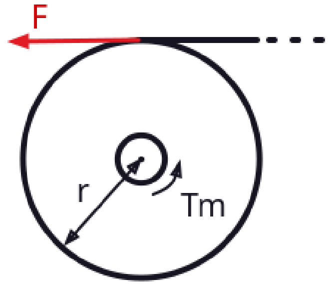
$$R(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

Then

$$\vec{a} = R(\theta) \cdot \begin{bmatrix} 0 \\ a \end{bmatrix}$$



(a) Elbow model



(b) Motor front view

Figure 4.9: Simplified orthosis model

Therefore

$$\vec{T_d} = \vec{a} \times \vec{F}$$
$$\vec{T_d} = ||\vec{a}|| \cdot ||\vec{F}|| \cdot \sin(\alpha)\vec{n}$$
$$\vec{T_d} = ||\vec{a}|| \cdot \left|\left|\frac{\vec{T_m}}{r}\right|\right| \cdot \sin(\alpha)\vec{n}$$
$$||\vec{T_m}|| = \frac{r \cdot ||\vec{T_d}||}{||\vec{a}|| \cdot \sin(\alpha)}$$

Since α is a function of θ, it can be calculated:

$$\alpha = \arcsin(\frac{||\vec{a} \times (\vec{b} - \vec{a})||}{||\vec{a}|| \cdot ||\vec{b} - \vec{a}||})$$

Finally, we obtain

$$||\vec{T_m}|| = \frac{||\vec{b} - \vec{a}|| \cdot r \cdot ||\vec{T_d}||}{||\vec{a} \times (\vec{b} - \vec{a})||}$$

Figure 4.10 illustrates torque predicted by the above model for different anchor points of the orthosis cable.

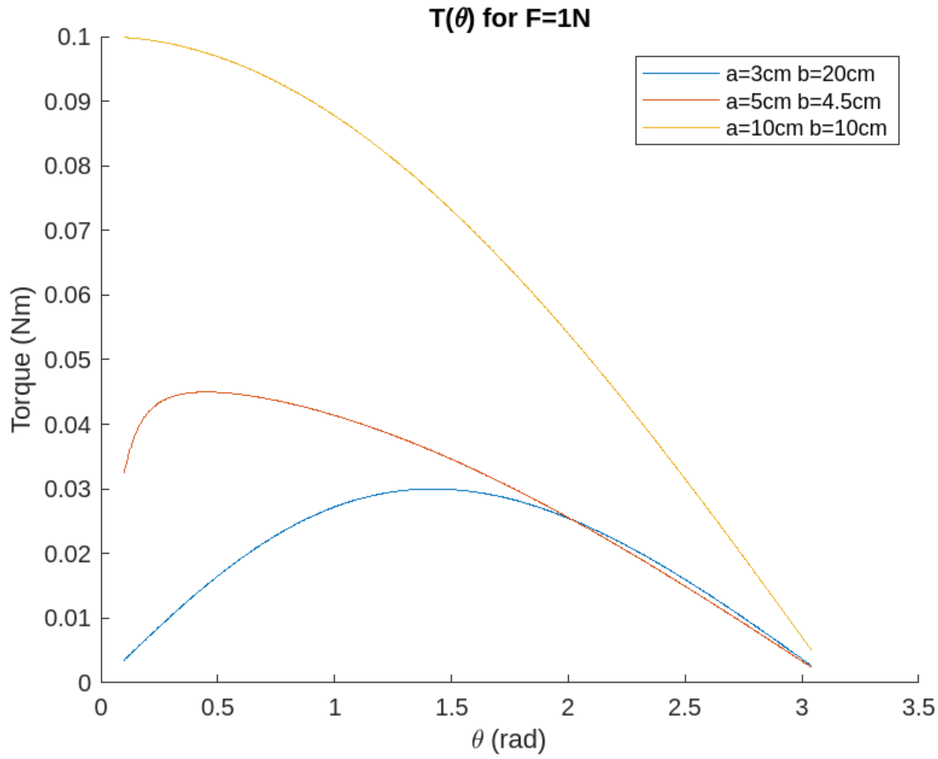


Figure 4.10: Torque generated by 1N of force according to cable anchor points

Motor driver

To make use of the EPOS4 70/15 driver, the Linux EPOS Command Library (ECL) was used. Three operation modes were developed: current, position and velocity. The position and velocity modes serve only for debugging purposes and experimental setup. The driver is implemented in C, using two threads. A *server* thread waits for an incoming connection and, when established, receives commands for the motor through a socket linked to port 8080. Upon reception, a real-time signal containing the command value is sent to the *command* thread. The *command* thread is responsible for handling the real-time signal and using the ECL to control the motor in accordance with the chosen mode.

Originally, the *command* thread was also responsible for saving data relative to the motor's position, velocity and current consumption. However, controlling the motor while simultaneously acquiring motor information through the ECL was not implemented successfully.

Sensor-Motor link

To link the sensor output to the motor driver, a python program was developed. This program first connects to the motor driver's server on port 8080, then initiates either a serial or BLE communication with the ESP32, repeatedly reading the sensor values. Torque estimation is subsequently performed and by using the orthosis inverse model and applying a K gain, the command is finally sent to the motor driver.

Experiment data is saved both as a CSV file and as an image.

Results

5.1 Torque estimation

All four NN models were tested against various datasets. Using the angle data and physiological parameters of the user, actual torque was calculated. Then, model predicted torque was plotted against it.

5.1.1 Single dataset models

As illustrated in Figures 5.1, 5.2 and 5.3 the *closed_model* beats the *open_model* in terms of accuracy but neither achieve satisfactory results for higher loads. In fact, these models were trained on a dataset where a load of 900g was used and they work best on datasets with similar loads. The biggest advantage of the *closed_model* over the *open_model* is its better performance on the dataset containing stops illustrated in Figure 5.2.

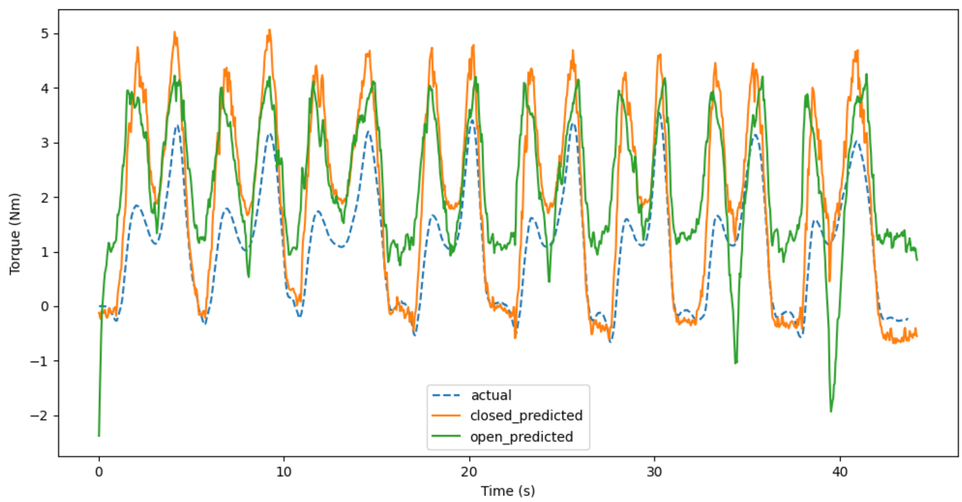


Figure 5.1: No load

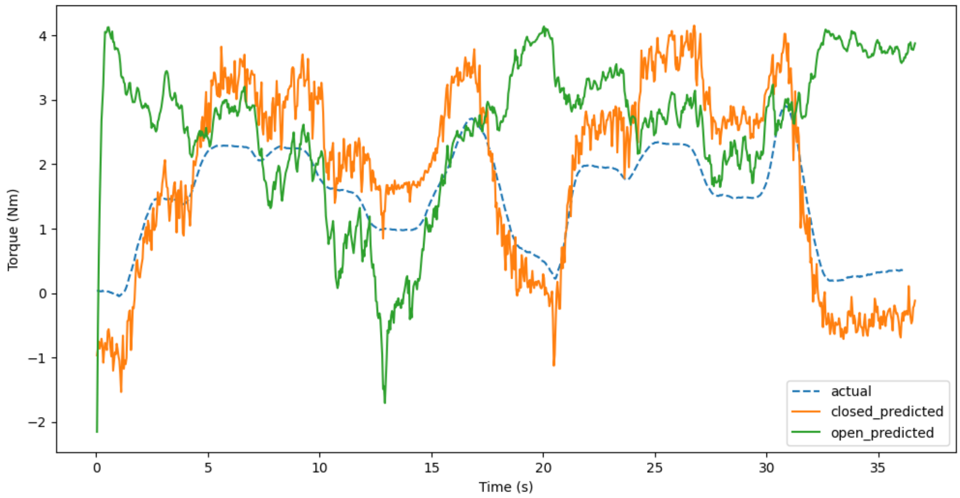


Figure 5.2: No load with stops

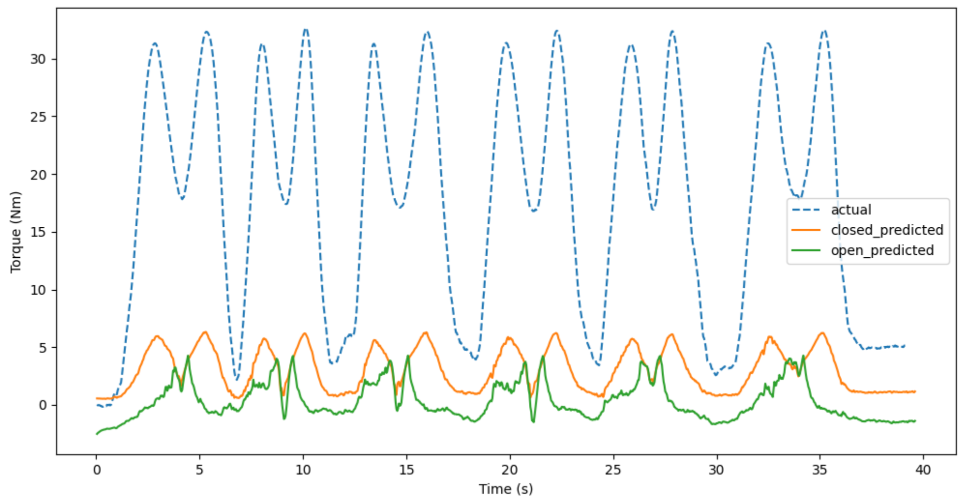


Figure 5.3: 10kg load

5.1.2 Multi-dataset models

Models trained on all initial datasets showed no advantage over the first two models. As illustrated in Figures 5.4, 5.5 and 5.6, *multi_open_model* and *multi_closed_model* performed worse than their single-dataset counterparts.

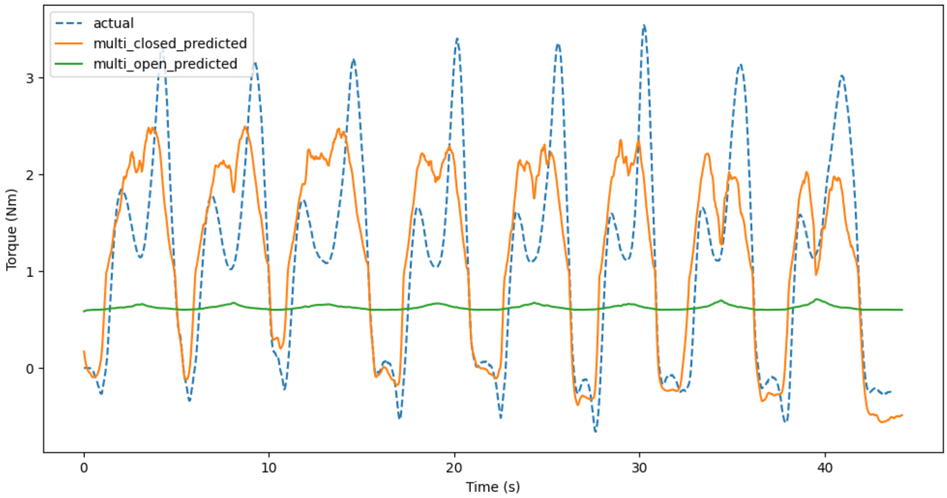


Figure 5.4: No load

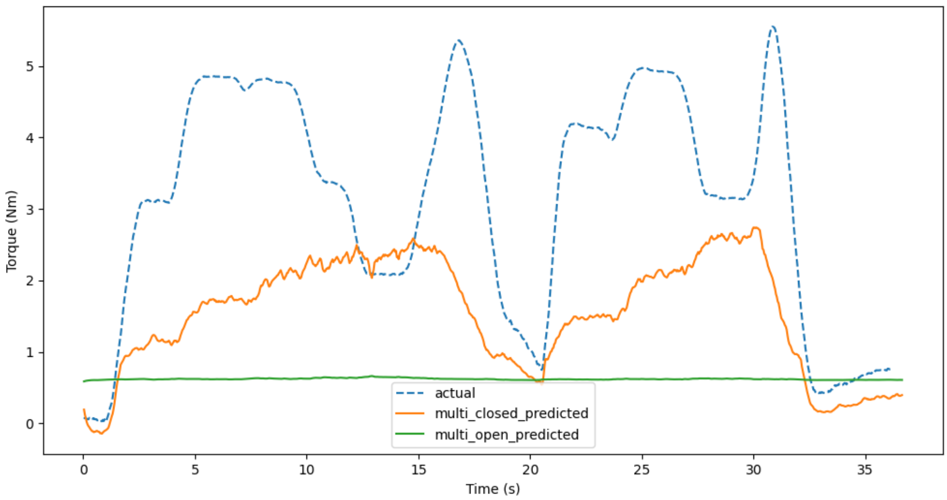


Figure 5.5: No load with stops

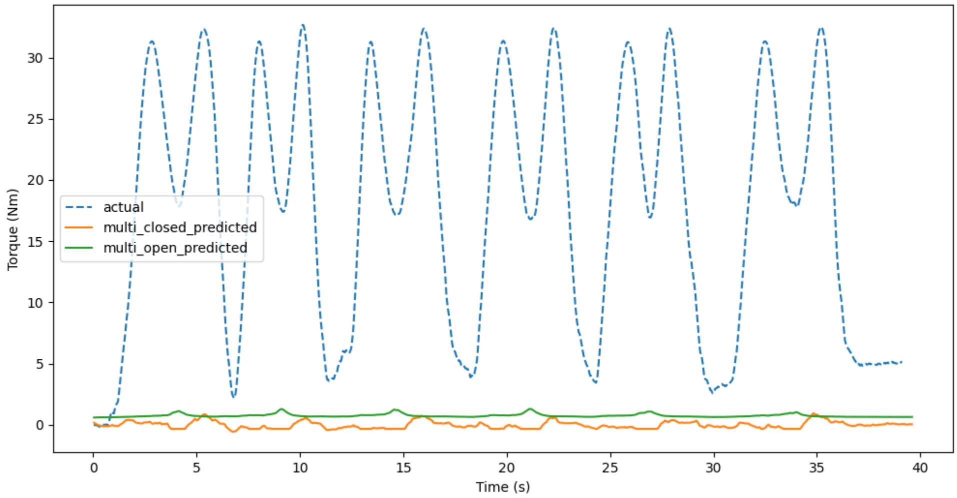


Figure 5.6: 10kg load

5.2 Orthosis control

5.2.1 Experimental setup

The experimental setup is illustrated in Figure 5.7. The motor driver is first connected to the PC via USB, then the motor is securely tightened to the table surface and connected to the driver. The power supply is then also connected to the driver via a twisted pair power cable. Finally, the rope is tightened to prepare the system for testing.

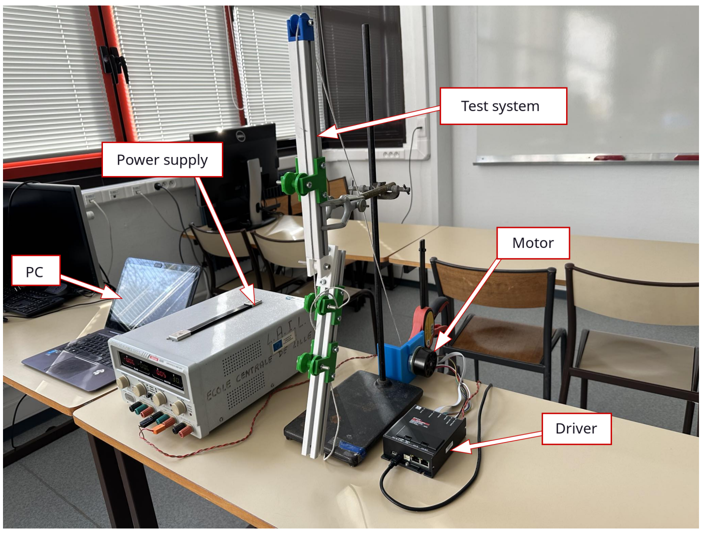
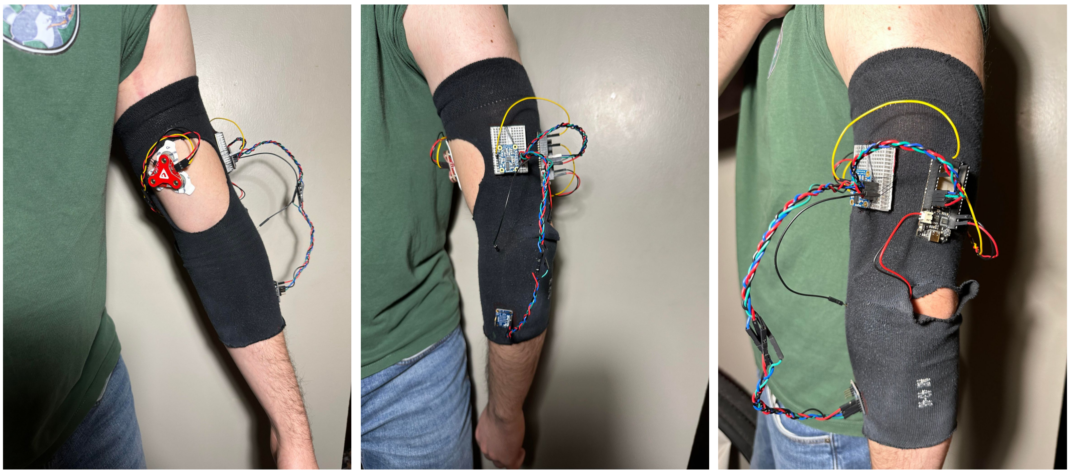


Figure 5.7: Experimental setup

As shown in Figure 5.8, the sensor system was installed on a sock by sewing each component individually. Thus, a soft orthosis system could be simulated for control purposes.



(a) Front (b) Side (c) Back

Figure 5.8: Control system on a sock

5.2.2 Experimental results

The control strategy shown in Figure 4.6 did not produce the desired results based on visual evaluation.

The goal was that the test system shown in Figure 5.7 starts applying an assistive torque as soon as motion intention is detected and stops providing this assistance as soon as that intention disappears. However, utilizing the devized control strategy only resulted in sporadic flexion of the test system as the identified torque values peaked too fast and too high, on top of arriving well after the user had already begun the motion.

A test where the test system was controlled by an even simpler control strategy shown in Figure **??** was successfully conducted. This test consisted in controlling the test system proportionally to the user's sEMG signal. Current command values were saturated for safety. The result was flexion of the test system that could be controlled by the contraction of the user's biceps.

Figure 5.9: Simplified Control Diagram - K is an arbitrary gain

Further work

6.1 Circuit board

6.2 NN

6.3 Model

6.4 Control strategy

6.5 Communication

Conclusion

Bibliography

[1] L. Lu, Q. Wu, X. Chen, Z. Shao, B. Chen, and H. Wu, "Development of a sEMG-based torque estimation control strategy for a soft elbow exoskeleton," vol. 111, pp. 88–98. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889018305128

[2] Q. Wu, B. Chen, and H. Wu, "Neural-network-enhanced torque estimation control of a soft wearable exoskeleton for elbow assistance," vol. 63, p. 102279. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415819301126

[3] Q. Wu and Y. Chen, "Adaptive cooperative control of a soft elbow rehabilitation exoskeleton based on improved joint torque estimation," vol. 184, p. 109748. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327022008160

[4] L. T. dos Santos, M. Kugler, and P. Nohama, "Signals, sensors and methods for controlling active upper limb orthotic devices: a comprehensive review," vol. 39, no. 3, pp. 759–775. [Online]. Available: https://doi.org/10.1007/s42600-023-00292-w

[5] X. Jiang, L.-K. Merhi, Z. G. Xiao, and C. Menon, "Exploration of force myography and surface electromyography in hand gesture classification," vol. 41, pp. 63–73. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350453317300176

[6] H. Begovic, G.-Q. Zhou, T. Li, Y. Wang, and Y.-P. Zheng, "Detection of the electromechanical delay and its components during voluntary isometric contraction of the quadriceps femoris muscle," vol. 5, p. 494. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4274888/

[7] Y. Xu, V. M. McClelland, Z. Cvetkovic, and K. R. Mills, "Delay estimation between EEG and EMG via coherence with time lag," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 734–738, ISSN: 2379-190X. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7471772

[8] R. E. Russo, J. G. Fernández, and R. R. Rivera, "Algorithm of myoelectric signals processing for the control of prosthetic robotic hands," vol. 18, no. 1, p. e04. [Online]. Available: https://journal.info.unlp.edu.ar/JCST/article/view/692