Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Project report in the courses
DVA490 and DVA474

SYSTEM DESIGN OF AN INTENT-DRIVEN ROBOTIC ARM WITH ADAPTIVE GRASPING USING EMG, VISION, AND PRESSURE SENSOR

Majid Azizi
mai20018@student.mdu.se

Jhon Eduar Garcia
jga24001@student.mdu.se

Mahlin Jansson
mjn21025@student.mdu.se

Johan Sandred
jsd21003@student.mdu.se

Elias Wesche
ewe20002@student.mdu.se

Oscar Ågren
oan21004@student.mdu.se

Examiner: Johan Hjort
Mälardalen University, Västerås

Stakeholder and Supervisor: Martin Ekström
Mälardalen University, Västerås

Stakeholder: Waqas Khalid
UC Berkeley, California

2026-02-07

Abstract

The development of bionic and assistive robotic arms requires dependable intention recognition and responsive control in noisy, variable, and safety-critical conditions. Multimodal sensing has emerged as a promising approach to overcome the limitations of individual sensors, but practical integration and validation of such systems remain a challenge. This project investigates the design, implementation, and validation of a multimodal system intended for intention recognition and control of a bionic arm. The work focuses on the mechanical design and construction of the arm, as well as on the integration of Electromyography (EMG), Inertial Measurement Units (IMUs), and vision to support movement-intention detection and object interaction. The main goal was not only to construct a bionic arm that could grip, move, and release a cup, but also to design a customised EMG acquisition system that could provide sufficient signal quality for the recognition of intentions, in combination with other complementary sensors. The system was implemented using custom analog front-end electronics, embedded processing on ESP32-C3 microcontrollers, and a vision pipeline based on depth-enabled object detection. The EMG system was validated across multiple trials. Results show clear separation between resting and gripping conditions, low baseline noise, minimal drift, physiologically plausible frequency content, and consistent signal-to-noise ratios across trials. IMU- and vision-based subsystems were evaluated as feasibility demonstrations, highlighting both their potential and their limitations under occlusions, drift, and environmental variability. Overall, the project demonstrates the feasibility of a multimodal, intent-driven control architecture and highlights practical considerations for integrating diverse sensors in assistive robotic systems.

Contents

List of Figures

List of Tables

List of Abbreviation

ML ..... Machine Learning
IMU ..... Inertial Measurement Unit
EMG ..... Electromyography
RMS ..... Root Mean Square
SNR ..... Signal-to-Noise Ratio
ECG ..... Electrocardiography
CMRR ..... Common-mode Rejection Ratio
RLD ..... Right Leg Drive
MCU ..... Microcontroller Unit
I/O ..... Input/Output
ADC ..... Analog-to-Digital Converter
PSU ..... Power Supply Unit
CAN ..... Controller Area Network
MAV ..... Mean Absolut Value
PNP ..... Perspective-n-point
FSR ..... Force Sensing Resistor
BLE ..... Bluetooth Low Energy
DOF ..... degree of freedom
CAD ..... Computer Aided Design
PWM ..... Pulse Width Modulation
ATT ..... Attribute Protocol
PDU ..... Protocol Data Unit
L2CAP ..... Logical Link Control and Adaptation protocol
MTU ..... Maximum Transmission Unit
LE ..... Low Energy
PHY ..... Physical Layer
FSM ..... Finite State Machine
LSTM ..... Long Short-Term Memory
EKF ..... Extended Kalman Filter
PCB ..... Printed Circuit Board
YOLO ..... You Only Look Once
ACK ..... Acknowledgment
CNN ..... Convolutional Neural Network
LL ..... Link Layer
PDU ..... Protocol Data Unit
RNN ..... Recurrent Neural Network
RS-485 ..... Recommended Standard 485
I^2C ..... Inter-Integrated Circuit
SPI ..... Serial Peripheral Interface
EMI ..... Electromagnetic Interference
PDU ..... Power Distribution Unit
IFS ..... Inter Frame Spacing

LIMB ..... Leveraging Intelligent Mechatronics for Bionics
RRD ..... Robotic Rehabilitation Device
ADL ..... activities of daily living
CNS ..... central nervous system
PNS ..... peripheral nervous system
PD ..... Proportional–Derivative
FFT ..... fast Fourier transform
EKF ..... Extended Kalman Filter
PLA ..... polylactic acid
TPU ..... thermoplastic polyurethane

1 Introduction

Stroke is a leading cause of adult disability worldwide, with one in four people having a stroke in their lifetime [1]. Globally, there are 12 million new stroke cases per year, resulting in 94 million people living with the effects of stroke, with profound and lasting consequences for individuals, families, and healthcare systems worldwide. The aftermath of a stroke often includes motor deficits that have a significant impact on quality of life. This, together with the growing global health burden, has increased the need for more effective rehabilitation processes to help patients regain motor function. Early evidence suggests that rehabilitation with robotics and bionics can help with high-intensity, task-specific exercises that traditional therapy alone cannot achieve, resulting in improved therapy outcomes [2]. In recent years, there has been significant progress in the advancement of prosthetic limbs [3], [4]. These advancements include more advanced robotics, improved sensors, and Machine Learning (ML) algorithms that provide better, more accurate, and human-like control. Such innovations open new opportunities for the use of robotic arms in stroke patient rehabilitation.

Traditional post-stroke rehabilitation is based on physiotherapy and occupational therapy techniques as the basis for recovery. While these therapies are crucial, they are only partially effective. They require extensive time and one-on-one sessions with a therapist [5]. As a result, many patients receive therapy that is constrained in duration or intensity. The conventional therapies often lead to slow and incomplete motor recovery, especially for people with severe stroke cases [6].

To address these limitations, modern rehabilitation has turned towards robotics, leveraging exoskeletons to automate task-specific training. A prominent example is the Upper Limb Exoskeleton Rehabilitation Robot (ULERR) [7], which optimises recovery by using a camera to capture the user's natural movement and create a personalised kinematic profile. From this profile, the system derives specific motor control commands that allow the robot to assist the patient in completing essential movements, such as shoulder abduction, adduction, and elbow flexion, within anatomically correct ranges. This guided assistance is crucial for stroke patients to regain functional autonomy.

Similarly, the Malaysia Exoskeletal Robot Assisted Therapy (MyERAT) [8] explores enhancing neuroplasticity through mirror therapy. This system uses a sensor-equipped glove on the healthy hand to capture finger flexion/extension, then transmits these commands via Bluetooth to a motorised exoskeleton on the impaired hand. While this approach promotes active patient engagement, it faces challenges such as a latency of 300–$400\,\mathrm{ms}$, which can disrupt the synchrony required for effective motor relearning.

Other modern alternatives are pivoting towards Machine Learning to achieve more intuitive synergy. Specifically, architectures based on Long Short-Term Memory (LSTM) networks have emerged as a superior solution for rehabilitation due to their ability to process the temporal dependencies of Electromyography (EMG) signals. For instance, recent studies on home-based rehabilitation systems [9] have demonstrated that hybrid Convolutional Neural Network (CNN)–LSTM models can achieve remarkable results in movement estimation. Unlike traditional methods that struggle with user variability, these models consistently achieve over 90% accuracy in gesture recognition, which demonstrates excellent results in predicting the user's intended movement for the robot's actual execution, regardless of the subject.

The predictive capacity of LSTM networks is fundamental to modern rehabilitation, enabling the system to adapt to diverse muscular patterns with minimal recalibration. By anticipating the intended trajectory, the robot transcends being a reactive agent following a fixed profile to become a proactive assistant. This shift toward intent prediction enables an almost instantaneous response, aligning the machine's execution with the patient's neurological command and effectively enhancing neuroplasticity [9].

Despite these advancements, a critical technical bottleneck persists: most current implementations rely on deterministic control or passive guidance, where the robot executes trajectories regardless of the user's real-time state [7]. This lack of synchrony between neurological command and machine response often restricts the patient to a passive role, limiting the potential for neuroplasticity [8]. Furthermore, existing platforms frequently operate in "environmental isolation", lacking the sensory awareness needed to adapt to real-world objects [9]. Consequently, there is a clear necessity for a multimodal architecture that can fuse user intention with environmental context to bridge the gap between simple mechanical assistance and intuitive, proactive rehabilitation.

In this context, the Leveraging Intelligent Mechatronics for Bionics (LIMB) project is established to evaluate the feasibility of a multimodal sensing architecture in bridging these clinical and technical gaps. This report assesses the development of a smart robotic platform designed to facilitate motor relearning in stroke patients' limbs through seamless interaction between user intent and machine execution. To this end, the project investigates first: how to develop a modular pipeline integrating LSTM networks for

gesture classification, Inertial Measurement Unit (IMU)-based directional tracking, and You Only Look Once (YOLO)-driven object detection, to support intuitive grip-intention; and second, how the hardware-software integration allows a user to manipulate everyday objects and perform daily tasks, such as moving a coffee mug.

By addressing these topics, the project aims to evaluate whether a context-aware, intent-driven system can significantly reduce cognitive load and enhance therapeutic feedback, offering a promising pathway toward more effective and accessible neuro-rehabilitation.

The remainder of this report is organised as follows: Section 3 provides the theoretical background, Section 4 reviews related work in bionic rehabilitation. Section 5 describes the research methodology and its appropriateness for this study. Section 6 discusses the ethical and societal considerations of the project. Section 7 details the design and hardware-software integration of the LIMB platform. Finally, the results are presented in Section 8, followed by a discussion and concluding remarks in Sections 9 and 10, respectively.

2 Problem Formulation

Current advancements in rehabilitation robotics have introduced a critical dichotomy between Robotic Rehabilitation Devices (RRDs), designed to guide the limb through repetitive exercises and assistive devices, which augment the user's residual motor function [10]. Despite technological advances in this field, a stark gap persists: high-end bionic solutions are often prohibitively expensive and difficult to control, while simpler systems are constrained by limited degrees of freedom (DOFs), limited environmental awareness, and a narrow repertoire of exercises. [10], [11]. This complexity, combined with a lack of intuitive interfaces, leads to high abandonment rates, as users struggle to integrate these devices into their activities of daily living (ADL) [11]. Furthermore, many existing Robotic Rehabilitation Devices (RRDs) inadvertently promote "passive training", where the patient becomes reliant on the machine's force rather than actively engaging their neural pathways, thereby hindering the neuroplasticity required for true recovery [10], [12]. To overcome these barriers and provide an accessible, high-performance bionic solution, the following pivotal technical challenges must be addressed:

- Cognitive-Intention bottleneck: There is a notable deficiency in rehabilitative systems that prioritise real-time response derived from EMG stimuli without imposing significant cognitive demands [12]. Traditional interfaces often require patients to consciously "drive" individual actuators, such as performing artificial co-contractions to switch modes, a process that creates high cognitive overload and mental fatigue [10], [11]. For a stroke patient, the mental effort required to trigger a simple grasp is so taxing that it distracts the brain from the quality of the movement itself, preventing the focus on functional recovery. Current solutions fail to provide an intuitive translation of raw physiological signals into fluid motions, which is essential for facilitating the reorganisation of impaired neural circuits [12].

- Environmental awareness and contextual information: Most current platforms operate in environmental isolation, lacking the real-time perception needed to identify and adapt to the objects they intend to manipulate [13]. This lack of context-awareness places the entire burden of trajectory correction on the user, who must visually and mentally compensate for every mechanical misalignment. Without integrating "perceptual technologies", such as computer vision for object recognition or IMUs, the interaction remains mechanically rigid. In the absence of a shared-control paradigm, in which the robot perceives the target's position and orientation, the system remains unable to proactively assist in the reaching and grasping phases, leading to frequent task failures and user frustration [10], [13].

- Multi-sensor fusion and low-latency integration: A major engineering hurdle lies in the effective fusion of heterogeneous data, including EMG patterns, vision sensors, and inertial data, to ensure consistency and accuracy in real-time [13]. To effectively promote neuroplasticity, the system must establish a "closed-loop" architecture where motor commands are executed with minimal delay, mirroring the natural timing of human biological response [12]. The challenge resides in integrating high-level inference (e.g., YOLO and LSTM) with a robust communication backbone (such as a Controller Area Network (CAN) bus) to move beyond simple teleoperation toward an autonomous, intent-driven assistance system [12], [13].

2.1 Assumptions and Limitations

To ensure the feasibility of the prototype within the project's scope, the following assumptions and limitations are established:

- **Target physiology and level of impairment:** The system is specifically designed for users with a transradial amputation (below the elbow) or stroke patients with equivalent motor control. It is assumed that the user retains sufficient residual muscular activity in the forearm to generate discernible EMG signals that can be classified into distinct intentional patterns.
- **Operational environment and setup:** The system is designed for controlled indoor settings with stable lighting to ensure the reliability of the YOLO-based object detection. For this prototype, the robotic base is fixed to a table, and the camera remains in a static, calibrated position. The operational workspace is predefined, limiting the camera's field of view and the arm's range of motion to a specific area. Furthermore, the control system is specifically programmed for a set of defined actions: identifying, grasping/releasing, lifting/lowering, and lateral displacement of the target object.
- **Payload and object specification:** The mechanical design is optimised for daily tasks. The prototype is intended to be used exclusively with standard coffee mugs or similar cylindrical objects that do not exceed the standard weight and dimensions of such items.
- **On-board computational architecture:** It is assumed that the local computational power provided by the NVIDIA Jetson AGX Orin is sufficient to handle high-level multimodal inference (YOLO and LSTM) in real-time. The system's feasibility relies on the assumption that executing these models at the edge, rather than on external servers, provides the necessary low-latency response required for synchronised rehabilitation. Furthermore, it is assumed that the integrated CAN bus architecture can maintain stable communication between the central node and distributed microcontrollers without data congestion, even when processing high-frequency input from EMG, IMU, and haptic sensors.

2.2 Purpose and Research Questions

The primary purpose of the LIMB project is to design, implement, and evaluate a multimodal robotic platform that bridges the gap between physiological intent and mechanical execution. By synchronising EMG signal classification with environmental context, such as computer vision, IMU directional tracking, and haptic feedback from pressure and piezoelectric sensors, the project aims to reduce the user's cognitive load during complex rehabilitation tasks.

This multimodal approach provides a semi-autonomous assistance system designed to facilitate neuroplasticity by ensuring that the robotic movement accurately, fluidly, and in real-time reflects the user's motor intentions. Through this "shared-control" paradigm, the system seeks to transform the patient from a passive recipient of force into an active participant in their recovery. Finally, this integration seeks to enable the intuitive execution of functional ADL, such as identifying, reaching, and manipulating a coffee mug through a seamless synergy between human intent and robotic assistance. Once this purpose is met, the project seeks to answer the following research questions:

- RQ1: How can a modular multimodal pipeline (LSTM + IMU direction + YOLO object detection) be designed to support grip-intention recognition for object interaction, and what are the observed performance characteristics of each module?
- RQ2: How can the integration of sensing and control components in a mechatronic prototype be optimised to ensure reliable and safe manipulation of objects, such as a coffee mug?

2.3 Objectives

To fulfil the project's purpose and address the research questions, the following technical objectives have been defined

1. **Multimodal Intent Recognition:** Collect an EMG dataset and train an LSTM network to classify grip and release gestures with over 90% accuracy, ensuring the system adapts to the temporal dependencies of muscle signals.

2. **Context-Aware Responsiveness:** Implement a synchronised software-hardware pipeline that integrates physiological intent (EMG) with environmental context (YOLO-based vision and IMU tracking) to provide continuous, real-time feedback, maintaining an end-to-end system latency of less than $300\,\mathrm{ms}$ to support neuroplasticity.

3. **Robust System Integration:** Establish a communication backbone utilising the CAN protocol and Bluetooth Low Energy (BLE) to synchronise the NVIDIA Jetson AGX Orin with distributed ESP32-C3 microcontrollers, ensuring low latency for reliable sensor fusion (including pressure and piezoelectric data).

4. **Functional Validation:** Demonstrate the successful execution of a complete "intent-to-grasp" cycle in a shared-control scenario, where the robotic platform autonomously identifies, reaches, and secures a coffee mug based on real-time user stimuli and haptic feedback.

3 Background

This section establish technical and theoretical context. Its coverage focuses on well-established and relevant theories and concepts.

3.1 EMG

EMG records muscle activity through electrodes that are placed on the skin [14]. The measured signal reflects the electrical activity generated by muscle fibers during contraction. The amplitude increases as more motor units are recruited and the contraction force rises. It is established that the EMG signal is stochastic in nature, meaning its waveform varies over time even for repeated executions of the same movement. It has an amplitude that varies between 0–$10\,\mathrm{mV}$ peak-to-peak or 0–$1.2\,\mathrm{mV}$ Root Mean Square (RMS) on the skin, with most of the energy in the range 10–$150\,\mathrm{Hz}$.

Compared with Electrocardiography (ECG), which measures the electrical activity of the heart, or Electroencephalography, which measures brain activity, EMG signals recorded from skeletal muscles are weaker and more variable. They are also more sensitive to electrode placement, signal conditioning, and external noise [15]. Proper acquisition, where the amplifiers should have high input impedance, low noise, and high Common-mode Rejection Ratio (CMRR), is therefore essential for reliable interpretation. Another major challenge is that muscle fatigue alters the amplitude and frequency characteristics of the EMG signal, which will reduce the control reliability [16].

The noise in the EMG signal can be divided into four main categories [14]:

- Ambient noise: Background noise from the environment that originates from sources of electromagnetic radiation. The dominant concern regarding ambient noise is the $50\,\mathrm{Hz}$ from power sources, and has an amplitude one to three orders larger than the amplitude of the EMG signal.
- Motion artifacts: It originates from movements of the sensor or subject and spans mainly over 0–$20\,\mathrm{Hz}$. The sensor picks up mechanical or electrical noise and is common in EMG applications.
- Inherent instability of the signal: The EMG signal is mostly unstable between 0–$20\,\mathrm{Hz}$ due to the quasirandom nature of the signal.
- Inherent noise in electronic components: All electrical components generate noise. This noise can not be eliminated, but can be reduced by intelligent circuit design and construction techniques.

EMG data can be used to control bionic or prosthetic devices [17]. Before analysis, EMG data are commonly preprocessed to enhance signal quality. Typical preprocessing includes band-pass filtering to remove noise and notch-filtering to suppress electrical interference. The resulting signal is often rectified and smoothed to derive the muscle activation envelope.

In many systems, features are extracted from the processed EMG signal to quantify muscle activity [17]. It is usually separated into time- and frequency-domain features [18]. Some common time-domain features are the RMS, Mean Absolut Value (MAV), and Zero Crossing Rate [17]. Some common frequency-domain features include Median Frequency and Mean Power Frequency that provide information about muscle fatigue and contraction dynamics [19].

Processed EMG features are commonly used as input to classifiers that infer muscle activation states or user intention, such as rest, movement or different grips [17]. These outputs can then be fused with complementary sensors data, such as IMU and vision input, to achieve robust multi-modal control [20].

When EMG processing is implemented in embedded systems, several real-time constraints must be considered. Sampling rates are chosen according to Nyquist's theorem to capture the full bandwidth of the signal, while short, overlapping windows are typically used to process data continuously and keep the response delay low [21], [22]. Digital filters are still applied at this stage, but their design requires caution since they can introduce phase shifts or aliasing that distort the signal [21]. Similarly, the feature extraction stage should remain simple enough to avoid adding noticeable latency to the system's response [18]. Efficient buffering and memory handling are also essential to ensure that samples are stored and processed without loss, while keeping the system within its limited computational and memory resources [20]. Overall, these considerations are key to maintaining stable, low-latency EMG control in real-time embedded applications.

3.2 Hardware

A bionic arm is an advanced electromechanical system that is designed to mimic the movement and functionality of a human arm [23]. The system consists of mechanical components (bearings, gears, etc.), actuators, sensors, and control electronics to interpret signals and generate precise and controlled movement. The actuators are the components that convert the electrical signals into movement. In this project, the relevant actuators are servo- and stepper motors. There are different advantages and disadvantages of using the two: for servo motors, one of the relevant advantages to this project are that they are small relative to their torque capacity, and a relevant disadvantage is that some servo motors are limited to a certain degree of motion, for example, $180°$. For stepper motors, an advantage is that they are suitable for applications with high holding torque; a disadvantage is their low efficiency due to their constant current draw.

Unlike the servo motor, the stepper needs another component to drive it. These components are called stepper drivers, and they convert the data signals received from a Microcontroller Unit (MCU) into current pulses that drive the phases of the stepper in sequence, providing controlled rotational steps[1]. Besides driving the stepper, a stepper driver can also provide microstepping, protection, and current control.

The MCU is a tiny computer integrated on a single chip[2]. It is designed to manage specific tasks within an embedded system without the need for a complex operating system. They incorporate their own processing, memory, Input/Output (I/O) peripherals such as timers, Analog-to-Digital Converter (ADC), depending on the MCU. The MCU will provide the stepper driver with data signals so that it will know what to do with the stepper.

Since the bionic arm is a type of robot, and for robots to interact with their environment, they need sensors [24]. Sensors in a robot can vary depending on the intended function. Examples include: potentiometer, IMU, temperature sensor, and pressure sensor. All of these sensors serve their different purposes for the robot application.

With all these components, the system needs power and a way to communicate. The components draw current, which means that it is necessary to have one or multiple Power Supply Unit (PSU). The PSU is responsible for providing the right voltage required by each component and be able to deliver enough current for the components to operate [25].

In the bionic arm, there is a main computer that handles the feedback and acts upon it. Then communication between the MCU and the computer is needed; this can be done wirelessly or non-wirelessly. Each of these MCU and the computer can be denoted as "nodes". CAN is a popular way of communicating between nodes in a robotic system [26]. CAN is a serial network information technology that communicates between the nodes within the network [27]. CAN transceivers are a piece of hardware that can be either built into the MCU and computer, or it can be added externally, as is the case in this project. The transceivers make sure the messages get sent and received at specific nodes [28]. Being able to make gestures and motions like a human hand is one of the most critical aspects of an anthropomorphic hand. To achieve this, each finger requires at least two joints per finger. When a person manipulates an object, they rely on their sensory inputs to interpret how to grasp it. By sensing the object and feeling it slipping, the hand will automatically adjust the grip to hold the object firmly. Likewise, a robotic hand will need a way to sense the object. Tactile sensors in a robotic hand can provide the essential information about contact with an object, enabling quick and precise slip detection. There exist different types of tactile sensors. One example is a Force Sensing Resistor (FSR), which is a robust Polymer thick film device. It shows a decrease in resistance as the force applied to its surface increases. This sensitivity to force is optimised for use in human touch control applications in electronic devices, such as automotive electronics, medical systems, and robotics.

3.3 3D modelling and 3D printing

Traditionally, robotic components were manufactured using time-consuming and expensive methods such as milling. This does not need to be the case anymore. With recent advancements in Computer Aided Design (CAD) and additive manufacturing technologies, the process of designing and developing mechanical systems, such as a robotic hand, has become more efficient and accessible. With modern CAD software, it is possible to create functional and detailed 3D models with great precision in dimension and mechanical constraints. The introduction of 3D printing enables a seamless transition from digital design to a physical prototype. Through the use of additive manufacturing, 3D models designed in CAD can be produced at a low cost and within a short period of time.

[1] https://www.electrokit.com/upload/product/41016/41016255/drv8825.pdf
[2] https://www.ibm.com/think/topics/microcontroller

3.4 Software

This section describes the software architecture and processing methods used to handle sensor data, perception, control logic, and communication within the system.

3.4.1 IMU data processing

IMUs are sensor modules that measure linear acceleration, angular velocity, and sometimes magnetic field strength. Additionally, temporal data and sensor temperature is usually provided but specific to the IMU. The IMU measurements enable estimation of an object's orientation, motion, and position over time. They are widely used in robotics, wearable systems, and motion tracking applications [29], [30], [31]. They are compact and able to operate independently of external references. A typical nine degrees of freedom IMU sensor consists of a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer.

While IMUs provides valuable motion data, each sensor component introduces characteristic errors: accelerometers are affected by vibration and dynamic acceleration, gyroscopes suffer from bias drift due to integration over time, and magnetometers are sensitive to electromagnetic disturbances. To obtain reliable motion estimates, these raw signals must be calibrated and filtered to reduce noise and bias accumulation. Additionally, the data from each sensor component can be combined or fused to account for the errors and provide a unified estimate of the orientation of a system. For example, complementary or Kalman-based sensor fusion algorithms are commonly used to combine gyroscope data with accelerometer and magnetometer measurements [30], [31]. This allows short-term stability from the gyroscope to be corrected by long-term reference information from the other sensors.

Processed IMU data can be used to derive velocity and position estimates by integration. Also, it can be used to characterise motion patterns. In human-machine interaction systems, IMUs are commonly used to monitor limb movement, detect gestures, or complement other sensor modalities [32]. The integration of IMU data enhances temporal continuity and robustness when an external sensor can not be used, for example, when a camera experiences occlusion or loss of tracking.

3.4.2 Vision and pose estimation

Vision-based systems play a central role in robotics and human-machine interaction, providing spatial and semantic information about the environment. By analysing images or video frames, such systems can detect objects, estimate their positions and orientation, or even track their motion over time [33]. Compared to inertial sensors, vision offers absolute reference information relative to the environment. This makes it a powerful complement in perception and control tasks [34].

There are many different models that vision-based estimation calculations are based upon, where the pin-hole model is one of the most common models [35]; it describes the projection of three-dimensional world points onto a two-dimensional image plane. Due to the absoluteness of the gathered vision-based information, calibration is needed. For that, intrinsic (focal length, principal point, and lens distortion coefficients) and sometimes extrinsic camera parameters are obtained to adapt the vision system to the contextual environment. Calibration is essential for accurate spatial measurements and is typically performed using known reference patterns such as a checkerboard or fiducial tags.

A common approach to obtain precise spatial information involves the use of fiducial markers such as ArUco or AprilTags [36], [37]. These tags encode unique IDs and geometric patterns in a square shape that can be robustly detected under varying lighting conditions. By identifying the corner of a marker and solving the Perspective-n-point (PNP) problem, the system can compute the marker's rotation and translation relative to the camera [38]. This is called its [39]. This enables reliable estimation of object positions or coordinate frames in the observed scene [36].

Another metric that is not trivially obtainable in using standard cameras is the depth at which an object is positioned relative to the camera [40]. Depth information can be received using stereo vision, structured light, or via depth-sensing cameras. Stereo cameras compute depth by triangulating corresponding points between two image sensors with known geometry [41]. A depth map can be produced that allows for estimation of object distances and thus facilitates three-dimensional scene understanding [40]. This can also be done with a depth-sensing camera. Obtaining three-dimensional information from such sensors is crucial for applications such as navigation, manipulation, and environment mapping [42].

Vision-based pose estimation is used extensively in robotics, augmented reality, and human-robot collaboration [39]. In such applications, visual data may be fused with inertial or other sensory inputs to enhance the robustness and temporal consistency [43].

3.4.3 Sensor fusion

Sensor fusion refers to the process of combining information from multiple sensors to obtain a more accurate, robust, and complete representation of a system's state that is more comprehensive than a single-sensor system could provide [44]. In robotics and human-machine interfaces, fusion is often used to integrate complementary sensing modalities, for example, kinematic, spatial, and neural information [45]. It is also possible to fuse sensor data from multiple instances of the same type of sensor [46]. Both types of fusion have multiple benefits with the cost of complexity.

Individual sensors are limited by noise, drift, or occlusion [47]. For instance, IMUs suffer from integration drift over time while vision-based systems may lose object tracking under poor lighting conditions. By combining data from multiple sensors, these weaknesses can be compensated. Fusion improves reliability, temporal stability, and precision in estimations [34].

Sensor fusion can be performed at different levels of data abstraction [48]:

- Low-level (data level): raw sensor measurements are combined directly, often requiring synchronisation and calibration.
- Mid-level (feature level): features extracted from each sensor are merged to create a richer representation.
- High-level (decision level): independent estimates or classifications are combined using voting or probabilistic reasoning.

The choice of level depends on the sensors involved and the target application [49]. There are many theoretical frameworks to explain the abstraction levels of sensor fusion, the framework explained above captures the essentials [46].

Several mathematical and computational methods are commonly applied in sensor fusion [47]. Classical approaches include complementary filters and Kalman filters. The complementary filter combines high- and low-frequency components from different sensors in a complementary manner [50]. For a linear system, the Kalman filter provides optimal state estimation under Gaussian noise assumptions [51]. More advanced variants include, among others, the Extended Kalman Filter (EKF) or the Unscented Kalman Filter, which are used to handle nonlinear dynamics [52]. Recent research explores learning-based approaches such as neural networks that implicitly learning fusion mappings from multimodal data [53].

3.4.4 Bluetooth Low Energy

BLE is a communication protocol intended for low-power wireless communication between electronic devices. There are two types of devices specified in BLE: Centrals and Peripherals. The Central establishes a connection to a Peripheral in what is called a piconet. The two devices in the piconet can then communicate with each other through the use of characteristics, which identify some type of data that the Peripheral provides. These characteristics are further grouped under what are called services, which are collections of characteristics that have some aspect in common. At the start of every connection event, which occurs periodically at a rate called the connection interval, the Central can send read- and write requests for a characteristic to the Peripheral, to which the Peripheral responds with either sending the value of the characteristic back to the Central or updating the value. The Central can also subscribe to a characteristic, which allows the Peripheral to send the value of the characteristic to the Central without a prior read request. Data sent as part of a subscription that expects an Acknowledgment (ACK) is called an indication. If no ACK is expected, it is instead called a notification.

3.4.5 State machine logic

A state machine, also known as a Finite State Machine (FSM), is a fundamental concept in control systems, robotics, and embedded software [54]. It provides a structured framework for representing system behavior as a sequence of discrete states and transitions triggered by internal or external events. It simplifies complex control logic by dividing it into smaller, well-defined operational modes.

An FSM consists of a finite number of states, transitions between states, and actions associated with either states or transitions. Each state represents a distinct mode of operation. Each transition defines the conditions under which the system changes from one state to another. Mathematically, a state machine can be described as a tuple (S, I, O, f, g) where S denotes the set of states, I the set of inputs, O the set of outputs, f the state transition function, and g the output function.

State machines are widely applied to define behaviour in robotic systems, where each state may represent a task [55]. Transitions are triggered by sensor inputs or decision logic, allowing the robot to respond predictably to environmental changes. FSMs are also often integrated with other decision-making frameworks, such as rule-based systems or learning algorithms. For instance, a reinforcement learning agent may determine the optimal transition between states, while the FSM enforces structured execution [56].

3.4.6 Bio-inspired control and haptic stability

Human motor control is organised into a hierarchical architecture where the central nervous system (CNS) orchestrates high-level intentions, while the peripheral nervous system (PNS) manages rapid, autonomous responses known as reflex arcs. In biological systems, these reflexes are triggered by afferent sensory signals, such as pressure or stretch, which are processed locally to produce an immediate efferent motor response, bypassing the latency of cortical processing [57]. This mechanism is fundamental for maintaining grasp stability during unexpected object slippage, where the reaction speed determines the success of functional manipulation.

However, translating this biological efficiency into myoelectric prostheses presents a significant challenge: EMG pattern recognition, although intuitive, introduces intrinsic computational latencies that can compromise real-time stability [57], [58]. Recent research suggests that it is imperative to implement closed-loop controllers based on tactile feedback to emulate the human reflex pathway, allowing for rapid and precise adjustments of the grasping force. This "shared-control" environment enables the device to consistently grasp delicate objects without causing damage or dropping them, effectively addressing the "slip and crush" dilemma through autonomous, neuro-mimetic algorithms [58].

The LIMB project adopts this decentralised paradigm by distributing computational tasks between an NVIDIA Jetson AGX Orin, acting as the CNS for intent decoding, and a dual-node ESP32-C3 architecture functioning as the PNS. This local subsystem implements a Proportional–Derivative (PD) control strategy that monitors the mean and variance of haptic data in $500\,\mathrm{ms}$ windows. By consolidating multi-finger tactile input into a unified mean pressure value, the controller emulates biological postural synergies, allowing a single DOF adjustment that simplifies the mechanical response while maintaining firm contact. Executing these "robotic reflexes" at the edge ensures that the adaptive hand can autonomously correct its grip on an object, such as a coffee mug, in response to incipient slip detection, providing a seamless synergy between the user's high-level intent and the machine's reactive stability.

4 Related Work

This section showcases previous work that is relevant to this project. The primary interests are regarding the intelligent hand, recording and classifying EMG data, as well as using a vision system in combination with robotic arms.

4.1 Intelligent Hand

Tian et al. [59] developed a 3D-printed, cable-driven, single-model humanoid robotic hand, designed for lightweight humanoid platforms. Their work aimed to strike a balance between mechanical realism, ease of fabrication, and cost efficiency. Unlike many traditional robotic hands that rely on complex assemblies or diverse actuators, their design integrates all 15 joints into a single printed structure. This eliminates post-printing assembly and significantly reduces production costs. Similar design philosophies have been explored in earlier works, where mechanical intelligence is leveraged to reduce control complexity while maintaining functional capabilities [60], [61].

The hand developed by Tian et al. [59] is actuated by six miniature servomotors using a tendon-driven mechanism that mimics how structures in humans combine muscles and tendons, which enables natural grasping motions with six degrees of freedom. Tendon-driven and cable-based actuation has been widely adopted in anthropomorphic and wearable robotic hands due to its lightweight nature and favourable force transmission characteristics [62], [63]. Comparable principles are also employed in synergy-based designs, such as the Pise/IIT SoftHand, where adaptive grasping is achieved through a reduced number of actuators while preserving human-like dexterity[64]. In contrast, highly dexterous systems such as the DLR Hand Arm System demonstrate the upper bound of mechanical complexity and sensing integration, which, however, comes with the cost of increased weight and system complexity [65].

Tian et al. [59] further evaluated grasping performance both with and without an artificial silicone skin, demonstrating the ability to perform a wide range of power and precision grasps consistent with Cutkosky's grasp taxonomy [66]. Their results show that human-like dexterity can be achieved without sacrificing affordability or manufacturability [59]. This aligns with broader research trends emphasising low-cost fabrication, underactuation, and tendon-driven mechanisms as viable alternatives to fully actuated anthropomorphic hands [61], [60].

Building on this mechanical foundation, Guo et al. [67] conducted a systematic review of recent advancements in bionic artificial hands with particular focus on control strategies, sensory feedback integration, and mechanical optimisation. Their review highlights a growing shift toward bio-electrical signal-based control, especially surface EMG, which has long been established as a robust input modality for multifunctional hand control [68], [20]. The adoption of machine learning techniques such as support vector machines and CNNs has further improved EMG signal classification accuracy, with reported performance often exceeding 90% [69], [70], [20]. Public datasets such as NinaPro have played a significant role in accelerating the development and benchmarking is such algorithms [69].

From a mechanical perspective, Guo et al. [67] confirm an ongoing trend toward lightweight, anthropomorphic designs using tendon-driven actuation and additive manufacturing, reinforcing the relevance of approaches such as those proposed by Tian et al. [59]. At the same time, Guo et al. [67] emphasise the need for future research in multimodal sensing, adaptive control, and user-centered design. They suggest that the integration of mechanical simplicity with intelligent, sensor-driven control remains a key challenge for next-generation robotic and prosthetic hands.

Broota et. al [71] tackles the challenge of creating a bionic arm to minimise the repetitive and banal tasks that humans face everyday. Using the Inmoov bionic arm model, composed of 3D printed material and controlled by servo motors, the Inmoov Nervo Board, and Arduino Mega 2560. The authors concluded that the arm was successful in pick and placing applications; they emphasise that the arm could be programmed to do more complex tasks than the tasks they presented to it.

4.2 EMG

Recording surface EMG signals is constrained by their low amplitude and the presence of high sensitivity to external electrical noise. De Luca et al. [14] demonstrate that differential amplification is mandatory for damping the $50\,\mathrm{Hz}$ power-line artifacts and for providing sufficient signal-to-noise ratio. However, they also argue that it is impossible to ideally remove common-mode signals, making high CMRR necessary for proper EMG recording.

Through these insights, Merletti and Cerone [15] offer one possible approach to the detection, conditioning, and processing of EMG signals. A significant contributor to impedance imbalance is the electrode and skin interface, which directly reduces the rejection of common-mode. Evidence supporting high input impedance and common-mode rejection ratios in the amplifier front-end is presented to address common problems like motion artifacts, electrode imbalance, and existing power line interference.

To further enhance the common-mode rejection, a Right Leg Drive (RLD) amplifier, as explained by Texas Instruments[3], can be employed. This was actually designed for use with ECG, but the principle can certainly be employed for EMG front ends as well. The feedback configuration improves the CMRR by a factor of $(1 + G)$, where G represents the loop gain for the RLD amplifier, and serves to further reject both $50\,\mathrm{Hz}$ noise and DC drift due to unbalanced input or cable coupling.

Despite careful front-end design, there are still properties of EMG signals that change with time because of physiological phenomena. Muscle fatigue causes both changes in amplitude and in frequency domains of EMG signals that make it difficult to rely upon control signals. To compensate for such problems, P. et al. [16] suggest multi-sensor fusion involving EMG signals together with other signals that help alleviate problems of noise sensitivity and fatigue-induced signal drifts. Recent studies also explore the application of deep learning techniques for processing EMG signals to get adaptive models that are robust to changes in signals [72], [73], [74].

These principles are presented in recent system-level works. Kaifosh et al. [75] designed and developed a 16-channel surface EMG wristband for data acquisition in various tasks: wrist movement control, gesture recognition, and handwriting recognition. In all tasks, data acquisition was used to train supervised neural network models. Offline analysis reveals more than 90% accuracy in gesture and handwriting tasks and an angular accuracy of less than $13\,^\circ\,\mathrm{s}^{-1}$ in wrist movement control.

4.3 EMG classification

EMG data has the particularity that needs to be analysed over time for detecting the patterns of muscle activation properly to determine what specific action a user wants to perform. Once the EMG signals are preprocessed, the literature shows the next step is signal segmentation into time "windows," from which features are extracted. A review made by Castruita-López et al. [76] highlights that these features are the primary input for classification. Specifically, as detailed in the survey by Sid'El Moctar et al. [77], Time-Domain features, such as MAV, RMS, and Zero crossing, are widely preferred. This preference is due to their low computational load, making them ideal for the real-time and embedded systems that are common in robotics and prosthetics.

Once extracted, these features are fed into classification algorithms. The literature shows two main families of models. On one hand, classic Machine Learning models like Support Vector Machines, k-Nearest Neighbors, and Linear Discriminant Analysis are commonly used. On the other hand, Deep Learning models have gained significant traction, especially CNNs and Recurrent Neural Networks (RNNs).

In particular, models like LSTM, a type of RNN, are proving to be highly suitable for EMG signals, aligning with the need to analyse data temporally. Unlike static classifiers that view each window in isolation, LSTM treat the signal as a sequence, allowing them to learn temporal dependencies. For example, a study by Millar et al. [78] used an LSTM to classify 12 individual finger movements, achieving high accuracy (90%) with a low-cost sensor. However, this also highlights the predominant focus of the literature: gesture recognition and optimizing classification accuracy. However, a significant gap remains between these software-based classification studies and the mechatronic integration needed for robust, functional task control in the real world.

4.4 Vision system with robotic arms

Recent research demonstrates significant progress in integrating computer vision with robotic arm control to improve automation and prosthetic functionality. Vasan and Pilarski [79] introduced a context-aware learning framework that fuses EMG, vision, and inertial data for synergistic prosthetic arm control. This enabled context-dependent grasping through reinforcement learning. Similarly, Deshmukh et al. [80] proposed a robust fusion model combining EMG and vision to enhance grasp classification accuracy and reduce reliance on unreliable muscle signals. Complementing this, Cognolato et al. [81] employed multimodal data, including eye-tracking and computer vision, to improve prosthetic hand control through visuomotor coordination, which resulted in accuracy gain for amputees. More recent work in the area has been done by

[3] https://www.ti.com/lit/an/sbaa188/sbaa188.pdf?ts=1768728987064

Manimaran et al. [16] who designed an adaptive prosthetic arm that integrated EMG, vision (YOLO-based object detection), and pressure sensors for real-time grip adjustment and fatigue recognition. This relates to the long-term objectives and interests of this project, seeing that stroke patients naturally experience fatigue in muscles quickly. Beyond the scope of prosthetic arms, there are multiple studies that explore general robotic arm application with visual intelligence [2], [82], [83]. These previous works highlight the trend toward multimodal perception by integrating vision, biosignals, and sensor feedback to achieve adaptive, intelligent robotic arms capable of human-like interaction and contextual decision-making.

A vision system can be fused with data from IMUs to get a better orientational and positional understanding of a bionic or robotic arm. Recent studies have demonstrated that combining vision and inertial sensing can significantly improve pose estimation accuracy for robotic systems. Nyqvist and Gustafsson [84] presented a high-performance tracking system integrating a camera and an IMU through an EKF that resulted in millimeter-level accuracy by compensating for IMU drift using visual landmarks. Similarly, Xie et al. [85] designed a handheld teleoperation device fusing monocular vision and a 6-axis IMU via an Error-State Kalmar Filter, which enhanced motion stability and mitigated visual motion blur during swift movements. In 2011, Madgwick et al. [86] introduced an efficient orientation filter for IMUs which has since been used popularly [31], [87]. Chan et al. [30] further applied the fusion approach for hand pose tracking using ArUco markers and an IMU to maintain real-time accuracy even under occlusion. These works highlight the robustness of vision-IMU fusion, where visual data provides a global reference while inertial data ensures high-frequency, drift-corrected motion estimation for robotic control.

4.5 Tactile-Driven and vision-based prosthetic grasping

Recent developments in "neuro-mimetic" grasping systems have demonstrated the importance of closed-loop control using three-axis tactile sensors. These systems, often implemented on high-speed field-programmable gate array-based embedded platforms, focus on autonomous reflex motions to improve robustness and prevent slippage without requiring prior knowledge of the object's physical properties [58]. While these architectures significantly enhance grasping stability, they are inherently reactive; the system cannot anticipate the required grasp posture until physical contact is established, which may lead to sub-optimal initial positioning.

On the other hand, other studies have prioritised affordability and proactivity through vision-driven models. Some implementations using YOLOv11 on Raspberry Pi 4 platforms allow prosthetics to autonomously detect household objects with high accuracy (90.3%) and low latency ($\approx 67\,\mathrm{ms}$) [88]. These systems initiate timed gripping sequences upon detection, simulating a biological reflex. However, as noted in their findings, these models often lack adaptive grip strength control and rely on open-loop actuation. This means that if the object's weight or surface friction changes mid-grasp, the system cannot adjust its torque, leading to potential grasp failure.

The LIMB project addresses the gaps identified in both approaches by proposing a multimodal shared-control architecture. Unlike the reactive-only tactile systems [58], LIMB utilises YOLOv8 on an NVIDIA Jetson AGX Orin to identify the object during the reach-to-grasp phase, pre-configuring the hand's posture. Furthermore, it overcomes the open-loop limitations of purely vision-based arms [88] by delegating the grip management to a decentralised CAN Bus network. In this architecture, a dedicated ESP32-C3 node executes a continuous PD-control loop that monitors pressure variance. This ensures that the interaction is not merely a "timed event" but a truly adaptive reflex that maintains stability based on real-time haptic data provided from the pressure sensor, effectively bridging the gap between proactive visual recognition and reactive physical stability.

5 Method

This section describes the scientific methods and approaches used to achieve the project objectives and answer the research questions. The project combines experimental design, system implementation, and quantitative evaluation.

5.1 Research design

The study follows an experimental engineering research design based on iterative prototyping, system integration, and empirical evaluation. The overall approach is aimed at assessing the feasibility and performance of a real-time, multimodal sensing and control system for a bionic arm–hand assembly performing functional grasping tasks.

Rather than relying on analytical modelling or simulation, the system was designed and implemented as a hardware-based proof-of-concept platform, allowing direct observation of sensor behaviour, control response, and system robustness under realistic operating conditions. This approach was chosen because the research questions address practical aspects such as intention recognition accuracy, timing and control reliability, which are strongly influenced by real-world noise, latency, and hardware interactions and are therefore difficult to evaluate using purely theoretical methods.

The development process was guided by principles of modularity, safety, and practical deployability. This enables individual subsystems to be tested both independently and in iterated configurations. Empirical data were collected during structured grasp-and-release experiments and analysed to evaluate the performance of the intention classification and control pipeline in relation to the stated research objectives.

The scope of the study is limited to system-level feasibility and performance evaluation in a controlled laboratory setting. The results are intended to demonstrate technical viability rather than clinical effectiveness, and conclusions are therefore restricted to proof-of-concept validation of the proposed approach.

5.2 EMG

The EMG circuit is built and evaluated in Multisim. This method allows theoretical optimisation of the circuit and adjustments of the components of the amplifier before manufacturing.

An RLD is used in the EMG sensor to reduce the common-mode noise in the EMG signal. It provides a reference voltage and feeds back the inverted common-mode signal to the user's body.

5.2.1 Signal processing

The DC offset was removed by subtracting the mean value of the analysed segment before further processing according to (1). This prevents a dominant component at $0\,\mathrm{Hz}$ in the frequency spectrum and improves numerical stability in amplitude estimates. Removing the DC component is a standard preprocessing step in surface EMG analysis and is recommended to improve interpretability of amplitude- and frequency-domain measures [14], [15].

$$x_0[n] = x[n] - \mu, \qquad \mu = \frac{1}{N}\sum_{n=0}^{N-1} x[n]. \tag{1}$$

The DC-free signal was then band-pass filtered to reduce drift artifacts associated with movements as well as high-frequency electronic artifacts. A fourth-order Butterworth filter with the cutoff points given by $f_{\mathrm{low}} = 20\,\mathrm{Hz}$ and $f_{\mathrm{high}} = 1000\,\mathrm{Hz}$ was used. A corner frequency of $20\,\mathrm{Hz}$ is a common general cutoff to remove movement artifacts [15], [89].

Since EMG is oscillating around the reference after DC removal, simple averaging is not a good measure of amplitude. An RMS, shown in (2), is an amplitude measure that is proportional to the overall level of the signal and a standard amplitude feature in surface EMG [14], [90].

$$\mathrm{RMS}(x_f) = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x_f^2[n]}. \tag{2}$$

While the RMS provides a single value for amplitude over a segment, muscle activation develops with time. To produce a smooth signal that describes the level of activation, a moving RMS envelope was computed.

The envelope was calculated using a sliding window of M samples, corresponding to $50\,\mathrm{ms}$, as shown in (3). RMS windows of 25–$50\,\mathrm{ms}$ are commonly used in surface EMG for tracking muscle activation changes without excessive smoothing [15], [90]. The envelope represents muscle activation over time and is less sensitive to quick oscillations.

$$e[n] = \sqrt{\frac{1}{M} \sum_{k=n-M+1}^{n} x_f^2[k]}. \tag{3}$$

5.2.2 Validation protocol

The validation protocol was designed to evaluate whether the developed EMG system produces signals that are clearly distinguishable from noise, stable over time, physiologically reasonable in the frequency domain, and repeatable across trials. These aspects were evaluated using complementary time-domain, frequency-domain, and statistical metrics, which are commonly used in surface EMG validation studies [14], [15].

Each trial consisted of a rest period followed by a voluntary gripping contraction. This task was chosen because it produces a clear contrast between inactive and active muscle states, which is well-suited for EMG validation [91]. Fixed rest and active time intervals were used for all trials to ensure fair comparison between trials.

A fundamental requirement for EMG measurements is that muscle activation can be clearly distinguished from background noise. Signal-to-Noise Ratio (SNR) is therefore widely used as a quantitative performance metric in EMG system evaluation [14].

Two complementary SNR measures were computed to capture different aspects of signal quality:

- *RMS-based SNR*: Compares the RMS amplitude of the bandpass-filtered EMG signal during muscle contraction to that during rest. This metric shows the electrical quality of the recorded signal and is sensitive to both muscle activation and noise [14], [15]. The RMS-based SNR is defined as

$$\mathrm{SNR}_{\mathrm{RMS}} = 20 \log_{10} \left(\frac{\mathrm{RMS}_{\mathrm{active}}}{\mathrm{RMS}_{\mathrm{rest}}} \right). \tag{4}$$

- *Envelope-based SNR*: This measure compares the mean value of the RMS envelope during active gripping and rest. It evaluates how clearly muscle activation is visible after smoothing and is particularly relevant for applications that rely on an activation-level signal rather than the raw EMG waveform [90]. The envelope-based SNR is defined as

$$\mathrm{SNR}_{\mathrm{ENV}} = 20 \log_{10} \left(\frac{\overline{e}_{\mathrm{active}}}{\overline{e}_{\mathrm{rest}}} \right). \tag{5}$$

Using both approaches increases confidence in the validation.

Baseline drift was also considered to evaluate changes of signal amplitude over longer periods of time, which can occur due to electrode polarisation, changes of temperature, and impedance changes [15], [89]. Drift was computed as the difference between the mean signal value at the end and at the beginning of the rest period:

$$\Delta_{\mathrm{drift}} = \mu_{\mathrm{end}} - \mu_{\mathrm{start}} \tag{6}$$

Also Frequency-domain analysis was performed in order to confirm that the recorded signal shows a spectral characteristic typical for surface EMG. Surface EMG is expected to have frequency content without narrow peaks [15], [91].

Fast Fourier transform (FFT)-based amplitude spectra were analysed during both rest and gripping. During contraction, broadband spectral content indicates muscle activity, while during rest, narrow peaks may reveal electrical interference.

To quantify frequency-domain differences between rest and contraction, the total spectral energy within the EMG frequency band was compared using a bandpower ratio. Bandpower metrics are commonly used in EMG analysis to summarise frequency-domain changes associated with muscle activation [14].

$$\mathrm{BandPowerRatio} = 10 \log_{10} \left(\frac{P_{\mathrm{active}}}{P_{\mathrm{rest}}} \right) \tag{7}$$

This indicates that the signal energy increases during gripping and that the signal is not dominated by low-frequency noise.

All the trial validation metrics were calculated for each trial separately, and the results are represented by mean and standard deviation values. Reporting variability across trials is recommended in EMG system evaluation to assess repeatability and robustness [91].

5.3 Multimodal intent recognition

To address research question 1 and the associated objectives concerning intention recognition within a multimodal sensing system, an experimental method based on EMG-driven intention classification combined with contextual sensing was employed. The primary objective was to evaluate whether reliable grip-intention recognition can be achieved using wearable EMG signals, and how this recognition can be integrated with additional sensing modalities to support robust system-level behaviour.

5.3.1 Experimental task and data collection

A structured functional task was designed in which participants performed repeated grasp-and-release movements of a coffee mug. This task was selected because it represents a realistic rehabilitation-relevant action involving coordinated muscle activation, limb motion, and object interaction.

Surface EMG signals were collected using electrodes placed on the forearm, capturing muscle activity associated with grip initiation and release. The EMG data were time-stamped and labelled automatically according to a specific sequence: two seconds of rest, four seconds of grip, and two seconds of rest. This dataset served as the sole source for training and evaluating the intention classification model. Ideally, a public dataset could have been used to complement the dataset, but the EMG signals collected in the public dataset did not meet the required frequency and amplitude.

In parallel, IMU data were collected in separate experiments to evaluate motion direction and orientation estimation performance. Vision-based perception was used to detect the target object and estimate its spatial pose during system operation but was not included in the training dataset for intention classification. This separation ensures that the learned model relies exclusively on EMG signals, while additional modalities contribute contextual information at the system level.

5.3.2 EMG signal preprocessing and temporal representation

The EMG signals underwent preprocessing to improve signal quality and reduce artifacts commonly encountered in surface EMG acquisition. These steps were designed to preserve temporal characteristics relevant to grip-intention onset while minimising noise that could degrade classifier performance. To enable temporal modeling, the EMG data were segmented into fixed-length time windows. This window-based representation captures the dynamic evolution of muscle activation patterns preceding and during grasp and release action. Then, features were extracted from the raw signals. These features were used as an input to the classifier.

5.3.3 Intention classification model and evaluation approach

Grip-intention recognition was performed using a LSTM neural network trained solely on the processed EMG data. A temporal deep learning model was selected because grip intention is not instantaneous but unfolds over time through characteristic muscle actuation sequences.

The trained model outputs a probabilistic classification of grip or release intention, which is then used as input to the system's higher-level control logic. By decoupling intention recognition from motion execution and environmental perception, the method enables independent evaluation of classification performance while maintaining compatibility with multimodal control architecture.

The performance of the intention classification method was evaluated using the labelled EMG dataset. Classification accuracy and temporal consistency were assessed to determine how reliably grip and release intentions could be detected. This evaluation strategy enables assessment of both standalone EMG-based intention recognition and its role within a broader multimodal sensing control framework.

5.3.4 Integration with contextual sensing

While intention recognition was based exclusively on EMG data, additional sensing modalities were incorporated at the system level to provide contextual information. IMU-based motion estimation was used to infer limb orientation and movement direction, and vision-based perception was used to detect and localise the target object. These modalities were integrated through rule-based logic and state transitions rather than data-driven fusion. This design choice allows each modality to contribute complementary information without requiring synchronised multimodal training data. It also improves interpretability and supports modular testing of individual subsystems.

5.3.5 Methodological justification

An EMG-only training approach was selected due to the availability of reliable labelled biosignal data and the well-established relationship between EMG pattern and hand-grip intention [92]. Collecting and synchronising high-quality multimodal datasets poses significant practical challenges, particularly in wearable and real-time systems [93]. By evaluating additional modalities separately and integrating them at the system level, the method balances scientific rigour with practical feasibility while still addressing research objectives related to robust, intention-driven control.

5.4 Mechatronic integration and control

To address research question 2, which concerns the integration and control of a mechatronic bionic arm system, an experimental system-integration methodology was adopted. The aim of this method was to evaluate whether a distributed sensing, actuation, and communication architecture can support stable, coordinated arm and hand motion under real-time control constraints.

Rather than optimising individual subsystems in isolation, the method focused on functional integration and empirical validation of mechanical, electrical, and control components within a unified robotic platform. This approach reflects the practical challenges inherent in rehabilitation-oriented robotic systems, where hardware interactions, communication latency, and sensor reliability strongly influence performance.

5.4.1 Mechatronic integration strategy

A physical robotic torso-arm-hand assembly was developed as a hardware-in-the-loop test platform to enable direct evaluation of joint actuation, tendon-driven hand mechanics, and sensor feedback under realistic operating conditions. The mechanical structure was designed to approximate human arm kinematics while remaining modular, allowing iterative refinement and replacement of individual components during development. A tendon-driven actuation concept was employed, enabling compliant grasping behaviour. Iterative prototyping was used to empirically assess joint range of motion, friction, and mechanical robustness with successive design iterations addressing observe red limitations. This iterative process was chosen in place of analytical modelling due to the difficulty of accurately modelling friction, compliance, and manufacturing tolerances in low-cost, 3D-printed mechanisms.

To address the constraints of component availability and integration time, a pragmatic availability-first hardware selection methodology was applied. Components were chosen based on immediate supply chain availability and compatibility with the central processing unit, which operates at $3.3\,\mathrm{V}$ logic levels.

5.4.2 Actuation and control logic

Joint actuation was implemented using a combination of servos and steppers that was selected to meet differing torque and speed requirements across the arm and hand. Instead of performing formal torque modelling, the selection of actuators was determined by factors such as acquisition cost, availability, and results from functional testing, while also considering similar work within the field of bionic arms.

All actuators were directly or indirectly controlled using Pulse Width Modulation (PWM) signals generated by distributed MCUs. This distributed control approach was chosen to reduce wiring complexity, improve modularity, and enable parallel sensing and actuation across the system.

A dedicated firmware calibration was developed for the hand to map the non-linear servo actuation to a normalised grip index [0,1]. A passive-compliance strategy was implemented to address the volumetric constraints of the wrists, combining active servo actuation with mechanical return springs. The calibration method involved determining the precise zero-tension angle for this tendon-driven mechanism to prevent

continuous current draw against the passive return spring, thereby reducing thermal load and power consumption during idle states.

5.4.3 Sensor feedback and state estimation

To enable closed-loop control and state awareness, the system integrates multiple complementary sensors, including IMUs along the arm and potentiometers for the joint rotations. This combination allows both local joint feedback and global limb motion estimation, increasing redundancy and improving robustness against individual sensor failure.

This approach prioritises sensor redundancy and complementary feedback, which is particularly important in rehabilitation robotics where safety and reliability are critical. Sensor data supports both low-level actuator control and higher-level state transitions within the control architecture.

5.4.4 Communication architecture

A distributed communication architecture was employed to coordinate sensing and actuation across the system. A central computing node handled high-level control logic, while multiple peripheral microcontrollers managed sensor acquisition and actuator commands. Communication between these units was implemented using the CAN protocol suitable for real-time embedded systems. The communication method was evaluated analytically to ensure that message sizes and update rates were compatible with real-time operation.

The selection of CAN as the primary system communication protocol was determined through a comparative analysis against standard embedded system protocols such as Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I^2C), and Recommended Standard 485 (RS-485). SPI and I^2C protocols are commonly used for board-level communication and rely on single-ended signalling, which increases their vulnerability to Electromagnetic Interference (EMI) from nearby stepper motors [27]. This can compromise the reliability of data transmission in environments where motors operate in close proximity to the data lines. Furthermore, SPI requires additional wires to be connected to each node, leading to increased wiring complexity and greater harness stiffness, specifically when routed through the arm joints. While the RS-485 provides differential signalling and effectively eliminates noise, it lacks hardware-level collision avoidance. In multi-master systems with asynchronous sensor data transmission, RS-485 would require complex software arbitration to prevent data corruption [94]. The CAN protocol was chosen due to its ability to address both of the previously identified challenges. Similar to RS-485, CAN provides robust differential noise immunity, which allows for reliable operation in environments with high-current motor cables. Additionally, CAN incorporates hardware-level message priority and arbitration mechanisms, ensuring that critical safety commands, such as emergency stop signals, are transmitted without being affected by lower-priority sensor data. While both SPI and I^2C were unsuitable to be used as the primary system communication protocol, they were implemented for local board-level communication. In this configuration, each distributed ESP32-C3 node functions as a local master, collecting data from IMUs through I^2C buses and controlling stepper motors through SPI buses.

5.4.5 Wireless data transmission

To support user mobility and wearable sensing, wireless communication was used to transmit bio-signal data from the user-side sensors to the central computing node. A low-latency transmission strategy based on periodic notifications and buffered data packets was employed to reduce communication overhead and ensure the timely delivery of sensor data. The communication parameters were selected based on protocol constraints and empirical latency considerations rather than theoretical throughput alone. This method reflects the practical limitations of wireless communication in real-time human-robot interaction systems.

5.4.6 Testing and validation approach

Validation was conducted incrementally through subsystem-level testing, including verification of sensor data acquisition, actuator responsiveness, and communication reliability. A theoretical analysis of the BLE- and CAN communication was performed to evaluate the feasibility of the system design with regard to latency and throughput. Due to incomplete mechanical assembly at the time of evaluation, full closed-loop system testing was not performed. Instead, integration readiness was assessed through partial system tests and functional demonstrations. This staged validation approach allows early identification of integration issues while acknowledging practical constraints on system completeness.

5.4.7 Methodological justification and limitations

The methodology for research question 2 prioritises functional integration and empirical evaluation over analytical optimisation. Given the system's complexity and the limited development time frame, simulation-based modelling was deemed insufficient to capture the interactions among mechanical compliance, sensor noise, communication latency, and control behaviour. The resulting method is therefore well-suited for feasibility assessment but does not aim to provide optimal mechanical or control performance.

6 Ethical and Societal Considerations

The LIMB project primarily involves the development of a robotic arm and multimodal sensing framework aimed at, in the long term, assisting stroke patients in regaining motor function. As such, it raises several ethical and societal considerations related to research integrity, user safety, data handling, and long-term social impact. From a research ethics standpoint, the project does not involve external human participants at this stage and therefore does not raise concerns related to privacy and informed consent or anonymity. By internal human participants, we refer to the project team. However, because the system is ultimately designed for human interaction and rehabilitation, future testing involving patients will require adherence to ethical research guidelines. This includes informed consent, data protection under GDPR, and oversight by relevant medical ethics committees. The team must ensure that collected data, such as EMG and vision information, are securely stored, anonymised where possible, and used strictly for research and development purposes.

Societally, the LIMB system supports several sustainability and inclusion goals. By offering a potential alternative to costly and time-intensive physiotherapy, it contributes to social sustainability through improved accessibility to rehabilitation. The design philosophy emphasises affordability and modularity, making it adaptable for healthcare systems. Economically, such technology could reduce long-term care costs by empowering patients to perform guided therapy independently. The platform also has implications beyond stroke rehabilitation. A bionic arm with integrated sensing and control could, with appropriate safeguards, assist individuals who experience involuntary movements or epileptic episodes. In such cases, automated stabilisation or motion suppression could prevent self-injury or accidental harm, enhancing personal safety and quality of life. Although this application lies outside the current project's primary focus, it highlights the broader societal value of safe, adaptive mechatronic assistive devices.

From an environmental perspective, the use of efficient embedded hardware and recyclable components helps reduce waste and energy consumption. Legally, future iterations of the LIMB system will need to comply with medical device regulations and standards concerning safety and reliability. Overall, while the present research stage poses minimal ethical risk, its societal implications are significant: it demonstrates how human-centered robotic systems can contribute not only to rehabilitation but also to safer and more inclusive assistive technologies.

7 Design and integration

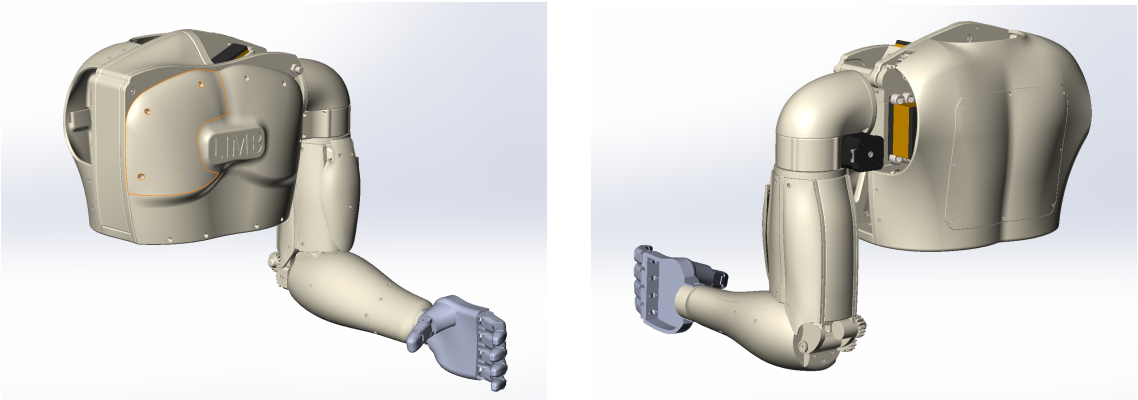
This section give insight in to the mechanical, electronic, and software implementation of the LIMB system, establishing the foundation required for evaluating system functionality, performance, and experimental results.

7.1 Mechanical Design of the Bionic Arm

The mechanical design of the bionic arm is based on the open-source InMoov arm, which served as the initial structural and kinematic reference. The original InMoov design has been adapted and modified to meet the requirements of the LIMB system, which includes integration with custom electronics and sensors. The arm provides five mechanically actuated DOF, enabling controlled positioning of the arm within its operational workspace. All joints are actively driven. The joints consist of:

- Shoulder abduction/adduction
- Shoulder extension/flexion
- Shoulder humeral rotation
- Elbow flexion/extension
- Wrist rotation

To facilitate easier maintenance and component replacement, screws are used instead of the interference fits used by the original InMoov design. This change reduces the risk of damaging adjacent parts or components during disassembly or replacing structural parts. The number of bearings used within the joints has been increased to improve alignment and minimise backlash. Further modifications have been made to enhance the durability of the mechanical transmission components. In the original design, plastic lead screws are present, which have instead been replaced with stainless-steel lead screws. This change improves wear resistance and ensures more consistent mechanical behaviour in the joints that require linear motion, which concerns the elbow and shoulder abduction/adduction joints. All structural components have been designed in SolidWorks. In figure 1, there is a display of the CAD assembly in SolidWorks. Structural parts have been 3D-printed in polylactic acid (PLA), enabling rapid prototyping and iterative refinement of the mechanical design while providing sufficient stiffness for the intended application. The arm is mechanically mounted to an InMoov torso which serves as a base for the arm assembly and as a housing for the majority of the embedded electronics. The position of the camera was decided to be above the shoulder. To achieve this, a mount for the camera has been designed. The mount stretches from the “neck” of the torso out towards the shoulder. With the mount, the camera can be angled up and down depending on what position of the camera would work the best.



(a) Front right side image of the torso and arm

(b) Back left side image of the torso and arm

Figure 1: This figure displays the assembly of the torso and the arm in 3D using SolidWorks. Note, that the hand in these images does not represent the final product in the hand.

7.1.1 Mechanical Design and Hand Actuation

The design of the hand is based on a five-finger structure, with each finger consisting of two joints providing one DOF per joint. The finger movements are actuated using a tendon-driven mechanism inspired by the flexor tendons of a human hand. Rigid fishing wire acts as the tendons to transmit force from the servo horns to the finger joints. The design was developed in CAD and 3D printed as an early-stage prototype. This additive manufacturing method enables rapid iteration while minimising cost when testing different mechanical configurations. Final iteration of the hand is seen in figure 2.

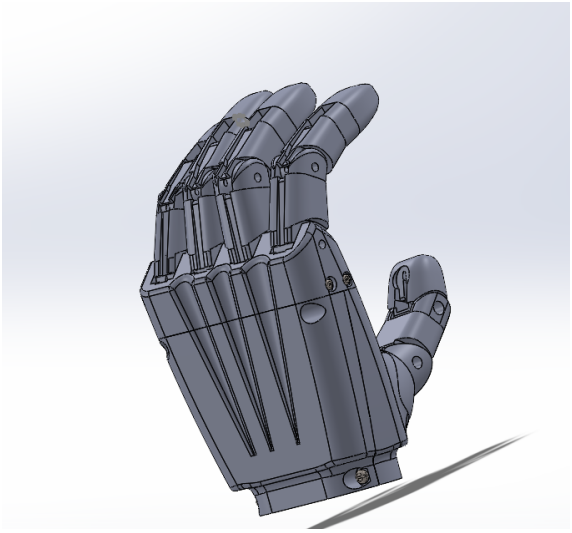


Figure 2: Last iteration of the hand represented in 3D, using SolidWorks.

The hand actuation presented additional challenges due to the limited space in the wrist area. To address this, a single-tendon mechanism was adopted for all five fingers, with actuation provided by HS-422 servos positioned in the forearm to optimise space utilisation and maintain reliable finger movement. In this design, a passive extension mechanism has been selected, in which springs are installed at each finger joint to generate a restoring force. The servo contracts the tendon to produce finger flexion for gripping, and when the servo is released, the stored energy in the spring extends the finger back to its original position.

7.1.2 Arm Actuation System

The HV2060MG servo motors are limited to 180° of rotation, but since the gear ratios between the motors and the joints are both 15:1, their effective range of motion ends up being $180^\circ/15 = 12^\circ$. This was considered inadequate, so the HV2060MG motors have been modified to provide continuous motion. This was achieved by removing the mechanical stop inside of the gearbox and the internal potentiometer from the controller within the servo case. Additionally, soldering a resistor on the controller between its V_{in} and V_{out} as well as between V_{out} and ground. This achieves the effect of making the motor believe it is at its centre position, as indicated by a pulse width of $1500\,\mu s$. The direction of its rotation is then determined by whether the pulse width is greater than or less than $1500\,\mu s$, and the speed is determined by the magnitude of the difference between the pulse width and $1500\,\mu s$. The relationship between the velocity of the servo motor and its corresponding pulse width is estimated by timing how long it takes to perform one rotation at the closest pulse width to $1500\,\mu s$ that produces motion and at the closest pulse width to $1500\,\mu s$ that no longer seems to produce any further increase in speed. Conversions between pulse widths and velocities can then be performed using linear interpolation between the two.

Since the HV2060MG servo motors and the NEMA-17 stepper motors were both controlled by setting their angular velocities, trapezoidal motor ramping was implemented in software to reduce mechanical stress from sudden stops.

7.1.3 Distributed Power Architecture

A centralised Power Distribution Unit (PDU) was developed to address the varying voltage requirements of the hybrid actuation system. As illustrated in Figure 3, the PDU converts the main $12\,V/26.7\,A$ bench

supply into three isolated voltage rails by using separate buck converters.

- $12\,\mathrm{V}$ rail supplies the NEMA-17 stepper drivers (DRV8825), which are used for high-torque joint holding.
- $7\,\mathrm{V}$ rail powers the HV2060MG shoulder servos.
- $6\,\mathrm{V}$ rails are used for the VMA601 wrist servo and the HS-422 finger servos.
- $5\,\mathrm{V}$ rail powers the distributed ESP32 nodes.

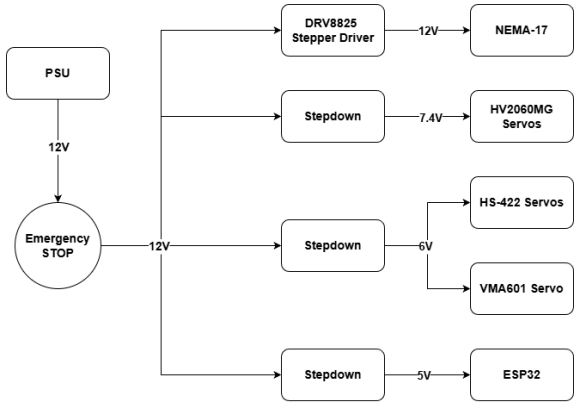


Figure 3: Block diagram of the main PDU designed to isolate inductive motor loads from sensitive logic components.

As a security measure, an emergency button was added between the PSU and the distributed voltage rails.

7.1.4 Embedded Hardware and Communication

The communication backbone of the system is established using a CAN bus, which interconnects the central NVIDIA Jetson AGX Orin with four distributed ESP32-C3 MCU nodes. In this configuration, each microcontroller is linked to the CAN bus through a daisy-chain topology, and twisted pair cabling is employed to effectively reduce EMI and ensure reliable data transmission.

One of the main integration challenges encountered was the voltage level mismatch between the standard $5\,\mathrm{V}$ CAN logic and the $3.3\,\mathrm{V}$ I/O levels of both ESP32-C3 and Jetson Orin controllers. Initial attempts to use Printed Circuit Boards (PCBs) based on the PCA82C251T transceiver were unsuccessful, as this component required a $5\,\mathrm{V}$ supply and logic voltage. To address this issue, a custom transceiver PCB was designed and fabricated using SN65HVD233D, which was specifically chosen for its compatibility with $3.3\,\mathrm{V}$ logic levels and its reliable availability. This solution ensured safe and effective communication between all nodes.

The custom PCB integrates the SN65HVD233D transceiver, and a $120\,\Omega$ termination resistor at each end of the bus to suppress signal reflection. This hardware solution enabled the establishment of a stable $1\,\mathrm{Mbit\,s^{-1}}$ communication link, which supports the real-time transmission of sensor data.

To optimise the system architecture and reduce wiring complexity, the control logic is distributed across four dedicated nodes, each responsible for specific actuation and sensing tasks within the overall system. An overview of the distributed embedded hardware architecture and the communication between the individual nodes is shown in figure 4.

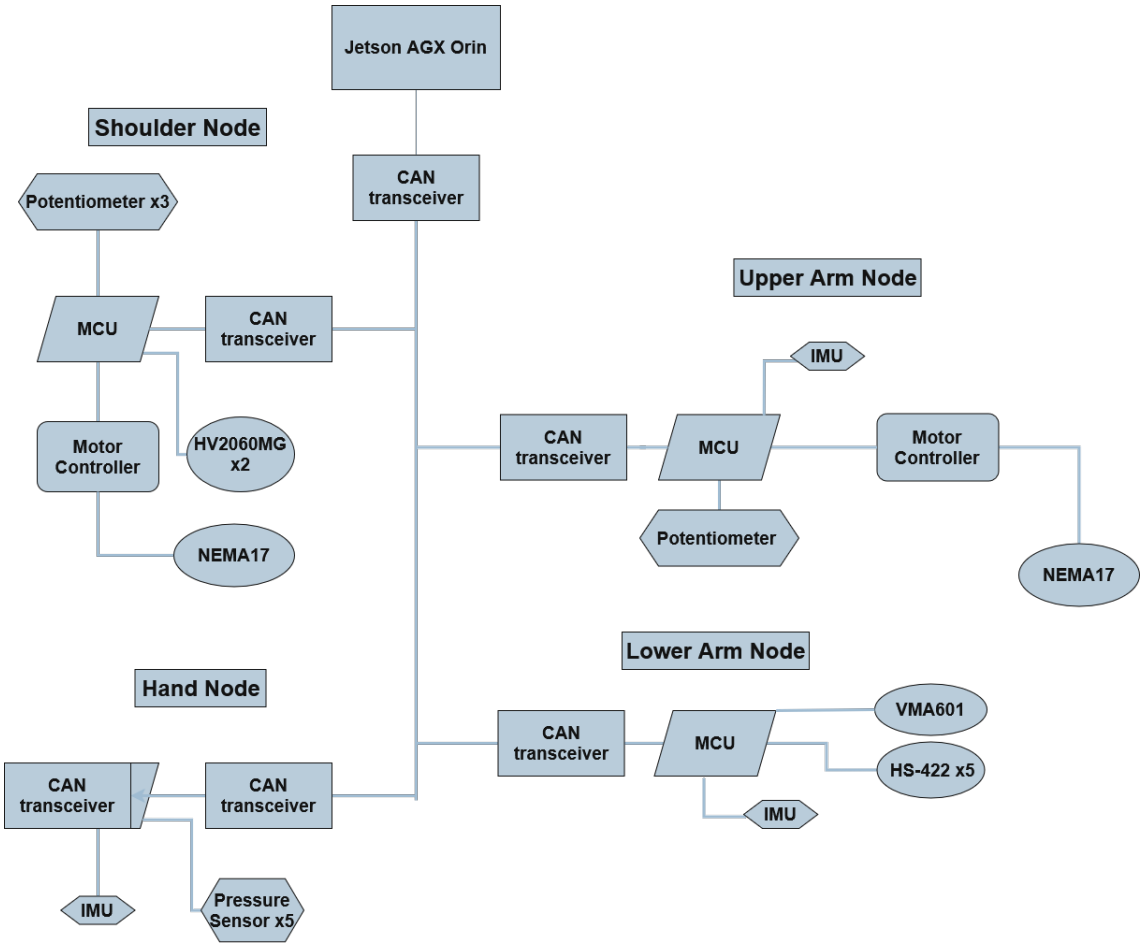


Figure 4: Distributed embedded hardware architecture of the bionic arm, showing the shoulder, upper arm, lower arm, and hand nodes interconnected via a CAN-bus and coordinated by the Jetson AGX Orin

- The shoulder MCU is connected to both HV2060MG servo motors controlling the shoulder joints, as well as the NEMA-17 stepper motor controlling the humeral rotation. Using ADC readings from the potentiometers attached to each of the joints, the angle of the joints could be obtained and used to periodically update the angular velocity of the motors based on the joint's target angle.
- The upper arm MCU is connected to the NEMA-17 stepper motor that controls the extension and flexion of the elbow. Similar to the shoulder MCU, the upper arm MCU reads ADC values from the joint's potentiometer to obtain its angle and then updates the motor's angular velocity based on its current angle and its target angle.
- The lower arm's MCU is connected to the VMA601 servo controlling the wrist rotation, along with the five HS-422 servos controlling each finger.
- Finally, the hand's MCU is connected to the FSR400 pressure sensors connected to the fingers. After receiving a CAN message to start gripping, the MCU will calculate how hard the hand should be gripping based on the values read from the pressure sensors and then send a CAN message to the lower arm's MCU with the angle that should be used by the servo motors controlling the fingers.

7.1.5 Adaptive grasping

The adaptive grasping routine is implemented using a decentralised architecture that mimics the biological hierarchy of motor control. In this scheme, the NVIDIA Jetson AGX Orin functions as the "central nervous system," performing high-level inference by fusing real-time vision system data with user intent from EMG signals. This high-level controller initiates the grasping sequence but delegates real-time stabilisation to the distributed ESP32-C3 nodes by sending a start command via the CAN bus.

The adaptive grasping strategy is implemented as a FSM, illustrated in Figure 5, that structures the interaction between high-level user intent and low-level tactile feedback, ensuring that the system responds efficiently and reliably to user commands. This logic is implemented as a state machine within the hand MCU, transitioning from an initial intent to a stabilised grip through continuous haptic monitoring. Upon receiving the activation command from the NVIDIA Jetson AGX Orin via the CAN bus, the system enters the **Idle (S0)** state, where the hand remains open, and a calibration sequence is initiated, capturing initial pressure values from the FSR sensors to establish the current set point. No object manipulation is performed at this state. When a grasp intent is received, the system transitions into the **Pre-Grasp (S1)** state, during which the fingers initiate closure with low applied force and contact detection remains active. Once the mean pressure surpasses a specified contact threshold, the controller advances to the **Contact Detected (S2)** state. Once calibrated, the controller transitions to a **Stable Grasping (S3)** state, where it continuously analyses the grip by processing haptic data over a $500\,\mathrm{ms}$ window to calculate the grip's mean pressure and variance. This statistical analysis is used to distinguish between secure hold and slippage. If the stability check detects an anomaly, such as a spike in variance indicating slippage, the system moves into an **Active Adjustment** state **(S4)**. In this state, a PD controller is executed to correct the motor position and restore grip integrity. Following the correction, the algorithm re-evaluates the stability metrics to determine if the pressure setpoint needs to be updated or if the system can return to its **S3**. Furthermore, the architecture includes a dedicated **Safety Grip State (S5)** as a fail-safe measure. In this state, adaptive adjustments are bypassed in favor of a constant, moderate-to-high torque hold, ensuring a secure grasp when sensor noise or environmental factors prevent reliable PD control. During the stable grasping, if a release command is detected, the system transitions to the **Release (S6)** state, where the fingers open to disengage the object. Once the fingers are fully open, the system returns to the **Idle (S0)** state, ensuring it is prepared for the next operation. The logical flow and the transitions between these states are illustrated in the diagram shown in Figure 5.

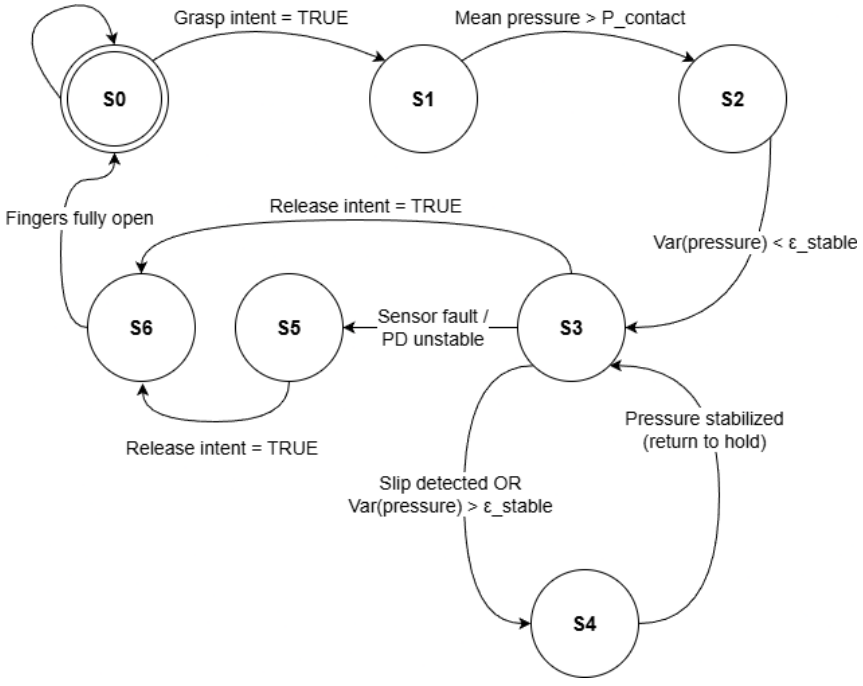


Figure 5: FSM of the adaptive grasping controller. The idle state (S0) includes system initialisation and sensor baseline calibration, while reflex-like adaptive correction (S4) enables autonomous grip stabilisation based on tactile feedback.

In further detail, the calibration sequence in **S0** involves initialising the haptic range by sampling the maximum and minimum pressure values. This is achieved by sweeping the system within its mechanical limits, sending minimum and maximum angular commands to normalise the sensor feedback before establishing the initial grasping angle and its corresponding pressure setpoint.

Once the grip is active, State 2 implements a continuous data acquisition routine that processes 0.5 second windows with a sampling interval of $50\,\mathrm{ms}$. Each data point represents the collective mean pressure from the FSR array distributed across the hand: one sensor on the thumb, two on the middle finger, and two

on the ring finger. This multi-point averaging provides a unified 1-DOF feedback signal that filters local noise while capturing the overall stability of the contact area.

Finally, State 3 incorporates critical safety interlocks to protect the mechanical integrity of the prosthesis. To prevent erratic movements and ensure that the servomotors can respond within their operational bandwidth, angular increments are strictly capped at 3 degrees per adjustment cycle. Furthermore, the algorithm continuously monitors the absolute angular limits of the HS-422 servos. By preventing the system from exceeding these boundaries, the controller avoids mechanical stalling and excessive current draw, thereby mitigating the risk of thermal damage or motor burnout during prolonged grasping tasks.

7.2 Cuff and EMG

This system is developed as a wearable sensor platform with the inclusion of an EMG sensor, an IMU sensor, and a piezoelectric sensor along with an embedded MCU. All these components are placed on a custom-made 3D-printed cuff.

7.2.1 Cuff design

The cuff is designed using SolidWorks software and manufactured with a 3D printer using thermoplastic polyurethane (TPU) material. TPU is selected for its flexibility and elasticity, and provides better comfort during extended wear while maintaining contact pressure during arm movement.

The cuff provides mechanical support for the electronic components and ensures the positioning relative to the limb. The electronics are enclosed on the cuff, while the electrode leads exit the enclosure and connect to the surface electrodes placed directly on the skin. Separating the electrode placement from the mechanical enclosure enables flexible positioning of electrodes for different users.

7.2.2 EMG sensor

The EMG circuit is designed and evaluated in Multisim. This allows for theoretical optimisation of the circuit and adjustments of the amplifier components prior to manufacturing. The circuit is then routed in Ultiboard and milled on a double-sided copper board in a Computer Numerical Control router machine.

This amplifier is an active system designed to acquire the low-amplitude EMG signal while reducing noise. It is mounted on the cuff, close to where the electrodes are placed.

Differential Amplifier The two electrodes are connected to an instrumentation amplifier (AD623ARZ) with $7\,\mathrm{cm}$ wires. The wires are kept as short as possible to minimise the power line noise which appears due to the small capacitance between the input wires and the power line.

The AD623ARZ is chosen because of its CMRR and input impedance.

- CMRR $\geq 100\,\mathrm{dB}$
- Input impedance $\geq 10\,\mathrm{G\Omega}$

To avoid distortion and attenuation of the signal, the input impedance of the instrumentation amplifier should be as large as possible. The gain of the AD623ARZ is programmed by two gain resistors (R_G) and set to a factor of 100 according to (8) found in its datasheet[4]. This gives a CMRR $> 110\,\mathrm{dB}$. A low gain value decreases the CMRR of the circuit. The value of the gain resistors is calculated in (9) and divided by 2 since the gain resistors are coupled in series.

$$R_G = \frac{100\,\mathrm{k\Omega}}{G-1} \tag{8}$$

$$R_G = \frac{100\,\mathrm{k\Omega}}{100-1} \approx 1.01\,\mathrm{k\Omega} \tag{9}$$

Between the output of the instrumentation amplifier and its reference pin, there is a feedback integrator circuit which serves the purpose of removing the DC offset at the output and to add the reference voltage to make the signal centred around $1.5\,\mathrm{V}$.

[4] https://www.analog.com/media/en/technical-documentation/data-sheets/ad623.pdf

Common-Mode voltage feedback To improve the circuit's CMRR, voltage feedback is applied according to Figure 18 in Appendix A. This works according to the same principle as RLD that is used when measuring ECG. By feeding back the inverted common-mode signal, it is possible to cancel the signals common to both electrodes and, in that way, eliminate the ambient noise of the EMG signal. It also works to bias the human body by being a common reference for the differential input. Due to this, the feedback electrode needs to be placed on electrically neutral tissue.

To protect the instrumental amplifier from parasitic capacitance on the gain resistor pins, a buffer is placed as a first step in the feedback circuit. The signal is then amplified to increase the CMRR further.

Amplification stage The amplification stage consists of two inverting amplifiers. This configuration makes it possible to connect the reference voltage to the non-inverting input of the operational amplifiers and can be seen in figure 18 in Appendix A.

The gain of the first amplifier is set to a factor of -1. To be able to tune the amplitude of the signal, a trimmer potentiometer is used for the second amplifier for a variable gain in the range -1 to -2. The total gain of the system is therefore 100 to 200.

The second inverted amplifier also includes a low-pass filter with a cutoff frequency of $1\,\mathrm{kHz}$.

The value of the capacitor C_f in the filter is given by (10), where R_f is the feedback resistor of the inverting amplifier and f_c is the cutoff frequency of the low-pass filter.

$$C_f = \frac{1}{2\pi R_f f_c} = \frac{1}{2\pi \times 20\,000 \times 1000} \approx 8\,\mathrm{nF} \tag{10}$$

Power section Figure 19 in Appendix A presents the power and reference section. Low-noise linear regulators generate stable supply rails. A buffered mid-supply reference establishes a virtual ground.

7.3 Software architecture and design

This section describes the software architecture of the bionic arm system. It focuses on the organisation of software components, data flow between modules, and the control logic that integrates perception, intention recognition, and actuation. The architecture is designed as a distributed system, where computationally demanding perception and learning tasks are executed on a central processing unit. Time-critical sensing and actuation are handled by embedded microcontrollers. This separation enables modular development, scalability, and robustness against partial system failures.

Figure 6 illustrates the high-level software architecture, highlighting the flow of sensor data through preprocessing, decision-making, and control layers.

7.3.1 Overall system architecture

The system follows a hierarchical and distributed software architecture composed of a central processing unit and multiple embedded nodes. An NVIDIA Jetson AGX Orin acts as the main computational unit, responsible for high-level perception, machine learning inference, sensor fusion, and decision-making. Low-level sensing and actuation are delegated to ESP32-C3-Zero microcontroller nodes, which interface directly with sensors and motors.

Sensor data from the EMG cuff, IMUs, vision system, and pressure sensors are acquired at the embedded or peripheral level and forwarded to the Jetson through wired (CAN) or wireless (BLE) communication links. On the Jetson, these data streams are processed asynchronously by dedicated software modules before being consumed by a central controller. The controller coordinates the system behavior through a state-based control logic and issues high-level motor commands, which are transmitted back to the embedded nodes for execution.

This architecture enables clear separation between perception, decision-making, and actuation, while allowing individual subsystems to be developed and evaluated independently.

7.3.2 Software components and responsibilities

The software system is divided into modular components, each with a well-defined responsibility:

- **Wearable sensing and acquisition:** Functionality related to collecting human-side data (EMG/IMU/piezo) and preparing it for analysis and transmission.

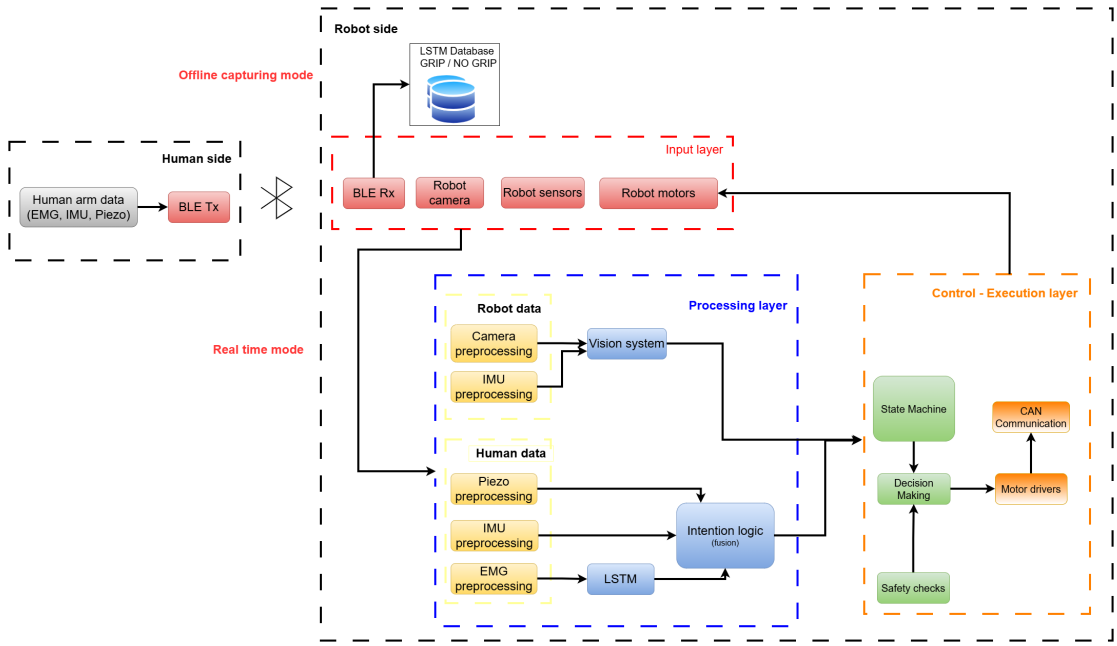


Figure 6: The high-level software architecture where the layers are separated by colors as follows: blue is the sensor layer, yellow is the preprocessing layer, red is the fusion and decision layer, green is the control layer, orange is the actuation layer, and grey is the feedback layer.

- **Preprocessing and feature extraction:** Includes filtering, normalisation, windowing, and temporal alignment strategies required for consistent machine learning inputs and comparable multimodal timing.
- **Perception subsystem:** Vision-based cup detection and pose estimation intended to provide environmental state and object-relative cues to the controller.
- **Intention recognition subsystem:** Temporal inference for grip initiation/release based on EMG, implemented as an LSTM-based pipeline.
- **Main controller:** The system-level "brain" that consumes sensor/perception outputs, applies control logic, and produces motor-level intents/targets.
- **Embedded node software:** Low-level drivers and control loops interfacing with actuators and local sensors, exposing motor and sensor state to the Jeston through the network.
- **Utilities and tooling:** Scripts support experiments, logging and reproducibility.

This component separation supports subsystem-level testing (e.g. EMG pipeline validation independent of vision), which is particularly important when the complete integrated system cannot be evaluated end-to-end.

7.3.3 Control architecture and state-based logic

Control is implemented using a state-based design in which system behavior is decomposed into discrete operational modes. A FSM in the main controller coordinates the overall flow, such as idle monitoring, intention-triggered grasping, grasp execution, and safety/error handling. Each state defines:

1. whose sensor inputs are relevant,
2. what conditions trigger transitions and,
3. what actuator commands are permitted.

Transitions are event-driven and can be triggered by multiple sources, such as

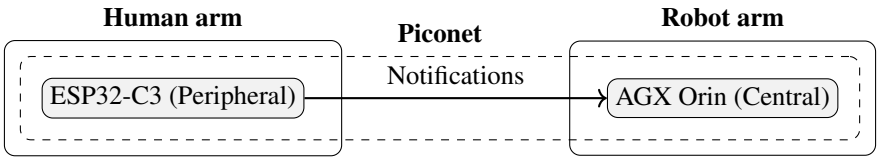


Figure 7: Human to robot piconet topology

- Intention events (grip/release) from the EMG inference pipeline,
- perception events from the vision subsystem,
- feedback events (pressure thresholds/joint limit conditions) from embedded sensing,
- system-level timeouts and communication health checks.

This structure allows the system to degrade gracefully; if one modality is unavailable, the controller can fall back to a reduced set of transitions or enter a safe state rather than producing undefined behavior.

7.3.4 Communication and data flow

The data flow is organised around a sensing-to-decision-to-actuation pipeline with feedback loops. The wireless communication from the human-side to the main controller is performed with BLE, while the communication between nodes on the bionic arm is performed using a CAN bus.

The AGX Orin uses the Python library bleak[5] to act as the Central in the BLE piconet. Initially, the plan was to use an NVIDIA Jetson Orin Nano as the Peripheral that would be strapped to the user. To achieve this, the Python libraries bless[6] and Bumble[7] were used in attempts to set up a BLE server on the Orin Nano. However, after running into issues where the Orin Nano's Bluetooth daemon crashed while trying to establish a connection to the server, regardless of the library used, it was decided to instead use one of the ESP32-C3s as the Peripheral using the ESP-NimBLE port in the ESP-IDF[8] development framework. The topology for the final piconet can be seen in Figure 7.

Since this project is developing a prototype that will be used in a controlled environment, the communication over BLE is not encrypted or protected against man in the middle attacks. This decision was made to simplify the implementation process and avoid potential complications during development.

In order to reduce latency and prevent retransmissions of outdated packets, a subscription using notifications is used. By using a subscription, the Peripheral can start sending packets as soon as the sensor data is read instead of waiting for a read request from the Central. Another thing that is done to reduce latency is to decrease the connection interval so that the queuing delay is decreased by sending queued packets more frequently, meaning all packets arrive at the Central earlier. The connection interval is set to $10\,\mathrm{ms}$ which aligns it with the $100\,\mathrm{Hz}$ sample rate for the IMUs, the lowest sample rate of all sensors. Finally, to decrease the transmission delay, Low Energy (LE) 2M Physical Layer (PHY) is used to increase the gross bit rate to $2\,\mathrm{Mbit\,s^{-1}}$. This has the effect of both reducing the overall latency and increasing the throughput.

In order to reduce potential overhead in the communication, the sensor data is buffered before transmission. Since each notification includes overhead in the form of different headers and packet fields, reducing the number of notifications sent also reduces the amount of overhead. Any type of data that is sent more frequently than the connection interval will end up being queued until the next connection event is reached, after which each piece of queued data will be sent in their own separate packets. Since the sensor data from the user are being collected continuously, the samples can be buffered for the same amount of time as the connection interval without affecting when the data are sent. Given that the connection interval used is $10\,\mathrm{ms}$, the resulting buffered sizes used are shown in Table I. It was originally planned to use an EMG sensor and an IMU sensor for the upper arm in addition to the ones on the lower arm, but they were removed during the course of the project. However, even though the sensors are no longer used, the data buffers have not been updated to reflect this case, and as such Table I includes space for the data from two EMG sensors and two IMU sensors.

[5] https://github.com/hbldh/bleak/tree/v0.22.3
[6] https://github.com/kevincar/bless/tree/v0.2.6
[7] https://github.com/google/bumble
[8] https://github.com/espressif/esp-idf/tree/v5.5.1

TABLE I
BUFFERED SIZE OF HUMAN SENSOR DATA

Sensor type	Buffered sensor data size
EMG	$2 \cdot 2\,\mathrm{B} \cdot 10\,\mathrm{ms} \cdot 4000\,\mathrm{Hz} = 160\,\mathrm{B}$
IMU	$2 \cdot 12\,\mathrm{B} \cdot 10\,\mathrm{ms} \cdot 100\,\mathrm{Hz} = 24\,\mathrm{B}$
Piezo	$2\,\mathrm{B} \cdot 10\,\mathrm{ms} \cdot 1000\,\mathrm{Hz} = 20\,\mathrm{B}$

TABLE II
BLE CHARACTERISTIC DATA SIZES

Sensor type	Characteristic data size
EMG	2 B + 4 B + 8 B + 160 B = 174 B
IMU	2 B + 4 B + 8 B + 24 B = 38 B
Piezo	2 B + 4 B + 8 B + 20 B = 34 B

Another factor that affects the amount of overhead is the size of the Link Layer (LL) Data Protocol Data Unit (PDU) payload. If a device tries to send a characteristic that is larger than the LL Data PDU allows, the data will be split into multiple LL packets. Bluetooth 5, the version used in this project, supports a feature called LE Data Packet Length Extension, by which the size of the LL Data PDU payload can be increased from the default of $27\,\mathrm{B}$ to a maximum of $251\,\mathrm{B}$[9]. By making sure that the LL Data PDU payload is large enough to hold both the largest characteristic data along with its headers, additional overhead is avoided by sending fewer packets. For this project, the LL Data PDU payload is set to its maximum value of $251\,\mathrm{B}$. Additionally, the Attribute Protocol (ATT) Maximum Transmission Unit (MTU) determines the largest size that a characteristic can be while still sending it as a single packet. Given an LL Data PDU payload of $251\,\mathrm{B}$, the maximum ATT MTU can be calculated by excluding the size of the Logical Link Control and Adaptation protocol (L2CAP) header and the ATT header, which are $4\,\mathrm{B}$ and $3\,\mathrm{B}$ respectively. This gives a maximum ATT MTU of $244\,\mathrm{B}$, which is what this project uses.

In this project, the Peripheral has one service with three characteristics for the EMG, IMU, and piezo data, respectively. Each characteristic represents $10\,\mathrm{ms}$ worth of samples for all sensors of its sensor type, along with a header, a sequence number, and a timestamp. The total size of the data for each characteristic being sent is shown in Table II. The Central subscribes to notifications from all three characteristics. Once the Peripheral has collected $10\,\mathrm{ms}$ worth of samples for a characteristic, it then sends them to the Central using a notification.

A CAN bus is used to send sensor information from the MCUs to the Jetson AGX Orin and to send motor commands from the Jetson AGX Orin to the MCUs. The bitrate used for the CAN bus is $1\,\mathrm{Mbit\,s^{-1}}$, which is the highest supported bitrate for a CAN bus on the order of $1\,\mathrm{m}$[10]. The types of sensor information being sent are from the IMUs and the potentiometers. The types of motor commands are actuation commands, stop commands, and a grip command. Since the data for an IMU sample is made up of two triplets of 16-bit values, the x, y, and z values for both the accelerometer and the gyroscope, the total amount of data that has to be sent is $12\,\mathrm{B}$. However, the maximum amount of data that can be sent in a single CAN 2.0 message is $8\,\mathrm{B}$, so the data is split up into separate messages for the accelerometer and the gyroscope. The messages for the motor actuation commands and the potentiometer sensor values both contain 32-bit floating point values representing an angle. The grip command contains a single byte boolean while the stop command contains no data. The sizes of the different commands are summarised in Table III.

[9] Bluetooth Core Specification 5.4
[10] `https://www.ti.com/lit/an/slla270/slla270.pdf`

TABLE III
SIZE OF CAN MESSAGE TYPES

Message type	Data size	Message size calculated using (13)
IMU gyro or acceleration	6 B	115 bit
Motor actuation or potentiometer	4 B	95 bit
Hand grip state	1 B	65 bit
Motor stop	0 B	55 bit

7.3.5 Sensor fusion and decision support

The implemented multimodal approach is best described as decision-level fusion rather than a single "all-sensors-into-one-model" pipeline. Each modality contributes a different type of evidence:

- **EMG:** Primary signal for predicting grip initiation and release intention (temporal inference).
- **IMU:** Provides kinematic context such as motion direction and arm orientation.
- **Vision:** Adds environmental context such as object (cup) presence and spatial information.
- **Pressure/force sensing:** Provides interaction feedback to confirm contact and detection slip.

Instead of requiring all modalities to be present at all times, the controller uses the available signals as decision support for state transitions and safety constraints. This approach makes it possible to validate combinations of modalities in isolation while preserving a clear path toward tighter fusion in future iterations.

7.4 Implementation details

This subsection gives a description of the implementation details regarding the vision system and the IMU integration, as well as the fusion of them both.

7.4.1 Vision system

The vision system is implemented using the DepthAI library in Python, which is specific to the DepthAI cameras. The vision system can be split into three sections: depth measurements, cup detection, and pose estimation. The full pipeline can be seen in Figure 8.

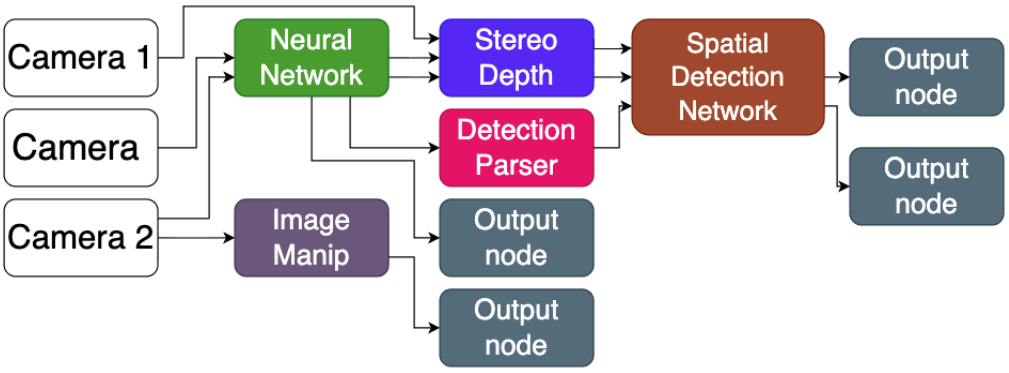


Figure 8: The pipeline graph of the system. Camera 1 and Camera 2 refer to the mono cameras used in the stereo system. Note that all three camera nodes (Camera, Camera 1, and Camera 2) are the same physical OAK-D camera. The Neural Network, Detection Parser, and the Spatial Detection Network are DepthAI-specific nodes that constitute the cup detection. In the DepthAI pipeline syntax, the AprilTag detection does not have its own specific graph node and thus is integrated in the systems that are shown in the figure. The Image Manip node is used to resize the window that is visualising the AprilTag detection.

Using the stereo vision feature of the camera, a depth map is produced. This map can be visualised using a hot color map. It is auto-scaled based on the min/max depth in the scene. The depth measurements are used in the other modules as well to get the position of the cup and the pose.

The cup detection is performed using YOLOv6. The detection is filtered for cups specifically. In combination with the depth map, the cup's coordinates in all three axes can be obtained.

Pose estimation is performed using AprilTags and the predefined AprilTag node in the DepthAI library. The TAG16H5 AprilTag family is used, although the software implementation allows for the use of different families. To estimate the six degrees of freedom pose from the AprilTag detection, the object points and corner points are extracted and fed into a PNP solver that returns the rotation and translation vectors. Fed to the solver is also the intrinsic camera matrix and the distortion coefficients. The translation vector is flattened and stored as the position of the tag. The rotation vector is converted to a quaternion that is stored as the orientation. The final return of the pose estimation algorithm also includes timestamps and the tag id since there can be multiple tags present simultaneously.

7.4.2 IMU implementation and fusion

System-wise, the pose estimation of the arm is performed using a combination of the AprilTag detection with the vision system and data from IMUs. There are two IMUs that are placed on the bionic arm that continuously read the linear and angular acceleration of the arm. The data is combined and fed into a EKF: a 13-state filter with gravity compensation, Jacobian-based linearisation for nonlinear dynamics, Joseph-form covariance update for numerical stability, and quaternion normalisation and sign ambiguity handling. The IMU data is captured at $100\,\mathrm{Hz}$ to predict the estimation of the position and orientation. The vision data is captured at around $30\,\mathrm{Hz}$ and updates the estimation. The final pose is used in the main controller to make decisions about motor control in the next step.

For the IMU in the cuff that is attached to the human arm, the IMU movement direction estimation is implemented as an event-driven Python pipeline that streams inertial data over BLE, performs gravity compensation, and infers dominant motion direction in real time. This is only implemented experimentally and not fully integrated to the system. IMU samples are received asynchronously via BLE notifications, decoded, and converted from raw values to physical units. Linear acceleration is obtained by removing gravity using full orientation estimation using a Madgwick filter with quaternion representation and gyro bias compensation. The gravity-compensated acceleration is temporally integrated to estimate velocity, stabilised using deadband thresholds, decay, and stillness detection. The dominant velocity axis is then identified using a threshold to classify discrete movement directions for higher-level control logic.

8 Results

The individual subsystems, including vision, EMG, and control logic, were each validated independently and demonstrated correct operation. However, during the final phase of closed-loop system integration, a critical hardware fault was identified, preventing complete system validation. The following section presents the performance results of the subsystems that remained operational and describes the observed system behaviour during partial integration testing.

In figure 9, the end result of how the physical product turned out can be seen. The colour scheme was not intended, it was due to lack of available supply.

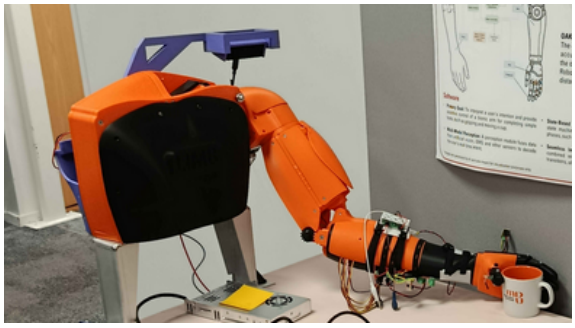


Figure 9: The physical representation of the project. Obviously, the product is incomplete so it appears messy. A better representation of the potential end result appearance is shown in figure 1 and 2

.

8.1 EMG system

Four trials of gripping, recorded with the designed EMG system were used for validation. The duration of each trial was eight seconds, with a resting interval followed by a voluntary contraction of the hand muscles for gripping. The signal was digitised with a sampling frequency of 4 kHz.

8.1.1 Raw EMG signal

Figure 10(a)–(d) shows the raw EMG signal for each trial. At rest, the signal is stable around the reference baseline with low fluctuations. During gripping, the signal amplitude clearly and consistently increases across all trials. The muscle activation is well defined and shows no saturation. This confirms that the analog frontend gain is appropriate and that the system captures muscle activity without distortion.

8.1.2 Validation

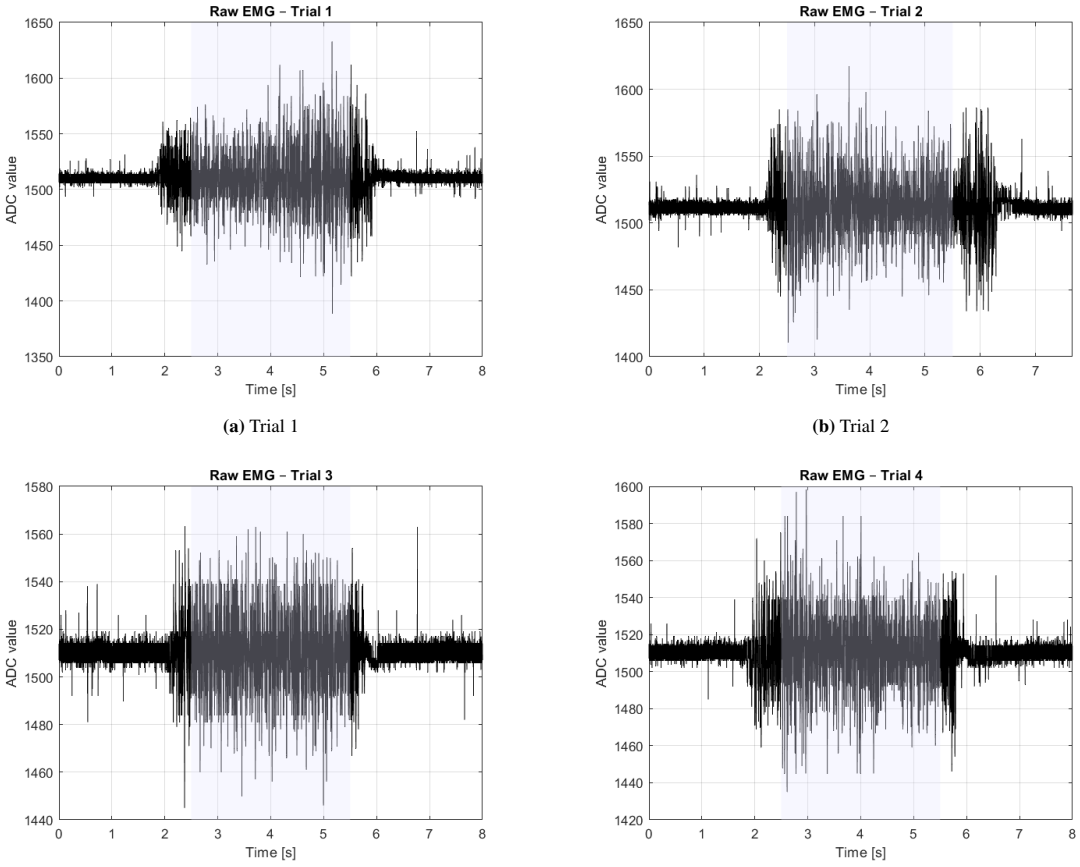
Mean and standard deviation of validations across four trials can be seen in Table IV.

TABLE IV
MEAN AND STANDARD DEVIATION OF VALIDATION METRICS ACROSS FOUR EMG TRIALS

Metric	**Mean ± Std**
RMS-based SNR [dB]	18.32 ± 1.67
Envelope-based SNR [dB]	18.03 ± 1.61
Bandpower ratio (20–1000 Hz) [dB]	18.45 ± 1.72
Baseline noise std [mV]	3.53 ± 0.07
Baseline drift [mV]	0.07 ± 0.37
50 Hz energy/total EMG signal energy (rest) [dB]	-10.64 ± 0.67
50 Hz energy/total EMG signal energy (active) [dB]	-5.24 ± 0.59

For all trials, the value for the RMS-based SNR is $18.32 \pm 1.67\,\mathrm{dB}$, while the value for the envelope-based SNR is $18.03 \pm 1.61\,\mathrm{dB}$. These values can be seen in Figure 11 and show that the muscular signal during gripping is several orders of magnitude stronger than the background noise during rest.

The small variation between the trials indicates that the system is consistent. That the two methods show nearly identical results supports the assumption that the outcome is independent of the analysis technique.



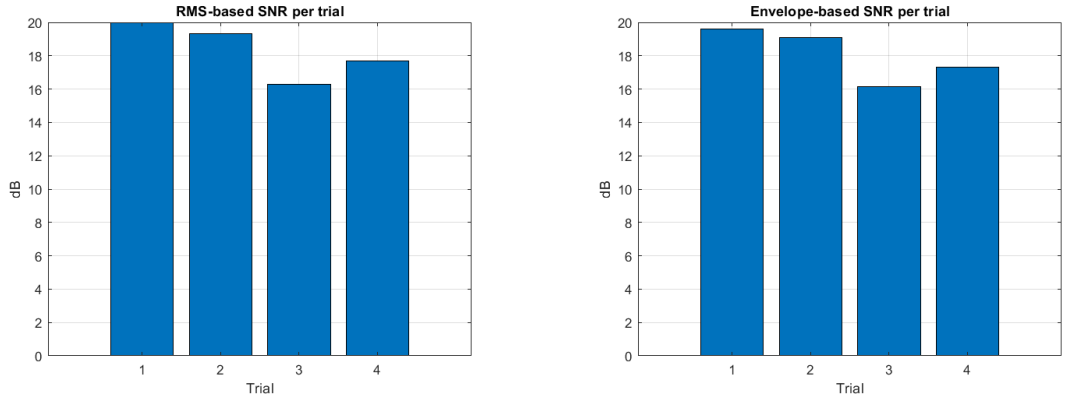
(a) Trial 1

(b) Trial 2

(c) Trial 3

(d) Trial 4

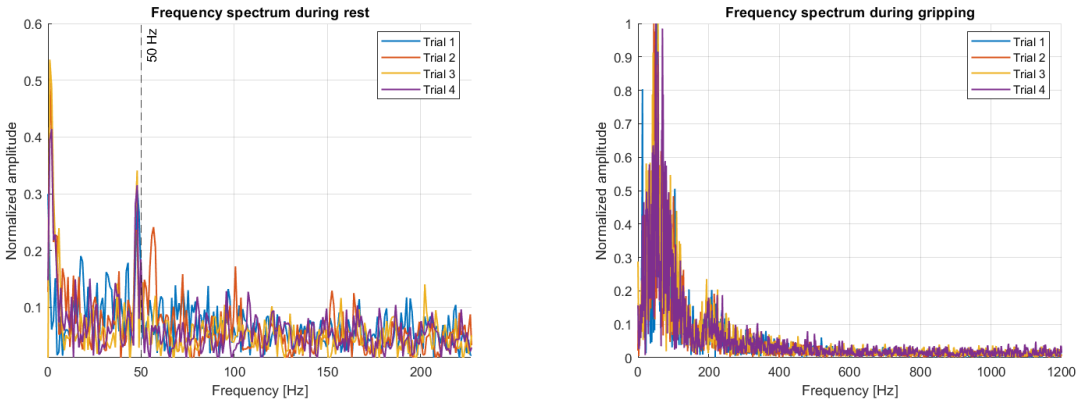
Figure 10: Raw EMG signals for four gripping trials. The shaded region indicates the active gripping phase that was used in validation.



(a) RMS-based SNR

(b) Envelope-based SNR

Figure 11: Comparison of signal-to-noise ratio representations. (a) RMS-based SNR computed from the bandpass-filtered EMG signal. (b) Envelope-based SNR computed from the RMS envelope, providing a smoothed activation-level representation.



(a) Rest

(b) Gripping

Figure 12: Normalized FFT magnitude spectra of the EMG signal during rest and gripping.

When no muscle activity is present, the EMG signal should remain stable over time. Two measures were used to evaluate this.

The baseline noise level, measured as the standard deviation during rest, was $3.53 \pm 0.07\,\mathrm{mV}$, indicating a low and stable noise floor. Baseline drift, which describes slow changes in the signal level over time, was $0.07 \pm 0.37\,\mathrm{mV}$.

These results show that the signal baseline does not vary much, which is significant when considering detectability within muscle activity.

Under the resting condition, the frequency components are much lower. There is a small peak at $50\,\mathrm{Hz}$ due to electrical interference from power lines, Figure 12(a). However, its strength is low compared to the corresponding signal related to the muscle activity of gripping.

The frequency spectra appear consistent across all trials, which means consistent behavior.

To further quantify the frequency-domain behavior, the amount of signal energy within the typical EMG frequency range was compared between rest and gripping Figure 13. The resulting bandpower ratio was $18.45 \pm 1.72\,\mathrm{dB}$.

This result confirms that the increase in signal energy during gripping occurs in the frequency range expected for muscle activity, rather than being caused by slow drift or narrow electrical interference.

During rest, the mains interference was $10.64\,\mathrm{dB}$ lower than the EMG signal. During gripping, it remained $5.24\,\mathrm{dB}$ lower. Negative values indicate that the interference is weaker than the muscle signal.

This shows that while some environmental noise is present, it does not dominate the measurement and is effectively suppressed by the system design.

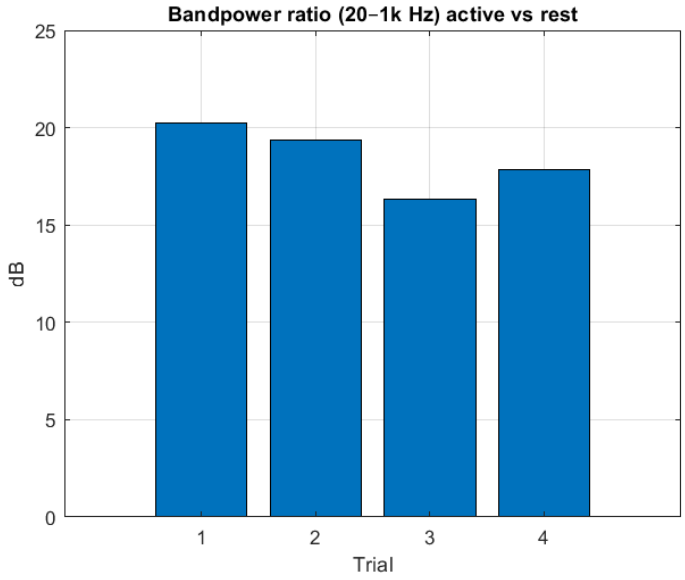


Figure 13: Bandpower ratio between gripping and rest computed in the EMG frequency band (20–1000 Hz). The ratio indicates a increase in signal energy during muscle activation compared to rest.

8.2 Intent recognition results

The EMG-based grip-intention classifier was trained using a labeled dataset consisting of repeated rest-grip-rest sequences collected during controlled grasping tasks. A LSTM model was trained to distinguish between grip and release intentions based on temporally segmented EMG features.

Table V summarises the training and validation performance over 100 epochs. Training accuracy increased from 85.5% in the first epoch to 96.0% at epoch 100, while the corresponding training loss decreased monotonically from 0.3625 to 0.1159. A similar trend was observed for the validation data, where validation accuracy improved rapidly during early epochs and stabilized above 94.0% after approximately 30 epochs.

TABLE V
TRAINING AND VALIDATION PERFORMANCE OVER EPOCHS

Epoch	Train Acc	Train Loss	Val Acc	Val Loss
1	0.855	0.3625	0.890	0.2827
10	0.932	0.2070	0.924	0.2239
20	0.938	0.1890	0.938	0.1843
30	0.941	0.1754	0.942	0.1764
40	0.945	0.1648	0.939	0.1827
50	0.948	0.1553	0.942	0.1736
60	0.951	0.1446	0.941	0.1746
70	0.952	0.1361	0.940	0.1791
80	0.956	0.1295	0.942	0.1837
90	0.957	0.1243	0.944	0.1801
100	0.960	0.1159	0.945	0.1836
Best Val	0.957	0.1306	**0.948**	**0.1718**

The highest validation accuracy of 94,8% was achieved during training, with a corresponding validation loss of 0.1718. The close alignment between training and validation accuracy across epochs indicates good performance on the collected dataset. Validation loss remained within a narrow range after convergence, further supporting training stability.

In addition to quantitative performance, the classifiers exhibited temporally consistent predictions during sustained grip and rest phases. Misclassifications were primarily observed during short transition intervals between rest and grip or where the data were particularly noisy due to EMG amplifier glitch. Outside these transition regions, predicted intention states remained stable across consecutive time windows.

Overall, these results demonstrate that the LSTM-based EMG classifier can reliably infer grip intention from surface EMG signals in a controlled experimental setting, fulfilling the performance requirements defined for intention recognition within the proposed multimodal pipeline.

8.3 Vision-based cup detection results

The performance of the vision subsystem was evaluated using two controlled recording scenarios designed to assess detection stability and robustness under stationary conditions. In both scenarios, the camera and target object remained fixed throughout the recording, and detections were performed continuously using a YOLO-based object detector within the DepthAI framework.

Stream A (only cup): In the first scenario, a single cup was placed within the camera's field of view and remained fully visible and stationary for the entire recording duration, placed at $x = -12\,\mathrm{mm}$, $y = 78\,\mathrm{mm}$, and $z = 297\,\mathrm{mm}$. This is shown in Figure 14.

The x and y coordinates are the position in the window frame, and the z coordinate is the distance from the camera to the cup. A summary of the results can be seen in Table VI, and the following results can be found in this table.

Over a 30-second recording interval, a total of 2383 detections were registered, corresponding to an average detection rate of 79.43 detections per second. The first valid detection occurred at $0.41\,\mathrm{s}$, after which detections were maintained continuously until the end of the recording window. The confidence statistics for this stream showed an average score of 0.638 with a minimum of 0.507 and a maximum of 0.836. A confidence threshold was set at 0.6, and the cup was not considered detected below this threshold. These values indicate consistent but moderately varying classification confidence across frames. Despite this

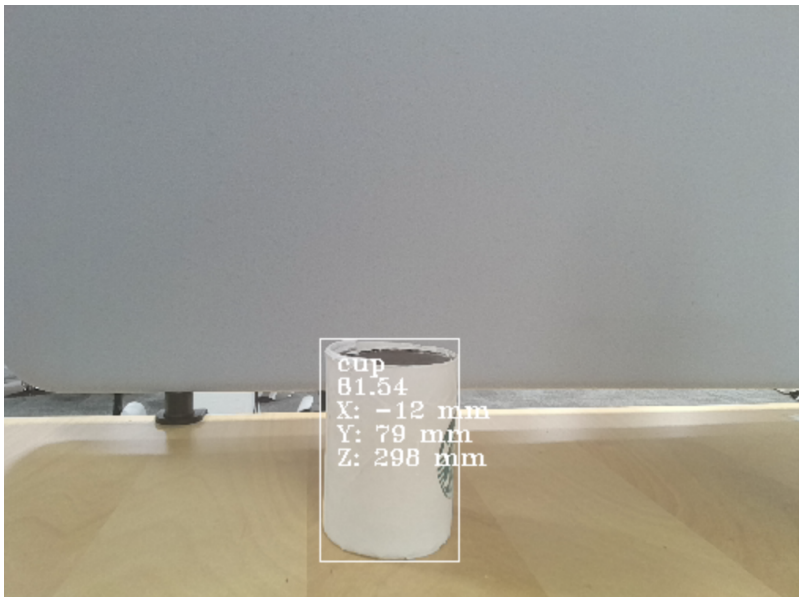


Figure 14: Frame from a stream of the vision system where the cup is detected. In this frame, only the cup is visible. The white bounding box of the detection is shown as well as the class (cup), confidence of detection (61.24), and the x, y, and z coordinates of the cup in mm.

variation, detections were sustained throughout the recording without extended dropouts. Spatial estimates derived from stereo depth measurements remained bounded over time. The average detected position of the cup was $-11.8\,\mathrm{mm}$ along the x-axis, $77.1\,\mathrm{mm}$ along the y-axis, and $295.0\,\mathrm{mm}$ along the z-axis. The observed ranges were limited to $[-12, -5]$ mm in x, $[36, 83]$ mm in y, and $[145, -325]$ mm in z. This variation reflects frame-to-frame estimation noise and depth quantization effects rather than loss of object tracking.

Stream B (hand and cup): In the second scenario, a human hand grasped the cup while both the hand and cup remained stationary and fully visible to the camera. This can be seen in Figure 15.

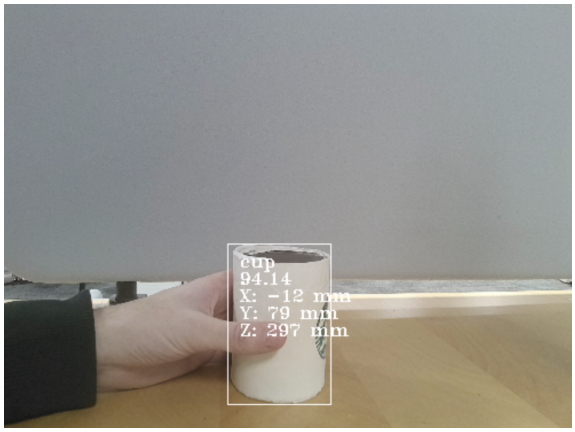


Figure 15: Frame from a stream of the vision system where the cup is detected. In this frame, the cup is grasped by a hand. The white bounding box of the detection is shown as well as the class (cup), confidence of detection (94.14), and the x, y, and z coordinates of the cup in mm.

This stream was intended to evaluate detection behavior under partial visual complexity introduced by the presence of the hand. During the 30-second recording, a total of 2425 detections were recorded, yielding an average detection rate of 80.83 detections per second. The first detection occurred at $0.42\,\mathrm{s}$, and detections were maintained continuously until the end of the recording interval. Compared to the stationary-cup-only scenario, confidence scores were significantly higher and more consistent. The average confidence was 0.942, with a minimum of 0.934 and a maximum of 0.947. This indicates highly stable classification confidence despite the additional visual structure introduced by the hand. The spatial estimates

also exhibited reduced variation. The mean detection position was $-11.1\,\mathrm{mm}$ along the x-axis, $76.5\,\mathrm{mm}$ along the y-axis, and $296.5\,\mathrm{mm}$ along the z-axis. The corresponding ranges were $[-13, -11]\ \mathrm{mm}$ in x, $[75, 84]\ \mathrm{mm}$ in y, and $[294, 330]\ \mathrm{mm}$ in z, indicating tighter clustering of depth-based position estimates across frames.

TABLE VI
VISION-BASED CUP DETECTION PERFORMANCE UNDER STATIONARY CONDITIONS

Metric	**Stream A: Cup only**	**Stream B: Hand grasping cup**
Recording duration [s]	30.0	30.0
Total detections	2383	2425
Detection rate [detections/s]	79.43	80.83
First detection time [s]	0.41	0.42
Average confidence	0.638	0.942
Minimum confidence	0.507	0.934
Maximum confidence	0.836	0.947
Mean X position [mm]	-11.8	-11.1
X position range [mm]	$[-12, -5]$	$[-13, -11]$
Mean Y position [mm]	77.1	76.5
Y position range [mm]	$[36, 83]$	$[75, 84]$
Mean Z position [mm]	295.0	296.5
Z position range [mm]	$[145, 325]$	$[294, 330]$

Across both streams, the vision system demonstrated stable real-time operation with sustained detection rates close to 80 detections per second. Object detections were maintained continuously once initialised, and spatial estimates remained bounded over time in both scenarios. The presence of a grasping hand did not degrade detection performance, it coincided with higher and more stable confidence scores and reduced spatial variation. These results indicate that the vision subsystem provides temporally dense and spatially consistent object detections under controlled conditions, supporting its role as a reliable source of environmental context within the proposed multimodal grip-intention recognition pipeline.

8.4 IMU direction estimation results

The IMU-based direction estimation module was evaluated using a controlled, sequential motion test designed to assess whether distinct movement directions could be reliably inferred from inertial measurements. During the experiment, the user wore the cuff containing the IMU and performed a predefined sequence of arm movements in the following order: up, down, back, forward, left, right, repeated twice consecutively.

The movements were performed in one direction at a time, with short pauses between each motion to allow the system to reset and register a new direction. The cooldown timer was 5 seconds. The detected directions were logged in real time and visualised alongside the instantaneous velocity components along the three axes. The interface is shown in Figure 16. The detected directions followed the commanded order consistently across both repetitions, with no missing observed during the test. This indicates that the direction-estimation logic correctly mapped the dominant velocity components to discrete directional labels. Overall, the results demonstrate that the IMU-based direction estimation module can reliably capture coarse hand movement directions in a structured and repeatable motion sequence. While the test does not quantify angular accuracy or robustness under complex trajectories, it confirms that the IMU provides meaningful directional information suitable for use as contextual input within the proposed multimodal intent recognition framework.

8.5 Mechatronic Kinematics and Actuation

The mechanical performance of the bionic arm was evaluated through a series of controlled motion and manipulation tests using preplanned motor actuation sequences. The mechanical arm was evaluated independently to assess its functional capabilities. All five DOFs of the arm were successfully actuated and demonstrated repeatable motion across multiple trials. Joint movements consistently reached their commanded positions, indicating reliable mechanical behaviour and sufficient structural rigidity for the intended tasks. The arm's effective workspace was intentionally constrained via software-defined joint limits.

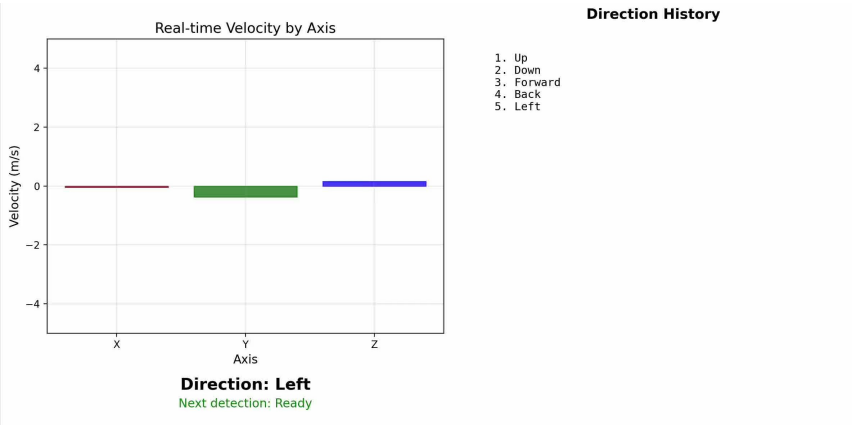


Figure 16: A snapshot from the IMU direction test where the y-axis is the dominant axis and the visualisation depicts left as the identified direction on the left side. The right side of the interface shows the logged history of the directions that were previously identified.

The limits were implemented as a safety measure to prevent structural parts from acting as mechanical end stops, thereby reducing the risk of damage. With these predefined limits, no other mechanical constraints were observed, and all the joints operated smoothly within their range of motion. If the arm was hanging freely, minor positional drift was observed in some joints. Additionally, mechanical backlash was observed in the humeral rotation joint. At low movement speeds, the joints controlled by the NEMA-17 steppers exhibited minor vibrations, primarily due to the use of steppers running without microstepping, which resulted in discrete motion steps.

The arm was further tested under load lifting and moving a cup by grabbing it with the attached hand, which weighed around 200 grams. The arm moved and held the cup without mechanical instability, with smooth movement when transferring the cup from one place to another. The integration between the arm and hand remained stable during grasping and manipulation tasks. The arm was mounted on the InMoov torso, which provided sufficient stability during all tested motions. The camera mount serves as a functional integration into the system, but is left unused due to a non-functional complete system.

The modular design proved beneficial during testing and calibration, especially during disassembly for redesigning structural parts and when adjusting potentiometer placement to ensure the desired joint range of motion.

In addition to the mechanical evaluation, the full mechatronic performance of the arm was assessed by observing actuator response, sensor feedback, and electrical reliability during operation. All of the actuators responded consistently to repeated command inputs during testing, no missed steps or undesirable direction changes was observed. The potentiometers joint feedback system worked moderately, at higher joint speeds the joint angle would surpass the targeted joint angle, this is proved to be an issue with the reading speed of the MCUs ADC. However, after the MCU notices that it has reached its target angle, it would change direction and continue toward it. The difference between the joint's current angle and the target angle is small, so when it observes its fault, it does not accelerate fast enough to miss the target angle a second time. From an electrical integration perspective, no critical power-related issues were observed during normal operation. During testing and development, the USB cable used to flash software onto the MCU powered the system via the 5V pin, even when the main power source was disabled. The wiring and electrical integration were generally robust and posed no significant challenges. However, coming the end of the project, the CAN-bus lost its functionality. Additionally, the custom PCBs (nodes) would be larger than intended to fit inside the arm structure.

The mechatronic performance of the hand was evaluated during repeated grasping and object manipulation tasks. The hand consistently opened and closed without electrical issues, and all fingers moved synchronously during grasping. The pressure sensors in the hand responded consistently during grasping, providing stable feedback as the object was held. Grip behaviour was consistent during trials, with minor delays in finger movement. These delays would not prevent successful object manipulation. Integrating the hand to arm did not show any noticeable sign of disturbance in the movement.

Overall, the arm and hand mechatronic system showed stable and predictable behaviour during manipulation tasks, supporting reliable grasping, lifting and transportation of the cup.

TABLE VII
LL PACKET SIZES AND TRANSMISSION DELAYS USING 2M PHY

Packet type	Packet size	Transmission delay
EMG	$174\,\mathrm{B} + 7\,\mathrm{B} + 2\,\mathrm{B} + 9\,\mathrm{B} = 192\,\mathrm{B}$	$192\,\mathrm{B}/2\,\mathrm{Mbit\,s^{-1}} = 768\,\mu\mathrm{s}$
IMU	$38\,\mathrm{B} + 7\,\mathrm{B} + 2\,\mathrm{B} + 9\,\mathrm{B} = 56\,\mathrm{B}$	$56\,\mathrm{B}/2\,\mathrm{Mbit\,s^{-1}} = 224\,\mu\mathrm{s}$
Piezo	$34\,\mathrm{B} + 7\,\mathrm{B} + 2\,\mathrm{B} + 9\,\mathrm{B} = 52\,\mathrm{B}$	$52\,\mathrm{B}/2\,\mathrm{Mbit\,s^{-1}} = 208\,\mu\mathrm{s}$
Empty	$2\,\mathrm{B} + 9\,\mathrm{B} = 11\,\mathrm{B}$	$11\,\mathrm{B}/2\,\mathrm{Mbit\,s^{-1}} = 44\,\mu\mathrm{s}$

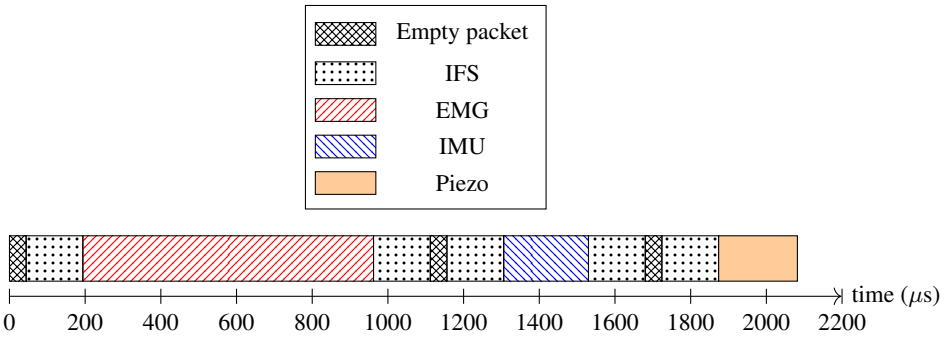


Figure 17: Theoretical scheduling of LL packets during a connection event.

8.6 Theoretical latencies and bandwidth in communication

The communication in the project has deadlines and bandwidth limits. A theoretical analysis of the data being sent was performed to validate the viability of the system.

8.6.1 BLE

The amount of data being sent for each characteristic is shown in Table II. This data is then encapsulated in an L2CAP PDU, adding $7\,\mathrm{B}$ of headers[11]. Then the L2CAP PDU is encapsulated in a LL Data PDU, which adds a $2\,\mathrm{B}$ header. Finally, the LL Data PDU is encapsulated in an LL packet, adding a final $9\,\mathrm{B}$ of headers. Empty LL packets do not add as many headers since they do not contain a payload in its Data Physical Channel PDU. Using the gross bit rate of $2\,\mathrm{Mbit\,s^{-1}}$ used by the 2M PHY, the transmission delay for the packets can be then be calculated. The resulting sizes and transmission delays for the different packets can be seen in Table VII.

Assuming that the Peripheral sends the packets in the order of EMG, IMU, and finally the piezo, the theoretical scheduling for a connection event is shown in Figure 17, with the Central initiating the communication before interleaving the transmission of packets between the Peripheral and Central. Each packet transmission is followed by Inter Frame Spacing (IFS), and the Central only sends empty packets since it has no data to send to the Peripheral. The resulting amount of packets is three empty packets, five IFS packets, and one packet each for the EMG, IMU, and piezo respectively. The time it takes to send all packets in a connection event is

$$3 \cdot 44\,\mu\mathrm{s} + 5 \cdot 150\,\mu\mathrm{s} + 768\,\mu\mathrm{s} + 224\,\mu\mathrm{s} + 208\,\mu\mathrm{s} = 2.082\,\mathrm{ms}. \tag{11}$$

Since the connection event lasts for $10\,\mathrm{ms}$, the packets should theoretically keep their deadline.

8.6.2 CAN

Standard CAN frames have 44 bits worth of headers, 34 of which are subject to bit stuffing[12]. Assuming the worst case where the maximum number of bits are stuffed, the calculation for the number of stuffed bits used in a CAN frame with n bytes of data is given by (12).

$$\text{Maximum possible bit stuffing} = \left\lfloor \frac{34 + 8n - 1}{4} \right\rfloor. \tag{12}$$

[11] Bluetooth Core Specification 5.4
[12] Bosch CAN Specification Version 2.0

TABLE VIII
TOTAL SENSOR BITS SENT PER PERIOD

Message	Bits sent per period
IMU	$6 \cdot 115\,\text{bit} = 690\,\text{bit}$
Potentiometer	$4 \cdot 95\,\text{bit} = 380\,\text{bit}$
Total	$690\,\text{bit} + 380\,\text{bit} = 1070\,\text{bit}$

This means the total size of a CAN frame that includes n bytes of data, headers, the worst case bit stuffing as well as 3 bits of inter-frame spacing is given by the equation

$$8n + 44 + \left\lfloor \frac{34 + 8n - 1}{4} \right\rfloor + 3. \tag{13}$$

The sensor messages are sent every $100\,\text{Hz}$ while the motor command and grip state messages are sent sporadically. The resulting number of bits sent per period from all sensors is given by Tab VIII. Given the frequency of the sensor messages, the minimum bit rate required to be able to handle the messages is $1070\,\text{bit} \cdot 100\,\text{Hz} = 107\,\text{kbit}\,\text{s}^{-1}$. Adding every other sporadic message; which includes ten motor actuation messages, ten motor stop messages, and one grip message; increases the number of bits sent by

$$10 \cdot 95\,\text{bit} + 10 \cdot 55\,\text{bit} + 65\,\text{bit} = 1565\,\text{bit}, \tag{14}$$

which in turn increases the minimum bit rate to $(1070\,\text{bit} + 1565\,\text{bit}) \cdot 100\,\text{Hz} = 263.5\,\text{kbit}\,\text{s}^{-1}$. The maximally delayed message during such a burst of messages, assuming no more sporadic messages are sent during the burst, would then be the hand's IMU acceleration message which has the lowest priority. The message would be delayed by $(2635\,\text{bit} - 115\,\text{bit})/1\,\text{Mbit}\,\text{s}^{-1} = 2.52\,\text{ms}$.

8.7 System Communication Failure

While the individual subsystems were successfully validated, the full closed-loop integration experienced a partial communication failure during the final testing phase. A critical failure occurred on the CAN communication bus linking the NVIDIA Jetson AGX Orin to the distributed ESP32-C3 nodes, while conducted tests confirmed that the internal CAN network between the ESP32-C3 nodes remained functional.

The hand node successfully transmitted pressure sensor data to the lower arm node, triggering the local reflex control loop. These results demonstrate the robustness of the daisy-chained bus topology and the custom SN65HVD232D transceiver design for node-to-node communication. As a result, the arm could execute local low-level reflexes such as grip stabilisation, but was unable to perform the autonomous, vision-guided intent-to-grasp sequence.

The failure did not occur during initial system startup. Instead, communication degradation appeared after a period of operation, indicating a fault that was dependent on time or system load rather than a fixed configuration issue.

A systematic diagnostic process was carried out to rule out common configurations and physical-layer issues. Termination resistors were confirmed to be installed and correctly positioned at both ends of the CAN bus. CAN bitrates were checked across all nodes to ensure uniformity. Ground reference continuity was verified between nodes, and cable lengths and routing were examined to rule out issues related to signal reflection or attenuation. Since the issue persisted after their measures, a more detailed investigation was initiated to examine possible interactions between the power distribution architecture and the communication subsystem. This analysis is presented in Section 9.7

9 Discussion

In this section, the results obtained from the individual subsystems are interpreted and discussed in relation to the project objectives, system design choices, and observed performance.

9.1 EMG system

From the validation results, the EMG system is sufficient to detect muscle activation when gripping. Based on the data obtained from the EMG signal for the 2×2 comparison shown in Figure 10, the signal increases significantly without saturating when the muscles are contracted compared to the resting position. This implies that the gain for the analog front-end is properly set.

The quantitative analysis with SNR is also consistent with these observations. Both RMS and envelope SNR are around $18\,\mathrm{dB}$, and there is an insignificant difference among trials. This shows that muscle activity during gripping is significantly larger than the noise during rest conditions. The fact that both RMS and envelope SNR are nearly equal shows that signal quality is independent of whether it is based on electrical or activation level.

Baseline stability is also a requirement of reliable EMG measurements. Low baseline noise during rest is measured, which shows a stable noise floor. The average value of baseline drift comes close to zero, showing that slow variations in signal level over time are minimal. These two findings together show that the system provides a stable reference level against which muscle activity can be detected.

The frequency-domain analysis also confirms that the recorded signals show characteristics expected of surface EMG. During gripping, the spectrum reveals broadband energy distributed across a wide frequency range, which reflects asynchronous motor unit activity, while during the resting condition, the spectral energy is low with only a narrow peak at $50\,\mathrm{Hz}$ due to power-line interference. This can be seen in Figure 12. The magnitude of this peak with respect to the overall signal energy suggests that environmental electrical noise does not dominate the measurement.

This is supported by the bandpower comparison and the ratios between the $50\,\mathrm{Hz}$ noise during rest and the total signal energy. The negative ratios for the rest and gripping states further verify that the power line signal is less dominant than the EMG signal.

9.2 Intent recognition

The results regarding the intent recognition demonstrate that EMG-based intention recognition can achieve stable and accurate classification of grip and release events under controlled conditions. The high validation accuracy observed during model training indicates that temporal patterns in EMG signal are well captured by the LSTM architecture. Although the classification is binary and there is a clear distinction between gripping and resting (see Figure 10), it is not surprising that a relatively complex model can distinguish between the two classes. Actually, a simple mean value average could probably identify the difference between gripping and resting. The choice of an LSTM model could then be considered excessive, but if the task were more complex, it might have been more convenient to use such an architecture. Examples of complex tasks could include classifying the transitional state between gripping and resting, or classifying different kinds of grips. The EMG circuit and amplifier that was designed and produced were too good to utilise the features of the LSTM.

While the classification performance is promising, the results also highlight limitations in generalisation and real-world deployment. The EMG model was evaluated primarily on offline or semi-online data, and its performance under full system integration remains partially unvalidated. Signal variability due to electrode placement, muscle fatigue, and user-specific differences may further impact robustness.

These results suggest that EMG alone can reliably indicate grip intention under controlled conditions, but multimodal fusion is required to ensure robustness and generalisation in realistic, variable-use scenarios.

9.3 Vision-based cup detection

The vision system results show that object detection and positional estimation using the DepthAI pipeline are feasible and sufficiently accurate for structured manipulation tasks such as cup grasping. YOLO-based cup detection, combined with stereo depth, enabled reliable estimation of object position in three-dimensional space. However, the controlled environment in which the test was conducted may not be representative of the environment in which the arm and vision system are supposed to operate when fully

integrated. Also, no test was performed with the hand in front of the camera, so the cup was barely visible. There is reason to think that cup detection would have been worse in that case, and the need to fuse with other sensors that could complement the camera's occlusions is thus important. It is also not possible to evaluate how the vision system behaves under different lighting conditions, camera placements, or other occlusions. It is also unclear whether the AprilTag implementation sufficiently supports and complements the vision system as intended, and more tests are needed to evaluate this fusion. Still, there is promise that a vision system could provide valuable information as to the position and orientation of the arm and cup, but perhaps not as a stand-alone system. A way to enhance the vision system's performance is to add a second camera from a different angle, which would minimise occlusions and complement the first camera. Although this may require controlled environmental conditions, which may not be feasible long-term if the bionic arm were used in a clinical setting. It would also be interesting to investigate whether it is possible to have a vision system that, from its camera, maps the movement of the human arm to the space and movement of the bionic arm.

The vision results highlight its value for spatial context, while also demonstrating that occlusions and environmental variability necessitate complementary sensing modalities to maintain reliable grip-intention recognition.

9.4 IMU direction estimation

The IMU-based movement estimation results indicate that inertial sensing can reliably capture short-term motion direction and dynamic arm behavior when appropriate gravity compensation and filtering are applied. The implemented gravity-removal strategies, particularly the Madgwick-based orientation estimation, enabled transforming the sensor frame-acceleration representation into a meaningful world-frame representation, allowing the dominant movement direction to be inferred. However, the results also confirm well-known challenges associated with inertial sensing, including integration drift and sensitivity to noise. While velocity-based direction estimation proved effective for discrete motion events, long-term position estimation was not reliable without external corrections. Regarding the system, at this time, the still-detection and corresponding correction remain viable strategies for the proof-of-concept functionality of the bionic arm. Although the human-like movement is compensated, the arm cannot seemingly follow the movement of a human arm, which could be a requirement in future iterations of the system. Albeit the results validate the design decision to use IMU data primarily for relative motion cues and high-frequency updates, relying on vision for absolute pose correction could be an option. Fused together, these results support the complementary role of IMUs within the multimodal system rather than their use as a stand-alone localisation solution.

From an overall system perspective, the IMU implementation faces challenges in coordination due to multiple IMUs attached to the bionic arm. Ideally, more IMUs would give a greater indicative picture of the position and orientation of the arm, but at the cost of complexity. However, due to the ease of implementation and use of IMUs, they are still viewed as a viable choice. Using a 9 DOF IMU instead of 6 degrees could potentially make the system more robust and should be considered in future iterations and work.

IMU-based direction estimation provides high-rate motion cues, but drift and coordination challenges reinforce the need for multimodal fusion to support robust intention recognition during functional tasks.

9.5 Mechatronic kinematics and actuation

The results of the arm indicate that the mechanical and mechatronic design choices were sufficient to achieve reliable basic manipulation. The repeatable actuation of all five DOFs showed that the underlying kinematic structure and distributed control strategy were appropriate for the intended application. The software-defined joint limits displayed an effective method of protecting structural parts from acting as mechanical end stops. This is particularly important due to the chosen material of PLA, where continuous impact could lead to accelerated wear or direct damage. While this reduced the reachable workspace, it increased overall system robustness during testing. Additionally, the workspace of the arm was already larger than required, so reducing the workspace by a small amount has little to no impact to the requirements.

The observed positional drift and backlash highlight limitations in the mechanical transmission, particularly under gravitational load. The backlash was especially noticeable in the humeral rotation node, the mechanical transmission that was used in this joint was a worm gear, at a later stage of the project it was observed that the worms pitch and the gears teeth did not match as well as seen in SolidWorks. One option

to reduce backlash is to increase the tooth size or decrease the pitch. However, these effects did not prevent successful manipulation of the arm. The inability to get the stepper motors to perform microstepping introduced discrete movement at low speed, resulting in reduced smoothness. This represents a trade-off between implementation simplicity and motion quality. While the arm does not need perfect smoothness, future improvements would include implementing microstepping. The joint overshooting at higher joint speeds suggests a limitation in sensor sampling rather than mechanical instability. The system's ability to recover from overshoot indicates that the control strategy was stable but limited by hardware. This limitation especially affects the speed at which the joints can move, but it does not break their functionality. The late CAN-bus failure, which prevented the mechatronic system from being tested as complete, underscores the sensitivity of distributed systems to reliable communication and emphasises the need for earlier system-level integration testing.

The hand subsystem demonstrated reliable sensing and actuation, and the integration with the arm did not introduce any mechanical or control issues. This shows that the coupling between the two mechatronic subsystems was well balanced and suitable for coordinated manipulation.

These results from the mechatronic system indicate that some subsystems within the arm achieved their primary functional objectives. It serves as a good foundation for further development. The identified limitations primarily affect smoothness and precision rather than basic functionality, suggesting targeted improvements rather than a fundamental redesign.

9.6 Theoretical analysis of communication

The result of the theoretical analysis for the BLE communication in section 8.6.1 show that, given ideal conditions, the BLE packets should be able to be sent within $2.082\,\mathrm{ms}$, which means they should be able to comfortably meet their deadlines of $10\,\mathrm{ms}$. However, since notifications are used, it means that the data transmission is not reliable, meaning that packets can be lost. This could introduce problems when trying to control the robot arm in an environment where the signal quality is low. However, notifications are still considered to be preferable to using indications since retransmissions introduce latency to the packets, which could cause them to miss their deadlines anyway, and would also introduce further latency to all packets that are queued during the retransmission.

Similar to the BLE communication, the result of the theoretical analysis for the CAN communication in section 8.6.2 also shows that messages should be able to be handled such that they do not miss their deadline of $10\,\mathrm{ms}$. The rate of periodic messages should be constant, so the only variable that changes and could affect the quantity of traffic is the sporadic messages. During normal operation, the frequency of motor messages should be small such that there are never more than one message each during one $100\,\mathrm{Hz}$ period. The amount of extra bits that would have to be transmitted in one period to saturate the bandwidth would be $10\,000\,\mathrm{bit} - 2635\,\mathrm{bit} = 7365\,\mathrm{bit}$, which is equivalent to $7365\,\mathrm{bit}/95\,\mathrm{bit} \approx 78$ actuation messages, which is large enough that the system should work without issue.

9.7 Root Cause Analysis of Communication Failure

Based on the observed failure behaviour and the elimination of common configuration and physical-layer faults, the most likely root cause was identified as an unintended interaction within the power distribution architecture rather than a protocol or firmware-level defect. During system debugging, it was found that supplying power to one ESP32-C3 node through its USB interface from an external PC caused partial back-powering across the system, affecting other ESP32-C3 nodes and actuators. This observation showed that the PDU lacked reverse current protection, which allowed current to flow upstream through unintended paths. Such reverse current flow can introduce uncontrolled voltage levels and transient conditions on the shared power rail, especially those supplying logic-level electronics and communication interfaces. Since the CAN transceivers were powered directly from the ESP32-C3 logic rails, disturbances from reverse-fed power paths could disrupt transceiver operation, biasing, or common-mode voltage levels. This interaction likely explains why CAN communication worked between ESP32-C3 nodes but failed at the Jetson interface, which used a separate power supply.

Although the precise electrical mechanism could not be conclusively isolated within the project timeframe, the observed behaviour strongly suggests that insufficient power domain isolation and the lack of reverse current protection contributed to unstable operating conditions for the CAN transceivers during extended system operation.

10 Conclusions and Future Work

This project investigates the feasibility of a multimodal sensing and control architecture for intent-driven robotic assistance in upper-limb rehabilitation. The LIMB platform integrates EMG-based intent recognition within a distributed mechatronic system. The results demonstrate that a multimodal approach can reliably capture complementary aspects of user intention and environmental context, providing a strong foundation for intuitive, context-aware robotic assistance.

It is clear from these EMG results that there is sufficient muscle activity, lack of noise, baseline activity, and frequency information preserved to conclude that this system is robust, so that intention and sensor fusion are possible.

The EMG subsystem showed that grip and release intentions can be classified with high accuracy under controlled conditions, confirming that surface EMG is a reliable indicator of user intent for simple functional tasks. However, the results also highlighted limitations in generalisation and robustness when EMG is used in isolation, particularly with respect to inter-user variability, electrode placement, and long-term signal drift. These findings support the central idea of this work: that EMG alone is insufficient for robust real-world deployment, and that multimodal fusion is necessary to achieve reliable and adaptive behaviour.

The vision system demonstrated reliable cup detection and spatial estimation under stationary and structured conditions, providing valuable environmental context for grasping tasks. Similarly, the IMU-based direction estimation successfully captured short-term motion direction when appropriate filtering and gravity compensation were applied. While each sensing modality exhibited limitations when evaluated independently, their combined use addresses complementary weaknesses. Vision contributes spatial awareness, IMU provides dynamic motion context, and EMG supplies direct information about user intent. Together, these results answer the research question by showing that multimodal sensor fusion improves the robustness and interpretability of grip-intention recognition compared to single-modality approaches.

For future system integration, it is important to focus on improving power-domain isolation and ensuring robust communication between subsystems. The PDU should include reverse-current protection, individual rail fusing, and transient suppression to avoid back-feeding and protect each subsystem. Separating the CAN transceivers from shared logic rails or using isolated transceiver stages can also improve communication stability, specifically under varying loads. These measures will increase the overall reliability during extended operation and are necessary for developing a practical rehabilitation platform.

This work is limited to a proof-of-concept evaluation conducted in a controlled laboratory environment and does not assess clinical effectiveness or long-term usability. Future work should therefore focus on full real-time multimodal fusion, closed-loop control with human-in-the-loop evaluation, and systematic testing under variable environmental conditions. Expanding the EMG dataset to include multiple users, grip types, and transitional states would enable more advanced intent models and better generalisation. Additionally, improving vision robustness under occlusion and integrating adaptive learning mechanisms could further enhance system reliability. Ultimately, these developments would move the LIMB platform closer to practical deployment in rehabilitation settings and contribute to more effective, user-centered robotic assistance.

11 Acknowledgment

The authors would like to express our sincere appreciation to our supervisor and stakeholder, Martin Ekström and Waqas Khalid, respectively, for their guidance, technical input, and continuous support throughout the course of the project.

We would also like to thank the collaboration teams from Universidad de Antioquia, Colombia, and Universidad Tecnológica de Panamá, Panama, for their valuable input, feedback, and collaboration throughout the project.

We further thank Harley Berglund, Rasmus Lindblom, and Ted Sköld, students at Mälardalen University for their valuable assistance with the 3D modeling and mechanical design of the system. In addition, we would like to thank Paul Briand from EFREI, Paris-Panthéon-Assas University for his significant contributions to the development of the simulation environment, as well as for his support and encouragement throughout the project.

Finally, we would like to express our gratitude to Thomas Birath and Carolina Sydén at Würth Elektronik for supplying essential electronic components that supported the development and testing of the prototype.

References

[1] V. L. Feigin, M. Brainin, B. Norrving, S. O. Martins, J. Pandian, P. Lindsay, M. F Grupper, and I. Rautalin, "World Stroke Organization: Global Stroke Fact Sheet 2025," *International Journal of Stroke*, vol. 20, pp. 132–144, Feb. 2025.

[2] M. Amirbekova, T. Kispayeva, A. Adomaviciene, L. Eszhanova, I. Bolshakova, and Z. Ospanova, "Systematic review and meta-analysis of effectiveness of robotic therapy in the recovery of motor functions after stroke," *Frontiers in Human Neuroscience*, vol. 19, p. 1622661, 2025.

[3] D. Quadrelli, M. Canepa, D. Di Domenico, N. Boccardo, M. Chiappalone, and M. Laffranchi, "Advances in HD-EMG interfaces and spatial algorithms for upper limb prosthetic control," *Frontiers in Neuroscience*, vol. 19, Sept. 2025.

[4] B. Abdikenov, D. Zholtayev, K. Suleimenov, N. Assan, K. Ozhikenov, A. Ozhikenova, N. Nadirov, A. Kapsalyamov, B. Abdikenov, D. Zholtayev, K. Suleimenov, N. Assan, K. Ozhikenov, A. Ozhikenova, N. Nadirov, and A. Kapsalyamov, "Emerging Frontiers in Robotic Upper-Limb Prostheses: Mechanisms, Materials, Tactile Sensors and Machine Learning-Based EMG Control: A Comprehensive Review," *Sensors*, vol. 25, June 2025. Company: Multidisciplinary Digital Publishing Institute Distributor: Multidisciplinary Digital Publishing Institute Institution: Multidisciplinary Digital Publishing Institute Label: Multidisciplinary Digital Publishing Institute.

[5] K. E. Lee, M. Choi, and B. Jeoung, "Effectiveness of Rehabilitation Exercise in Improving Physical Function of Stroke Patients: A Systematic Review," *International Journal of Environmental Research and Public Health*, vol. 19, p. 12739, Oct. 2022.

[6] J. H. Villafañe, G. Taveggia, S. Galeri, L. Bissolotti, C. Mullè, G. Imperio, K. Valdes, A. Borboni, and S. Negrini, "Efficacy of Short-Term Robot-Assisted Rehabilitation in Patients With Hand Paralysis After Stroke: A Randomized Clinical Trial," *Hand (New York, N.Y.)*, vol. 13, pp. 95–102, Jan. 2018.

[7] S. Li, Z. Wang, Z. Pang, Z. Duan, and M. Gao, "Design and analysis of an upper limb exoskeleton robot for stroke rehabilitation," in *2022 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 573–578, 2022.

[8] A. D. Othman, N. A. Che Zakaria, N. M. Hashim, and S. Mohamaddan, "Exoskeleton Robotics Intervention as an Adjunctive Treatment in Enhancing Post-Stroke Upper Limb Neurorecovery," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pp. 1391–1396, 2021.

[9] H. Li, S. Guo, D. Bu, H. Wang, and M. Kawanishi, "Subject-Independent Estimation of Continuous Movements Using CNN-LSTM for a Home-Based Upper Limb Rehabilitation System," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6403–6410, 2023.

[10] A. Karam, F. Alnajjar, and M. Gochoo, "Assistive and Rehabilitation Robotics for Upper Limb Impairments in Post-Stroke Patients: Evaluation Criteria for the Design and Functionality," in *2020 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–4, 2020.

[11] M. Irshaidat, M. Soufian, A. Elkurdi, and S. Nefti-Meziani, "Soft and Hard Robotics for Movement Rehabilitation, Analysis and Modelling," in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 964–969, 2019.

[12] W. Wang, X. Lyu, X. Li, H. Zhou, J. Huang, X. Wang, and L. Wang, "EMG/EEG Based Control Scheme of Rehabilitation Robots for Stroke Patients: A Systematic Review," in *2025 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 328–333, 2025.

[13] L. Zhang, Y. Peng, X. Qi, S. Bao, and J. Yuan, "The Development Status, Challenges and Development Trends in Rehabilitation Robotics: A Review," in *2024 17th International Convention on Rehabilitation Engineering and Assistive Technology (i-CREATe)*, pp. 1–6, 2024.

[14] C. J. De Luca, "The Use of Surface Electromyography in Biomechanics," *Journal of Applied Biomechanics*, vol. 13, pp. 135–163, May 1997.

[15] R. Merletti and G. Cerone, "Tutorial. Surface EMG detection, conditioning and pre-processing: Best practices," *Journal of Electromyography and Kinesiology*, vol. 54, p. 102440, Oct. 2020.

[16] M. P, A. JB, S. S. Kaimal, and S. R S, "Alleviating Muscle Fatigue in Robotic Prosthetic Arms Through Multi-Sensor Fusion and Adaptive Real-Time Grip Assistance," in *2024 International Conference on Communication, Computing, Smart Materials and Devices (ICCCSMD)*, pp. 1–12, Dec. 2024.

[17] B. Hudgins, P. Parker, and R. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, pp. 82–94, Jan. 1993.

[18] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Systems with Applications*, vol. 39, pp. 7420–7431, June 2012.

[19] D. Farina, R. Merletti, and R. M. Enoka, "The extraction of neural strategies from the surface EMG," *Journal of Applied Physiology*, vol. 96, pp. 1486–1495, Apr. 2004.

[20] E. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *The Journal of Rehabilitation Research and Development*, vol. 48, no. 6, p. 643, 2011.

[21] E. Clancy, E. Morin, and R. Merletti, "Sampling, noise-reduction and amplitude estimation issues in surface electromyography," *Journal of Electromyography and Kinesiology*, vol. 12, pp. 1–16, Feb. 2002.

[22] H. Landau, "Sampling, data transmission, and the Nyquist rate," *Proceedings of the IEEE*, vol. 55, pp. 1701–1706, Oct. 1967.

[23] D. J. Atkins, D. C. Y. Heard, and W. H. Donovan, "Epidemiologic Overview of Individuals with Upper-Limb Loss and Their Reported Research Priorities," *JPO: Journal of Prosthetics and Orthotics*, vol. 8, no. 1, p. 2, 1996.

[24] K. Balakrishnan and V. Honavar, "On sensor evolution in robotics," in *Proceedings of the 1st annual conference on genetic programming*, (Cambridge, MA, USA), pp. 455–460, MIT Press, July 1996.

[25] M. Lister and T. Salem, "Design and implementation of a robot power supply system," in *Proceedings IEEE SoutheastCon 2002 (Cat. No.02CH37283)*, (Columbia, SC, USA), pp. 418–421, IEEE, 2002.

[26] H. M. Boland, M. I. Burgett, A. J. Etienne, R. M. S. Iii, H. M. Boland, M. I. Burgett, A. J. Etienne, and R. M. S. Iii, "An Overview of CAN-BUS Development, Utilization, and Future Potential in Serial Network Messaging for Off-Road Mobile Equipment," in *Technology in Agriculture*, IntechOpen, July 2021.

[27] S. Karthick, S. Venkatesh, P. S. Chaithanya, and G. C. S. Royal, "Implementation and Functional Verification of I2C, SPI and CAN frame protocols," in *2025 International Conference on Emerging Smart Computing and Informatics (ESCI)*, (Pune, India), pp. 1–5, IEEE, Mar. 2025.

[28] W. Chen, C. Hong, X. Wen, M. A. Thornton, and P. Gui, "Controller Area Network (CAN) Bus Transceiver with Enhanced Rail Converter," in *2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS)*, (Springfield, MA, USA), pp. 64–67, IEEE, Aug. 2024.

[29] P. Agarwal, S. Kumar, J. Ryde, J. Corso, V. Krovi, N. Ahmed, J. Schoenberg, M. Campbell, M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, R. Siegwart, J. Brookshire, S. Teller, M. Bryson, M. Johnson-Roberson, O. Pizzaro, S. Williams, N. Colonnese, A. Okamura, D. Dey, T. Y. Liu, M. Hebert, J. A. Bagnell, M. R. Dogar, K. Hsiao, M. Ciocarlie, S. Srinivasa, B. Douillard, N. Nourani-Vatani, C. Roman, I. Vaughn, G. Inglis, A. Dragan, J. Gillula, C. Tomlin, K. Gilpin, D. Rus, G. Gioioso, G. Salvietti, M. Malvezzi, D. Prattichizzo, M. Goodrich, S. Kerman, B. Pendleton, P. B. Sujit, C. Hartmann, J. Boedecker, O. Obst, S. Ikemoto, M. Asada, K. Hauser, T. Kanda, L. Jaillet, J. M. Porta, A. Jain, C. Crean, C. Kuo, H. V. Bremen, S. Myint, D. Joho, G. D. Tipaldi, N. Engelhard, C. Stachniss, W. Burgard, B. Julian, S. L. Smith, M. Kazemi, J.-S. Valois, N. Pollard, M. Kovac, M. Bendana, R. Krishnan, J. Burton, M. Smith, R. Wood, M. Kuderer, H. Kretzschmar, C. Sprunk, S. Kuindersma, R. A. Grupen, A. G. Barto, T. Kunz, M. Stilman, A. Kushleyev, V. Kumar, D. Mellinger, M. Laffranchi, N. Tsagarakis, D. Caldwell, Y. Latif, C. C. Lerma, J. Neira, M. Li, A. Mourikis, S. Lim, C. Sommer, E. Nikolova, L. Liu, D. Shell, A. Long, K. Wolfe, M. Mashner, G. Chirikjian, I. Lysenkov, V. Eruhimov, G. Bradski, W. Maddern, M. Milford, G. Wyeth, B. Marthi, J. Nakanishi, S. Vijayakumar, E. Olson, P. Agarwal, I. Paprotny, C. Levey, P. Wright, B. R. Donald, Q.-C. Pham, Y. Nakamura, M. Phillips, B. Cohen, S. Chitta, M. Likhachev, F. Qian, T. Zhang, C. Li, A. Hoover, P. Masarati, P. Birkmeyer, A. Pullin, R. Fearing, D. Goldman, K. Rawlik, M. Toussaint, C. Robin, S. Lacroix, E. Rombokas, M. Malhotra, E. Theodorou, Y. Matsuoka, E. Todorov, S. Sarid, B. Xu, H. Kress-Gazit, A. Schepelmann, H. Geyer, M. Taylor, L. Sentis, J. Petersen, R. Philippsen, C. Taylor, A. Cowley, S. Tellex, P. Thaker, R. Deits, D. Simeonov, T. Kollar, N. Roy, P. Vernaza, V. Narayanan, H. Wang, G. Hu, S. Huang, G. Dissanayake, Z. Wang, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, J. Peters, R. Wilcox, S. Nikolaidis, J. Shah, E. Wolff, U. Topcu, R. Murray, C. Yoo, R. Fitch, S. Sukkarieh, D. Zarubin, V. Ivan, T. Komura, D. Zelazo, A. Franchi, F. Allgöwer, H. Bülthoff, and P. R. Giordano, "State Estimation for Legged Robots: Consistent Fusion of Leg Kinematics and IMU," in *Robotics: Science and Systems VIII*, pp. 17–24, MIT Press, 2013.

[30] T. K. Chan, Y. K. Yu, H. C. Kam, and K. H. Wong, "Robust Hand Gesture Input Using Computer Vision, Inertial Measurement Unit (IMU) and Flex Sensors," in *2018 IEEE International Conference on Mechatronics, Robotics and Automation (ICMRA)*, (Hefei), pp. 95–99, IEEE, May 2018.

[31] S. A. Ludwig and K. D. Burnham, "Comparison of Euler Estimate using Extended Kalman Filter,

Madgwick and Mahony on Quadcopter Flight Data," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1236–1241, June 2018.

[32] I. H. López-Nava and A. Muñoz-Meléndez, "Wearable Inertial Sensors for Human Motion Analysis: A Review," *IEEE Sensors Journal*, vol. 16, pp. 7821–7834, Nov. 2016.

[33] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 743–761, Apr. 2012.

[34] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *The International Journal of Robotics Research*, vol. 30, pp. 56–79, Jan. 2011.

[35] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, Nov. 2000.

[36] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, pp. 2280–2292, June 2014.

[37] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, May 2011.

[38] S. Pan and X. Wang, "A Survey on Perspective-n-Point Problem," in *2021 40th Chinese Control Conference (CCC)*, (Shanghai, China), pp. 2396–2401, IEEE, July 2021.

[39] E. Marchand, H. Uchiyama, and F. Spindler, "Pose Estimation for Augmented Reality: A Hands-On Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 2633–2651, Dec. 2016.

[40] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, (Kauai, HI, USA), pp. 131–140, IEEE Comput. Soc, 2001.

[41] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, pp. 146–157, Nov. 1997.

[42] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, pp. 1309–1332, Dec. 2016.

[43] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, pp. 1004–1020, Aug. 2018.

[44] D. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, pp. 6–23, Jan. 1997.

[45] T. Xue, W. Wang, J. Ma, W. Liu, Z. Pan, and M. Han, "Progress and Prospects of Multimodal Fusion Methods in Physical Human–Robot Interaction: A Review," *IEEE Sensors Journal*, vol. 20, pp. 10355–10370, Sept. 2020.

[46] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, pp. 28–44, Jan. 2013.

[47] R. Luo and M. Kay, "Multisensor integration and fusion in intelligent systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 901–931, Sept. 1989.

[48] B. Dasarathy, "Sensor fusion potential exploitation-innovative architectures and illustrative applications," *Proceedings of the IEEE*, vol. 85, pp. 24–38, Jan. 1997.

[49] M. Liggins, C.-Y. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85, pp. 95–107, Jan. 1997.

[50] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control*, vol. 53, pp. 1203–1218, June 2008.

[51] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, Mar. 1960.

[52] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems,"

[53] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, "Deep Multispectral Semantic Scene Understanding of Forested Environments Using Multimodal Fusion," in *2016 International Symposium on Experimental Robotics* (D. Kulić, Y. Nakamura, O. Khatib, and G. Venture, eds.), (Cham), pp. 465–477, Springer International Publishing, 2017.

[54] D. Harel, "Statecharts: a visual formalism for complex systems," *Science of Computer Programming*, vol. 8, pp. 231–274, June 1987.

[55] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*. July 2018. arXiv:1709.00084 [cs].

[56] A. G. Barto and S. Mahadevan, "Recent Advances in Hierarchical Reinforcement Learning," *Discrete Event Dynamic Systems*, vol. 13, pp. 41–77, Jan. 2003.

[57] G. D. Valderrama-Gonzalez, M. N. Gasson, K. Warwick, and B. J. Whalley, "Prosthesis grasp reflex via peripheral nerve control — An in vitro study," in *2010 IEEE 9th International Conference on Cyberntic Intelligent Systems*, pp. 1–6, 2010.

[58] T. Zhang and L. Jiang, "Biomimetic Tactile Data Driven Closed-loop Control of Myoelectric Prosthetic Hand-," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1738–1742, 2018.

[59] L. Tian, N. Magnenat Thalmann, D. Thalmann, and J. Zheng, "The Making of a 3D-Printed, Cable-Driven, Single-Model, Lightweight Humanoid Robotic Hand," *Frontiers in Robotics and AI*, vol. 4, Dec. 2017.

[60] A. Bicchi, M. Gabiccini, and M. Santello, "Modelling natural and artificial hands with synergies," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, pp. 3153–3161, Nov. 2011.

[61] A. M. Dollar and R. D. Howe, "The Highly Adaptive SDM Hand: Design and Performance Evaluation," *The International Journal of Robotics Research*, vol. 29, pp. 585–597, Apr. 2010.

[62] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, pp. 161–185, Jan. 2016.

[63] H. In, B. B. Kang, M. Sin, and K.-J. Cho, "Exo-Glove: A Wearable Robot for the Hand with a Soft Tendon Routing System," *IEEE Robotics & Automation Magazine*, vol. 22, pp. 97–105, Mar. 2015.

[64] M. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the Pisa/IIT SoftHand," *The International Journal of Robotics Research*, vol. 33, pp. 768–782, Apr. 2014.

[65] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, H. Höppner, S. Jörg, M. Nickl, A. Nothhelfer, F. Petit, J. Reill, N. Seitz, T. Wimböck, S. Wolf, T. Wüsthoff, and G. Hirzinger, "The DLR hand arm system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3175–3182, May 2011.

[66] M. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 269–279, June 1989.

[67] K. Guo, J. Lu, Y. Wu, X. Hu, and H. Yang, "The Latest Research Progress on Bionic Artificial Hands: A Systematic Review," *Micromachines*, vol. 15, p. 891, July 2024.

[68] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 50, pp. 848–854, July 2003.

[69] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific Data*, vol. 1, p. 140053, Dec. 2014.

[70] M. Atzori, M. Cognolato, and H. Müller, "Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands," *Frontiers in Neurorobotics*, vol. 10, Sept. 2016.

[71] B. Sumir, "Building of Inmoov Robotic Arm for Performing Various Operations,"

[72] M. Jafarzadeh, D. C. Hussey, and Y. Tadesse, "Deep learning approach to control of prosthetic hands with electromyography signals," in *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, (Houston, TX, USA), pp. A1–4–1–A1–4–11, IEEE, Sept. 2019.

[73] Y. Guo, P. Yao, G. Gou, C. Liu, J. Liu, J. Zhou, J. Cheng, M. Zhao, and N. Xue, "sEMG-Based Wearable HMI System For Real-Time Robotic Arm Control With Edge AI," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, (Toronto, ON, Canada), pp. 1–5, IEEE, Oct. 2023.

[74] M. Sharma, H. Malhotra, and N. Kumar, "Deep Learning with Convolutional Neural Network for Controlling Human Arm Prosthesis using sEMG signals,"

[75] P. Kaifosh and T. R. Reardon, "A generic non-invasive neuromotor interface for human-computer interaction," *Nature*, vol. 645, pp. 702–711, Sept. 2025.

[76] J. F. Castruita-López, M. Aviles, D. C. Toledo-Pérez, I. Macías-Socarrás, and J. Rodríguez-Reséndiz, "Electromyography Signals in Embedded Systems: A Review of Processing and Classification Techniques," *Biomimetics*, vol. 10, p. 166, Mar. 2025.

[77] S. M. Sid'El Moctar, I. Rida, and S. Boudaoud, "Time-domain features for sEMG signal classification," in *JETSAN 2023*, (Paris, France), June 2023.

[78] C. Millar, N. Siddique, and E. Kerr, "LSTM Classification of sEMG Signals For Individual Finger Movements Using Low Cost Wearable Sensor," in *2020 International Symposium on Community-centric Systems (CcS)*, pp. 1–8, Sept. 2020.

[79] G. Vasan and P. M. Pilarski, "Context-Aware Learning from Demonstration: Using Camera Data to Support the Synergistic Control of a Multi-Joint Prosthetic Arm," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, (Enschede), pp. 199–206, IEEE, Aug. 2018.

[80] S. Deshmukh, V. Khatik, and A. Saxena, "Robust Fusion Model for Handling EMG and Computer Vision Data in Prosthetic Hand Control," *IEEE Sensors Letters*, vol. 7, pp. 1–4, Sept. 2023.

[81] M. Cognolato, M. Atzori, R. Gassert, and H. Müller, "Improving Robotic Hand Prosthesis Control With Eye Tracking and Computer Vision: A Multimodal Approach Based on the Visuomotor Behavior of Grasping," *Frontiers in Artificial Intelligence*, vol. 4, p. 744476, Jan. 2022.

[82] A. S. Badrinath, P. B. Vinay, and P. Hegde, "Computer Vision based semi-intuitive Robotic arm," in *2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, (Chennai, India), pp. 563–567, IEEE, Feb. 2016.

[83] "Automated Tomato Maturity Classification: Integrating Computer Vision System and Robotic Manipulation | Request PDF," in *ResearchGate*.

[84] H. Nyqvist and F. Gustafsson, "A high-performance tracking system based on camera and IMU," in *Proceedings of the 16th International Conference on Information Fusion*, pp. 2065–2072, July 2013.

[85] L. Xie, D. Huang, Z. Lu, N. Wang, and C. Yang, "Handheld Device Design for Robotic Teleoperation based on Multi-Sensor Fusion," in *2023 IEEE International Conference on Mechatronics (ICM)*, (Loughborough, United Kingdom), pp. 1–6, IEEE, Mar. 2023.

[86] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–7, June 2011. ISSN: 1945-7901.

[87] S. A. Ludwig, "Optimization of Control Parameter for Filter Algorithms for Attitude and Heading Reference Systems," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, July 2018.

[88] D. Saha, M. Lakshmipathi, N. Brijesh, C. Chitnis, N. K. Jisy, and R. Vinu, "Vision-Guided Prosthetic Arm with Autonomous Object Interaction," in *2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1656–1661, 2025.

[89] C. J. De Luca, L. Donald Gilmore, M. Kuznetsov, and S. H. Roy, "Filtering the surface EMG signal: Movement artifact and baseline noise contamination," *Journal of Biomechanics*, vol. 43, pp. 1573–1579, May 2010.

[90] A. Mark Burden, S. E. Lewis, and E. Willcox, "The effect of manipulating root mean square window length and overlap on reliability, inter-individual variability, statistical significance and clinical relevance of electromyograms," *Manual Therapy*, vol. 19, pp. 595–601, Dec. 2014.

[91] L. McManus, G. De Vito, and M. M. Lowery, "Analysis and Biophysics of Surface EMG for Physiotherapists and Kinesiologists: Toward a Common Language With Rehabilitation Engineers," *Frontiers in Neurology*, vol. 11, p. 576729, Oct. 2020.

[92] M. Asghari Oskoei and H. Hu, "Myoelectric control systems—A survey," *Biomedical Signal Processing and Control*, vol. 2, pp. 275–294, Oct. 2007.

[93] J. Ni, H. Tang, S. T. Haque, Y. Yan, and A. H. H. Ngu, "A Survey on Multimodal Wearable Sensor-based Human Action Recognition," Apr. 2024. arXiv:2404.15349 [eess].

[94] P. D. Hung, V. V. Chin, N. T. Chinh, and T. D. Tung, "A Flexible Platform for Industrial Applications Based on RS485 Networks," *Journal of Communications*, pp. 245–255, 2020.

A Additional figures

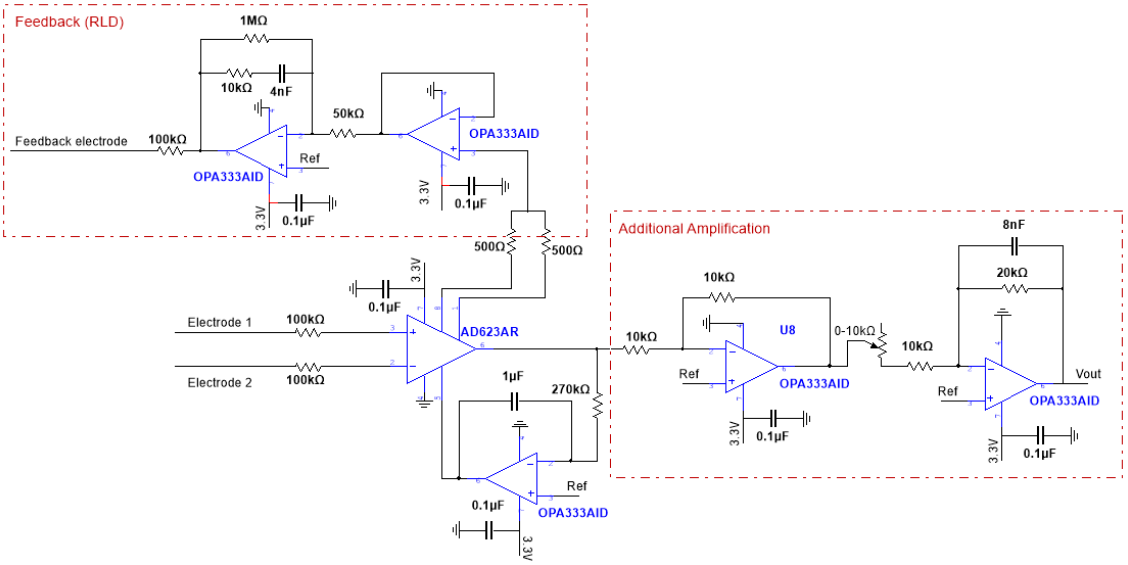


Figure 18: Electrical schematic of the front-end EMG sensor.

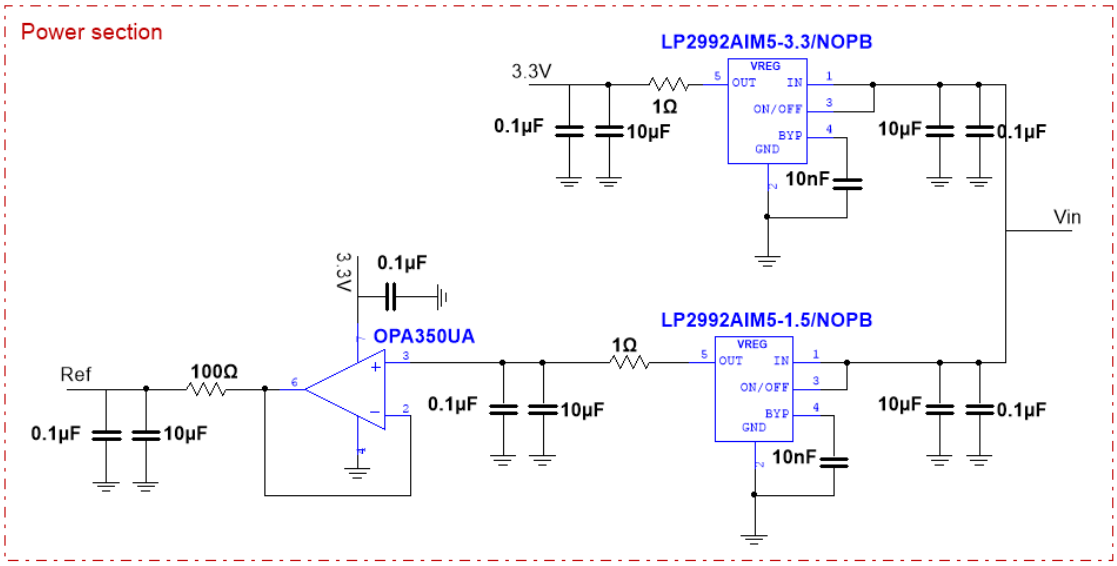


Figure 19: Electrical schematic of the power section.