MITACS X Toronto Metropolitan University

MITACS Globalink Research Internship - 2024

Design, Development and Testing of DC servomotor controller for precise torque control

Submitted by:
Chirag Kotian,
B.Tech in Electrical Engineering, IIT Tirupati

Supervised by:
Prof. Kourosh Zareinia,
Department of Mechanical Engineering
Associate Professor, TMU

Contents

1 Abstract 3

2 About Research Lab 3

3 Introduction 3
- 3.1 Motivation 3
- 3.2 Project Aims and Specifications Desired 4

4 Features of the system 4
- 4.1 One Single-Sided Power Supply 4
- 4.2 Wide Range of Input Power Supply 4
- 4.3 Efficient Communication Interface 5
- 4.4 Analog Current Source (VCCS) 5
- 4.5 High-Resolution Encoder counter 5
- 4.6 Custom ROS-Messages 5
- 4.7 Expandable I/O 5
- 4.8 Integrated Power Supply 5
- 4.9 Modular PCBs 5
- 4.10 Robust Protection 5
- 4.11 Direction Indicators 5
- 4.12 Current Monitoring 6
- 4.13 Control and Status Monitoring 6
- 4.14 Single Clock Oscillator 6
- 4.15 Ease of Manufacturing 6

5 Approach 6
- 5.1 Why Brushed Motors over Brushless Motors? 6
- 5.2 Controlling Torque 6
- 5.3 Controlling Motor Current 6
  - 5.3.1 Microcontroller-Based Feedback 6
  - 5.3.2 Current Source with Analog Feedback 6
- 5.4 Motor Positioning 7
- 5.5 Low Voltage Supplies 7
- 5.6 Reducing complexity in wiring for the ease of the user 7
- 5.7 Current Sensing 7
- 5.8 Interfacing with ICs 7

6 Version 1 8

7 Version 2 8

8 Version 3 9
- 8.1 Version 3.0 9
- 8.2 Version 3.1 9

9 Problems Faced and Solutions Adopted 11

10 Codebase and Package 12

11 Tests **12**
11.1 Current Source . . . 12
11.1.1 Testing only the current source, No H-bridge: Video . . . 12
11.1.2 Testing the current capabilities of Current Source:Video . . . 12
11.1.3 Testing the current source with UVLO:Video . . . 13
11.1.4 Adding Resistor Helps with UVLO:Video . . . 13
11.1.5 Testing the current source with UVLO solved:Video . . . 13
11.2 Testing other ICs . . . 13
11.2.1 Testing DAC:Video . . . 13
11.2.2 Testing ADC:Video . . . 13
11.3 Quadrature Encoder Counter . . . 13
11.3.1 Encoder with Attachment:Video . . . 13
11.3.2 Testing Quadrature Output:Video . . . 13
11.3.3 Testing Encoder Counter Against DAQ and Matlab:Video . . . 13
11.3.4 Testing Encoder Counter on Signal Processing PCB:Video . . . 13
11.4 Testing the complete system . . . 13
11.4.1 Test Of Entire Board V3.1 with ROS2 Humble: Video . . . 13
11.4.2 System Application:Video . . . 13
11.4.3 Testing the fully assembled system:Video . . . 13

12 Limitations, probable solutions and further developments **13**
12.1 ROS2 Publishing Issue . . . 13
12.2 High Inductance Motors . . . 13
12.3 IC Footprint Optimization . . . 13
12.4 Alternative H-Bridge Solutions . . . 13
12.5 Communication Simplification . . . 14
12.6 LED Indicators for Digital Signals . . . 14
12.7 Threading in Python . . . 14
12.8 Enhanced Control Topology . . . 14
12.9 DAC Zero-Voltage Issue . . . 14
12.10 Expanding ROS Message Scope . . . 14

13 Conclusion **14**

14 References **15**

Design, Development and Testing of DC servomotor controller for precise torque control

August 22, 2024

1 Abstract

This project develops an advanced motor control system to enhance haptic feedback in robotics, which is crucial for tasks requiring precision like surgery. The system manages the torque output of six brushed motors with high accuracy and a response time of 1 kHz. Key features include a single-sided power supply, support for a wide input voltage range, and efficient communication via SPI and I2C.

The controller integrates a high-precision analog current source and six encoder decoder chips with a 32-bit counter for accurate motor position measurement. Custom ROS2 messages facilitate intuitive data handling and analysis, while modular PCBs and robust protection mechanisms ensure reliability and ease of manufacturing.

Future improvements will focus on resolving ROS2 publishing issues, optimizing motor inrush current management, and enhancing control algorithms. The codebase, designed for Ubuntu 22.04 with ROS2 Humble, uses Python's lgpio library and SPIDEV for SPI communication, offering a streamlined setup and execution process. This project aims to advance haptic feedback capabilities in robotics, providing a versatile and high-performance solution for various applications.

2 About Research Lab

The Haptics and Telerobotics Laboratory (HapTel) at Toronto Metropolitan University, under the direction of Prof. Kourosh Zareinia, conducts advanced research in haptics and teleoperation of robotic manipulators. The lab's focus is on developing cutting-edge human-robot interfaces that enable operators to experience a sense of touch, with applications in areas such as robot-assisted surgery. Equipped with state-of-the-art haptic devices like PHANToM OMNI and Desktop, along with robotic manipulators, microcontrollers, actuators, sensors, and a comprehensive mechatronics workstation, HapTel is dedicated to making medical interventions more efficient and less traumatic through innovative robotic technologies.

3 Introduction

3.1 Motivation

Haptics technology has become indispensable in fields such as surgery, where the sense of touch is crucial for precise manipulation. This tactile feedback enables us to intuitively gauge the force exerted on objects. However, this sense is often lost in robotic applications, posing a significant challenge. Robotic hands, designed to enhance strength and accuracy, typically lack the ability to provide real-time, intuitive feedback about the environment they interact with. Advances in this field have led to the integration of force sensors and mechanical contraptions with motors (haptic devices), which mimic tactile sensations by scaling the forces applied by robotic hands.

Despite these advancements, existing haptic devices are plagued with issues such as jittery outputs, inaccurate torque control, high costs, cumbersome designs, and slow response times. Additionally, most commercially available motor controllers are primarily designed for position control of servomotors rather than precise torque control.

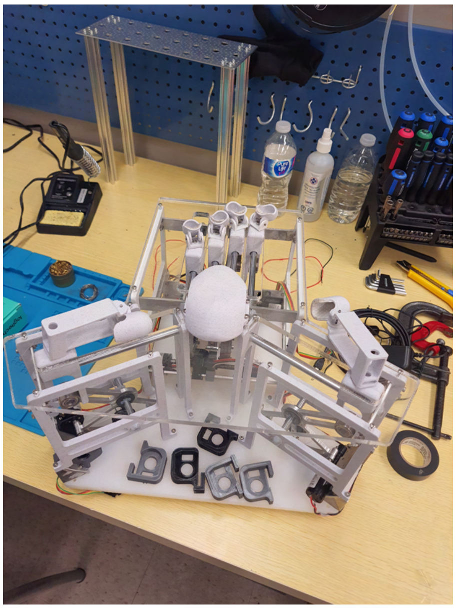

Figure 1: Mechanical part of Haptic Device designed by TMU students

3.2 Project Aims and Specifications Desired

This project aims to design, develop, and test a comprehensive system capable of controlling the torque output of six brushed motors with a minimum accuracy of 12 bits **from scratch**. The system will also measure the absolute rotation angle using a quadrature encoder set at 1024 pulses per revolution, achieving a resolution of 4096. This will be accomplished without compromising the system's response time of 1 kHz and a maximum motor RPM of 3000. The motors will be bi-directional, powered by a single 24V single-sided supply, with the motor controller capable of delivering up to 10A per motor. While in operation, the user shouldn't feel any sudden kicks. The rationale for selecting brushed motors will be elaborated upon in the report. The user must be able to see different stats related to the system in real-time through GUI tools with the possibility to record all the stats for future analysis. This project seeks to address the limitations of current haptic devices, providing a more accurate, responsive, and adaptable solution.

4 Features of the system

4.1 One Single-Sided Power Supply

The system requires only one single-sided power supply, simplifying the power management and reducing complexity.

4.2 Wide Range of Input Power Supply

The system supports a wide range of input power supply voltages, enhancing its versatility across various applications.

4.3 Efficient Communication Interface

Only six wires are needed to connect the system to an external computer(SPI and I2C capable), SPI (3 wires), I2C (2 wires), and common ground. Tested communication speeds are 400 kHz for SPI and 100 kHz for I2C, with potential speeds of up to 50 MHz for SPI (DAC) and 4 MHz (Quadrature encoder counter), and up to 1.7 MHz for I2C (ADC), 1 MHz (Level Shifter), and 400 kHz (Digital I/O Expanders).

4.4 Analog Current Source (VCCS)

The system features a completely analog current source, ensuring faster response times to disturbances and greater accuracy. This is achieved through the use of precise instrumentation amplifiers, with adjustable gain via a potentiometer. Theoretically, the maximum current output can be adjusted from 0A to 9.25A, with a resolution determined by dividing the maximum current by $(2^{14} - 1)$. Note that realistic operation at low resistance values may have limitations.

4.5 High-Resolution Encoder counter

The system includes six encoder counter ICs with a 32-bit counter each, configurable for motor position measurement through various methods. By default, it is set for use with a quadrature encoder at 1024 pulses per revolution, providing a resolution of 4096 per revolution. The system has been tested with a motor running at 1920 RPM, and an encoder capable of handling up to 3000 RPM.

4.6 Custom ROS-Messages

The system utilizes custom ROS messages that are intuitive to understand. Leveraging ROS 2, these messages can be stored, plotted, and processed for further analysis using various publicly available ROS tools with user-friendly interfaces and GUIs.

4.7 Expandable I/O

All unutilized I/O pins are available for users to expand the scope of their projects, offering flexibility for additional functionalities.

4.8 Integrated Power Supply

Each current source PCB includes a 5V buck converter power supply operating at 1.1 MHz, capable of supplying 3A of current. This can power the computer, cooling fans, or other external devices.

4.9 Modular PCBs

The system's PCBs are modular, allowing for independent use, and are designed for ease of manufacturing with hand-solderable, 2-layer boards.

4.10 Robust Protection

The system features well-protected digital pins for independent operation. Protection diodes are in place to safeguard the power supply and overall system from voltage spikes caused by inductive discharges from the motors.

4.11 Direction Indicators

Direction LEDs are included to visually indicate the rotation direction of the motors.

4.12 Current Monitoring

Using the onboard 8-bit ADC, the system can monitor the current passing through the operational amplifier (OP-AMP) and/or H-bridge.

4.13 Control and Status Monitoring

The system employs two 16-channel digital I/O ICs to control the H-bridge, manage communications, and monitor the status of the current OP-AMP.

4.14 Single Clock Oscillator

The entire system is synchronized using just one clock oscillator, ensuring cohesive operation.

4.15 Ease of Manufacturing

The design prioritizes manufacturability, with each PCB being easily hand-solderable and consisting of only two layers.

5 Approach

5.1 Why Brushed Motors over Brushless Motors?

It's a well-known fact that brushless motors are way superior to brushed motors in various aspects like efficiency, but the decision to use brushed motors instead of brushless motors was driven by the need for smoother operation, which provides superior haptic feedback for the user. Expensive Brushless Motors can somewhat achieve this but even cheaper brushed motors achieve this pretty easily. Also, another lesser important reason was the fact that the control circuit of a brushed motor is less complex than a brushless motor.

5.2 Controlling Torque

The primary objective of this project is to control motor torque. There are two main methods to achieve this: using a torque sensor or controlling the current passing through the motor, as motor current is directly proportional to torque. Developing or sourcing a torque sensor is challenging due to the difficulty, complexity, and inaccuracy involved, making the latter method more viable.

5.3 Controlling Motor Current

To control motor current, there are two approaches:

5.3.1 Microcontroller-Based Feedback

This method involves using a microcontroller to provide feedback that controls the PWM input to the motor. By controlling the voltage across the motor, the current—and thus the torque—can be regulated in a closed-loop system (version 1 of the project). However, while functional, this approach has limitations such as slower response times and current spikes.

5.3.2 Current Source with Analog Feedback

This method eliminates the need for a microcontroller, thereby reducing response time, though it introduces more complexity (versions 2 and 3 of the project).

5.4 Motor Positioning

Precise motor positioning was another crucial aspect of the project. To achieve this, an IC capable of storing the motor's position for a sufficiently long period without overflowing (32-bit) was selected. This ensures that no data is lost even if the computer samples the encoders after extended intervals. Before finalizing the IC, a development board was ordered and tested using an Arduino Uno R4, a highly accurate DAQ, and MATLAB. Tests involved running the motor at different RPMs for up to 5 minutes, with custom 3D-printed attachments designed to accommodate various motor shaft sizes. The results confirmed that the IC performed flawlessly without skipping any pulses from the quadrature incremental encoder.

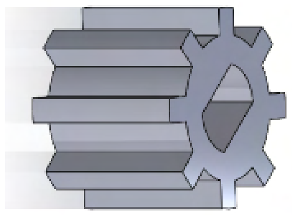

Figure 2: Attachments for different Motor Types

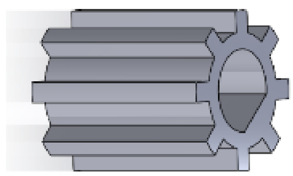

Figure 3: Attachments for different Motor Types

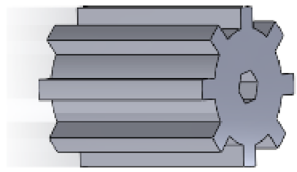

Figure 4: Attachments for different Motor Types

5.5 Low Voltage Supplies

For powering the system's low-power electronics, a switching buck converter was chosen over an LDO, as the required 5V output is significantly lower than the maximum input voltage, leading to inefficiencies if an LDO were used. However, a 3.3V LDO was used to step down from 5V for powering a few ICs on the signal-processing PCB.

5.6 Reducing complexity in wiring for the ease of the user

To minimize the number of wires required to connect to the computer, ICs that could be interfaced using SPI or I2C protocols were selected. Instead of using digital GPIOs from the computer, I2C-based GPIO expander chips were utilized, providing 32 digital I/O pins for various purposes, including chip selection for SPI communication.

5.7 Current Sensing

To gather information about the current passing through the system for fail-safes or for plotting actual vs. intended states, a 12-channel 8-bit ADC was employed. This ADC provides voltage readings at both the current source and H-bridge, offering redundancy in current measurement.

5.8 Interfacing with ICs

All ICs on the signal processing PCB were operated at the register level, using specific I2C and SPI modes tailored to each IC. This required a thorough understanding of the ICs' datasheets, including their configuration, setup, and data register structures, which were then implemented in the code.

6 Version 1

In this version, current control was attempted using constant feedback from a current sensor to a microcontroller (Arduino Uno R4 Wi-Fi). The microcontroller controlled an H-bridge motor controller via PWM at frequencies of 20 kHz and above, using timers instead of prebuilt libraries to achieve higher control and frequencies. This ensured the user didn't hear any high-frequency irritating noises. The H-bridge used was the LMD18200, with a schematic inspired by reference designs in its datasheet. The current sensor in the LMD18200 itself was used for feedback. Although the PCB design for this version was completed, tests conducted on a similar H-bridge (VNH5019) before manufacturing the PCB revealed issues such as slower response times, current spikes, and constant fluctuations in current. Based on these findings, it was decided to proceed with a current source approach that does not rely on microcontroller-based feedback.

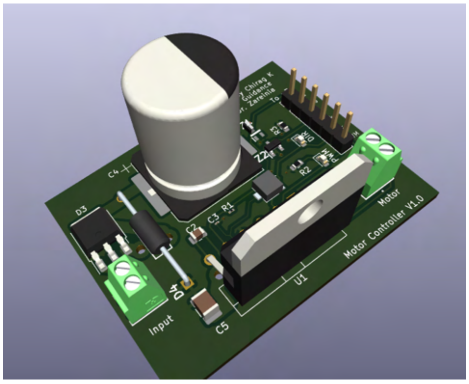

Figure 5: 3D view of Version1

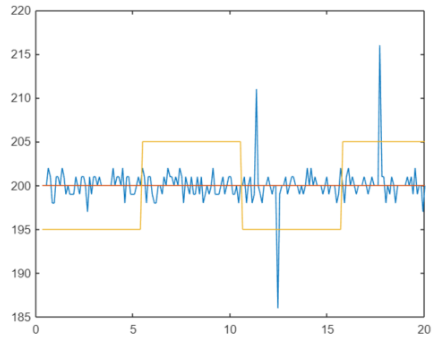

Figure 6: Set Current(Orange), Actual Current(Blue) and Direction of Rotation(Yellow) VS Time

7 Version 2

Version 2 served as a crucial step toward the final design (Version 3), focusing on controlling a single motor with specific requirements. The primary goal was to address the shortcomings of Version 1 and reduce dependency on any sort of computing device in the feedback loop, requiring only one USB connection to send commands and receive system status updates. A 7-segment display was integrated to show real-time current consumption, controlled by a multi-LED driver.

An Atmega 328P microcontroller was used to manage the entire system, providing access to unused pins, ISP pins, and the option to use micro USB via a USB to Serial/UART bidirectional IC for programming and monitoring. All USB pins were protected. The user could select between signed PWM or signed analog input via an SPDT switch. In the case of PWM, a duty cycle from 0 to 50 was considered to be for motor rotation in one direction with speed increasing from 0 to 50, whereas other direction from 50 to 100 with speed increasing from 50 to 100. In analog mode, similar was the case; just the divisions were from 0 to 2.5V and 2.5V to 5.0V. For analog mode, a 16-bit ADC was used, while for PWM, an interrupt pin programmed to control an internal timer was utilized. A good H-bridge IC controlled the motor's direction, chosen for its high performance, numerous features, ease of use, and high current handling capacity without excessive heating, as determined from previous experiments.

All internal IC communications were conducted via I2C protocol. The microcontroller controlled a 14-bit DAC, which served as the input for a completely analog Voltage Controlled Current Source (VCCS). This VCCS, inspired by the protection mechanisms in Version 1, was well-protected against motor voltage surges. It featured an output proportional to the current passing through it, enable/status pins, and sufficient protection for all input pins connected to the microcontroller. The VCCS also had adjustable gain to control the maximum current, rated for 5A with a maximum of 7A. Additionally, the system included a 5V buck converter power supply to power low-voltage components.

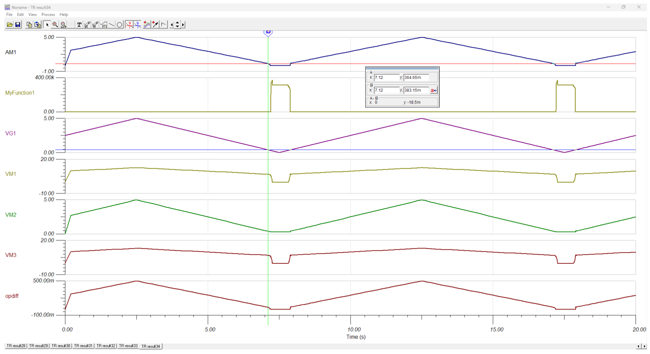

Figure 7: Results of Current Source Simulation with approximate models of H bridge and motor

This version was feasible but was ultimately abandoned due to its complexity and the impracticality of requiring a microcontroller on board for each motor, considering the final requirement to control six motors simultaneously.

8 Version 3

8.1 Version 3.0

Version 3.0 represented a significant evolution, designed to manage six motors simultaneously. Many circuits from Version 2 were retained, but the overall approach was simplified by removing the onboard microcontroller and introducing a 32-bit quadrature encoder counter for all six motors. The design leveraged a computer like a Raspberry Pi to interface with the user through a GUI and communicate with the motor controller via I2C and SPI, minimizing the number of wires and enhancing the user experience.

The current source (with an increased capacity of up to 10A per motor), H-bridge, and protection circuits from Version 2 were directly carried over. Additional components such as digital GPIO expanders, multichannel 8-bit ADCs, and 16-bit DACs were used. The user was provided with pins to utilize the unused GPIOs to expand the project scope if needed. They were also given the option to utilize the index channel of the quadrature encoder by simply shorting two jumper pads.

All six current source H-bridge combos and six 32-bit quadrature encoder counters, each with its own oscillator, were integrated onto a single PCB, making the design more compact than Version 2, though still relatively large.

8.2 Version 3.1

Version 3.1 further refined Version 3.0 by making the system more compact and modular. All encoder counters were synchronized using a single oscillator, and the board was divided into ten separate PCBs: six current source + H-bridge boards, one signal distribution board, one signal processing board, and two diode array boards. This modular approach enabled a more compact system design, allowing it to be packed into a box more efficiently by stacking them on top of each other rather than having a single large PCB. It also improved heat dissipation, allowing the use of only a single large heatsink for all current sources, reducing the number of cooling fans required. Moreover, it provided the possibility of replacing a single PCB rather than the entire system in case of damage. Specific connectors were chosen to make the system plug-and-play, minimizing the risk of incorrect connections or short circuits. Diode arrays were introduced to resolve the issue of UVLO triggering.

The majority of the components on PCBs of this version were handsoldered.

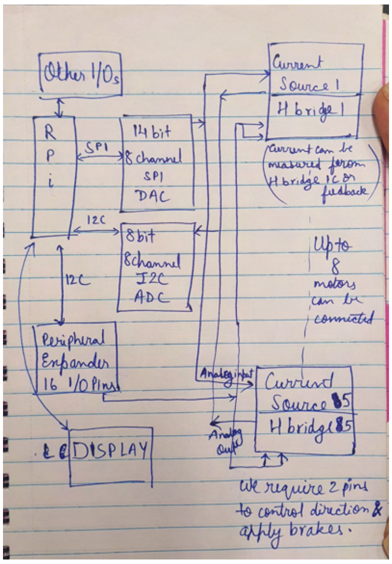

Figure 8: Plan for Version 3.1

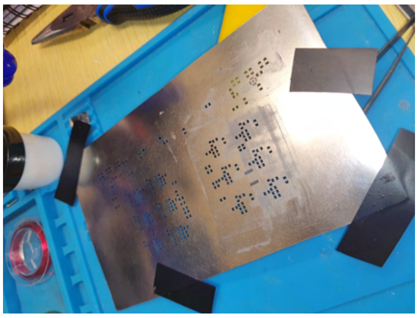

Figure 9: Soldering components on PCB using stencil

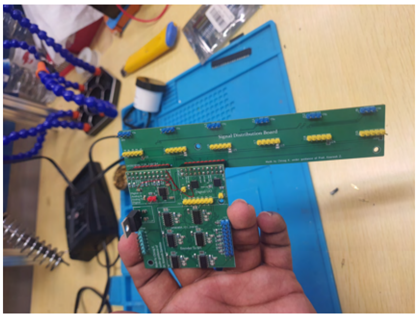

Figure 10: Easy attachment of signal processing board to signal distribution board

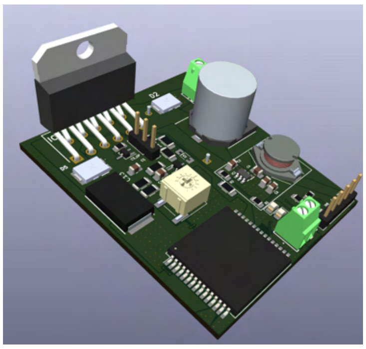

Figure 11: Front 3D view of Version 3.1 Current Source

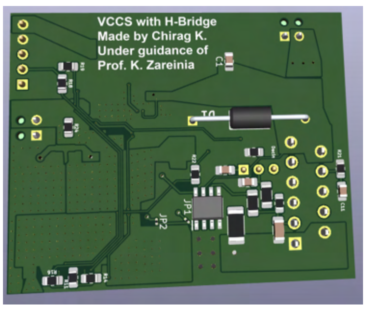

Figure 12: Back 3D view of Version 3.1 Current Source

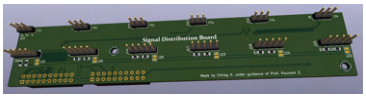

Figure 13: Front 3D View of Version 3.1 Signal Distribution Board

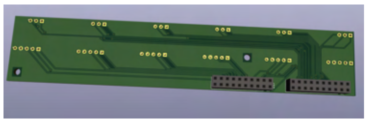

Figure 14: Back 3D View of Version 3.1 Signal Distribution Board

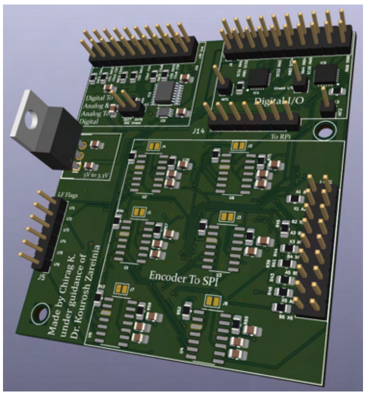

Figure 15: Front 3D View of Version 3.1 Signal Processing Board

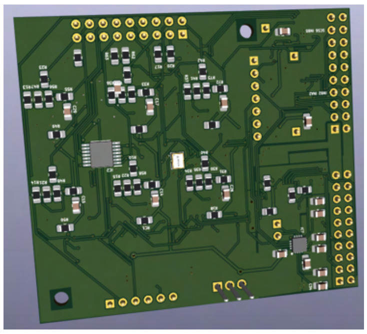

Figure 16: Back 3D View of Version 3.1 Signal Processing Board

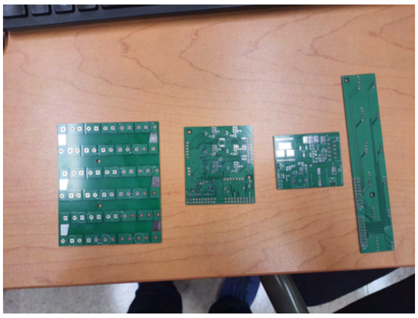

Figure 17: Manufactured V3.1 PCBs

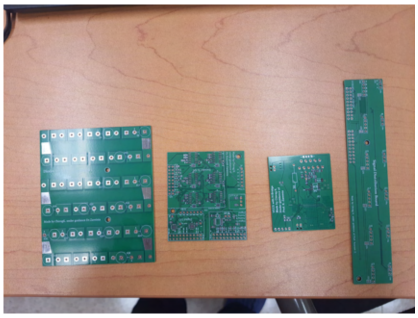

Figure 18: Manufactured V3.1 PCBs

9 Problems Faced and Solutions Adopted

1. **Missing Ground/supply voltage connections in PCB**: Due to the presence of too many connections and traces, a few of the ground and supply voltage connections were missed, which was found in the continuity tests on the PCB. Temporarily, the issues were resolved by jumpers on defective PCBs, and the designs were immediately updated to rectify the issue.

2. **H-Bridge Supply**: One of the pins on the H-bridge was initially connected to supply voltage instead of the current source due to a lack of information on the datasheet. But this issue has been resolved in updated designs, and for defective PCBs, it is recommended to desolder the Vbat pin and short it to the adjacent Vcc pin.

3. **UVLO Trigger**: Due to the low resistance of motor coils, UVLO (Under Voltage Lock-Out Protection) triggers in the H-bridge, shutting it down. To prevent this, diode forward voltage drop was used.

4. **GPIO Expander Pull-Up Issue**: It was discovered that the digital GPIO expander had issues pulling up the pulled-down digital input pins of the H-bridge. To solve this issue, another resistor

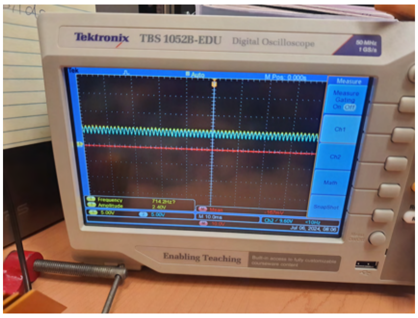

Figure 19: UVLO triggering

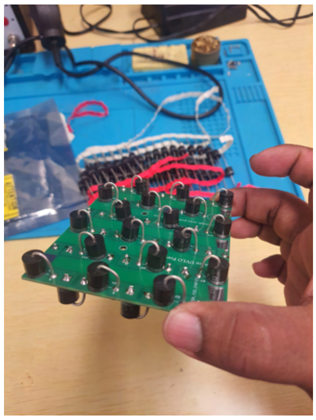

Figure 20: Diode array

of the same value as the pull-down resistor was connected between the input of the H-bridge and 5V, producing a net pull-up effect on digital input, considering any voltage greater than 2.1V is considered high by the H-bridge. This issue has also been resolved in the PCB design.
Full system test with pull-up resistors: Link

10 Codebase and Package

The codebase primarily relies on the Python ‘lgpio‘ library and ROS2. It has been designed and tested on Ubuntu 22.04 with ROS2 Humble and Python3. However, since no commands specific to the Humble distribution are used, it should be compatible with any ROS2 distribution.

Additional Libraries: Apart from the mentioned libraries, it is essential to install the ‘SPIDEV‘ library, which is used to handle all SPI communication.

Once the ROS package is installed and built. Just a launch file needs to be run with the command **‘ros2 run MotorControllerV1 MotorControllerV1’**.

11 Tests

11.1 Current Source

Figure 21: Attachment used to test motor with high torque

Figure 22: Test setup to test current sensor

11.1.1 Testing only the current source, No H-bridge: Video

11.1.2 Testing the current capabilities of Current Source:Video

11.1.3 Testing the current source with UVLO:Video

11.1.4 Adding Resistor Helps with UVLO:Video

11.1.5 Testing the current source with UVLO solved:Video

11.2 Testing other ICs

11.2.1 Testing DAC:Video

11.2.2 Testing ADC:Video

11.3 Quadrature Encoder Counter

11.3.1 Encoder with Attachment:Video

11.3.2 Testing Quadrature Output:Video

11.3.3 Testing Encoder Counter Against DAQ and Matlab:Video

11.3.4 Testing Encoder Counter on Signal Processing PCB:Video

11.4 Testing the complete system

11.4.1 Test Of Entire Board V3.1 with ROS2 Humble: Video

11.4.2 System Application:Video

Generate Torque in the motor proportional to the angle of rotation (-90 degrees to 90 degrees) such that motor torque direction always wants it to bring the shaft to 0 degrees.

11.4.3 Testing the fully assembled system:Video

12 Limitations, probable solutions and further developments

12.1 ROS2 Publishing Issue

The current code has a bug where custom messages, although correctly saved in variables and printed without issues, fail to publish. This issue needs to be rectified.

12.2 High Inductance Motors

Motors with high inductance experience high inrush current, leading to high-frequency noise and repeated UVLO triggering due to the slow response of diodes. Possible solutions include adding a load capacitor to the motor, using faster diodes, or removing the diodes entirely. However, the latter solution may cause UVLO to trigger at low DAC output voltage.

12.3 IC Footprint Optimization

During the design phase, a small footprint IC was used for the I/O expander to minimize PCB size. However, it was later determined that a larger IC could be accommodated in the current design, reducing the risk of short circuits and simplifying the soldering process.

12.4 Alternative H-Bridge Solutions

Consider using H-bridges or other methods that do not have UVLO issues, improving overall system reliability.

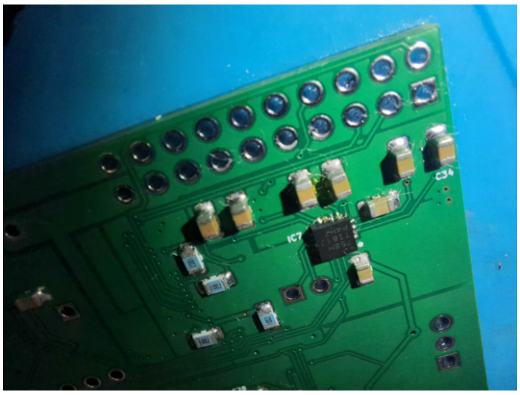

Figure 23: An IC soldered to signal processing board with QFN package.

12.5 Communication Simplification

Transitioning all communications to I2C or SPI would reduce the number of wires connected to the computer (e.g. Raspberry Pi), simplifying the setup.

12.6 LED Indicators for Digital Signals

Adding LED indicators on the digital signal board would enhance visual feedback and troubleshooting capabilities.

12.7 Threading in Python

Implementing threading in Python could increase the speed of handling six motors simultaneously, improving system performance.

12.8 Enhanced Control Topology

The current codebase uses a P controller. Introducing more complex control topologies like PID would likely improve the system's overall performance and stability.

12.9 DAC Zero-Voltage Issue

When the DAC is commanded to output zero volts, it still produces a small voltage in the millivolt range. While this is not an issue for larger motors, it causes smaller motors to rotate slowly. Implementing brakes on the H-bridge could address this issue.

12.10 Expanding ROS Message Scope

Currently, the ROS message scope has been reduced for the initial testing phases. It can be expanded to incorporate the status of various ICs, providing more comprehensive system feedback.

13 Conclusion

This project represents a comprehensive exploration into the design, development, and refinement of a precise DC motor control system with an emphasis on haptic feedback, efficiency, and reliability. The evolution from the initial microcontroller-based feedback system in Version 1 to the more advanced, modular current source approach in Versions 2 and 3 demonstrates a commitment to improving performance and addressing real-world challenges encountered during development.

The decision to use brushed motors over brushless motors was driven by the need for smoother operation, which is essential for delivering high-quality haptic feedback. By focusing on controlling motor torque through current regulation, the project avoided the complexities and inaccuracies associated with torque sensors. The move from microcontroller-based PWM control to a current source with analog feedback not only simplified the design but also significantly enhanced the system's response time and stability.

Through meticulous testing and iteration, the system was refined to handle multiple motors simultaneously, with enhanced power management, precise motor positioning, and minimized wiring. The introduction of modular PCBs in Version 3.1 further improved the system's scalability, ease of assembly, and maintenance, while addressing critical issues such as UVLO triggering and heat dissipation.

The challenges faced during this project, such as high inrush currents, GPIO expander pull-up issues, and missing grounds, were systematically addressed, leading to a robust and reliable motor control system. The careful selection of components, from the 32-bit quadrature encoders to the choice of SPI and I2C protocols, ensured that the system met the project's requirements while remaining adaptable for future expansions.

Throughout this project, I extensively applied the principles and techniques learned in my coursework. Theoretical concepts from my electrical engineering studies were essential in designing and optimizing the circuits, while the practical lab work honed my ability to troubleshoot and resolve issues effectively. The hands-on experience gained in this project has deepened my understanding of motor control systems and reinforced the value of integrating academic knowledge with real-world applications.

In conclusion, this project not only achieved its primary objectives but also laid a strong foundation for further developments in the field of precise motor control. The insights gained and the solutions developed through this work have the potential to contribute significantly to applications requiring high precision, reliability, and user feedback, particularly in haptics and telerobotics.

Note: *The project specifications were determined by professor Kourosh Zareinia and based on it, the project was completely started from scratch by me. And all the progress mentioned above was achieved by me under the guidance of my mentor, Professor Kourosh Zareinia.*

Wiring The Entire System: Link

14 References

1. https://www.youtube.com/watch?v=yO9xIVv8ryc
2. https://www.youtube.com/watch?v=O5qMxY4apoI
3. https://www.pololu.com/product/1451
4. https://www.mikroe.com/counter-click
5. https://www.allaboutcircuits.com/technical-articles/how-to-design-simple-voltage-controlled-bidirectional-analog-current-source/

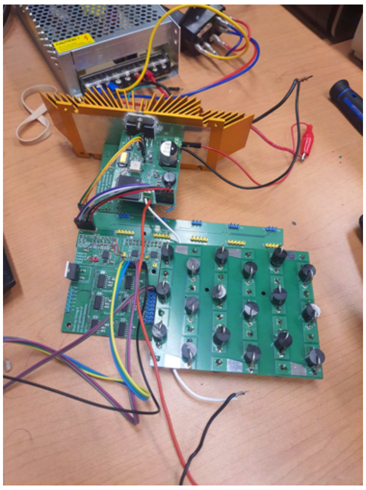

Figure 24: Fully Assembled PCBs

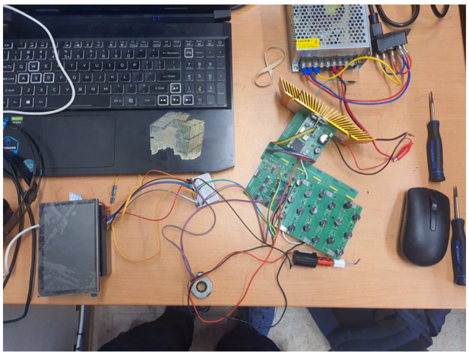

Figure 25: Fully Assembled PCBs with computer, screen, motor and encoder

Figure 26: Me with my mentor Professor Kourosh Zareinia.

Declaration:
I hereby grant permission to Chirag Kotian to submit this report to IIT Tirupati for evaluation as part of the compulsory internship course required for credit in his B.Tech in Electrical Engineering program. Additionally, I authorize him to publish this report on his public GitHub repository as evidence of his work.

Dr. Kourosh Zareinia