Sensorless Contact Force Estimation on Raven II Surgical Robot Platform

Tin Chiang

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2024

Committee:

Blake Hannaford

Eric Seibel

Sawyer Fuller

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Sensorless Contact Force Estimation on Raven II Surgical Robot Platform

Tin Chiang

Chair of the Supervisory Committee:
Blake Hannaford
Department of Electrical & Computer Engineering

Haptic feedback in surgical robotic arms has long been a challenging area due to factors such as the need for sterilization, which complicates the mounting of force sensors on the end-effector, and the complex, nonlinear nature of cable-driven systems. In this work, I present a data collection procedure that enables contact force estimation without direct sensor measurements, using a learning-based approach. Additionally, a 6-degree-of-freedom (DOF) cable-driven haptic device was developed to apply known contact forces over a wide workspace. The device includes multiple direct-current (DC) motors and shear beam load cells, and a real-time feedback control system implemented using the Zephyr Real-Time Operating System (RTOS). A high-level controller synthesizes the desired contact forces.

The device was tested on a Raven II Surgical Robot Arm, demonstrating its capability in applying precise contact forces, though certain limitations in the setup were identified. A one-directional contact force estimation experiment was conducted with a random force applied along the Y-axis, while the Raven-II followed a random trajectory. Three neural network models—MLP, LSTM, and GRU—were trained using the collected data, achieving mean absolute errors (MAE) of 0.5859, 0.6664, and 0.7486 N, respectively, within a force range of ±4.5 N. Further analysis revealed a correlation between joint velocities and estimation error, providing insights for designing more efficient training trajectories.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

First of all, I would like to express my sincere appreciation to my parents for their unwavering encouragement and support throughout my education at the University of Washington. Secondly, I would like to thank Professor Hannaford and Haonan Peng for giving me the opportunity to work on this project and for their mentorship throughout the process. Last but not least, I would like to thank all the members of the BioRobotics Lab at the University of Washington for their emotional and academic support. Thank you all for being a part of this amazing journey.

Chapter 1

INTRODUCTION

This chapter briefly summarizes the problem addressed in this thesis, and summarizes the thesis's objectives.

1.1 Problem Statement

Haptic feedback has always been a popular research area for surgical robots. Currently, the majority of endoscopic surgical robots on the market don't have the haptic feedback feature available. As a result, operators have no knowledge of how much force they apply to patients' tissue, relying heavily on their experience and visual feedback to control the robot. The loss of sensory feedback introduces difficulties and risks to both surgeons and patients.

To enable haptic feedback, knowing the contact force at the robot end-effector is crucial. On an industrial robot, a force sensor can be placed on the end-effector to get a direct measurement of the contact force. However, the end-effector of an endoscopic surgical robot is usually a purely mechanical structure due to the need for sterilization. Any attempt to mount a force sensor on the grasper must address the problem of sensor calibration after harsh sterilization procedures.

An alternative approach is indirect estimation through the inverse dynamics model of the robot, which can be derived from the forward kinematic model. The servo system of Raven-II operates in position control mode and adjusts the motor torque by controlling the current until the motor reaches the desired position. If external forces act on the end-effector, the motor torque will increase correspondingly to maintain the position. With this characteristic, contact forces can be estimated indirectly, allowing the surgical tool to remain electronics-free for easier sterilization. However, the unique design of endoscopic surgical robots introduces many uncertainties and complexities in modeling.

To reduce size and improve sterilizability, endoscopic surgical robots usually adopt a cable-driven design, meaning the actuators are not placed on the joints but remotely on the base frame. Each actuator controls a joint through a pair of cables running through a series of pulleys, which guide them from motor to joint. This design reduces the bulkiness of the arm structure but introduces many undesired characteristics for modeling the robot. Firstly, the cable itself is elastic and will stretch under motion. The amount of stretch depends on many factors, such as cable tension, joint motion, and the plastic deformation of the cable over time. Additionally, each pulley along the cable path introduces friction, which is highly non-linear and can also be affected by cable tension. Lastly, joints are coupled with each other. For joints that are further from the motors, for example, joint 4, its cable pair needs to run through joints 1, 2, and 3. Any motion of joint 4 can affect joints 1 to 3, and vice versa. As a result, modeling the robot through traditional methods is very complicated and subject to significant uncertainties.

Due to the complexity and uncertainty of the robot, we decided to take a data-driven approach to model the robot, instead of using a physics-based approach. The hypothesis here is that by providing proper contact force ground truth on the end-effector and the robot state information, the neural network will be able to learn the inverse dynamics relationship of the robot and tackle the non-linearity and uncertainty of the cable-driven mechanism.

1.2 Objectives

The primary objective of this thesis is to develop a process of collecting robot state data of Raven-II with known contact forces acting on the robot grasper. One of the most important aspect of data-driven modeling is to ensure the diversity and coverage of the training data, which in our case, means the training data should include scenarios of robot subjected to a wide variety of contact forces(magnitudes and directions) under various of robot configurations(joint poses, joint velocities).

Since there's no existing device that can provide such "haptic feature" to the surgical robot with sufficient coverage in robot workspace, developing such device becomes a necessary step for this research. With all the equipment and resources available at BioRobotics Lab, driving a cable connected to the robot grasper with motor to provide forces was the most viable option. To further elaborate, objectives below are set for this thesis:

1. To develop a process of controlling cable tension using motor.
2. To develop a process of simulating contact force using multiple set of motors and cables.
3. To construct such system, develop associated software and hardware.
4. To test the system on Raven-II surgical robot platform, collect and process training data.
5. To train a neural network with collected data and evaluate the model performance on contact force estimation.

1.3 Summary of Contribution

The following contributions to the state of the art are delivered with this thesis:

- I designed, built, and tested the performance of the "force unit", a cable tension control device. The device consists of a DC motor, a shear beam load cell, and a 3D printed cable reel.
- I combined 6 force units to apply controlled forces in Cartesian space to the manipulator end-effector.
- I designed, built, and tested a motor control board to interface with the servo system of Raven-II Surgical Robot.
- A low-level controller that can maintain cable tension under passive motion was developed.
- A high-level controller that can synthesize contact force with 6 force units was developed.

- A data collection procedure for collecting robot state information with contact force applied were developed.

- The preliminary results for the learning-based inverse dynamics modeling are also provided.

Chapter 2

BACKGROUND

This chapter briefly introduced the minimally invasive surgical robot platform used in this thesis and summarized different approaches on estimating the external force acting on the robot grasper.

2.1 Robot-Assisted Minimally Invasive Surgery(RAMIS) and Raven II Research Platform

Minimally invasive surgery (MIS) has gained significant popularity due to its smaller incisions and shorter recovery times. Traditional MIS techniques require the surgeon to operate beside the patient, utilizing long laparoscopic instruments, as depicted in Figure 2.1. The motion of the instrument's end-effector is inverted relative to the surgeon's hand movements, making the procedure highly counterintuitive and challenging.

The introduction of endoscopic surgical robots has brought significant benefits to both surgeons and patients. The robotic controller precisely scales the surgeon's hand movements to a finer level, eliminating the cognitive burden of compensating for the reverse motion of traditional surgical tools. Additionally, these systems enable surgeons to perform procedures in an ergonomic position, reducing fatigue and allowing for longer operations. The enhanced precision and dexterity provided by surgical robot systems minimize blood loss and lower the risk of infection at the incision site.

The Raven-II, the RAMIS research platform utilized in this thesis, was developed in 2012[8] to provide an open-source software and hardware architecture, facilitating advancements in robotic surgery research. Built upon the successful design of the Raven-I[15], which was informed and optimized using clinical laparoscopic surgery data, the Raven-II features a link geometry tailored to the workspace typically required by surgeons. Its unique cable-actuation system significantly reduces the size of the arm structure, enabling a compact design that can be easily deployed in various settings.

Figure 2.1: Traditional Minimally Invasive Surgery

Source: https://precisionsurgeryaz.com/wp-content/uploads/2024/11/minimally-invasive.webp

The Raven-II further improved upon its predecessor by simplifying cable wiring and motor base design. The adoption of brushed DC motors streamlined the control electronics, enhancing system robustness. Moreover, the platform incorporates the ROS (Robot Operating System) and an Open TCP/IP communication interface, enabling greater functionality expansion and improved accessibility to robot control.

2.2 Force Estimation on Surgical Robot

2.2.1 Direct Measurement

To detect external forces acting on the end-effector, the most straightforward approach involves placing sensors on the gripper to directly measure forces. The Miro surgical robot[7] utilizes a resistive strain gauge design[24] on the wrist of the end-effector to achieve force feedback during teleoperation. A research group from Harvard[23] introduced a novel printing technique for strain gauges, enabling the direct application of low-cost, low-profile sensors onto the surface of the end effectors of da Vinci surgical robots (Intuitive Surgical, Inc.), capable of detecting deflection and providing force feedback. Additionally, researchers at UW[11] developed an innovative surgical tool incorporating a four-degree-of-freedom (DOF) force sensor based on capacitive transduction principles. This sensor directly measures both normal and shear forces at the tips of surgical instruments.

Despite the promising advancements of sensor-embedded surgical tools, sterilizability remains a significant challenge due to the materials and electronics involved. Researchers have been investigating protective materials that can endure multiple autoclave sterilization cycles while maintaining sensing performance[28]. Furthermore, advancements in low-temperature sterilization methods, such as hydrogen peroxide and ethylene oxide sterilizers, present viable options for reprocessing heat- and moisture-sensitive sensorized surgical tools[29]. Ongoing efforts in developing new coatings, adhesive materials, manufacturing processes, and sterilization technologies continue to support the viability of sensorized surgical tools for force estimation in clinical settings in the future.

2.2.2 Model-Based Force Estimation

Due to the limitations of sensor-based surgical tools, sensorless force estimation has become a prominent research direction. When one of the earliest prototypes of surgical robots, the Black Falcon, was developed at MIT[16], researchers sought to achieve force feedback using a model-based approach. By utilizing joint torques and joint poses, forces were directly reflected to the operator through an inverse dynamics model. However, most of the reflected forces in this method were inertial forces necessary for manipulating the robot arm, which were reported to be disruptive.

Subsequently, researchers at JHU attempted to address this issue by compensating for Coulomb friction and inertia at each joint. Nevertheless, their approach did not account for the elasticity of the cable tendons driving the robot. Researchers at UW[6] later adopted the dynamic model described in [5], incorporating joint friction, cable stiffness, and damping. Despite these advancements, identifying the physical parameters of cable elasticity and system friction requires extensive experimentation. These parameters can vary between different robots and even within the same robot over time due to cable wear and stretching under tension. Recalibrating these parameters for every robotic configuration demands significant effort, posing a substantial challenge to this approach.

2.2.3 Data-Driven Force Estimation

In recent years, with the increasing attention given to machine learning (ML) and data-driven techniques, researchers have begun employing these technologies for system identification in cable-driven surgical robots. For non-parametric machine learning methods, [13] proposes a Gaussian Process Regression (GPR) based scheme to address gripping force estimation. GPR utilizes a kernel function to compute the covariance matrix between input data and makes predictions based on a Gaussian distribution. This approach effectively handles uncertainty and non-linearity. The author further combined GPR with the Unscented Kalman Filter (UKF) for force estimation, demonstrating superior performance compared to model-based approaches.

Aside from non-parametric ML methods, academia has also explored the advantages of neural networks in directly extracting relevant features from datasets to identify the system dynamics of surgical robots. In an early work [25], force estimation was achieved using a neural network-based approach for a 3-DOF Planar Twin-Pantograph haptic interface. While this mechanism is not cable-driven, it demonstrated the feasibility of neural network approaches. For cable-driven surgical robots, several researchers have utilized Feed Forward Neural Networks (FFNN) to model the nonlinear relationship between motor behaviors, robot kinematics, and dynamics. For instance, the research group at Johns Hopkins University [30] employed FFNNs to identify the torque requirements for robot dynamics, gravity compensation, and internal forces such as friction and cable elasticity in the dVRK surgical robot. Using joint poses, velocities, and applied torque information during free motion, the researchers isolated internal joint torques from applied torques, enabling the calculation of external forces through the robot's kinematics model.

This physics-informed machine learning approach delivered accurate force estimation on the dVRK platform with a straightforward data collection procedure, allowing the robot to be modeled before each use. However, the method remains vulnerable to accuracy loss if the underlying robot model is flawed. For example, in the Raven-II robot, joint position and torque information are derived from encoder readings and motor torque commands, which are located far from the actual joints. The unique cable-actuation system introduces complex and nonlinear errors into the robot state information, complicating the acquisition of an accurate robot model.

As a result, the purpose of this thesis is to further explore the potential of learning-based approaches and evaluate the feasibility of achieving sensorless contact force estimation without relying on underlying assumptions about the robot model.

Chapter 3

DESIGN OF CABLE-DRIVEN HAPTIC DEVICE

The device has 6 ”Force Units”, each unit consists a DC motor, combined with a shear beam load cell, a 3D-printed motor frame, and a cable reel. The force unit will control the tension on one cable connected to the end-effector of the Raven-II.

The device is controlled by the servo system of the Raven-II Surgical Robot Platform developed in the BioRobotics Lab at the University of Washington[8, 15]. The device repurposed DC motors, servo drivers, and power system from the Raven-II.

Through out the research, two versions of low-level feedback controller were designed. The first version gained control to servo motors by modifying the control software of the Raven-II. The second version gained direct control on the hardware through a newly-designed control board.

A high-level inverse dynamic controller was developed to calculate the desire tension on each cable to synthesize the desired contact force at the end-effector.

3.1 Requirements

The following are requirements that the device should satisfy:

1. Each force unit shall maintain the cable tension regardless of the motion of the end-effector for feasible displacements and velocities.

2. The device shall have a calibration procedure to obtain the location and orientation of each force unit with respect to the Raven-II’s coordinates.

3. The device shall be aware of the location of the end-effector within the device work space.

4. The device needs to determine the tension on each cable based on the current end-effector position and desired contact force.

5. The device should always maintain a minimal tension on each cable to prevent slackness.

3.2 Force Units

”Force unit” is the fundamental component that provides force to the end-effector of the Raven-II (Figure 3.1). Each force unit has a cable connected to the robot, and is providing force by driving a cable reel using DC motor.

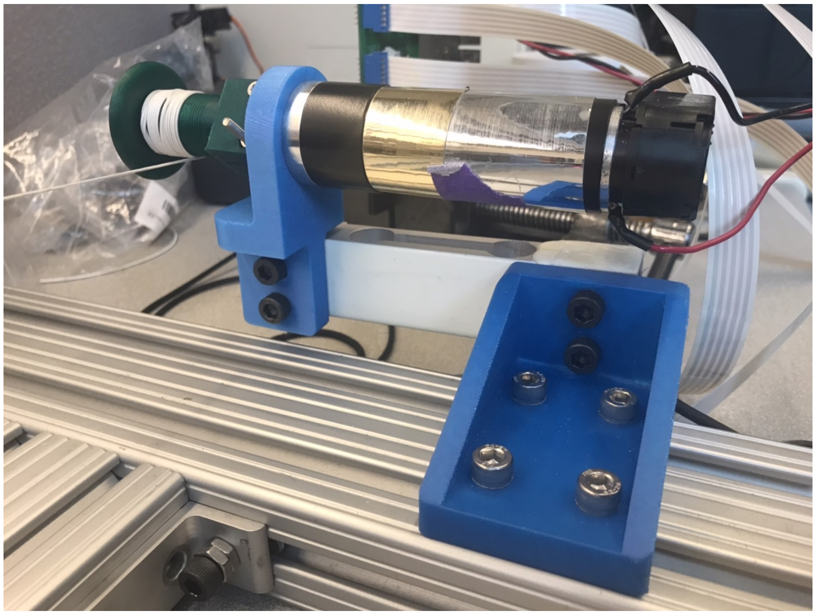

Figure 3.1: A single Force Unit

The current device has 6 force units mounted around the left arm of Raven-II to provide multiple force components from different directions. Each cable provides a force vector, its direction points from robot grasper to the force unit, and its magnitude is the cable tension. All six force vectors are acting on the same point on the grasper to simulate a contact force in the Cartesian space.

To assure the system can simulate forces with a wide variety of directions and magnitudes, cables shall pull the grasper from 3 orthogonal directions in space, and every two force vectors shall be approximately antiparallel to each other to provide a full degree of freedom on that particular direction. An illustration in CAD along with Raven-II's left arm can be seen in Figure 3.2, 3.3, and 3.4. The actual system setup is shown in Figure 3.5 and 3.6.

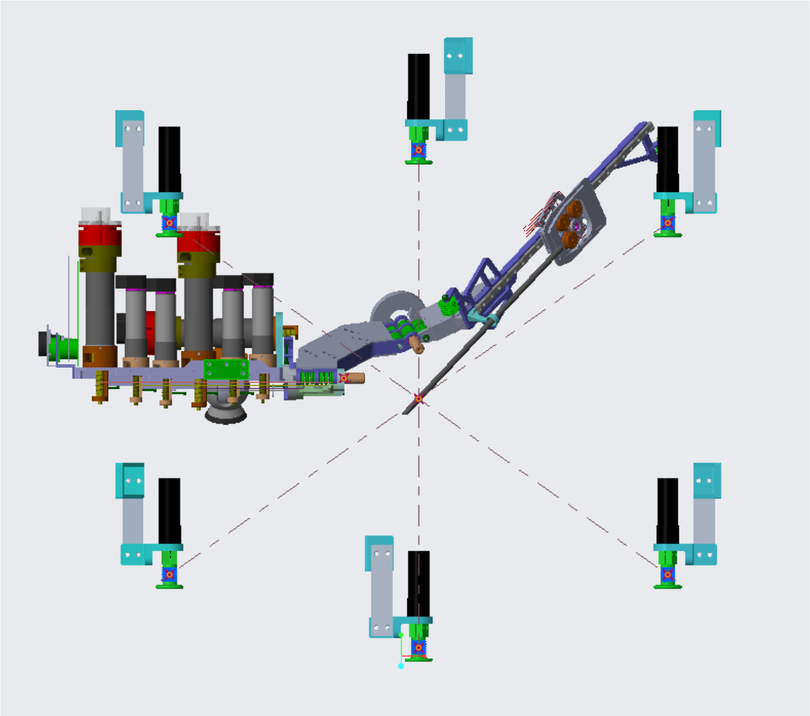

Figure 3.2: Front view of the force units setup(without the supporting frames for each unit) with the left arm of Raven-II.

3.2.1 Structure

A force unit is composed of 4 main components: a DC motor, a cable reel, a motor frame, and a shear beam load cell. The structure of the force unit allows the direct control of the cable tension through controlling the torque of DC motor. A shear beam load cell sits between the cable reel and the base frame, so that the tension of the cable can be measured

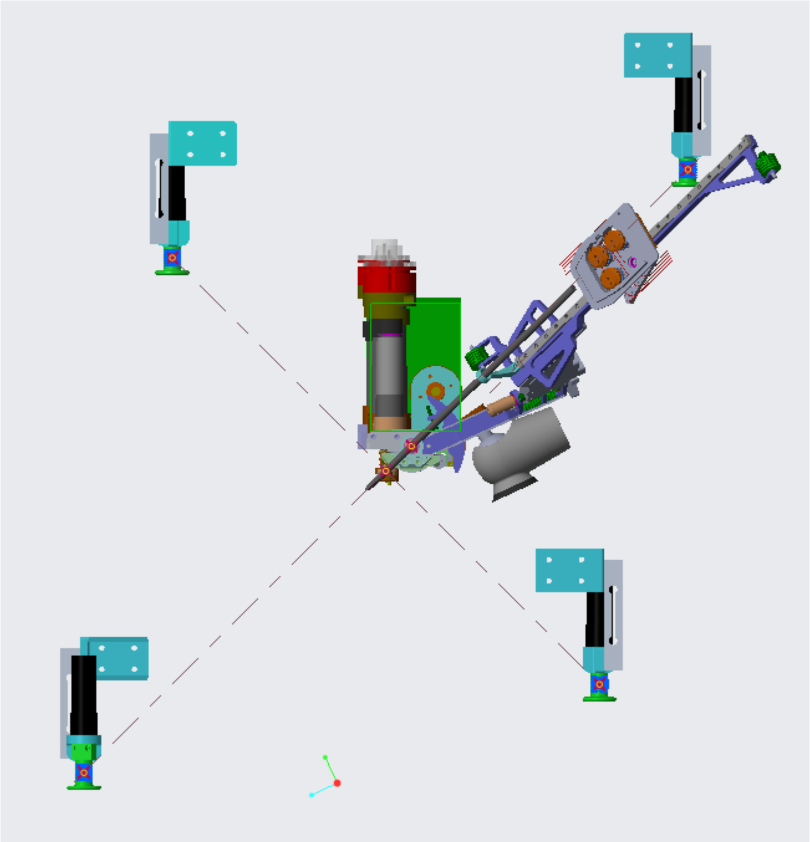

Figure 3.3: Side view of the force units setup(without the supporting frames for each unit) with the left arm of Raven-II.

by the shear beam load cell. However, the shear beam load cell can only measure the force in 1 direction, which the cable won't always be aligned with. The angle difference between cable and the measurement direction will be compensated by the controller.

3.2.2 DC Motor

Each force unit requires a DC motor to convert electric energy to cable tension. The quality of the motor is critical to the force control application, because the internal friction and the torque ripple can all directly affect the control of cable tension.

The current force units utilize same the DC motors from joint 4 to joint 7 on the Raven-II. The DC motor is custom-made by Maxon, the coreless design of the rotor allows the motor

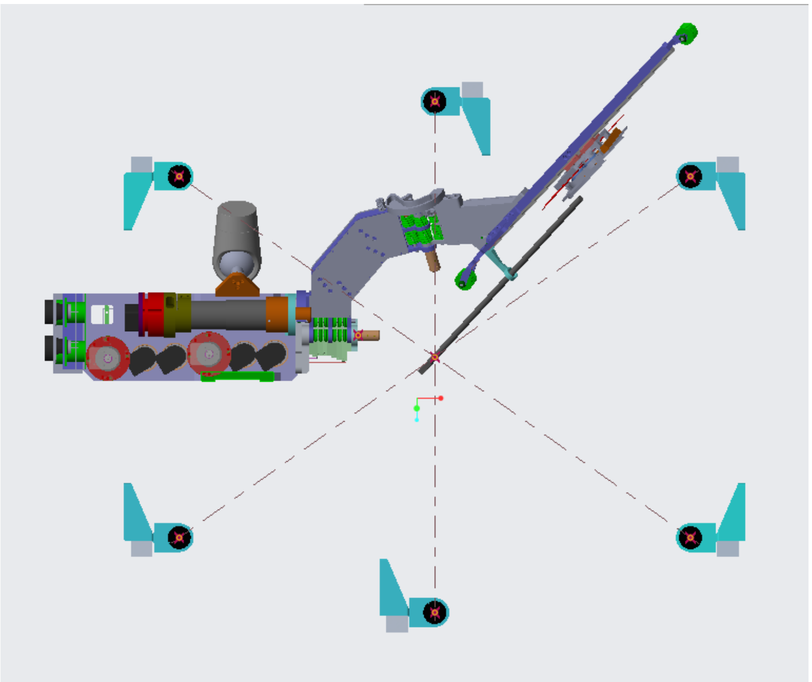

Figure 3.4: Top view of the force units setup(without the supporting frames for each unit) with the left arm of Raven-II.

to operate with minimal torque ripple. The motor offers 52.1 mNm/A of force constant, and has a planetary gear head of ratio 3.7:1.

3.2.3 Sensor

The shear beam load cell embedded in the motor frame provides a feedback signal for the control system to adjust cable tension. This load cell is a type of Wheatstone bridge sensor, consisting of four strain gauges acting as resistors to form a bridge circuit. These strain gauges are integrated with a metal bending beam structure. When external forces are applied, they deform the bending beam, causing the strain gauges to react and change their resistance. This change disrupts the balance of the bridge circuit, resulting in a measurable potential difference across the bridge.

Depending on the application, shear beam load cells come in a wide variety of measurement ranges. For the purpose of this thesis, which involves applying forces to the Raven-II

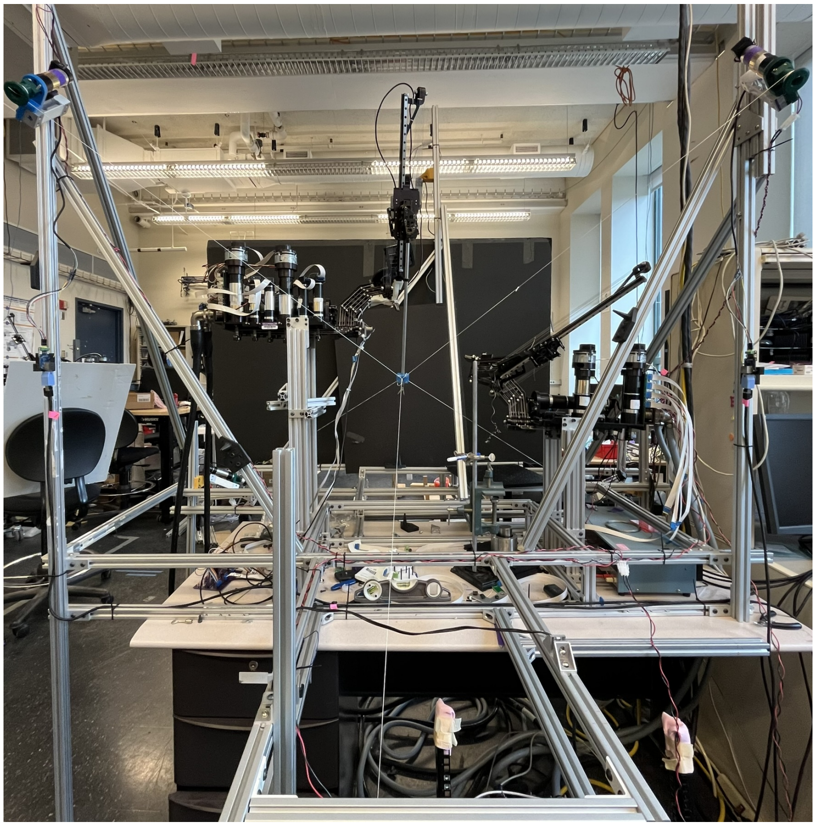

Figure 3.5: Actual force units setup with the left arm of Raven-II.

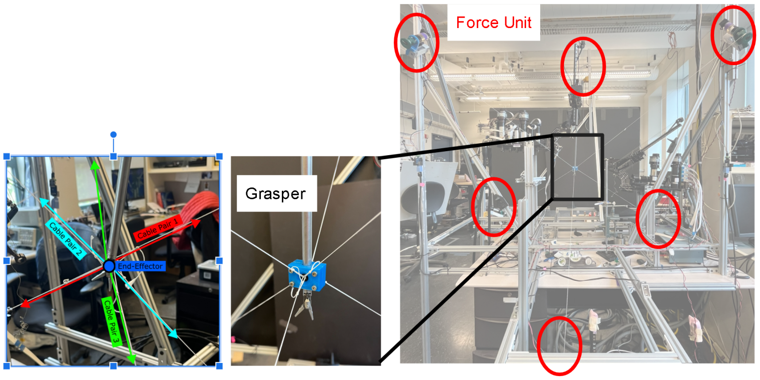

Figure 3.6: Cable connection setup.

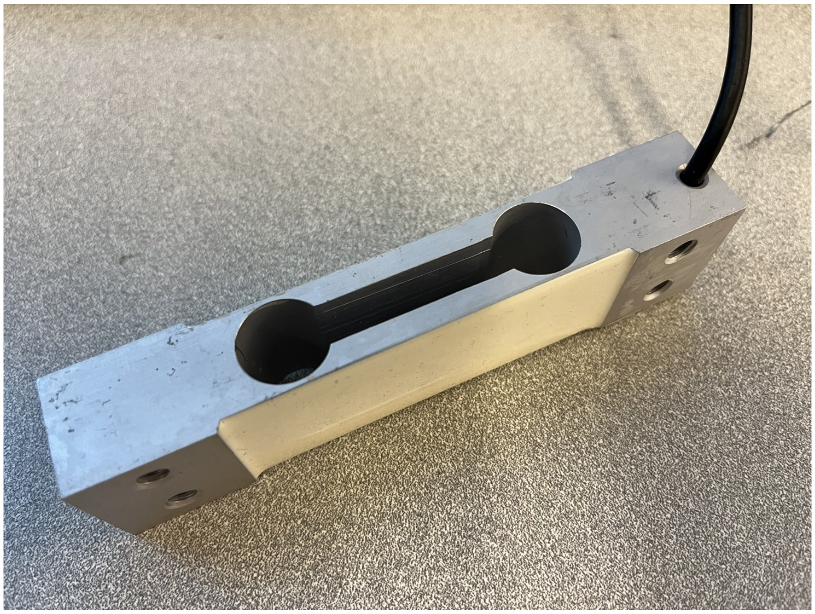

Figure 3.7: A shear beam load cell

end-effector, the tension on each cable will not exceed 1 kg to avoid damaging the surgical robot. Considering this requirement and allowing for a margin to compensate for angled cables, we decided to use a load cell with a measurement range of 3 kg.

3.2.4 Motor Frame and Cable Reel

All of the components mentioned here are designed using Creo Parametric by PTC, and 3D-printed with PLA (Poly lactic acid) using Prusa i3 MK3s.

The motor frame is composed of two parts: a motor mount that connects the DC motor to the load cell, and a base that connects the load cell to the fixed frame. The motor mount assures the orientation of motor shaft

The cable reel converts the motor torque to cable tension. The current combination of Raven's DC motor, gearbox, and driver can easily provide torque that far exceeds our need. Therefore, the radius of the cable wheel should be small to increase the control resolution of the system.

Current version of the cable wheel has a radius of 1 cm,

3.3 Sensor System

The shear beam load cell on the force unit is an analog sensor, and it requires dedicated electronics to amplify and convert its signal to digital for the controller. The selected analog-to-digital(ADC) chip is the NAU7802 by Nuvoton. The IC offers a universal I2C interface, a relatively high sampling rate of 320Hz compared to other load cell ADCs, and is both accessible and well-supported.

The system shall meet the following requirements:

- Read 6 loadcells at 320Hz.
- Transfer the readings to PC at the same rate.
- The ADC chip will be located close to the load cell, but far from the microcontroller. Signal integrity must be maintained over long cables connecting the two.
- The system should be resistant to electromagnetic interference (EMI) present in the lab environment, particularly near motors and power cables.

The system consists of two main parts: a microcontroller and a sensor module. The microcontroller side includes an Arduino Mega board with an Ethernet shield and an I2C multiplexer (since the NAU7802 chips have a fixed device address). The sensor module features a NAU7802 breakout board and a custom PCB that receives power and I2C signals via an ethernet port. It also includes an LTC4311 bus accelerator from Analog Devices to maintain signal integrity.

A hardware timer triggers the system to read data from six load cells every 1/320 of a second. Subsequently, the system transmits these six readings to a PC via the serial port.

The first prototype was built using unshielded cables and without a bus accelerator, the device frequently failed to initialize either the multiplexer (MUX) or the analog-to-digital converter (ADC). Additionally, the I2C bus would often lock up while the robot was in operation.

Upon investigating the issue, I determined that the problem stemmed from poor signal integrity. The oscilloscope revealed two main issues with the I2C signal:

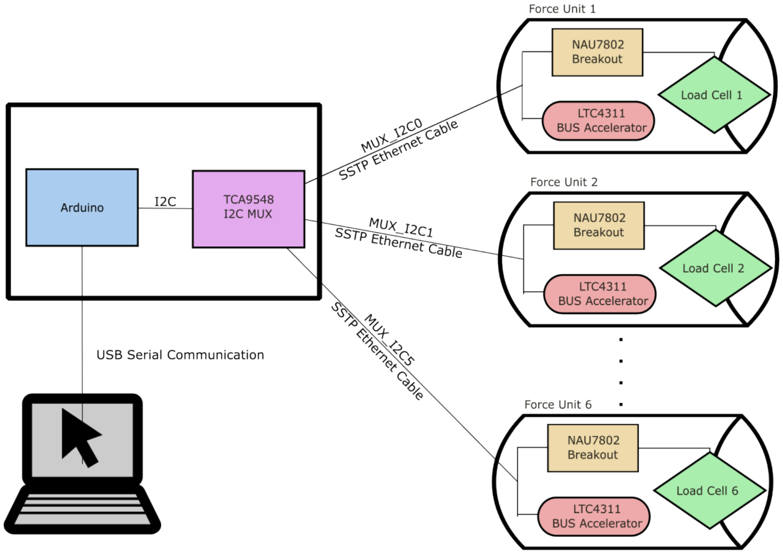

Figure 3.8: Sensor System Diagram

- Slow rise times due to the capacitance of long cables.
- Cross-talk between the clock and data lines.

The exact cause of the I2C bus lockup was difficult to pinpoint, but the failure consistently occurred when the signal cable was near any powered motor. The unshielded signal cable acted as an antenna, picking up electromagnetic interference (EMI), which could lead to bit-flips and cause the I2C controller to enter an incorrect state.

To address the slow rise time issue, I first evaluated the I2C pull-up resistors on my system. With three sets of 10k ohm resistors in parallel (on the Arduino board, TCA9548 breakout, and NAU7802 breakout), the effective resistance was 3.3k ohm. Any attempt to further decrease the resistance caused the bus to fail.

To improve performance, we opted to use the LTC4311 from Analog Devices to actively pull up the I2C bus near the ADC chip. The available LTC4311 breakout boards require extensive jump-wiring, adding to cable length and complexity. Therefore, a custom PCB was designed and built to integrate the NAU7802 breakout, LTC4311, and cable connections seamlessly. The combined module will be placed next to the force unit.

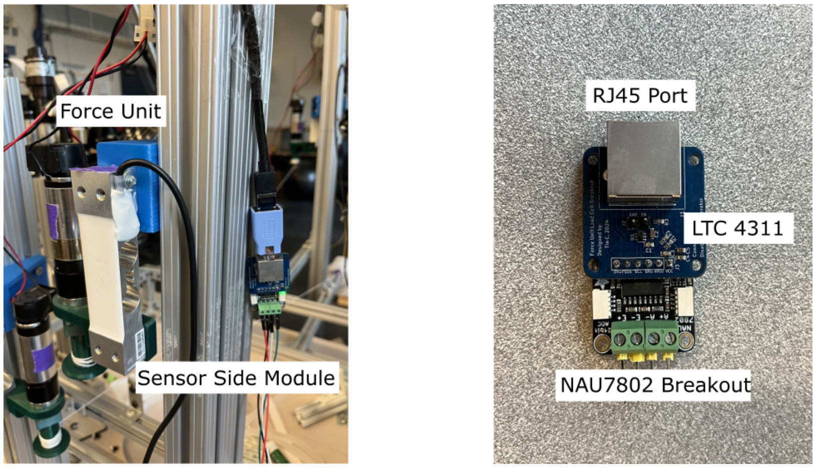

Figure 3.9: Sensor-side Module

Each sensor-side module requires six wires: two power lines, a clock line, a data line, a DRDY trigger line, and a ground line. To minimize noise and crosstalk, the power lines should be isolated from the signal lines, and each signal line should be properly shielded. Additionally, the entire cable should be shielded from environmental EMI.

A Screened Shielded Twisted Pair (SSTP) RJ45 cable is ideal for this application. Each twisted pair is individually shielded with metal foil, allowing power lines to run without affecting the signal lines. Each signal line can use its own twisted pair along with a ground line. An additional shield over the entire cable further reduces environmental EMI. There aren't any existing products that allow such usage of SSTP RJ45 cable. As a result, a custom Arduino shield board was built to interface with a TCA9548 I2C Mux and 6 sensor-side modules.

With the LTC4311 bus accelerator and SSTP cable, the I2C signal integrity is significantly enhanced, providing ideal rise times and minimizing crosstalk. Additionally, the proper shielding around the cables ensures that the system can reliably operate in proximity to the robot.

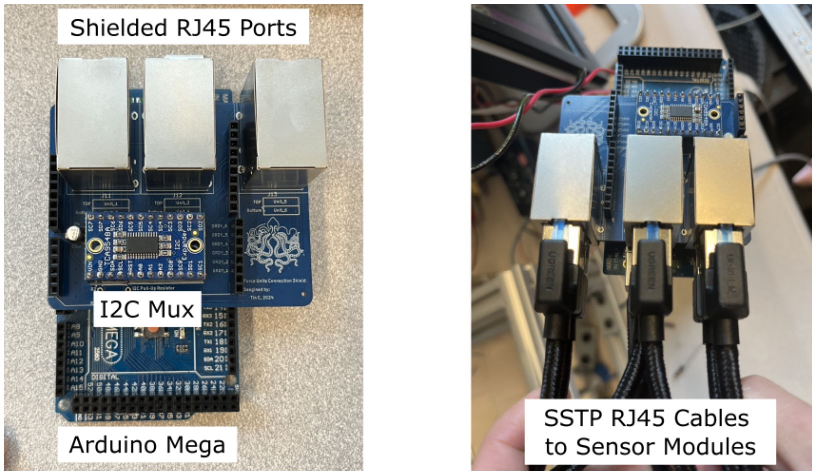

Figure 3.10: Arduino shield for connecting the sensor-side modules.

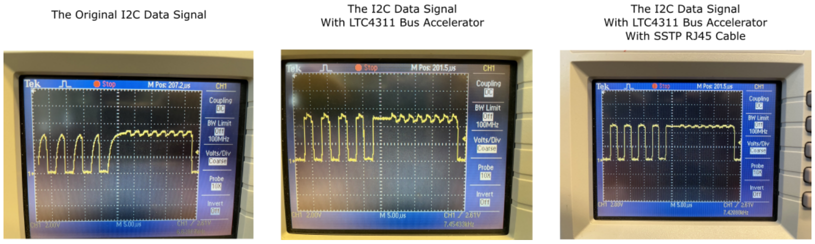

Figure 3.11: I2C Signal Integrity Improvement

3.4 Servo System Electronics

The DC motors on force units are driven by the same servo system of Raven-II Surgical Robot. The Raven-II system features high-quality current-control electronics and DC motors, making it ideal for force control applications. The electronics are highly modular and can be divided into three main components:

1. Power Box: Contains dual power supplies and a PLC for safety functions.
2. Control Box: Houses motor drivers and a USB board that handles motor control and feedback signals to the control PC.
3. Control PC: An RT Linux machine that runs Raven-II's control software and the ROS

API for research purposes.

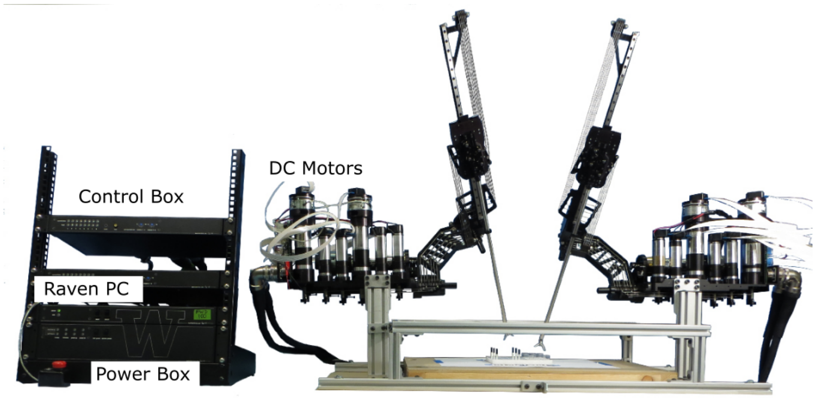

Figure 3.12: Servo System Components of Raven-II

3.4.1 Motor Driver and Power System

Each Raven-II system has one power module, as known as "the power box". With in the box, there are 3 main components(figure 3.13):

1. A 48V power supply that drives 14 DC motors on 2 robot arms.
2. A +/-15V power supply for the logic units and digital-to-analog converter chips that control the servo motors.
3. A programmable logic controller(PLC) unit that controls the power relay to 48V power supply and enables/disables the motor drivers based on input such as switches, E-stop button, and surgeon pedals.

The relationship between each components are shown in figure 3.14. The power box convert the AC power into 3 DC power sources with different voltages, and then send to the control box of each arm along with the PLC control signals through a Mil-Spec power cable.

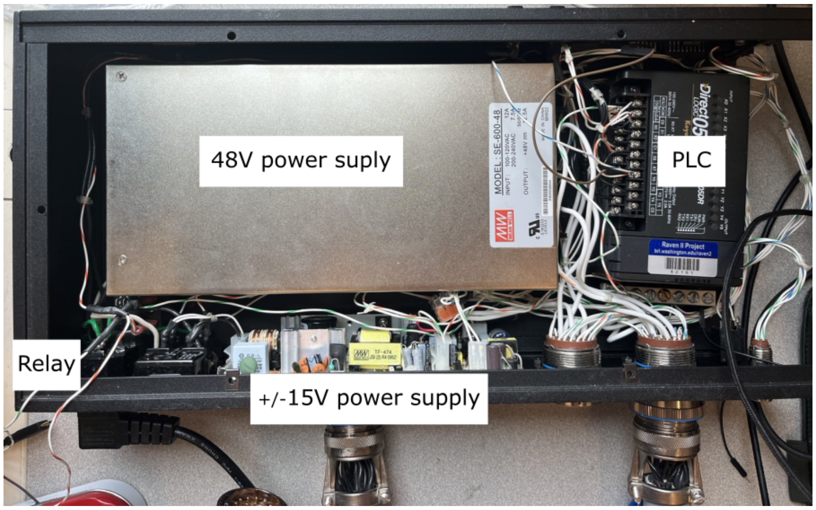

Figure 3.13: Power Box of Raven-II

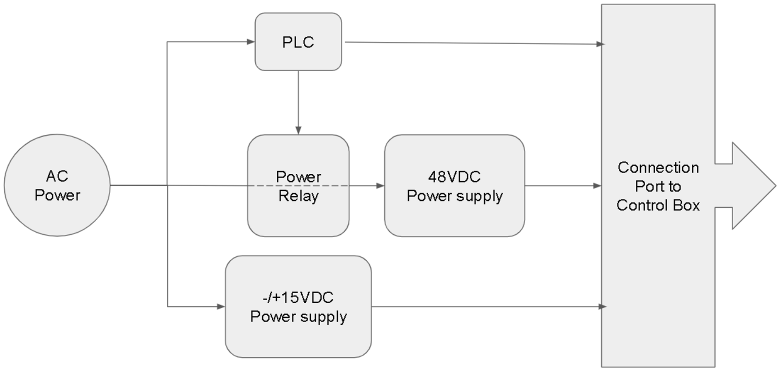

Figure 3.14: Power Box of Raven-II

There are two arms in each Raven-II system, each arm is controlled by an ”controller box”. The controller box receives power from the power box, and receives control signals sent from the control PC. Within the box, there are 3 main PCB components(fig 3.15):

1. Connection Board: The board interfaces with three Mil-Spec connectors to distribute power sources, control signals, and feedback sensor signals to other boards in the box.

2. Motor Driver Board: The board is in charge of the high-power section of the system. It has 7 servo amplifier modules (Advanced Motion Control, Z6A6, Z12A8) mounted

on it, which control the amount of 48V current that flow into DC motors of the robot arm.

3. USB Board:

The board is responsible for the communication, control and feedback signals of the system. It has an 8-bit microcontroller unit that communicates with control PC through USB, an array of bipolar digital-to-analog converters(TI, DAC7731) that control the motor driver, and an array of quadrature counter(LSI/CSI, LS7266) for receiving signals from optical encoders on motors.

The relationship between each components are shown in figure **??**.

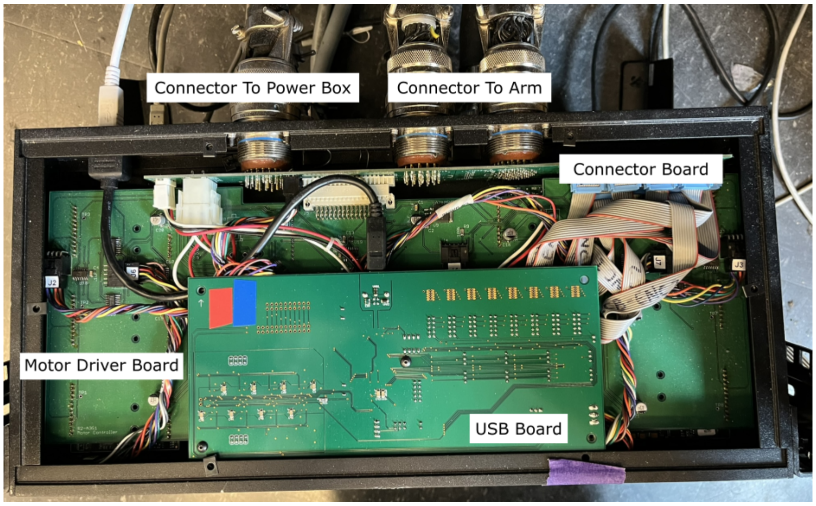

Figure 3.15: Controller Box of Raven-II

3.4.2 Motor Control Process

The motor control process for the Raven-II servo system begins with receiving joint position commands from either the inverse kinematics module or direct joint space control. These commands are processed by a low-level feedback position controller, which integrates encoder feedback. The controller outputs motor torque commands (Nmm), which are then

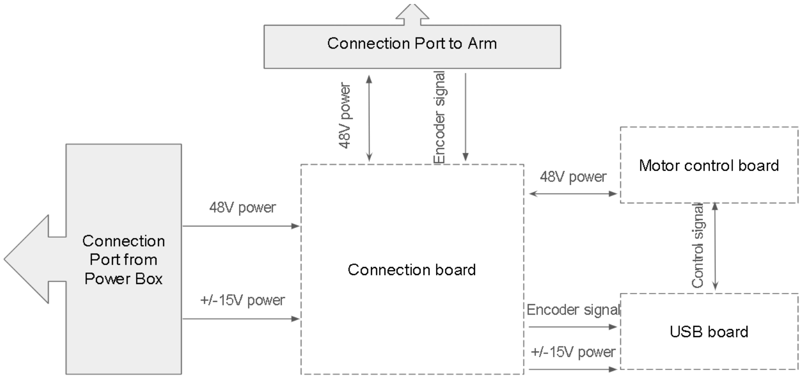

Figure 3.16: Component Relationship in Controller Box of Raven-II

converted into current commands (A) based on motor specifications and hardware configurations, such as gear ratios. The current commands are subsequently transformed into appropriate DAC values, determined by the DAC chip's output range, resolution, and the voltage-to-current conversion ratio of the servo amplifier.

This process occurs on Raven's control PC. Once the DAC values are obtained, they are transmitted to the USB board in the control box via USB. The MCU on the board uses SPI(Serial Peripheral Interface) to communicate with the DAC7731 IC, generating analog voltage signals ranging from ±10V. These signals are then sent to the motor driver board, where the servo amplifier modules regulate the 48V current supplied to the DC motors.

3.5 System Calibration

The force sensor system requires multiple calibration procedures to simulate the desired contact forces accurately. These include calibrating the force sensors and determining the precise location of each force unit relative to the right arm of the Raven-II system.

3.5.1 Force Sensor

The force sensor uses the NAU7802 module to excite and measure the differential output from its Wheatstone Bridge circuit. The sensor provides a linear response between the applied load and the voltage difference within its measurement range, which the NAU7802's

ADC samples and converts into a digital signal.

Since the digital output is unit-less and represents only the signal-to-output ratio of the ADC, two types of calibration are required to assign physical meaning to the measurement:

1. **Zero Offset:** Zero offset is an error that shifts the sensor output from its true value. This can result from various factors, including temperature fluctuations, manufacturing tolerances in both the force sensor and ADC module, and sensor configuration.

   In this thesis, an additional source of zero offset arises from the installation angle of the force unit. The motor and rack assembly weighs approximately 500g, and each unit is positioned at a unique angle so that the load cell's measurement axis is aligned with the workspace center of the Raven-II's left arm. Depending on the setup, the weight of the motor and rack can differently affect the force sensor's readings.

   To eliminate the zero offset, a set of ADC readings should be collected when the system is considered unloaded. The average of these readings is then subtracted from future measurements. This calibration must be performed each time the system is powered on with the cables being slack.

2. **Gain Offset:** Gain offset represents the deviation in the input/output relationship between the ADC output and the physical measurement, essentially reflecting the slope of this relationship. Proper calibration of the gain offset is crucial for translating the measurement values into physical units. Factors affecting gain offset include manufacturing variations between sensors, signal amplification gain, and ADC resolution.

   To calibrate the gain offset, a set of weight scales is required for ground truth. The force sensor should be placed horizontally, and weights of 10g, 20g, 50g, 100g, 200g, 500g, and 1kg should be applied sequentially. ADC readings for each weight are recorded and averaged. Seven data points are then fitted with a line using linear regression, and the slope of this line determines the gain offset calibration factor. All force sensors have been individually calibrated using this method before mounting on the force units.

3.5.2 Force Unit Configuration

The force unit system requires accurate position information of each unit with respect to Raven's surgical tool to calculate the cable directions. In addition, the direction of force sensor measuring axis on each unit is also needed for the controller to compensate the misalignment of cable direction and measurement axis.

With a 5-DOF coordinate measurement machine(MicroScribe MX, Immersion), accurate position of a point with respect to the MicroScribe's coordinate can be obtained. However, several additional requirements should be met to achieve configuration calibration for force units:

1. A method to locate the base frame of Raven-II's left arm.

2. A method to effectively scale up the measurement range to cover the entire force unit system.

To find out the location of force units with respect to Raven, defining the spatial relationship between coordinates of Raven and the measurement device is crucial. Such relationship can be expressed in homogeneous transformation matrix ${}^{Device}_{Raven}\mathbf{T}$, which describe the Raven's base frame in terms of the coordinate of measurement device. The location of Raven's base frame coordinate is difficult to access using the measurement device, as it doesn't have any physical feature that directly defines the origin and orientation(figure 3.18). Instead, a set of screw holes on the base frame, refer as $\mathbf{P_{Base}}$, are easy to locate with the measurement device, and their location with respect to Raven's base frame coordinate can be obtained from the CAD model(figure 3.19).

By properly positioning the measurement device, we can obtain the location of P_{Base} in both the Raven's base frame and the measurement device's frame, refer as ${}^{Device}\mathbf{P}_{\text{Base}}$ and ${}^{Raven}\mathbf{P}_{\text{Base}}$. Then, the ${}^{Device}_{Raven}\mathbf{T}$ can be obtained through Kabsch algorithm[10]. With the transformation matrix, any point within the workspace of MicroScribe can be measured and represent in Raven's Base frame. In our case, there are at least 4 points that we need to measure on a force unit. One point will be the contact point of cable and cable reel, which defines the cable direction along with the end-effector's location. The other three

Figure 3.17: MicroScribe MX 3D Digitizer

Source: https://shop.gomeasure3d.com/cdn/shop/products/microscribe-portable-cmm-m-series-in-action.jpg?v=1629412615

points will be on the load cell, which define a surface whose normal vector represents the measurement axis of the load cell. These points are referred as $\mathbf{P_{Cable}}$ and $\mathbf{P_{Loadcell}}$(figure 3.20). If the measurement device can reach both Raven base and the force unit, we will get ${}^{Device}\mathbf{P}_{\text{Base}}$, ${}^{Device}\mathbf{P}_{\text{Cable}}$, and ${}^{Device}\mathbf{P}_{\text{Loadcell}}$. Then, combine with the information from Raven-II's CAD model, ${}^{Raven}\mathbf{P}_{\text{Base}}$, transformation matrix ${}^{Device}_{Raven}\mathbf{T}$ can be calculated. By multiplying ${}^{Device}_{Raven}\mathbf{T}$ with ${}^{Device}\mathbf{P}_{\text{Base}}$ and ${}^{Device}\mathbf{P}_{\text{Cable}}$, we get ${}^{Raven}\mathbf{P}_{\text{Base}}$ and ${}^{Raven}\mathbf{P}_{\text{Cable}}$.

However, some force units locate further away from the Raven base, and the measurement device couldn't reach both without moving. In this scenario, a calibration plate was designed and built to extend the measurement range(figure). The main purpose of these

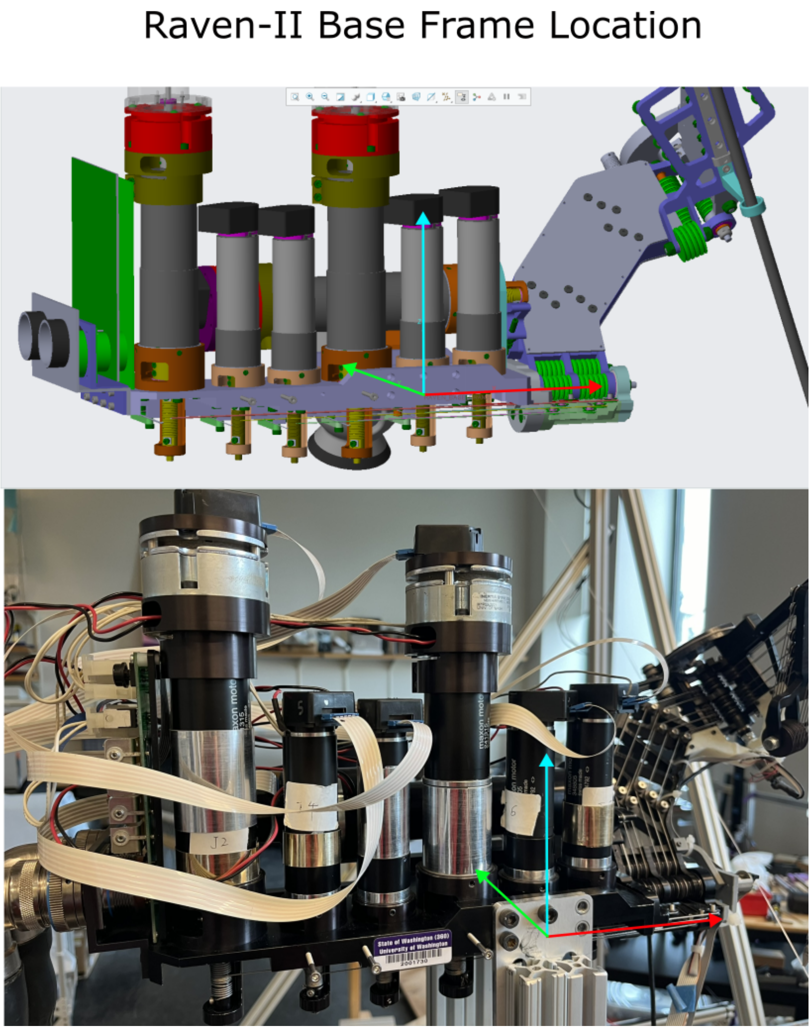

Figure 3.18: The Base Frame Location of Raven-II's Left Arm

plates is to establish a chain of reference coordinates, each of which has a known transformation with the previous and next coordinates. The establishment of the reference coordinate is done by measuring the location of three holes, denoted as $\mathbf{P}_{\text{Plate-1}}$, $\mathbf{P}_{\text{Plate-2}}$, and $\mathbf{P}_{\text{Plate-3}}$. The reference coordinate can be represented as below(figure 3.21):

1. Origin: $\mathbf{P}_{\text{O}} = \mathbf{P}_{\text{Plate-1}}$
2. X axis: $\vec{x_o} = \frac{\overrightarrow{\mathbf{P}_{\text{Plate-1}}\,\mathbf{P}_{\text{Plate-2}}}}{|\overrightarrow{\mathbf{P}_{\text{Plate-1}}\,\mathbf{P}_{\text{Plate-2}}}|}$

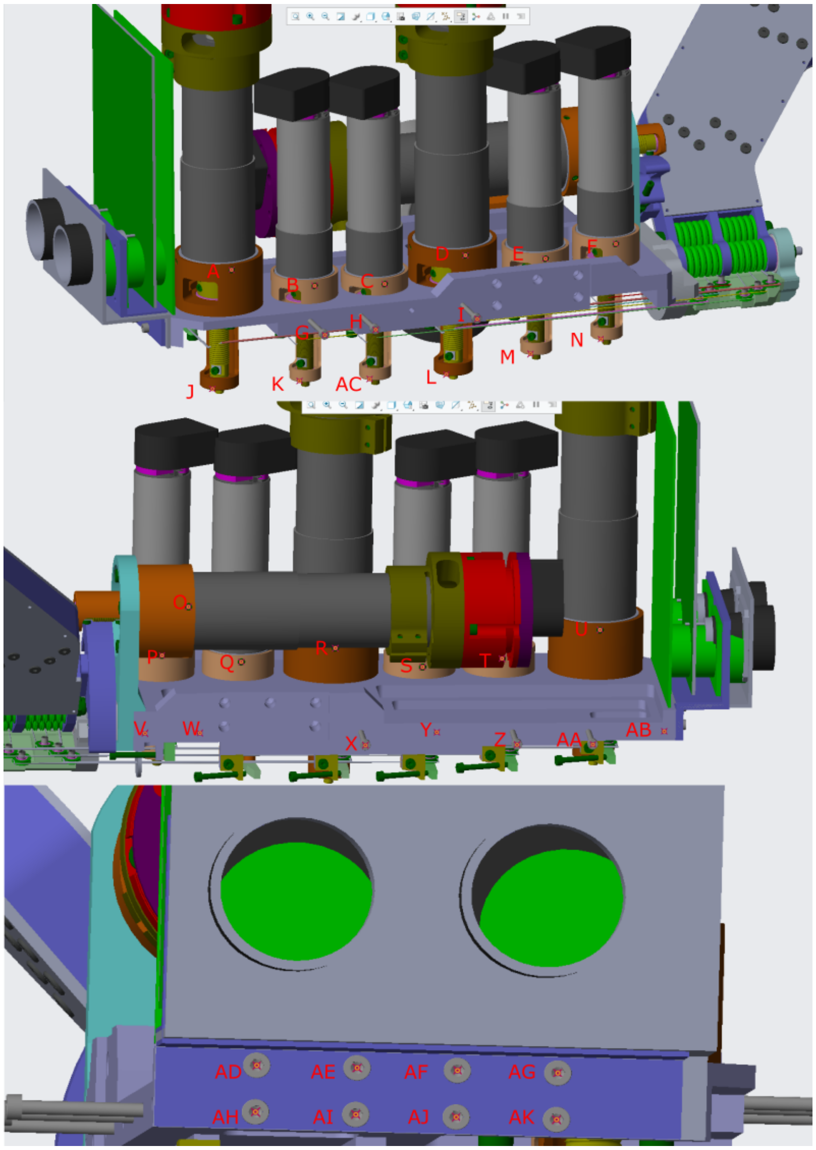

Figure 3.19: The $\mathbf{P}_{\mathbf{Base}}$ of Raven-II's Left Arm

3. Z axis: $\vec{z_o} = \frac{\overrightarrow{\mathbf{P}_{\text{Plate-1}}\,\mathbf{P}_{\text{Plate-2}}}}{|\overrightarrow{\mathbf{P}_{\text{Plate-1}}\,\mathbf{P}_{\text{Plate-2}}}|} \times \frac{\overrightarrow{\mathbf{P}_{\text{Plate-1}}\,\mathbf{P}_{\text{Plate-3}}}}{|\overrightarrow{\mathbf{P}_{\text{Plate-1}}\,\mathbf{P}_{\text{Plate-3}}}|}$

4. Y axis: $\vec{y_o} = \vec{z_o} \times \vec{x_o}$

As long as the measurement device can access $\mathbf{P}_{\text{Plate-1}}$, $\mathbf{P}_{\text{Plate-2}}$, and $\mathbf{P}_{\text{Plate-3}}$, the transformation matrix ${}^{Device}_{Plate}\mathbf{T}$ can be defined.

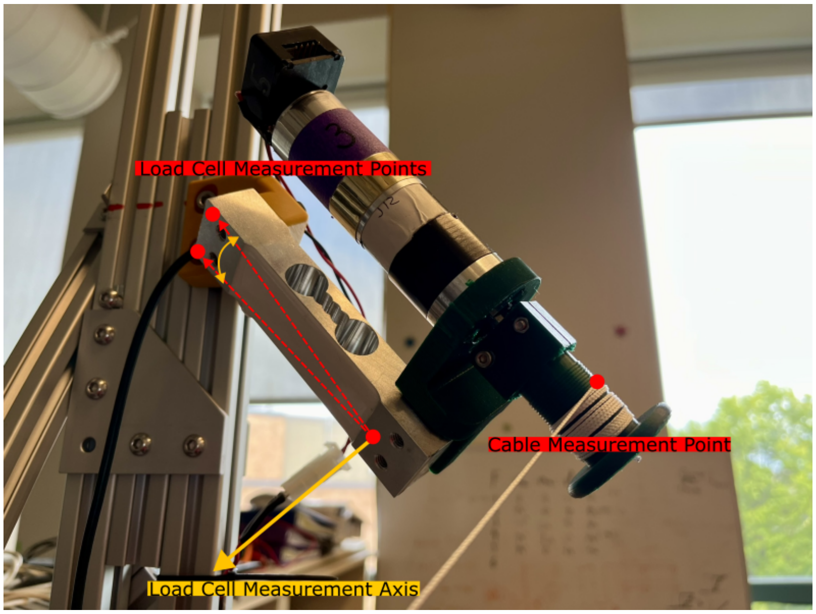

Figure 3.20: The $\mathbf{P_{Cable}}$ and $\mathbf{P_{LoadCell}}$ of Force Unit

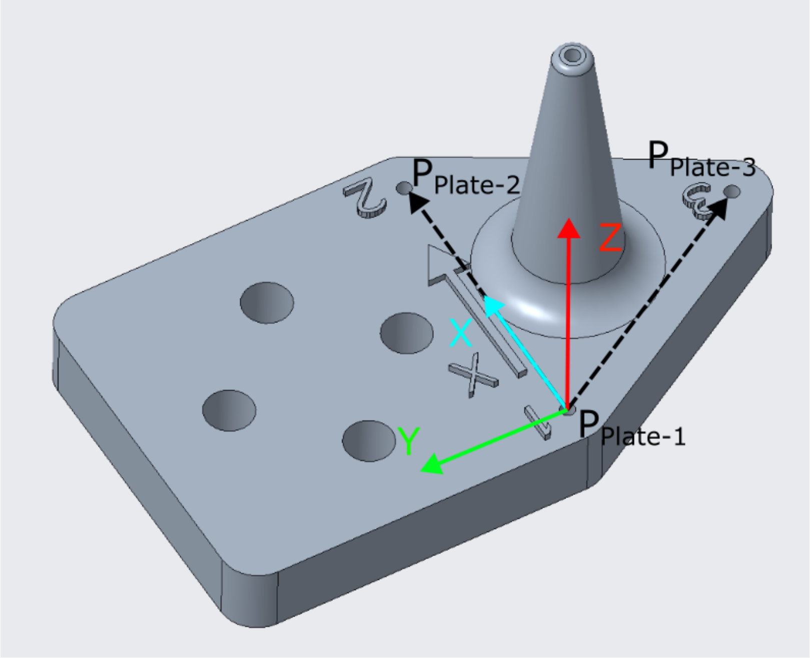

Figure 3.21: The reference coordinate and $\mathbf{P_{Plate}}$ on the Calibration Plate

One or more calibration plates will be placed in between the Raven base and the measuring force unit with proper spacing so that the measurement device can always reach the reference points on both sides. Suppose there are n measurement plates in between Raven base and the force unit, the measurement device need to move through $n + 1$ locations in order to establish the transformation relationship between the Raven base and the force unit(figure 3.22).

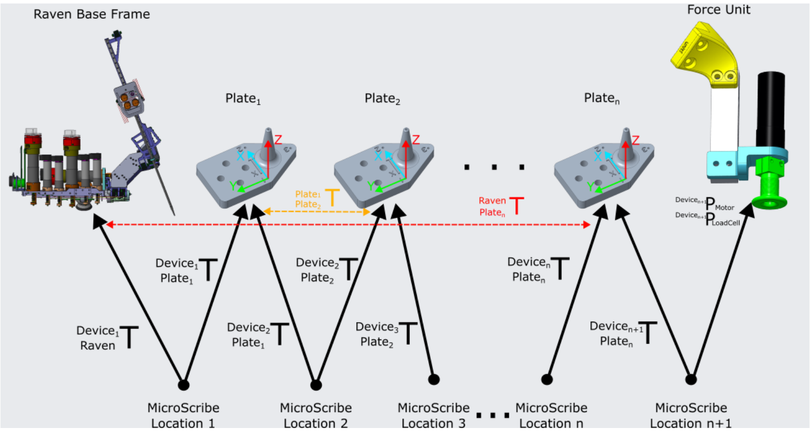

Figure 3.22: Measuring the Force Unit Using Calibration Plates

3.6 Low-Level Controller

A low-level feedback controller is essential for maintaining cable tension in the force unit system. Its primary function is to compensate for internal motor friction. While torque is generally proportional to motor current, the controller must account for disturbances caused by non-ideal motor characteristics and unpredictable behavior, as the force units passively follow the movements of the Raven-II surgical robot.

Several factors can impact force control performance using a DC motor, including driver circuit design, motor winding, and temperature. However, observations to the force control system indicate that the most significant factor is internal motor friction. Based on the friction model, three primary types of friction influence torque output:

1. Static friction: This occurs when the motor transitions from rest to motion. The force

required to overcome the adhesion between two surfaces is known as static friction, and it is typically greater than Coulomb friction.

2. Coulomb friction: This type of friction occurs when the motor is in motion. It refers to the constant kinetic friction between the motor shaft and bearings.

3. Viscous friction: Caused by the lubricant in the motor, viscous friction increases with the speed of relative motion between surfaces. The magnitude of this friction is dependent on the angular velocity.

Among the three types of friction, viscous friction is less of a concern because the motor in the force unit is driven by the surgical tool's movement, which is relatively slow. The main challenge is that the tool's motion causes frequent changes in motor direction and sometimes requires the motor to remain static.

When the motor reverses direction, the torque offset from Coulomb friction changes significantly. Although the magnitude of Coulomb friction can be experimentally determined and its direction is tied to motor velocity, compensating for this force is difficult due to its nonlinear nature. During the Raven's motion, the motor velocity of the force unit is so low that even slight noise can cause velocity sign flips, leading to drastic changes in the control signals and potential system instability. Furthermore, when the motor stops, controlling cable tension without feedback becomes challenging, as the tension can vary without overcoming static friction.

Challenges mentioned above necessitate the feedback controller to actively handle the internal friction and monitor the cable tension. The feedback control diagram is shown in figure 3.23. For each force unit, the controller will receive force command from a higher-level controller, and the NAU7802 IC will sample the load cell on force unit at 320Hz. The readings will first pass through a digital filter to eliminate the high frequency noise. Noise analysis on closed-loop system shows that the force unit is particularly sensitive to the noise around 20 -25Hz(figure 3.24). Currently, a moving average filter can effectively eliminate the noise in the feedback system, but more sophisticated signal processing can be applied in the future.

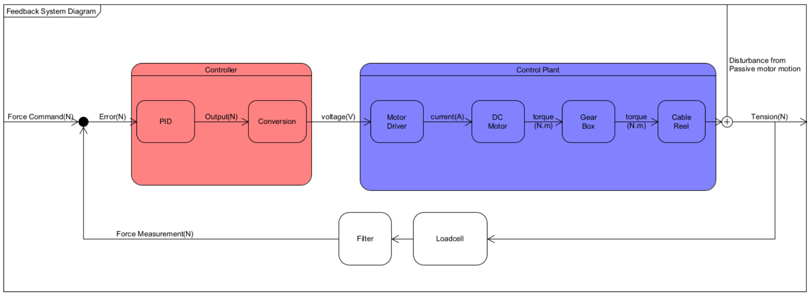

Figure 3.23: Force Unit Control Feedback Loop

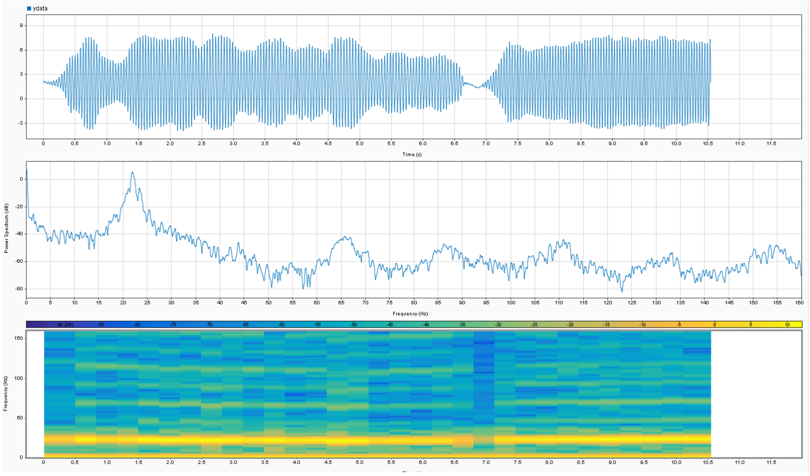

Figure 3.24: Feedback Signal Noise Analysis

The filtered signal will then be fed into a PID controller, which calculate the desired force output as motor torque command. The controller has been tested with one end of cable connected to the Raven-II's end-effector. The force unit output a sinusoidal force profile while the grasper of Raven-II following a random trajectory in its workspace(Figure3.25). The discontinuity in the torque command showing in the picture indicates directional change of the force unit, and the controller quickly reacted to compensate for the change of friction within the system.

For the control of DC motor on force units, two types of control process were developed in this thesis. The first one utilizes the control pathway of the original Raven-II system, and the second one utilizes a newly designed hardware component to directly control the

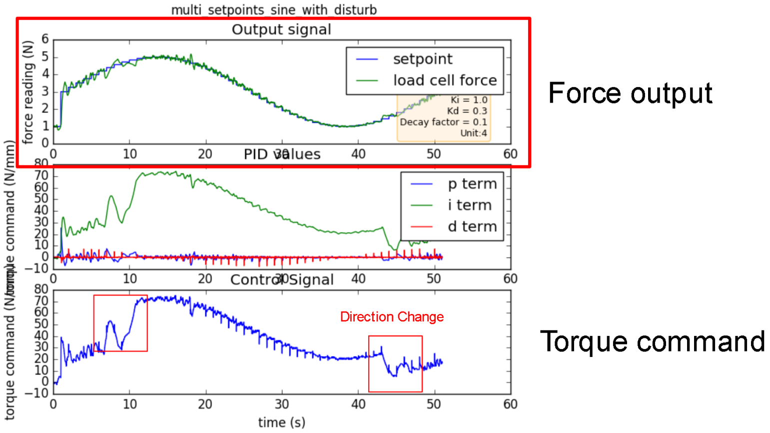

Figure 3.25: Single Force Unit Test

servo amplifiers of the Raven-II system.

3.6.1 Control through Raven-II's control software

Prior to gaining direct hardware access to Raven-II's servo amplifiers, torque control of the force units was achieved by modifying the control software on the Raven control PC. The Raven-II operates in position control mode, utilizing a proportional-derivative (PD) controller that adjusts motor torque based on encoder feedback. This control loop runs at 1000Hz, providing a software-based entry point for motor torque control.

This approach required significant modifications to the Raven-II software. By default, the Raven-II software controls 14 DC motors as 2 regular surgical robot arms. However, for the purpose of this thesis, the system was modified so that the motors of right arm are for force units and the left arm remains its surgical functions. To enable direct torque control on the right arm, several modules needed to be disabled:

1. Homing: The incremental encoders on Raven's motors require homing to calibrate joint positions. Since the motors do not operate in position control mode but as force units to apply force to the left arm, the homing process could cause damage to the left arm's gripper.

2. Inverse Kinematics: The inverse kinematics module is unnecessary for the force units, as torque control does not require end-effector position calculations.

3. Gravity Compensation: Without initialized joint positions, the gravity compensation module could generate unpredictable torques, particularly in joint 3, a prismatic joint, leading to instability.

After disabling these modules and modifying the CRTK API, the right arm can receive torque commands through ROS (Robot Operating System).

This software-only method offers the advantage of requiring no electronic modifications, making it easily replicable for institutions with a Raven-II platform. However, there are significant drawbacks. Integrating other hardware, such as the NAU7802 load cell sensor, is challenging due to the Raven PC's lack of proper interfacing IO ports. This necessitates an external sensor system, increasing feedback loop delays. Additionally, the synchronous update requirement for all motor torques forces the system to process all six load cells at once, creating overhead, particularly for the first load cell, which must wait for the others to be processed. Lastly, the torque control resolution is limited. Raven-II software allows motor torque to be specified only in integer "N.mm" values. Given the current cable reel radius of 1 cm, the minimum controllable cable tension is 0.1N, despite the 16-bit resolution of the DAC7731 chip, indicating potential for improvement.

3.6.2 Control through dedicated motor control board

Different from the previous software-based approach, a dedicated motor control board is designed to gain direct access to the servo amplifiers on the Raven's servo system. The current version of the prototype is able to control 2 force units.

The motor control board is essentially a modified USB board of Raven-II with force sensor system integrated in it(figure 3.26). The main features are:

1. Changed the interfacing of DAC7731 chips from daisy-chain to parallel SPI configuration to enable independent control of each chip.

2. Changed the output range configuration for the DAC7731 chips.

3. Integrated a TCA9458 I2C multiplexer chip and external sensor ports for NAU7802 ICs, with data-ready trigger lines.

4. Upgraded the microcontroller unit to a 32-bit STM32H753 MCU, which offers far more superior computational power than the original USB board and enables the possibility of decentralized controller.

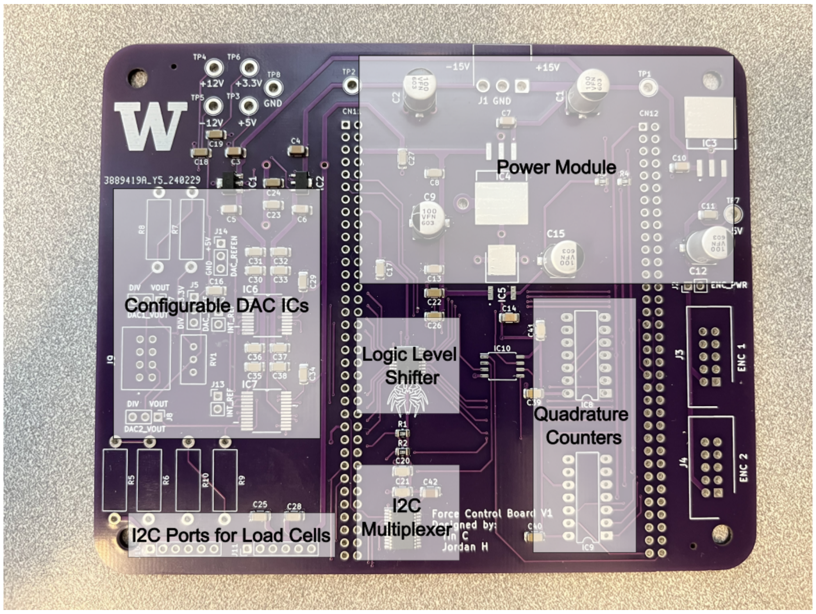

Figure 3.26: Motor Control Board(Without MCU)

The hardware design of the motor control board allows the feedback force control process to be streamlined and integrated into a single device, eliminates the overhead causing by communication between multiple platforms. In addition, by integrating data-ready trigger line of NAU7802 and separating the DAC7731 chips, the torque of each force unit can be updated right after the feedback force signal for that particular unit is ready. Such controller design effectively minimize the delay in the feedback loop to 0.7ms(figure 3.27.

The low-level controller is implemented on the Zephyr RTOS(real-time operating system) and deployed to the MCU on the motor control board. The firmware has 6 threads in total and can be structured as below(figure):

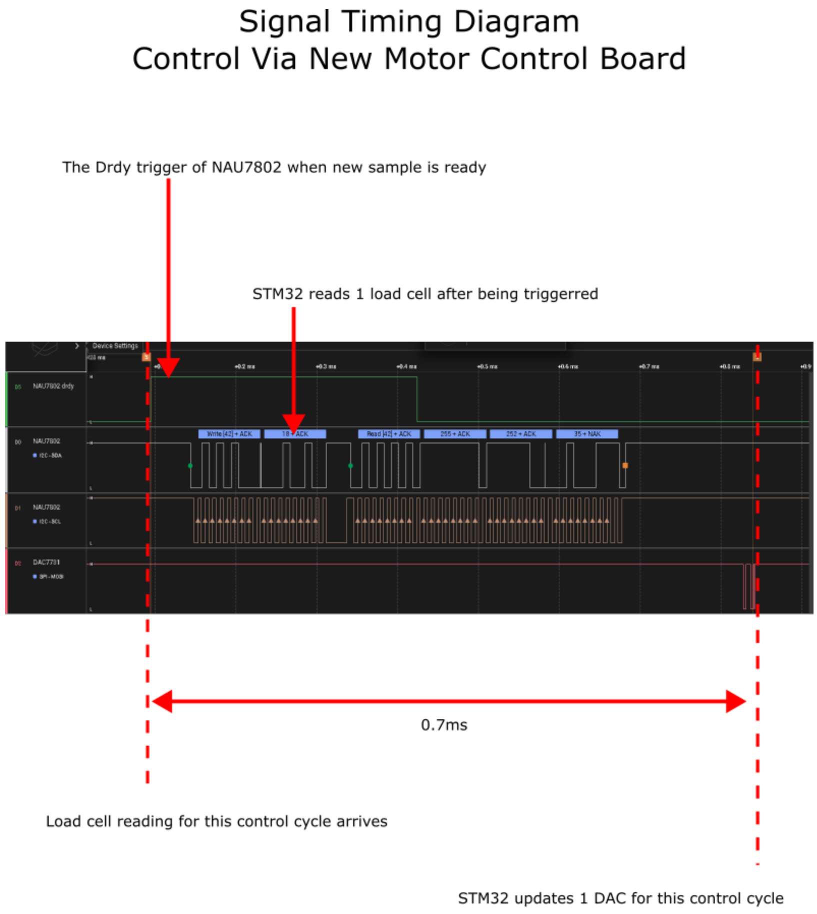

Figure 3.27: Signal Timing Diagram on Motor Control Board

1. **USB Communication thread:** The thread communicates with the host PC through USB, and will only start communication after the system is initialized. It receives control command from the host PC, and send the system information back to host PC.

2. **System State Machine thread:** The SystemStateMachine thread is an additional layer sits above all MotorControl Modules. The SystemStateMachine represents the higher level behavior of the system as a whole.

3. **Motor Control module:** The Motor Control Module will ensure the correct motor behavior based on the system state, for example, run/stop state. When the module

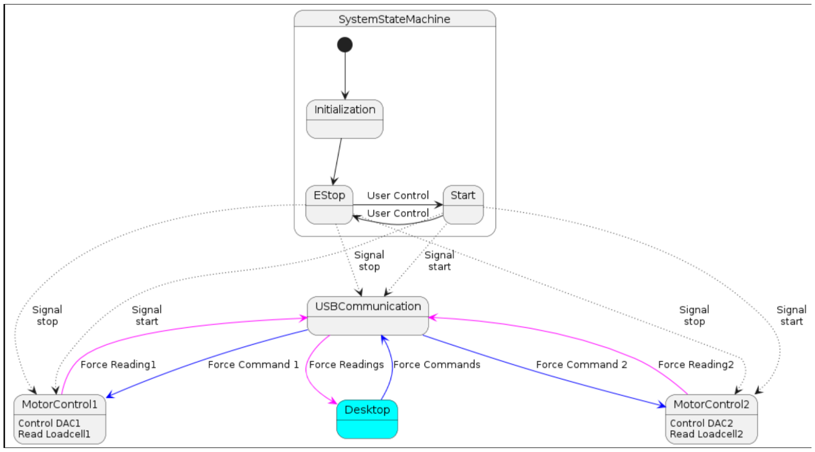

Figure 3.28: Firmware Structure of Motor Control Board

is controlling the motor, it will generate the DAC output based on the load cell readings. The Motor control module is composed of two parts: MotorStateMachine and MotorTorqueControl threads.

- The MotorStateMachine thread actively receives the external event and change the behavior of motor. The MotorStateMachine thread can block or unblock the MotorTorqueControl thread using a binary semaphore. It is also in charge of initializing the filter buffer, PID controller, and setting up the trigger handler for the load cell module when the system is powered on.
- The MotorTorqueControl thread is essentially a super-loop that does the actual signal processing and PID control. The thread needs to be triggered by NAU7802, filter the load cell readings, generate output signal using PID controller, and convert the output signal to DAC value.

3.7 *High-Level Controller*

A high-level controller is designed to coordinate six force units, ensuring that the desired force vector is accurately applied to the end-effector based on its position within the workspace. At each time step, the controller calculates the orientation of the six cables and determines the optimal distribution of tension required to achieve the specified force vector. The system utilizes load cell feedback to measure the applied force, allowing real-time adjustments to maintain accuracy. Implemented in ROS, the controller employs a Proportional-Integral (PI) controller to address steady-state errors and incorporates a numerical solver to compute cable tensions based on their respective orientations (Figure **??**).

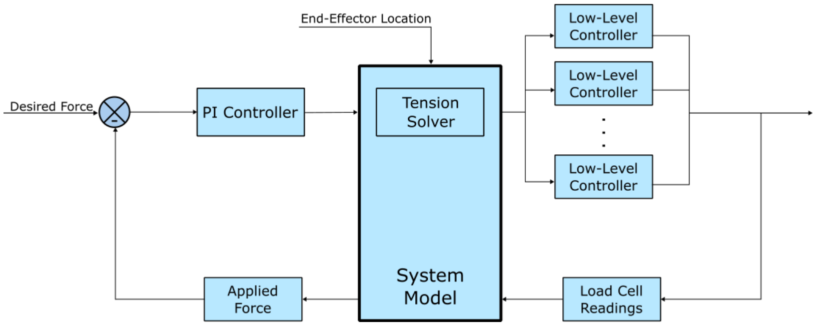

Figure 3.29: High-Level Controller Diagram

The parameters through out the process are listed as below, where $\mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{6}$:

- Input Parameters:
    - $\mathbf{P_{ForceUnit}i}$: Force Units Location, where the cable contact with the cable reel.
    - $\tilde{\mathbf{V}_{\mathbf{LoadCell}}}\mathbf{i}$: Load Cell Measurement Direction.
    - $\mathbf{P_{EndEffector}}$: End-Effector Location, provided in the robot state information of Raven-II.
    - $\tilde{\mathbf{F}_{\mathbf{Desired}}}$: Desired Force, provided in a predefined trajectory.
    - $\mathbf{F_{LoadCell}i}$: Load Cell Readings.
- Intermediate Parameters
    - $\tilde{\mathbf{V}_{\mathbf{Cable}}}\mathbf{i}$: Cable Direction
    - $\theta_{\mathbf{Cable}}\mathbf{i}$: The angle between $\tilde{\mathbf{V}_{\mathbf{LoadCell}}}\mathbf{i}$ and $\tilde{\mathbf{V}_{\mathbf{Cable}}}\mathbf{i}$
    - $\mathbf{T_{Cable}i}$: The Estimated Cable Tension based on $\theta_{\mathbf{Cable}}\mathbf{i}$ and $\mathbf{F_{LoadCell}i}$

  - $\tilde{\mathbf{F}}_{\mathbf{Comp}}$: The Compensated Force calculated by the PI controller
  - $\mathbf{T}_{\mathbf{Desired}}\mathbf{i}$: Desired Cable Tensions

- Output Parameters:
  - $\mathbf{F}_{\mathbf{LoadCellDesired}}\mathbf{i}$: The Desired Load Cell Readings, serve as the set point for the low-level controller.
  - $\tilde{\mathbf{F}}_{\mathbf{Applied}}$: Applied Force

The controller runs at 100Hz. Each round, the controller will first try to establish the physical model of the system using the updated $\mathbf{P}_{\mathbf{EndEffector}}$ and $\mathbf{F}_{\mathbf{LoadCell}}\mathbf{i}$. The cable direction will be calculated as below:

$$\tilde{\mathbf{V}}_{\mathbf{Cable}}\mathbf{i} = \frac{\mathbf{P}_{\mathbf{ForceUnit}}\mathbf{i} - \mathbf{P}_{\mathbf{EndEffector}}}{|\mathbf{P}_{\mathbf{ForceUnit}}\mathbf{i} - \mathbf{P}_{\mathbf{EndEffector}}|}$$

With the $\tilde{\mathbf{V}}_{\mathbf{Cable}}\mathbf{i}$, the controller can calculate the feedback force value, $\mathbf{F}_{\mathbf{Applied}}$, using the $\mathbf{F}_{\mathbf{LoadCell}}\mathbf{i}$ and $\tilde{\mathbf{V}}_{\mathbf{LoadCell}}\mathbf{i}$. The $\tilde{\mathbf{V}}_{\mathbf{Cable}}\mathbf{i}$ and $\tilde{\mathbf{V}}_{\mathbf{LoadCell}}\mathbf{i}$ will not be perfectly align as the cable direction will change with the $\mathbf{P}_{\mathbf{EndEffector}}$. Load cell measurement value is the projection value of the cable tension value(figure 3.30).

The relationship between the two vector can be obtained through vector dot product:

$$\cos\theta_{\mathbf{Cable}}\mathbf{i} = \frac{\tilde{\mathbf{V}}_{\mathbf{Cable}}\mathbf{i} \cdot \tilde{\mathbf{V}}_{\mathbf{LoadCell}}\mathbf{i}}{|\tilde{\mathbf{V}}_{\mathbf{Cable}}\mathbf{i}||\tilde{\mathbf{V}}_{\mathbf{LoadCell}}\mathbf{i}|}$$

With the relationship, we can compensate the load cell reading and estimate the cable tension:

$$\mathbf{T}_{\mathbf{Cable}}\mathbf{i} = \frac{\mathbf{F}_{\mathbf{LoadCell}}\mathbf{i}}{\cos\theta_{\mathbf{Cable}}\mathbf{i}}$$

The applied force can be calculated as:

$$\tilde{\mathbf{F}}_{\mathbf{Applied}} = \sum_{\mathbf{i=1}}^{\mathbf{6}} \mathbf{T}_{\mathbf{Cable}}\mathbf{i} \times \tilde{\mathbf{V}}_{\mathbf{Cable}}\mathbf{i}$$

With the $\tilde{\mathbf{F}}_{\mathbf{Applied}}$ and $\tilde{\mathbf{F}}_{\mathbf{Desired}}$, a PI controller will calculate the compensated force, $\tilde{\mathbf{F}}_{\mathbf{Comp}}$. Then, an optimize solver will solve the equation:

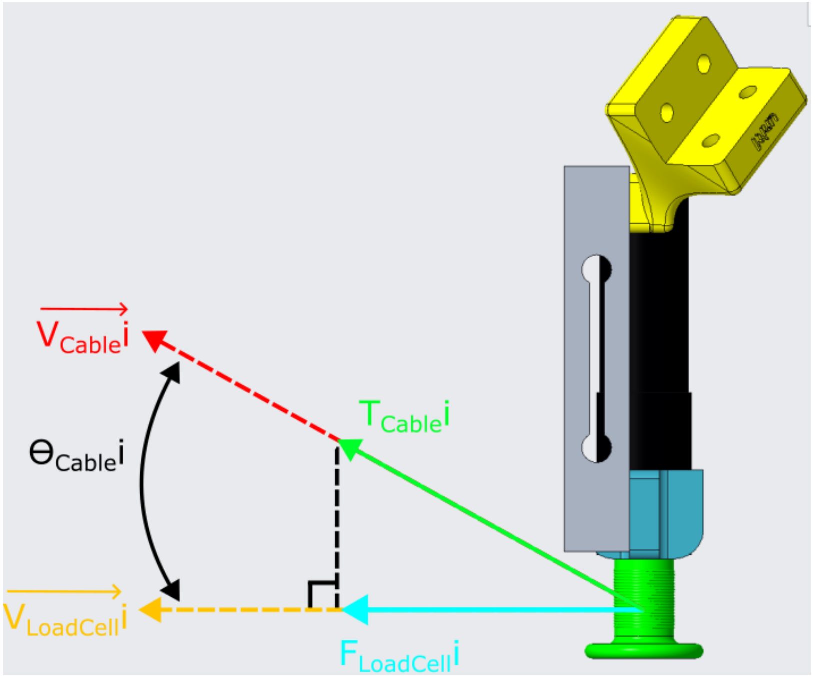

Figure 3.30: Relationship between cable tension and load cell measurement value

$$\begin{aligned} &\text{minimize} \quad \mathbf{f}(\mathbf{T_{Desired}}) = |\mathbf{F_{\tilde{C}omp}} - \sum_{\mathbf{i=1}}^{\mathbf{6}} \mathbf{T_{Desired}i} \times \mathbf{V_{\tilde{C}able}i}|^{\mathbf{2}} \\ &\text{subject to} \quad \mathbf{2N} \leq \mathbf{T_{Desired}i} \leq \mathbf{12N} \end{aligned}$$

The desired cable tension will then be converted to the desired load cell readings by compensating the angle between the cable and load cell measurement axis:

$$\mathbf{F_{LoadCellDesired}i} = \mathbf{T_{Desired}i} \times \cos\theta_{\mathbf{Cable}}\mathbf{i}$$

Currently, we don't have a proper sensor system to validate the performance of the entire system.

Chapter 4

DATA-DRIVEN INVERSE DYNAMICS LEARNING ON RAVEN-II SURGICAL ROBOT

The cable-driven design of the Raven-II surgical robot introduces significant uncertainty and challenges in accurately modeling its dynamics. To address these issues, a learning-based calibration method using a neural network is proposed to identify the robot's dynamic model. In this thesis, a preliminary experiment was conducted using a force unit prototype to validate this approach.

The experiment aims to demonstrate that the neural network can learn the inverse dynamic relationship inherent in the robot's state information. As this is a proof-of-concept experiment, the precise performance of the model in measuring external forces within Cartesian space is not critical. Additionally, any inaccuracies resulting from the installation of the force units are considered negligible for this initial validation.

4.1 Experimental Setup

Two force units are positioned along the Y-axis in the Raven-II's workspace to apply a one-dimensional contact force (Figure 4.1). The cables from these units are attached to the wrist of the Raven-II's end-effector. The force units are aligned such that the cables are approximately parallel when the end-effector is in its initial position after completing the homing procedure. Each force unit is equipped with a load cell, facing opposite directions, to measure the cable forces along the positive and negative Y directions.

The controllers of the force units are configured to maintain a pre-tension force of 2N in each cable, ensuring they remain taut and preventing slack. When a force command is applied in the Y direction, the appropriate force unit adjusts the force output by adding or subtracting from this pre-tension value based on the command's direction.

The experiment collects data from the Raven-II robot while varying the contact forces applied to its end-effector using two force units. Specifically, two types of trajectories are

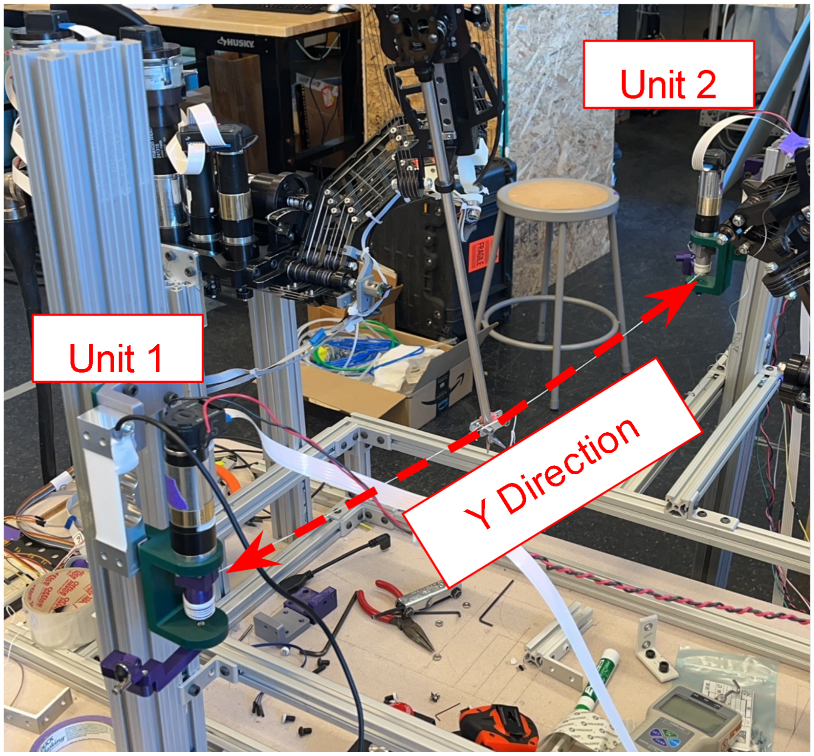

Figure 4.1: 1D Contact Force Estimation Experiment Setup

generated: one representing the Raven-II's end-effector movement within its workspace and the other detailing the force profile, consisting of a series of force commands applied in the Y-direction.

To ensure the dataset captures a wide range of configurations, the end-effector trajectory follows a random path based on the movement of the first three joints. Each joint operates under a sinusoidal positional trajectory with randomly assigned amplitudes and periods. This random trajectory is visualized in both the workspace, joint space, and joint velocity space (Figure 4.4). For the force profile, a similar random trajectory is used, with the contact force ranging between $\pm 4.5N$.

When combining both trajectories, evaluating coverage in this high-dimensional space remains challenging. Nevertheless, visualizing the force profile alongside Raven-II's trajectory indicates a fairly uniform distribution in the workspace, joint space, and joint velocity space (Figure 4.4).

4.2 Data Processing Procedure

For data collection, the Raven-II executes a random trajectory with varying forces applied over 9 trials, each lasting 20 minutes. During the operation, the robot's state information and the measured forces along the Y-axis of Raven-II's Zero frame are recorded in a bag file. As the Raven-II state data and the measured forces are logged at different sampling rates, these datasets are synchronized using timestamps from the recordings. The synchronized data are then compiled into a CSV file for further analysis (Figure 4.2).

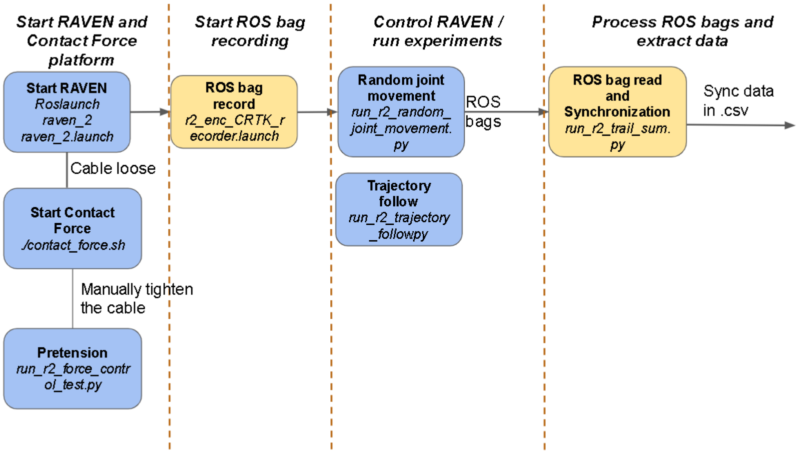

Figure 4.2: The data collection procedure for contact force estimation.

The Raven-II state includes information from both the right and left arms, covering kinematic details such as joint positions, joint velocities, end-effector position, orientation, and velocity. Additionally, it incorporates dynamic information, including joint torques and Jacobian forces calculated using an ideal kinematic model. All these data are derived from optical encoder readings on the motors and the motor control commands, which are also recorded as part of the Raven-II state. To optimize neural network performance, only the relevant features in the joint and task spaces are retained for training. Furthermore, noisy data such as joint velocities, end-effector velocities, joint torques, and end-effector forces are smoothed to reduce the computational load on the neural network.

The training dataset consists of 160 minutes of random trajectories, comprising 427,403 data points. An additional 40 minutes of random trajectories, containing 121,909 data points, is used for validation. To investigate the significance of sequential data, both datasets are converted into 3D tensors with a window size of 50 samples, corresponding to approximately 1 second.

Three types of neural networks are employed in this thesis: the multi-layer perceptron (MLP), the gated recurrent unit (GRU), and the long short-term memory (LSTM) network. The MLP is used to evaluate how well the model handles uncertainty based on the current robot state, while the LSTM is utilized to assess whether incorporating sequential data improves model performance. Model structures were selected empirically(Figure 4.3).

The MLP network consists of five dense hidden layers with neuron counts of 1200, 1000, 800, 600, and 400, respectively. A dropout rate of 20% is applied to each layer to mitigate overfitting. The GRU and LSTM networks each have three hidden layers, including two recurrent layers and one dense layer, with 100 neurons per layer. An input time window of 50 time steps (approximately 1 second) is used. To prevent overfitting, a 10% dropout rate is applied to the recurrent layers.

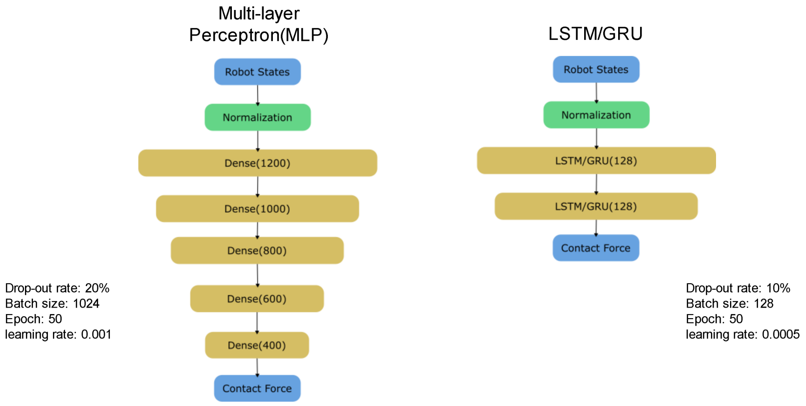

Figure 4.3: Model Structure Selected and Training Parameters

Each model is trained with a batch size of 1024 for the MLP and 128 for the GRU and LSTM networks. The learning rate is set to 0.001 for the MLP and 0.0005 for the GRU and LSTM. Training was conducted over 50 epochs. The training and testing datasets are depicted in Figure 4.4, illustrating the 1D contact force encoded as a colormap on the Raven-II's trajectory.

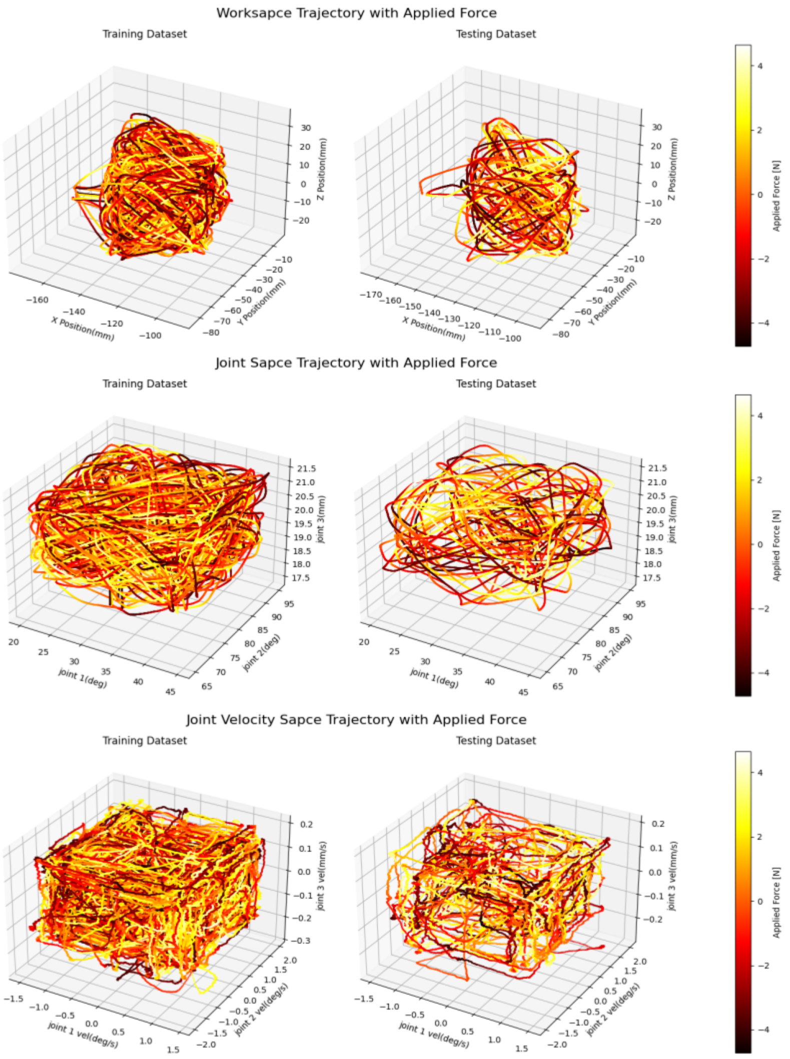

Figure 4.4: Training and testing trajectory for Raven-II and contact force profile. The Raven-II's trajectories are visualized in both task space and joint space.

4.3 Experimental Result

The contact force prediction results of the MLP, GRU, and LSTM models are presented in figure 4.5 and Table 4.3. The error distribution for each model is shown in the corresponding box plot. The results indicate that all models accurately predict the trends and the maximum/minimum values of the contact force, with most error values falling within ±0.5 N of the ±4.5 N contact force range. Notably, while the MLP shows overall better performance compared to the GRU and LSTM, this does not necessarily indicate its superiority in contact force estimation. Both GRU and LSTM models demonstrate comparable performance with less than 10% of the size of the MLP model. Given comparable model sizes, GRU and LSTM could potentially achieve even better performance. However, the shorter inference time of the MLP makes it particularly advantageous for real-time applications.

	MLP	GRU	LSTM
MAE of $\mathbf{F_y}$(N)	0.5859	0.6664	0.7486
NRMSE of $\mathbf{F_y}$(%)	9.1562	9.8122	11.3075
Execution Time Cost	2 ms	5ms	4ms
Total Parameters	2,777,203	182,531	236,803

Table 4.1: Mean absolute error of contact force prediction for each model

The data collection method in this thesis facilitates the analysis of the relationship between the performance of learning-based contact force estimation and the Raven-II's configuration. Figure 4.7 visualizes the prediction error concerning the Raven-II's trajectory in both task and joint spaces. The task space visualization provides limited insight, but the joint space visualization reveals that larger errors concentrate on both sides of the joint 1 axis, where the joint reverses its motion direction. This pattern is also evident in the joint velocity space: when joint 1 changes direction, its velocity decreases, leading to increased errors concentrated at the trajectory center in the velocity space.

This phenomenon aligns with the physical characteristics of the Raven-II, where joint 1 primarily contributes to the end-effector's motion in the y-direction and compensates for

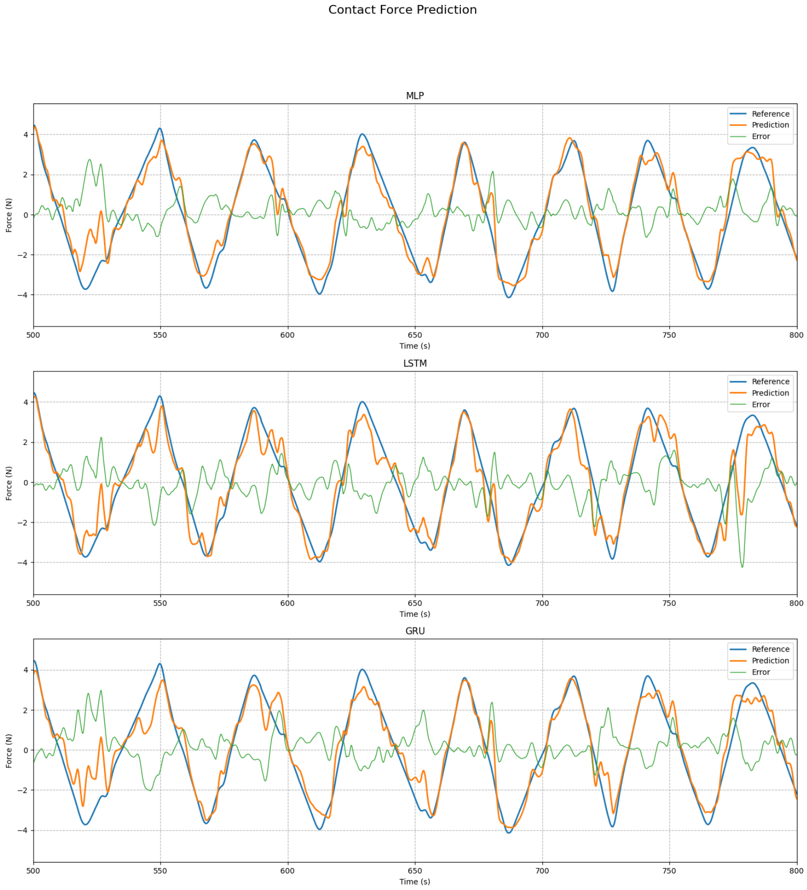

Figure 4.5: 1-Directional Contact Force Prediction Result

most external forces along this axis. Consequently, the model heavily relies on joint 1 data for accurate y-direction contact force prediction. When joint 1's velocity decreases, joint friction becomes dominant. Once joint 1 comes to a complete stop, static friction creates a "dead zone" for force estimation until the motor torque exceeds the breakaway force.

Both joint 2 and joint 3, in addition to joint 1, contribute to compensating the external force in the y-direction. These joints also encounter the ”dead zone” of static friction when they come to a complete stop. The relationship between joint velocities and prediction error is further illustrated in Figure 4.8. The velocities of the first three joints of the Raven-II robot are used to define the end-effector speed, which is categorized into six groups ranging from 0 mm/s to 2.4 mm/s.

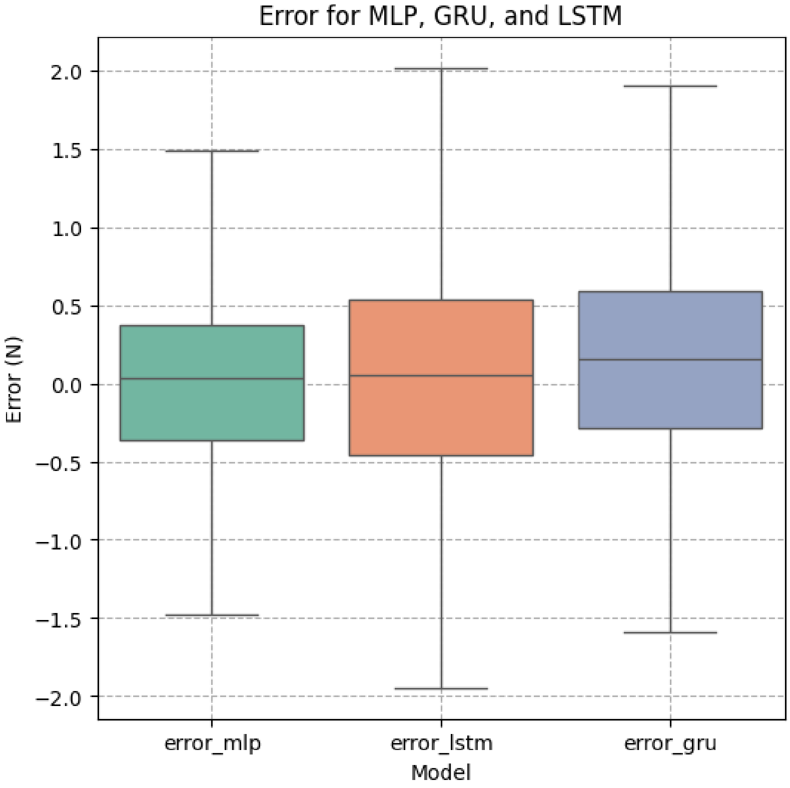

Figure 4.6: 1-Directional contact force prediction error for each model

Visualizing the prediction error across these speed groups reveals that all models perform better when the end-effector is moving at higher speeds and exhibit poorer performance at lower speeds. This degradation occurs because one or more joints may stop and be influenced by the static friction "dead zone." These findings suggest that the training dataset should include more instances where the end-effector operates at low speeds or remains stationary to better account for the effects of static friction and improve model performance.

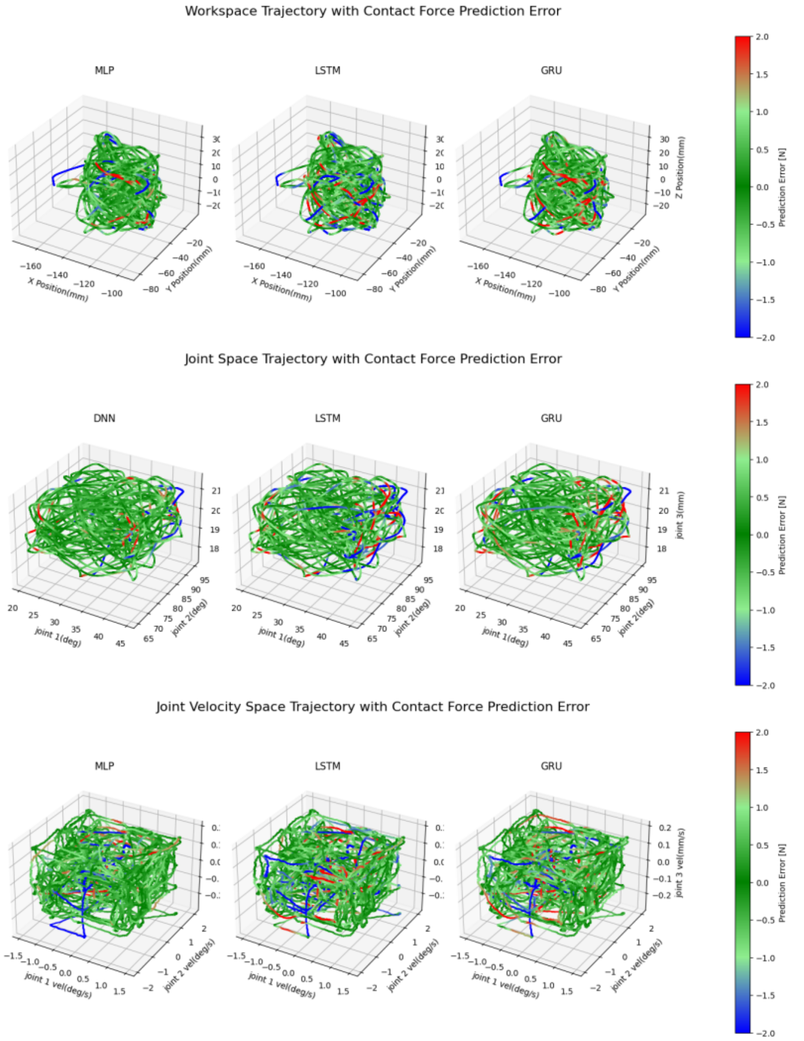

Figure 4.7: 1-Directional contact force prediction error with Raven-II trajectory

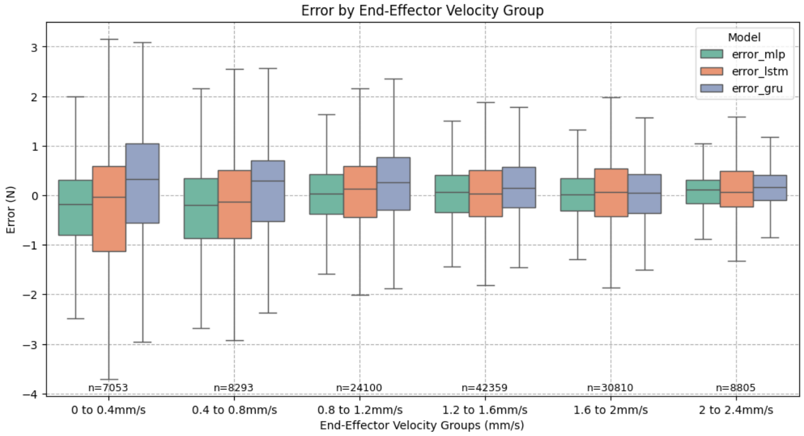

Figure 4.8: 1-Directional contact force prediction error with respect to end-effector's velocity

Chapter 5

CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

5.1 Conclusion

In this thesis, we propose a learning-based contact force estimation method for the Raven-II, a cable-driven laparoscopic surgical robot system. The proposed method leverages neural networks to capture the nonlinear relationships within the data, reducing the need to model the complex tendon structures of the Raven-II.

A "Force Unit" was developed to apply force on a target by precisely controlling cable tension through a motorized reel. This unit, along with an integrated force sensor and mechatronics system, ensures robust force control. Using the Force Unit, a 6-degree-of-freedom (6-DOF) cable-driven actuation device was designed to apply Cartesian forces to the end-effector across the Raven-II's workspace. Additionally, a localization procedure for the force unit was developed, and a hierarchical controller was designed and fine-tuned for effective force actuation.

Preliminary 1-dimensional contact force estimation experiments were conducted, where data was collected by applying known external forces to the end-effector in the Y-direction using two force units. Three neural network models—MLP, LSTM, and GRU—were trained on the dataset using the TensorFlow framework. The results demonstrate that all models can accurately predict contact forces. Moreover, the analysis highlights a relationship between joint velocities and estimation error, offering insights for future trajectory designs to improve data efficiency during training.

5.2 Recommendations for Future Work

Future work focuses on enhancing the force actuation device and expanding the scope of learning-based contact force estimation. For the estimation process, the data collection will be extended from 1-dimensional to 3-dimensional contact forces within the Cartesian space.

An ablation study will be conducted to determine which input features most significantly influence model performance. Additionally, we will investigate the model's generalization capability, particularly its ability to predict contact forces beyond the robot configurations used in the training dataset. This analysis will inform the development of optimized training trajectories, enabling efficient data collection that maximizes the representativeness of the training data while minimizing collection time.

Regarding the force actuation system, improvements can be made by utilizing a motor with a smaller torque constant and a cable reel with a reduced diameter to increase control resolution. Upgrading the interfacing electronics of the force sensor to components with a higher sampling rate would enhance the feedback control system's bandwidth. Furthermore, the placement of the force units can be strategically optimized based on the robot's workspace, ensuring that the system provides the maximum possible force coverage.

BIBLIOGRAPHY

[1] Angelica I Aviles, Samar M Alsaleh, Eduard Montseny, Pilar Sobrevilla, and Alicia Casals. A deep-neuro-fuzzy approach for estimating the interaction forces in robotic surgery. In *2016 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pages 1113–1119. IEEE, 2016.

[2] Daniel Bruder, Xun Fu, R Brent Gillespie, C David Remy, and Ram Vasudevan. Data-driven control of soft robots using koopman operator theory. *IEEE Transactions on Robotics*, 37(3):948–961, 2020.

[3] Zonghe Chua, Anthony M Jarc, and Allison M Okamura. Toward force estimation in robot-assisted surgery using deep learning with vision and robot state. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12335–12341. IEEE, 2021.

[4] Fan Feng, Wuzhou Hong, and Le Xie. A learning-based tip contact force estimation method for tendon-driven continuum manipulator. *Scientific Reports*, 11(1):17482, 2021.

[5] Mohammad Haghighipanah, Yangming Li, Muneaki Miyasaka, and Blake Hannaford. Improving position precision of a servo-controlled elastic cable driven surgical robot using unscented kalman filter. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2030–2036. IEEE, 2015.

[6] Mohammad Haghighipanah, Muneaki Miyasaka, and Blake Hannaford. Utilizing elasticity of cable-driven surgical robot to estimate cable tension and external force. *IEEE Robotics and Automation Letters*, 2(3):1593–1600, 2017.

[7] Ulrich Hagn, Rainer Konietschke, Andreas Tobergte, Mathias Nickl, Stefan Jörg, Bernhard Kübler, Georg Passig, Martin Gröger, Florian Fröhlich, Ulrich Seibold, et al. Dlr mirosurge: a versatile system for research in endoscopic telesurgery. *International journal of computer assisted radiology and surgery*, 5:183–193, 2010.

[8] Blake Hannaford, Jacob Rosen, Diana W Friedman, Hawkeye King, Phillip Roan, Lei Cheng, Daniel Glozman, Ji Ma, Sina Nia Kosari, and Lee White. Raven-ii: an open platform for surgical robotics research. *IEEE Transactions on Biomedical Engineering*, 60(4):954–959, 2012.

[9] Minho Hwang, Brijen Thananjeyan, Samuel Paradis, Daniel Seita, Jeffrey Ichnowski, Danyal Fer, Thomas Low, and Ken Goldberg. Efficiently calibrating cable-driven surgical robots with rgbd fiducial sensing and recurrent neural networks. *IEEE Robotics and Automation Letters*, 5(4):5937–5944, 2020.

[10] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[11] Uikyum Kim, Dong-Hyuk Lee, Woon Jong Yoon, Blake Hannaford, and Hyouk Ryeol Choi. Force sensor integrated surgical forceps for minimally invasive robotic surgery. *IEEE Transactions on Robotics*, 31(5):1214–1224, 2015.

[12] Min Cheol Lee, Chi Yen Kim, Bin Yao, William J Peine, and Young Eun Song. Reaction force estimation of surgical robot instrument using perturbation observer with smcspo algorithm. In *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 181–186. IEEE, 2010.

[13] Yangming Li and Blake Hannaford. Gaussian process regression for sensorless grip force estimation of cable-driven elongated surgical instruments. *IEEE robotics and automation letters*, 2(3):1312–1319, 2017.

[14] Yangming Li, Shuai Li, David Caballero, Muneaki Miyasaka, Andrew Lewis, and Blake Hannaford. Improving control precision and motion adaptiveness for surgical robot with recurrent neural network. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3538–3543. IEEE, 2017.

[15] Mitchell JH Lum, Diana CW Friedman, Ganesh Sankaranarayanan, Hawkeye King, Kenneth Fodero, Rainer Leuschke, Blake Hannaford, Jacob Rosen, and Mika N Sinanan. The raven: Design and validation of a telesurgery system. *The International Journal of Robotics Research*, 28(9):1183–1197, 2009.

[16] Akhil J Madhani, Günter Niemeyer, and J Kenneth Salisbury. The black falcon: a teleoperated surgical instrument for minimally invasive surgery. In *Proceedings. 1998 ieee/rsj international conference on intelligent robots and systems. innovations in theory, practice and applications (cat. no. 98ch36190)*, volume 2, pages 936–944. IEEE, 1998.

[17] Mohsen Mahvash, Jim Gwilliam, Rahul Agarwal, Balazs Vagvolgyi, Li-Ming Su, David D Yuh, and Allison M Okamura. Force-feedback surgical teleoperator: Controller design and palpation experiments. In *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 465–471. IEEE, 2008.

[18] Jörn Malzahn, Rene Felix Reinhart, and Torsten Bertram. Dynamics identification of a damped multi elastic link robot arm under gravity. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2170–2175. IEEE, 2014.

[19] Arturo Marban, Vignesh Srinivasan, Wojciech Samek, Josep Fernández, and Alicia Casals. Estimation of interaction forces in robotic surgery using a semi-supervised deep neural network model. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 761–768. IEEE, 2018.

[20] Muneaki Miyasaka, Mohammad Haghighipanah, Yangming Li, and Blake Hannaford. Hysteresis model of longitudinally loaded cable for cable driven robots and identification of the parameters. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4051–4057. IEEE, 2016.

[21] John J O'Neill, Trevor K Stephens, and Timothy M Kowalewski. Evaluation of torque measurement surrogates as applied to grip torque and jaw angle estimation of robotic surgical tools. *IEEE Robotics and Automation Letters*, 3(4):3027–3034, 2018.

[22] Haonan Peng, Andrew Lewis, and Blake Hannaford. Ablation study on features in learning-based joints calibration of cable-driven surgical robots, 2023.

[23] R Pena, MJ Smith, NP Ontiveros, FL Hammond, and RJ Wood. Printing strain gauges on intuitive surgical da vinci robot end effectors. 2018 ieee. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–5.

[24] Ulrich Seibold, Bernhard Kuebler, and Gerd Hirzinger. Prototypic force feedback instrument for minimally invasive robotic surgery. *Medical robotics*, 44:377–400, 2008.

[25] Andrew C Smith, Farid Mobasser, and Keyvan Hashtrudi-Zaad. Neural-network-based contact force observers for haptic applications. *IEEE Transactions on Robotics*, 22(6):1163–1175, 2006.

[26] Nam Tran, Jie Ying Wu, Anton Deguet, and Peter Kazanzides. A deep learning approach to intrinsic force sensing on the da vinci surgical robot. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 25–32. IEEE, 2020.

[27] A. L. Trejos, A. Escoto, Dustin Hughes, M. Naish, and Rajnikant V. Patel. A sterilizable force-sensing instrument for laparoscopic surgery. *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 157–162, 2014.

[28] A. L. Trejos, A. Escoto, M. Naish, and Rajni V. Patel. Design and evaluation of a sterilizable force sensing instrument for minimally invasive surgery. *IEEE Sensors Journal*, 17:3983–3993, 2017.

[29] C. A. Wallace. New developments in disinfection and sterilization. *American journal of infection control*, 44 5 Suppl:e23–7, 2016.

[30] Nural Yilmaz, Jie Ying Wu, Peter Kazanzides, and Ugur Tumerdem. Neural network based inverse dynamics identification and external force estimation on the da vinci research kit. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1387–1393. IEEE, 2020.

Appendix A

APPENDIX A

A.1 Circuit Boards Schematic

The motor control board was designed in KiCad7.0. The motor control board was designed collaboratively with Jordan Hendrix. The Arduino shield and breakout boards for force sensor system were designed in Flux AI. Flux AI doesn't provide any export feature for schematic sheet, so the schematic sheets provided here were captured directly from the editor view.

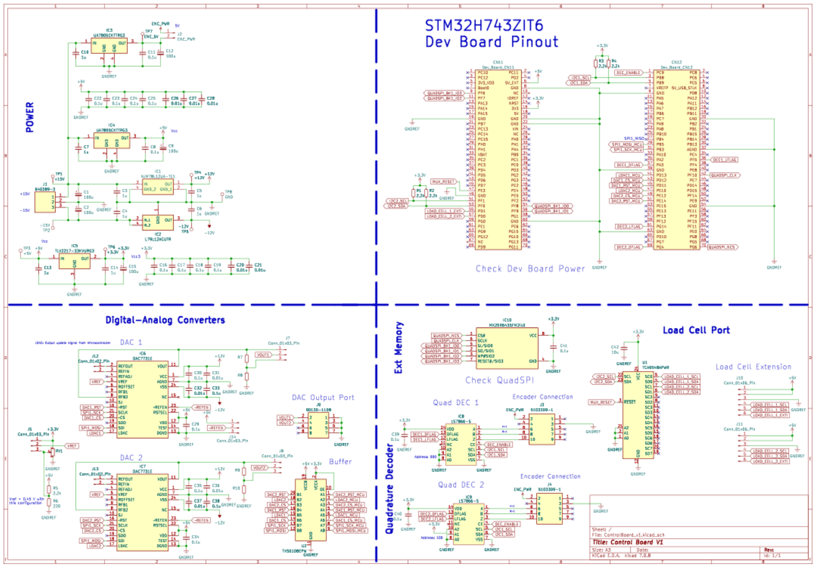

Figure A.1: Motor Control Board Schematic sheet

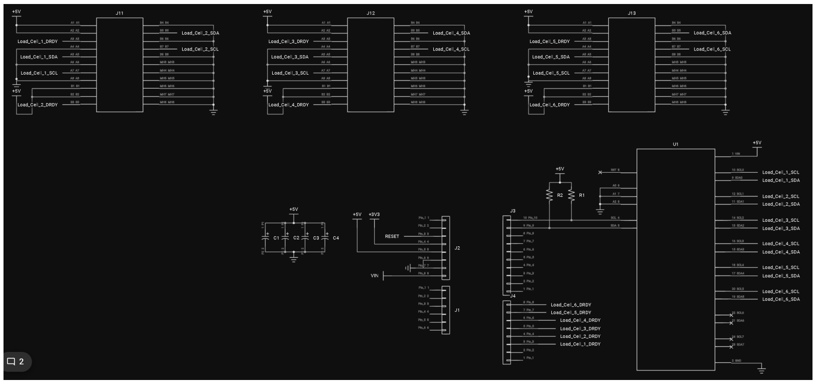

Figure A.2: Force Sensor System Arduino Shield Board Schematic sheet

A.2 Circuit Boards Layout

The motor control board layout was designed in KiCad7.0. The motor control board layout was designed by Jordan Hendrix. The Arduino shield and breakout boards layout for force sensor system were designed in Flux AI. Flux AI doesn't provide any export feature for board layout, so the board layout provided here were captured directly from the editor view.

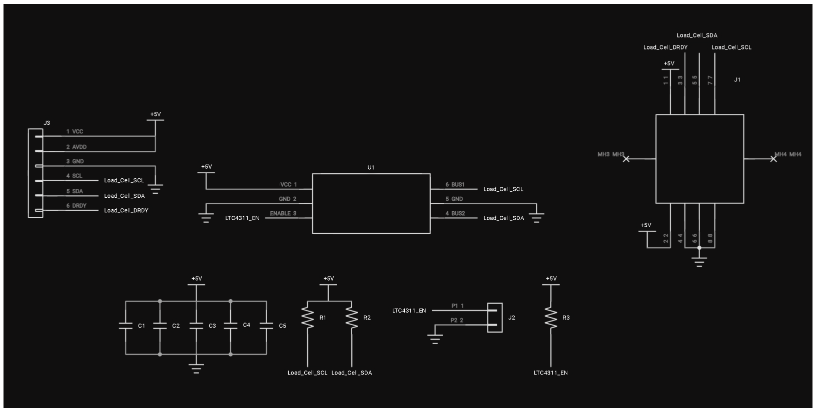

Figure A.3: Force Sensor System Breakout Board Schematic sheet

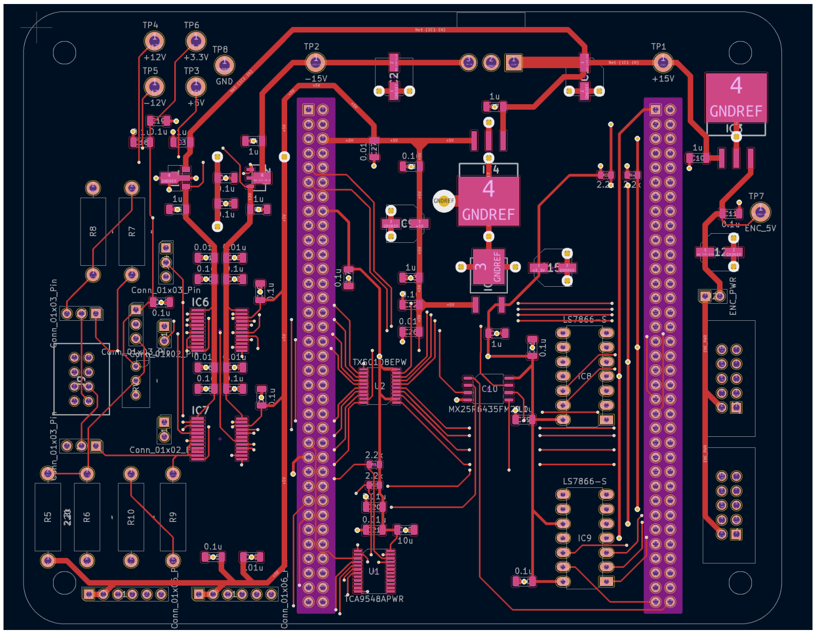

Figure A.4: Motor Control Board Layout - Top Layer

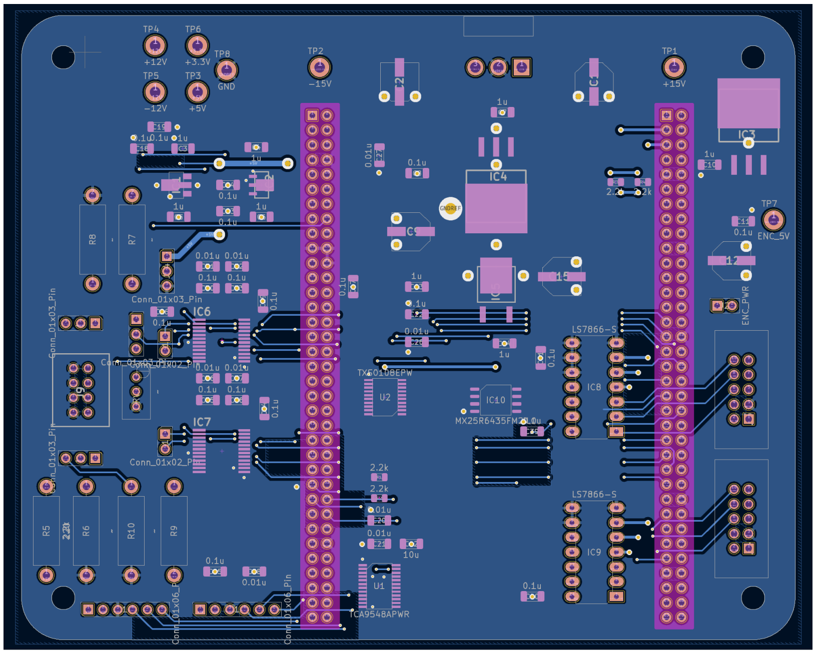

Figure A.5: Motor Control Board Layout - Bottom Layer

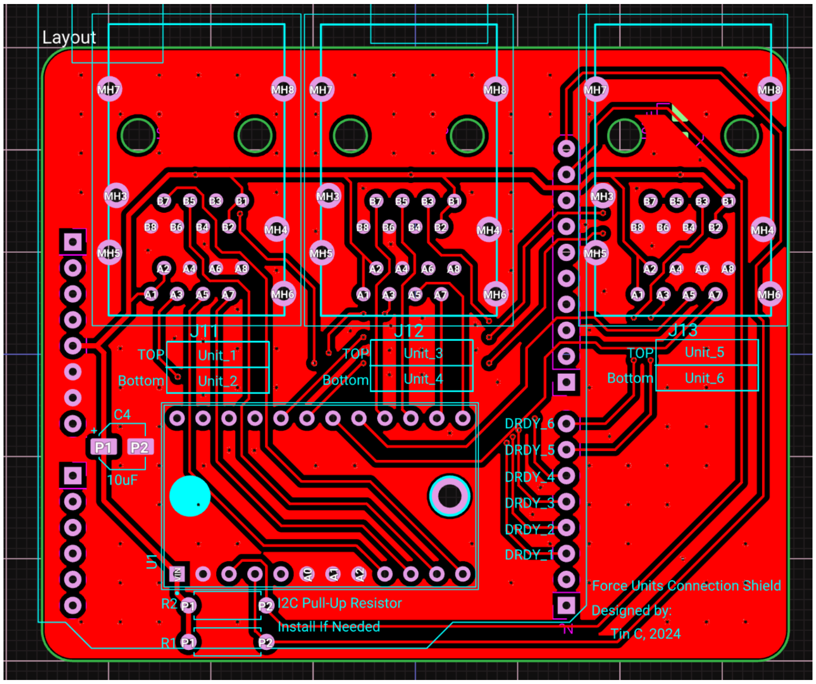

Figure A.6: Force Sensor System Arduino Shield Board Layout - Top Layer

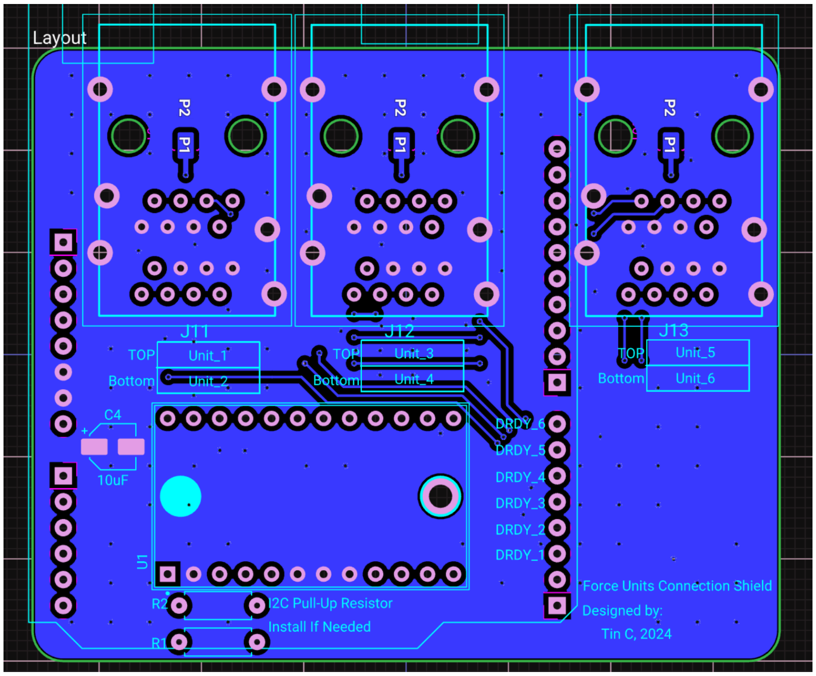

Figure A.7: Force Sensor System Arduino Shield Board Layout - Bottom Layer

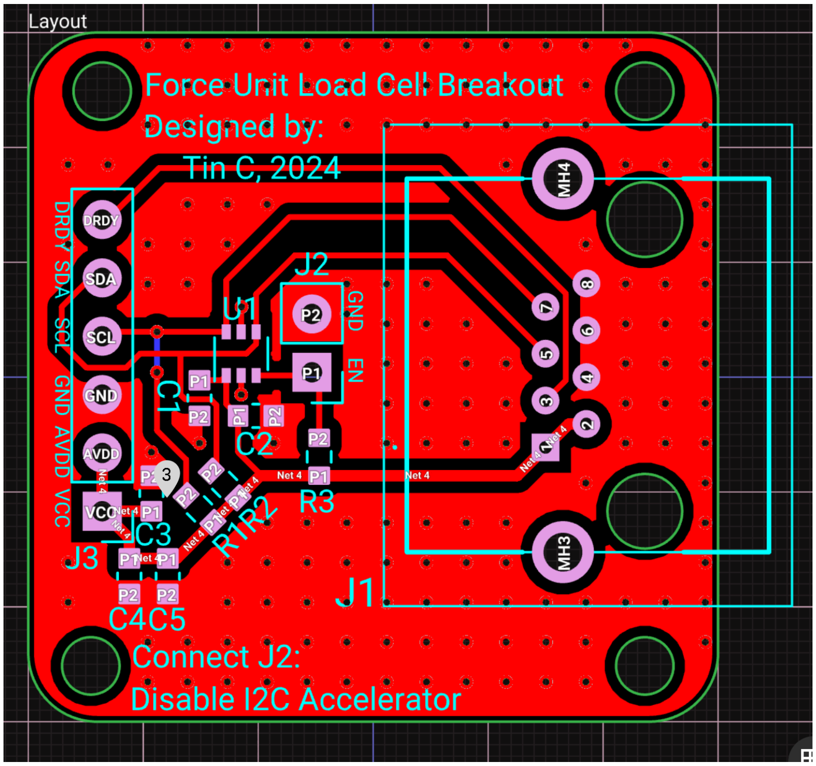

Figure A.8: Force Sensor System Breakout Board Layout - Top Layer

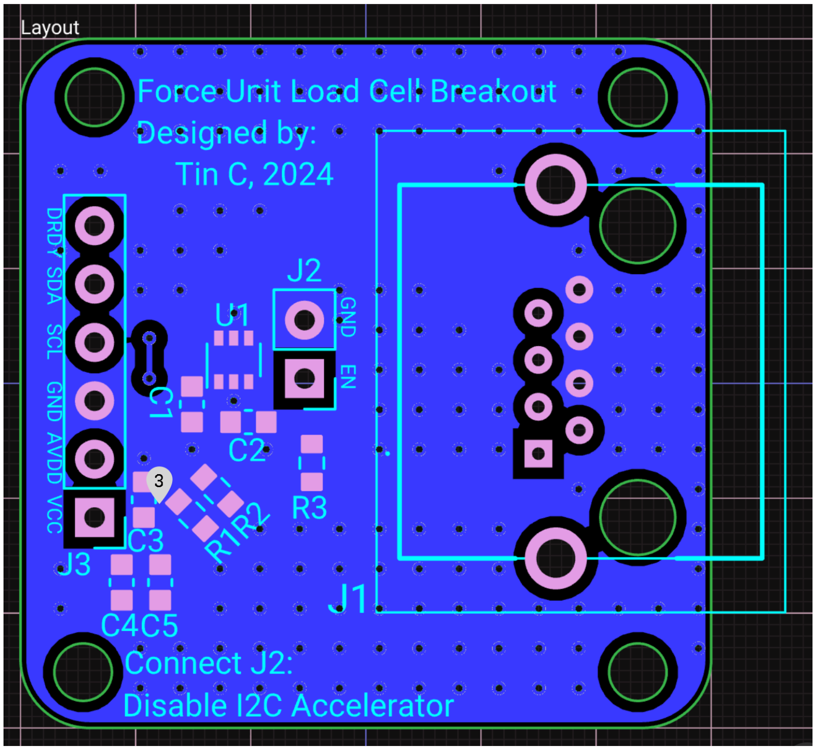

Figure A.9: Force Sensor System Breakout Board Layout - Bottom Layer