Design of Electromechanical Attachments for Improved Ultrasound Imaging Repeatability

by

Ryan Koeppen

S.B., Massachusetts Institute of Technology (2019)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author .....
Department of Mechanical Engineering
May 14, 2021

Certified by .....
Brian W. Anthony
Principal Research Scientist
Thesis Supervisor

Accepted by .....
Nicolas Hadjiconstantinou
Graduate Officer

Design of Electromechanical Attachments for Improved Ultrasound Imaging Repeatability

by

Ryan Koeppen

Submitted to the Department of Mechanical Engineering
on May 14, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Ultrasound imaging (or ultrasonography) is a common tool used for medical diagnostics. It has many advantages over other imaging modalities (such as MRI and CT) such as being more portable, less expensive, and lower power. Ultrasound imaging is emerging as a noninvasive diagnostic alternative in many applications that traditionally rely on biopsies.

Ultrasound imaging also has notable limitations, such as being highly operator dependent and having low resolution at large imaging depths. In recent years, several engineering solutions have been designed to overcome these limitations, such as force-coupled ultrasound, external mechanical vibration (EMV) for shear wave elastography (SWE), and volume ultrasound. Each of these technologies also has its limitations and some have not been optimized for clinical settings.

In this work, these technologies are developed further into attachments to allow for *easier* and *simultaneous* use in clinical ultrasound settings. A more compact force coupling attachment was designed using a linear DC servomotor and validated with external sensors. An external vibration system for SWE, designed in previous work, was developed to improve resistance to debris and its dynamic performance was experimentally validated. An optical tracking module was incorporated for estimating the probe's 6 degrees of freedom and its performance was quantified. Electronic hardware and a Robot Operating System (ROS) network were developed to synchronize the three attachments for control through a single, custom MATLAB application.

The ultrasound probe attachments were used in experiments on calibrated phantoms and human subjects. Initial experimental results validated the effectiveness of force coupling on improving imaging variability. The combination of force coupling and optical tracking enabled force-coupled, elastogram volumes to be created in post-processing.

Thesis Supervisor: Brian W. Anthony
Title: Principal Research Scientist

Acknowledgments

I have so many people to thank for helping me throughout my research. First, many thanks to Brian Anthony for being a great advisor over the past few years. You encouraged me to take ownership of and pave my own path with my research, and that's an invaluable skill to bring with me through the rest of my career.

I also want to thank several labmates for responding to random questions, requests, and discussions over the years. First, thanks to Shawn for being my go-to person for almost every technical question I've come up with throughout my research and encouraging me every step of the way. Thanks to Rebecca for answering all of my random questions about ultrasound machines and teaching me the art of writing IRB's. Thanks to Melinda for partnering with me in human studies and putting up with the hectic timeline near the end. Thanks to Ivan for being my go-to person for random manufacturing questions and for making the transition to this project as easy as possible. Finally, thanks to everyone in my mini-group for giving me feedback throughout all of my research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis motivates and details the design and performance characterization of a system of three attachments for a hand-held ultrasound probe for use during medical ultrasound exams. The system enhancing attachments aim to improve repeatability and/or quality of image acquisition by using one of three techniques: i) controlling contact force between the probe and patient's tissue, ii) low-power, mechanical excitation of tissue (external mechanical vibration, or EMV) for shear wave elastography, and iii) orientation and position tracking using an optical sensor for volume reconstruction.

This thesis begins with an overview of the technologies used in these attachments followed by several clinical use cases for these attachments.

1.1 Medical Ultrasound Imaging

Ultrasound imaging is a commonly used diagnostic tool in a wide range of clinical applications. It can be used for identifying onset of disease, monitoring disease progression, and assisting with invasive procedures. Its main advantages are that it is non-invasive, low power (relative to other imaging modalities), low cost, and portable.

Ultrasound also has a number of notable disadvantages that limit its diagnostic potential. First, ultrasound signals are attenuated by the medium through which they travel, limiting how far into the tissue one can image. Attenuation limitations

can be mitigated either by using a transducer with higher power or a lower frequency. However, acoustic power transmitted through human tissue is limited by the FDA to avoid tissue damage, and lower frequency operation results in decreased spatial resolution. Second, handheld ultrasound requires a human operator (sonographer), introducing variability into image collection that is difficult to reproduce. For example, ultrasound images only produce 2-D slices of underlying tissue which yield different views when imaging tissue with different probe orientation, position, and contact pressure. Noise in the operator's motor control (such as hand tremors) can lead to varying contact force between the patient's body and the probe and cause distortion in images [4]. The sonographer may perform other tasks, such as a biopsy, simultaneously and cannot stabilize the ultrasound probe as easily. Sonographers can also become fatigued due to the duration and number of scans performed over time, leading to further destabilization and increased potential for injury to the sonographer [4].

1.2 Technology Overview

1.2.1 Force Control for Hand-Held Ultrasound Imaging

As discussed in section 1.1, hand-held ultrasound is susceptible to natural variation from the sonographer. Previous work by Gilbertson found that significant variability in contact force and probe orientation exists during a typical ultrasound scan, and this variability limits the ability to collect repeatable images [4]. Gilbertson also designed several attachments for an ultrasound probe which reduced variations in contact force.

Controlling contact force is especially important for quantitative ultrasound imaging techniques, like elastography (discussed more in section 1.2.2), which attempt to estimate tissue properties. In general, biological tissue has a nonlinear stress-strain relationship:

$$\varepsilon = f(\sigma) \tag{1.1}$$

where ε is the local strain in the tissue and σ is the local stress. Young's Modulus (E) is defined as:

$$\frac{1}{E} = \frac{d\varepsilon}{d\sigma} = f'(\sigma) \tag{1.2}$$

Therefore, any measurement of Young's Modulus for a tissue depends on how the tissue is loaded.

Gilbertson's work demonstrated proof of concept for the use of force coupling (i.e. force measurement and/or control) in ultrasound imaging, but force-coupled ultrasound has not been widely used outside of research settings. One possible reason is the lack of ergonomic optimization. Force-controlled ultrasound in particular tends to require bulky hardware and motors to function as intended. Another possible reason is that force control adds less value to qualitative ultrasound (such as B-mode) than for quantitative ultrasound (such as elastography). More research devoted to scaling down force coupling hardware and exploring optimal use cases will inform the avenue for greater commercial adoption.

1.2.2 External Mechanical Vibration (EMV) for Shear Wave Elastography

In recent years, elastography has emerged as a useful ultrasound imaging mode. In traditional B-mode ultrasound scanning, a 2D grayscale slice of tissue displays feature outlines and their relative reflectivity. One limitation of this imaging mode is that image interpretation is subjective and requires a specially-trained clinician (sonographer). Another limitation is that it depends on contrast within the tissue. For example, a slice of tissue could have no pathological features while the bulk tissue could be pathological. Elastography overcomes these limitations by providing quantitative color images in which the color relates to the tissue's elastic modulus. The

quantitative elastic modulus estimate is then compared to known values for healthy and pathological tissue.

The premise of elastography is to dynamically perturb tissue to induce a slow-moving shear wave, then measure the propagation speed of that wave at various points in the tissue using an ultrasound probe. This shear wave speed is directly related to the elastic modulus. In practice, the relationship for converting shear wave speed (V_s) to the local Young's Modulus (E) is:

$$E = 3\rho V_s^2 \tag{1.3}$$

where ρ is the density of the tissue (assumed to be 1000 kg/m^3) [3].

Most current commercial and research systems achieve these perturbations in one of two ways. The first method, traditionally used by elastography-equipped ultrasound machines, is using acoustic radiation force (ARF). ARF excites tissue by sending high power, focused acoustic beams into the tissue. ARF elastography typically requires specialized hardware within the ultrasound machines, but the ultrasound probes do not require much additional hardware, allowing them to maintain a compact form factor. However, ARF-based elastography has many significant disadvantages. Acoustic power is still limited by FDA regulations (just as with B-mode imaging), so measurements may be limited in depth, resolution, or accuracy. This high acoustic power also causes heating in tissue, and consequently requires cooling time in each imaging sequence. The additional time significantly limits the frame rate of elastography to less than 1Hz, making it an inconvenient imaging mode for large or dynamic scan regions. Additionally, the slow frame rate permits more sonographer variation. Nonetheless, ARF-based elastography persists as the current gold standard, used in high-end ultrasound systems such as the Philips Epiq and GE Logiq.

The second method for perturbing tissue for SWE is using external mechanical vibrations (EMV). EMV excites tissue by applying low-amplitude vibrations at the surface of tissue, which then propagate into the underlying tissue. Unlike ARF, EMV does not cause significant heating in the tissue and does not require a large cooling

time. Therefore, compared to ARF systems, EMV systems have the potential to achieve higher frame rates, greater measurement accuracy, deeper tissue excitation, and lower cost implementation. The main disadvantage of EMV-based systems is the added hardware to the ultrasound probe, which can be bulky and ergonomically inconvenient. This disadvantage has largely discouraged wide-scale implementation in clinical settings. One example of a commercial system using EMV for imaging is FibroScan [5]. Chavez discussed existing EMV-based systems in more detail in [3].

1.2.3 Orientation and Position Tracking using V-SLAM

As discussed in section 1.1, ultrasound images are often limited to 2-D images collected during freehand scans. One method for overcoming this limitation is to collect many ultrasound images and stitch them together into a 3D volume [1]. Other imaging modalities, such as MRI and CT, accomplish this stitching with ease because the positions of the 2D slices are known, automated, and relatively fast. This same approach is difficult for ultrasonography because: 1) the ultrasound probe is a freehand tool that is controlled by a human operator instead of a machine, and 2) the tissue being imaged is not necessarily fixed in space (ex. the patient breathes or otherwise moves during the scanning session).

The approach taken by Benjamin to overcome these limitations involved two parts [1]. First, an optical tracking module was rigidly attached to the ultrasound probe to estimate its 6 degrees of freedom in space. The system implemented a vision-based simultaneous localization and mapping (V-SLAM) algorithm to perform this estimation. This solution solves problem (1) listed above by localizing the probe with respect to a fixed coordinate frame. But on its own, this solution cannot solve problem (2) because a patient's may move with respect to that fixed coordinate frame. Benjamin also explored using near-infrared (NIR) imaging to identify superficial vein structures to help reorient images.

1.3 Motivating Clinical Applications

1.3.1 Non-Alcoholic Fatty Liver Disease (NAFLD)

Non-alcoholic fatty liver disease (NAFLD) is a chronic liver disease affecting approximately one billion people worldwide [7]. It ranges in severity from a small accumulation of fat (steatosis) to severe tissue scarring (cirrhosis and fibrosis). In many cases, NAFLD is associated with diabetes and metabolic dysfunction, and if left untreated it can cause end-stage liver disease, hepatocellular carcinoma, and death.

Early detection and continuous monitoring of NAFLD progression is crucial for managing the disease. To date, there are no standardized treatments or therapies to cure NAFLD [5]. Treatment primarily focuses on prevention and eliminating risk factors. Liver biopsy is the gold standard for evaluating disease progression, but recent developments in medical imaging (such as CT, MRI, and ultrasound) aim to create a non-invasive alternative for diagnosis [10, 12, 7]. Among these options, ultrasound is attractive due to its greater accessibility and low cost.

However, ultrasound imaging for NAFLD has its limitations. First, liver imaging occurs through thick layers of fat in the abdomen, requiring low frequency and resolution probes to overcome significant attenuation of the acoustic signal. Many patients with NAFLD are obese, and the resulting excessive attenuation reduces image contrast and the ability to differentiate important features [12]. Second, ultrasound is sensitive to probe positioning, orientation, and contact relative to the patient, which may vary non-repeatably due to dependence on a sonographer [7]. These variations are worsened when scanning obese patients, who may require higher contact forces to acquire adequate images [4]. Third, interpretation of traditional B-mode images by a sonographer is highly subjective and qualitative. Subjectivity is especially problematic for NAFLD diagnosis where images are low contrast.

Augmenting ultrasound with EMV for elastography and optically tracking the probe could significantly improve diagnosis and monitoring of NAFLD by overcoming these limitations. EMV may allow for imaging deeper into tissue and with higher frame rates than ARF-based elastography. Optical tracking can help mitigate the

concern of uncontrolled probe motion by constructing ultrasound volumes of the liver.

1.3.2 Thyroid Nodule Detection and Classification

Thyroid nodules are common in the general population. One major clinical challenge, however, is the ability to distinguish between benign and malignant nodules. The current gold standard for diagnosing thyroid nodules is fine needle aspiration (FNA), a biopsy method. In practice, up to 25% of FNA samples are classified as indeterminate, and in many cases this is due to insufficient tissue samples [9, 8].

Ultrasound imaging, specifically elastography, has been studied and used in recent years as a noninvasive alternative to FNA for thyroid nodule diagnosis. Several studies have found that both strain and shear wave elastography have high diagnostic potential for thyroid nodules, especially when combined with B-mode imaging [9]. Shear wave elastography (SWE) is particularly advantageous because it is quantitatively measures tissue stiffness. In the case of follicular thyroid cancer, SWE was more predictive than B-mode imaging [9].

There are several commonly noted problems with ultrasound elastography for thyroid imaging. First, tissue nonlinearity can lead to higher stiffness measurements when tissue is compressed more [9]. While this is generally a problem for any kind of tissue, it has been shown that thyroid tissue is especially susceptible to this phenomenon due to its close proximity to the probe surface [8]. Second, larger nodules may be more difficult to compress in a controlled manner necessary for elastography [9].

These problems may be overcome with a combination of force coupled ultrasound and volume ultrasound. Previous studies have shown that force coupling can help track the average increase in thyroid tissue stiffness in response to load [8]. Volume imaging may help overcome issues with varying compression by combining images taken at multiple positions and orientations along tissue. Additionally, thyroid volume has in some cases been used as a diagnostic indicator of thyroid pathology [2]. Volume ultrasound may serve as yet another quantitative measure for diagnostics in addition to noninvasively augmenting visualization of tissue.

Chapter 2

Force Coupling Module

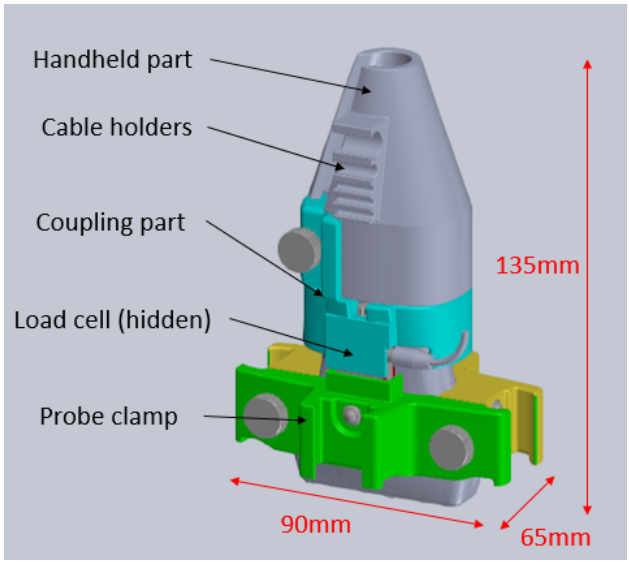
This chapter details the design and engineering validation of force coupling attachments. The force measurement attachment uses a load cell to measure the contact force exerted between the user's hand and the tissue. The force control attachment uses a linear DC servomotor to actively compensate for variations in contact force.

2.1 Force Measurement

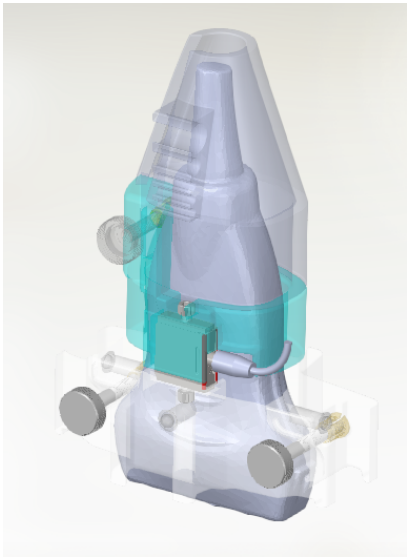
2.1.1 Hardware Design

The diagram of the mechanical design and electronics architecture for the force measurement attachment is shown in Figures 2-1, 2-2, and 2-4. A load cell (Futek LSB200) with a 50-lb rating is mounted on one side to the ultrasound probe clamp and on the other side to a coupling part. In the force measurement configuration, a single handheld plastic component is screwed directly onto the coupling part. The user holds onto this part while performing scans (see Figure 2-2). All plastic components were 3D printed from PLA material using an Ultimaker 3 printer. There is clearance between the coupling part and the ultrasound probe such that the force exerted by the user is transmitted in series to both the load cell and the ultrasound probe. This series configuration gives the most direct measurement of contact force.

The handheld part consists of cable holders to reduce errors due to cables tugging



(a) Labeled CAD model of force measurement components.



(b) Transparent CAD model with ultrasound probe.

Figure 2-1: Mechanical components of the force measurement attachment. The load cell is mounted to the probe clamp on one side and to a coupling part on the other side. The coupling part attaches to a handheld part, which the user holds while using the system. Cable holders allow cables to be managed and also minimize tugging on the load cell.

on the load cell (see Figure 2-3). This cable relief is most important for the ultrasound probe cable, which is heavy relative to the other cables. Any tension or tugging on this cable is directly transmitted to the load cell through the probe. The cable holders allow changes in cable tension to be transmitted instead to the user's hand.

The load cell is connected to an amplifier (Futek IAA100) powered by the system's 12VDC supply. The amplifier contains various DIP switches to easily change the amplifier settings. The DIP switches were set to have a +5V load cell excitation, gain of 509, reverse polarity (so that compressive forces are positive output voltages), and shunt resistance 60.4 kΩ. These settings ensure that compressive forces ranging from 0 to 50-lb result in an output voltage from 0 to 5.095V, close to the max voltage that a 5V microcontroller can read.

The load cell amplifier output is connected to a diode clipping circuit to prevent excessive negative voltages from being applied to the microcontroller analog-to-digital converter (ADC). Negative voltages may occur if the load cell is accidentally pulled,

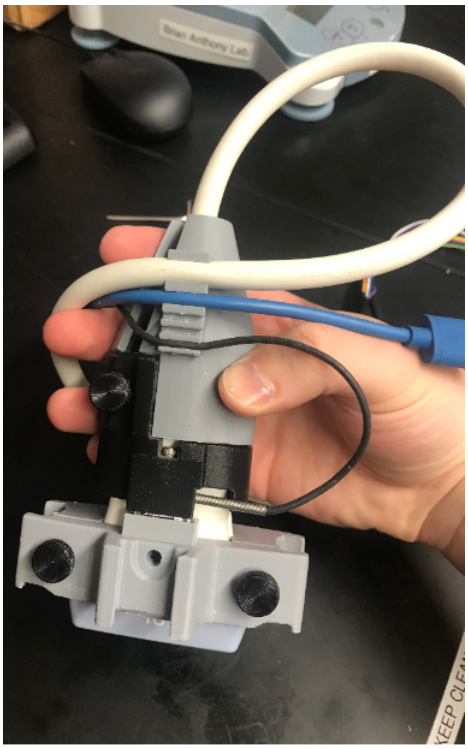


Figure 2-2: Real-life mechanical components of the force measurement attachment while fixed to a probe and held by a user.

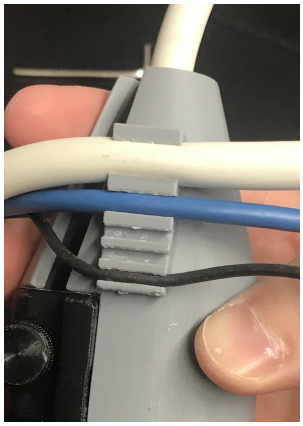


Figure 2-3: Cable relief features on the handheld part of the force measurement assembly.

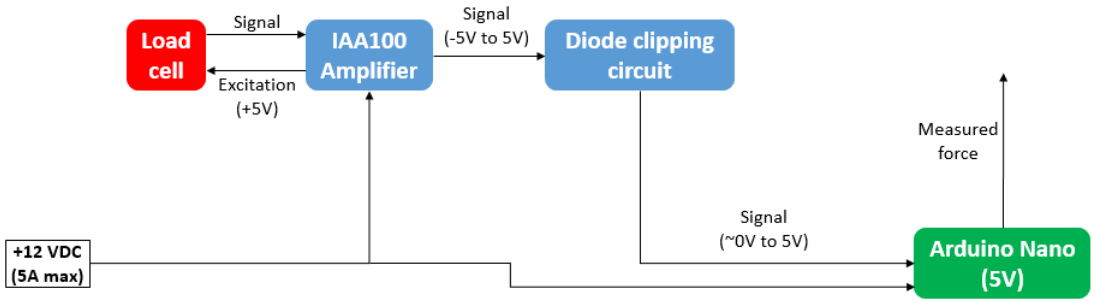


Figure 2-4: Schematic of electrical components for the force measurement attachment. The system consists of a Futek LSB200 load cell, Futek IAA100 amplifier, diode clipping circuit to prevent pin overload with negative voltages, and an Arduino Nano microcontroller.

for example. The diode clipping circuit uses a 47 kΩ resistor in series with a reverse-biased, small-signal diode (IN6263). The resistor value was chosen to limit the voltage drop to below the microcontroller's ADC resolution, while the small-signal diode was chosen to limit the negative output voltage to greater than -0.5V (the maximum negative voltage that the microcontroller's pins can withstand safely).

The diode clipping circuit output is connected to an analog pin on the Arduino Nano microcontroller. The Arduino board is connected via USB to a laptop for data transfer. However, because USB power is often inaccurate and noisy, the board is powered on its "VIN" pin by the system's 12VDC power supply. The 12V power is then regulated down to 5V by the onboard voltage regulator, ensuring the 5V reference for the ADC is more accurate and steady than with USB power alone.

2.1.2 Force Measurement Calibration

The force measurements are calibrated through two separate steps. First, the load cell is calibrated to ensure accurate, linear readings over the ±50-lb range, independent of what it is mounted to. Second, the force measurements are zero-ed and gravity-compensated continuously using a calibration sequence that is triggered manually in the system's MATLAB app. This second calibration accounts for bias in the amplifier, any changes in weight on the end of the probe (ex. attachments were added or taken off), changes in tension on the ultrasound probe cable, and bias due to ultrasound

probe being tilted.

The load cell first needed to be positively biased to ensure the voltage output was positive when mounted to the probe clamp equipped with attachments. These attachments put the load cell in tension, resulting in negative voltage at the amplifier output. To overcome this, the "zero" potentiometer on the amplifier was adjusted while the system was in its heaviest configuration (i.e. ultrasound probe, both thumpers, and camera were all attached) until the output voltage was greater than zero.

The load cell amplifier output was then fine-tuned using calibrated weights. The amplifier output was connected to a multimeter. A 500g calibrated weight was placed onto the load cell while unconnected to 3D printed parts. The voltage increases by an amount determined by:

$$\Delta V = (G)(S)(V_{exc})\left(\frac{Mg}{F_{max}}\right) \tag{2.1}$$

where ΔV is the increase in voltage, G is the amplifier gain as specified in section 2.1.1, $S = 0.002$ is the sensitivity of the load cell (2 mV/V for this LSB201 load cell), $V_{exc} = 5V$ is the excitation voltage of the load cell, M is the mass of the calibrated weight (in kg), $g = 9.81\frac{kg}{m^2}$ is gravitational acceleration, and $F_{max} = 50lb = 222.4N$ is the maximum rated load for the load cell. The "span" potentiometer on the load cell amplifier was adjusted until the multimeter reading was ΔV volts higher than the reading without the weight.

The second calibration step was implemented in the system's MATLAB app. Based on the error sources listed above, it is assumed that each raw reading of the load cell is of the form:

$$F_{LC} = F_{applied} + C + C_g cos(\theta)cos(\phi) \tag{2.2}$$

where F_{LC} is the force reading from the load cell, $F_{applied}$ is the contact force exerted on the probe, C and C_g are calibration constants, θ is the probe's pitch angle, and ϕ is the probe's roll angle. Therefore, the true, compensated contact force was estimated by subtracting an offset from the raw load cell readings:

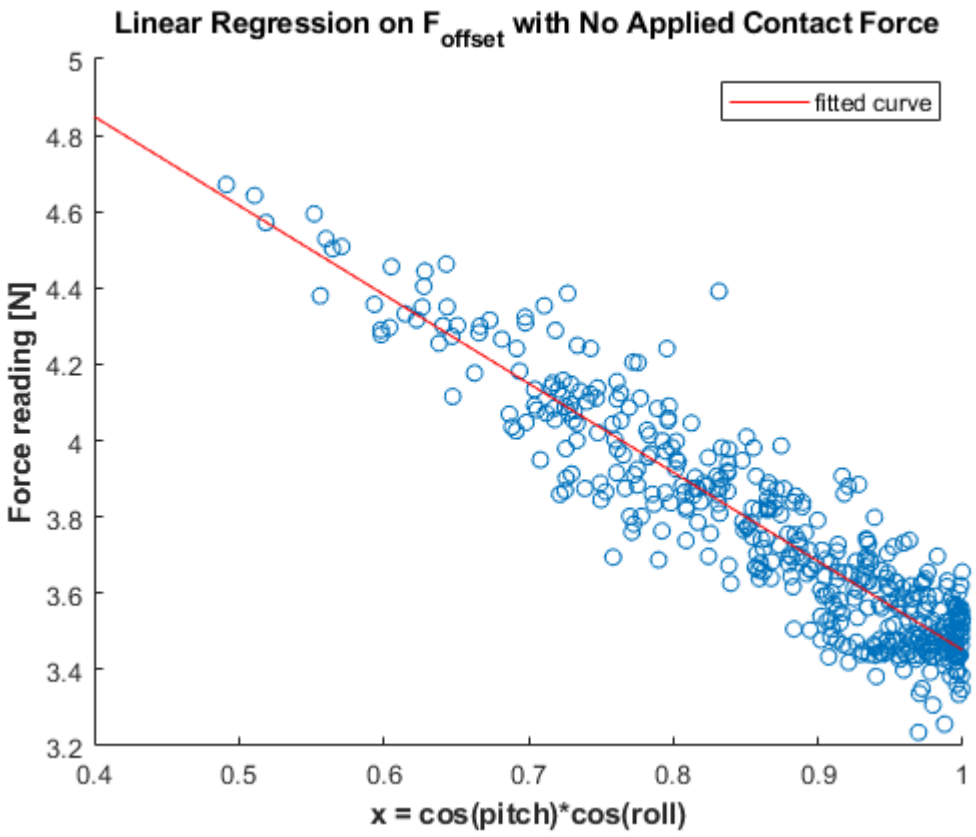


Figure 2-5: Raw force readings from a test when $F_{applied} = 0$ plotted against the variable $x = cos(\theta)cos(\phi)$. The force offset needed to zero the readings follows a linear form $F_{offset} = C + C_g x$ with correlation coefficient $R = 0.926$

$$F_{offset} = C + C_g cos(\theta)cos(\phi) \tag{2.3}$$

To verify this form of F_{offset}, a test was conducted in which the probe was held in the air (i.e. $F_{applied} = 0$) and continuously rotated. Raw force and orientation measurements were taken from the force measurement system and optical tracking system (see Chapter 4). Letting $x = cos(\theta)cos(\phi)$, F_{offset} is a linear function of x. A linear regression was performed on the data and the result is show in Figure 2-5. A clear linear relationship is seen in the plot. The linear regression validates that the linear fit is appropriate, with a correlation coefficient of $R = 0.926$.

The calibration sequence consists of the following steps:

1. The user requests a load cell calibration using the "calibrate" button in the

MATLAB app

2. The system requests raw force readings from the force measurement system

3. The system records the pitch and roll angles of the probe using the camera

4. The system performs a fast linear regression and saves estimates for C and C_g

5. After each update from the camera, the force offset F_{offset} is updated and sent to the force measurement system

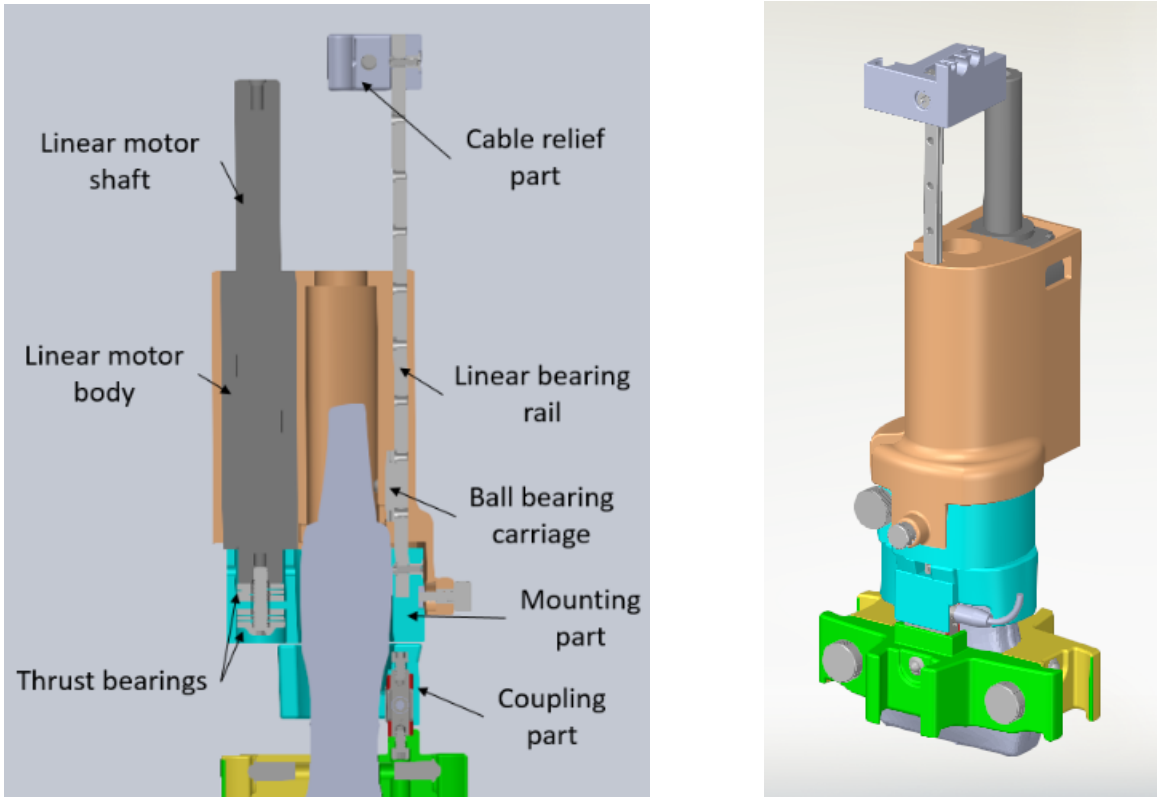
Note that without the camera module (or another orientation measurement sensor), the calibration sequence cannot accurately compensate the force readings. The user may calibrate the probe in an orientation in which they plan to perform scans, but the system would only calculate the constant offset C and assume $C_g = 0$.

2.2 Force Control

2.2.1 Hardware Design

The force control attachment consists of a motorized assembly, which attaches to the coupling part as discussed in section 2.1.1, combined with a dedicated motor driver. The assembly consists of 3D printed PLA plastic parts, a 150mm-long linear ball bearing (McMaster part 8438K1) and rail (McMaster part 6725K23), and a linear DC servomotor (Faulhaber part LM2070-040-01). The motor functions similarly to a brushless DC motor, but with a linear shaft moving through the motor's housing. The motor is connected to the motor driver (Faulhaber part MCLM3003 P RS), which is commanded by a PWM signal from the Arduino microcontroller. The motor driver uses the PWM signal to drive the motor phases, similar to the commutation sequence used in a brushless rotary motor. The motor driver is powered by the system's 12VDC supply.

Figure 2-6a shows a cross-section of the force control attachment assembly. The motor shaft and linear rail are both attached to a part which transmits motion from



(a) Cross-sectional view of assembly

(b) 3D view of assembly

Figure 2-6: CAD renderings of force control attachment assembly with ultrasound probe.

the motor to the ultrasound probe. The motor housing and ball bearing carriage are attached to the handheld part. In this configuration, the linear ball bearing constrains the probe's motion in 5 degrees of freedom. The motor's linear motion provides control of the remaining degree of freedom.

To reduce overconstraint between the rail and motor, a pair of thrust bearings are placed between the mounted end of the motor's shaft and the 3D printed part. If lateral load is applied to the 3D printed part, the thrust bearings help minimize load transmitted to the motor and instead transmit that load to the ball bearing. This configuration also helps reduce friction. Because the sleeve bearings in the motor have higher friction than the ball bearing, transmitting side loads to the ball bearing reduces the likelihood of jamming and damage to the motor.

Position feedback for the linear motor is provided through three built-in analog

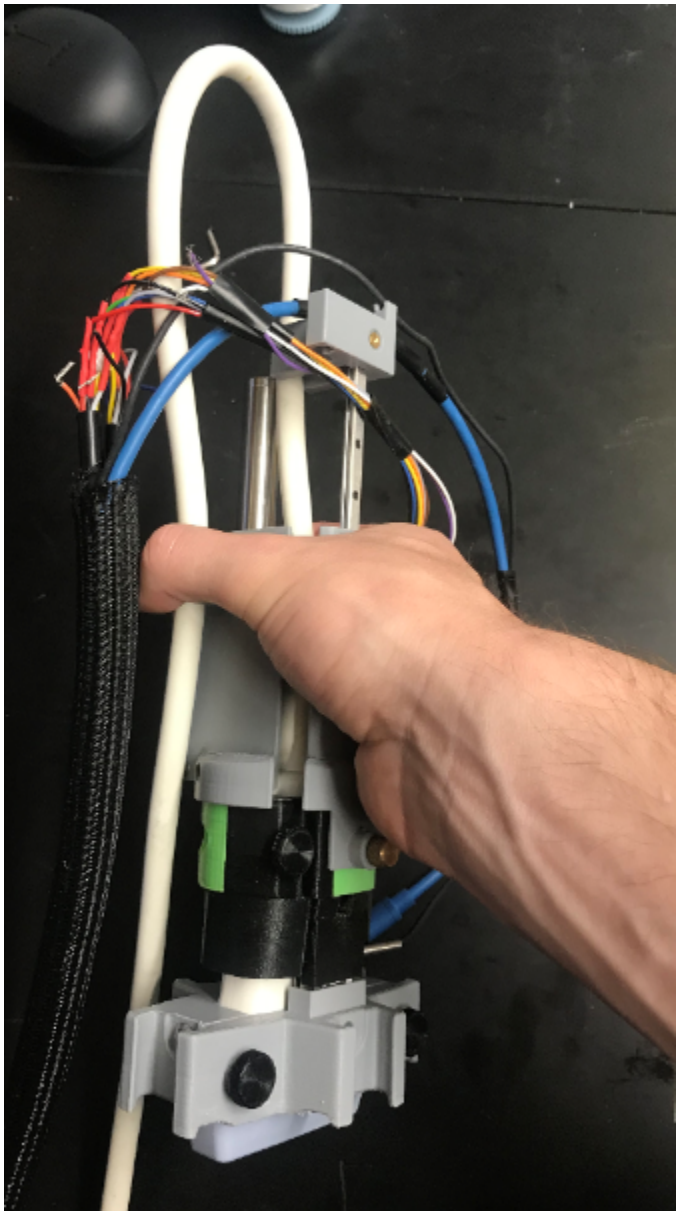


Figure 2-7: Force control assembly being held by a user.

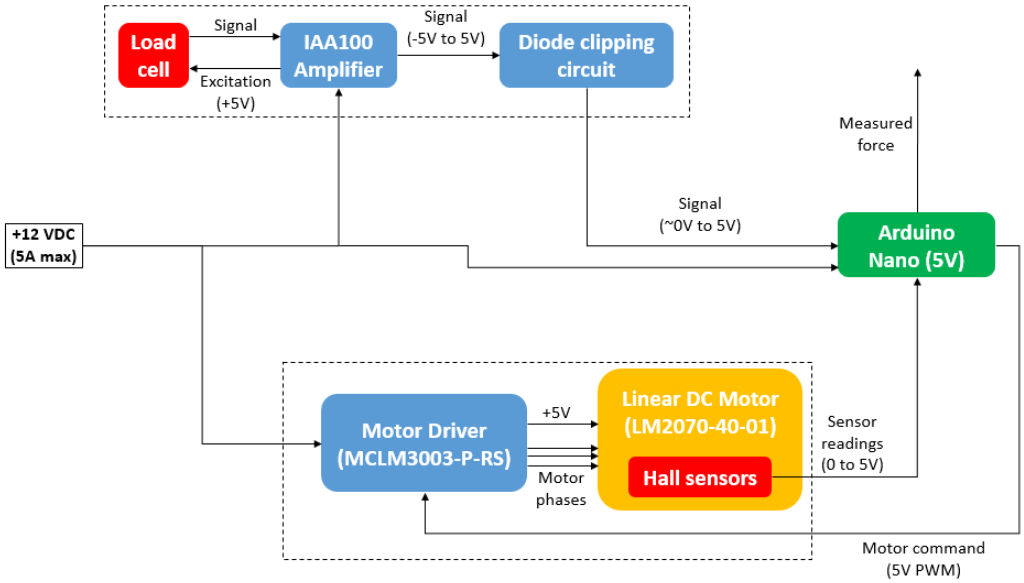


Figure 2-8: Block diagram schematic of the two force attachments.

hall effect sensors. For simple control schemes, these sensors can be directly connected to the motor driver and automatically used for feedback. However, the force control functionality needed for this attachment is beyond the capabilities of the motor driver. Therefore, the analog hall sensor signals were passed to the Arduino and processed using methods provided by Faulhaber. The position feedback, combined with the force feedback from the load cell, is used to command a PWM signal to the motor driver.

Figure 2-8 shows a schematic of the signals used in the force control attachment in addition to the force measurement signals.

2.2.2 Linear Rail Friction

To maximize performance of the linear motor, it was crucial that the degree of freedom controlled by the motor was friction-less as possible. To better understand the magnitude of the friction force due to the bearing, a static analysis was performed using the free body diagram shown in Figure 2-9.

From the free body diagram:

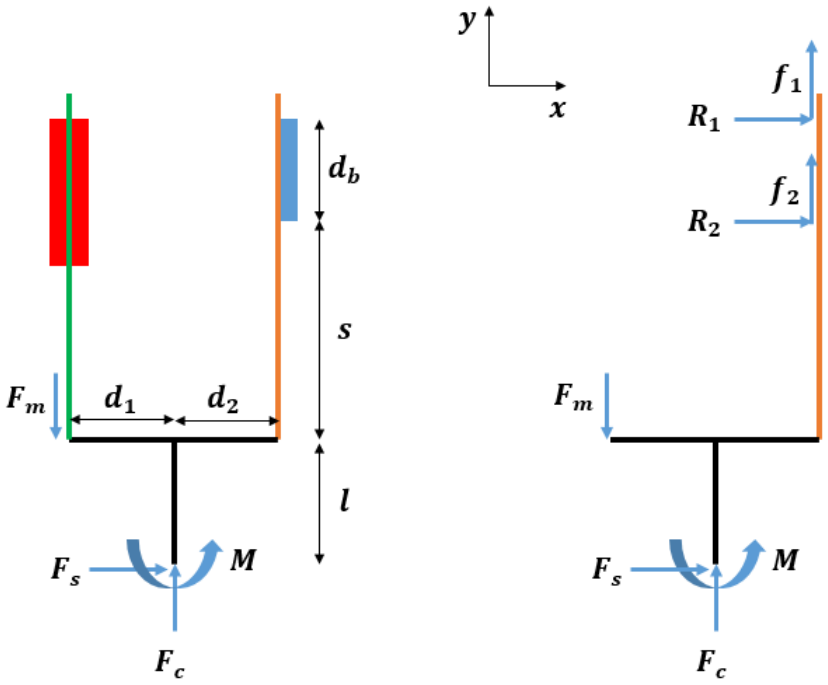


Figure 2-9: Free body diagram of the linear rail assembly for the force control attachment.

$$\Sigma M_{z,1} = d_b R_2 + (d_1 + d_2)F_m + (d_b + s + l)F_s - d_2 F_c + M = 0 \tag{2.4}$$

$$\Sigma M_{z,2} = -d_b R_1 + (d_1 + d_2)F_m + (s + l)F_s - d_2 F_c + M = 0 \tag{2.5}$$

Assuming Coulomb friction for simplicity, the total friction force is:

$$f = f_1 + f_2 = \mu(|R_1| + |R_2|) \tag{2.6}$$

where R_1 and R_2 are reaction forces in the ball bearing carriage, F_m is the force exerted by the motor onto the probe, F_c is the contact force between the probe and the tissue, F_s is side load on the probe, and f_1 and f_2 are friction forces in the ball bearing carriage.

The external forces and moments were estimated based on previous work by Gilbertson. In this work, the forces and moments applied to an ultrasound probe

during a typical abdominal exam were quantified. These measurements are therefore a "worst-case" estimate for this system, which is designed for lower force applications.

From this analysis, it was found that the frictional force from the linear ball bearing was on the order of 0.1N under worst-case loading. This analysis justified the use of a ball bearing carriage to constrain five of the six degrees of freedom.

2.2.3 System Modeling and Simulation

While designing the force control attachment, the assembly was modeled using a 1-D spring and damper model, then simulated using Simulink. This simulation was used to:

- examine the limits of the actuator,
- iterate on various control schemes to optimize performance, and
- ensure stability of the system

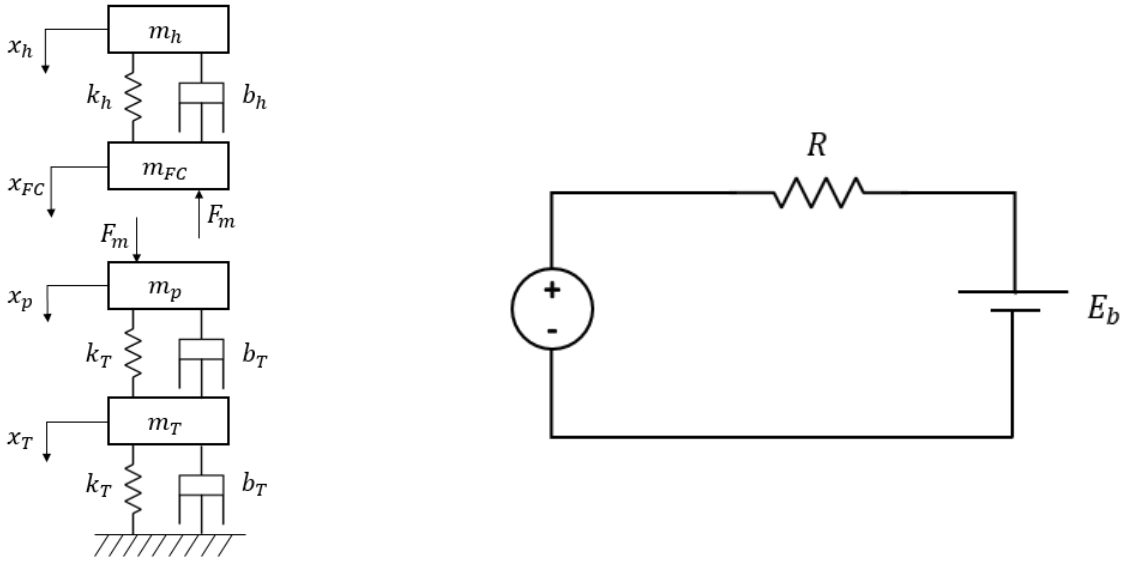
Diagrams of the mechanical model (adapted from [4]) and electrical model are shown in Figure 2-10. The subscripts "h", "FC", "p", and "T" correspond to the user's hand, force control assembly, ultrasound probe, and underlying tissue (respectively). The motor data sheet provides estimates of electrical properties such as force constant (K_F), back-EMF constant (K_E), and winding resistance (R), allowing system behavior to be reasonably predicted with the model and fine-tuned through testing.

The motor's behavior can be reasonably described by the following relations:

$$F_m = K_F\, i \tag{2.7}$$

$$E_b = K_E\, (\dot{x}_{FC} - \dot{x}_p) \tag{2.8}$$

$$V_{in} = iR + E_b \tag{2.9}$$



(a) Mechanical model

(b) Electrical model

Figure 2-10: Diagrams of the mechanical and electrical models used for Simulink simulations of the force control system

$$F_m = m_{shaft}\,(\ddot{x}_{FC} - \ddot{x}_p) \tag{2.10}$$

where F_m is the force exerted by the motor on the shaft, i is the current running through the motor, x_{FC} is the position of the handheld force control assembly, x_p is the position of the probe, E_b is back-EMF voltage, V_{in} is input voltage applied to the motor, and m_{shaft} is mass of the motor shaft.

As a first prototype, this system uses a simple proportional-integral (PI) force feedback controller of the form:

$$V_{in} = \frac{F_{set}}{K_F} R + K_E\, v + K_p(F_{set} - F_i) + K_I \sum_{n=0}^{i} (F_{set} - F_n)\Delta t \tag{2.11}$$

where F_{set} is the force setpoint, v is the measured velocity of the motor shaft, K_p and K_I are controller gains, F_i is the current force sample, and Δt is the time between force samples.

2.3 Error Budgeting

The force attachments have a variety of error sources that may reduce the system's ability to measure or control force accurately and repeatably. An error budget was created to predict the effect that design changes may have on measuring or applying contact force. Those errors include:

- Nonlinearity, hysteresis, and non-repeatability in the load cell
- Inaccuracy in the Arduino's ADC
- Non-constant or inaccurate ADC reference voltage
- Error from the control scheme
- Inaccurate tilt compensation

To predict the overall error in the force coupling system, the contributions from each error source were combined by calculating the root mean square error (RMSE):

$$\delta F = \sqrt{(\delta F_{LC})^2 + (\delta F_{ADC})^2 + (\delta F_{ref})^2 + (\delta F_{control})^2 + (\delta F_{tilt})^2} \tag{2.12}$$

The following equations convert the variables above to a corresponding force applied to the load cell:

$$F = V_{out}\left(\frac{F_{max}}{S\,V_{exc}}\right) \tag{2.13}$$

$$V_{out} = \frac{V_{amp}}{G} \tag{2.14}$$

$$V_{amp} = V_{ref}\left(\frac{b}{2^r}\right) \tag{2.15}$$

where F is the force applied to the load cell, $F_{max} = 50lb$ is the load cell force rating, $V_{exc} = 5V$ is the excitation voltage of the load cell, $S = 0.002$ is the sensitivity of the load cell V_{out} is the voltage output from the load cell into the amplifier, V_{amp} is the

Error Source	Predicted Error in Force (N)
Load cell nonlinearity, non-repeatability, etc.	0.33
Arduino ADC random fluctuation	0.43
Inaccurate ADC reference voltage	0.04
Control scheme steady state error	0.05
Incomplete tilt compensation	0.11
Total error	**0.56**

Table 2.1: Force coupling attachments error sources and their contributions to the total error in measured force

voltage output from the amplifier, G is the amplifier gain, $V_{ref} = 5V$ is the reference voltage of the Arduino ADC, b is the number of bits registered by the Arduino ADC, and $r = 10$ is the resolution (in bits) of the ADC.

The propagation of errors method was used for each variable to determine its contribution to errors in force estimates:

$$\delta F_x = \frac{\partial F}{\partial x}\delta x \tag{2.16}$$

where x is the variable of interest, and δx is the error in the variable. For example, the error due to the load cell (F_{LC}) is calculated as:

$$\delta F_{LC} = \frac{\partial F}{\partial V_{out}}\delta V_{out} = \left(\frac{F_{max}}{S\,V_{exc}}\right)\delta V_{out} \tag{2.17}$$

where δV_{out} can be calculated from specifications on the load cell's data sheet.

Table 2.1 summarizes the error sources and their predicted contributions to the total error in force.

2.4 Experimental Validation

Several experiments were performed to validate the performance of the force coupling attachments. In these tests, the two attachments were attached to the ultrasound probe and pressed against an external force sensor (Vernier Dual-Range Force Sensor). Measurements were collected from both the system's load cell readings and the external force sensor. The readings from the two sensors were compared to determine

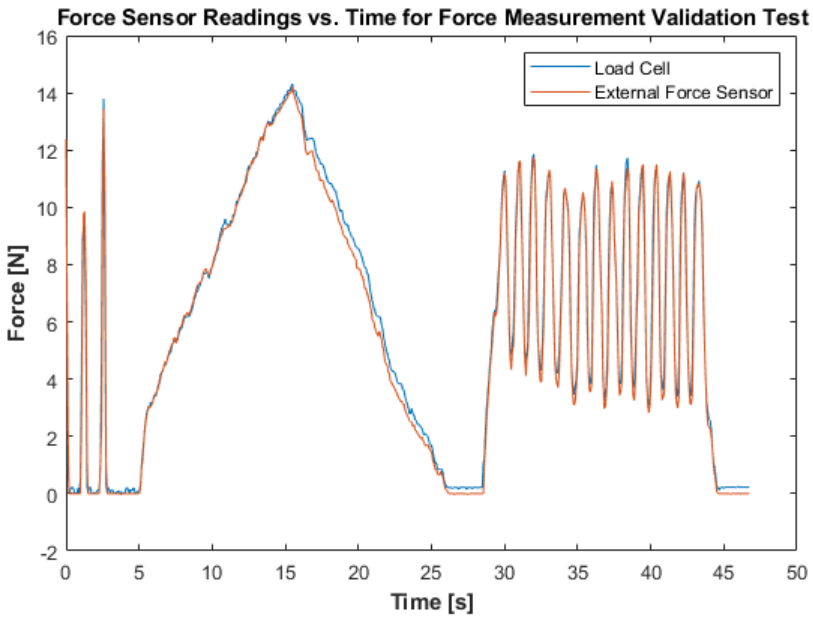


Figure 2-11: Time series data of the force sensors.

system accuracy and performance under quasi-static and dynamic loading.

First, the force measurement system was validated to ensure the load cell readings were accurate after calibration. Figure 2-11 shows the loading on the sensors over time during the test. At the beginning of the test, three sharp taps were performed to allow the two datasets to be synchronized. The data were then resampled and plotted in MATLAB and linear regression was performed. A scatter plot and the fit from the regression are shown in Figure 2-12. The results from this test show that the load cell readings closely match an external force sensor, indicating that the force measurement system is accurate. The zero offset from the regression $b = 0.23N \pm 0.05N$ is statistically different from zero, indicating that the load cell was not fully zero'ed properly. This is a known issue with the force coupling system and may be addressed in future work.

Second, the force control system was validated during quasi-static operation using a slow force sweep test. The force setpoint was continuously varied between two force levels. Figure 2-13 shows the force profile (varied between 2N and 8N) as measured by both the load cell and the external force sensor. A linear regression between the two sensor readings was then performed and the result is shown in Figure 2-14. As

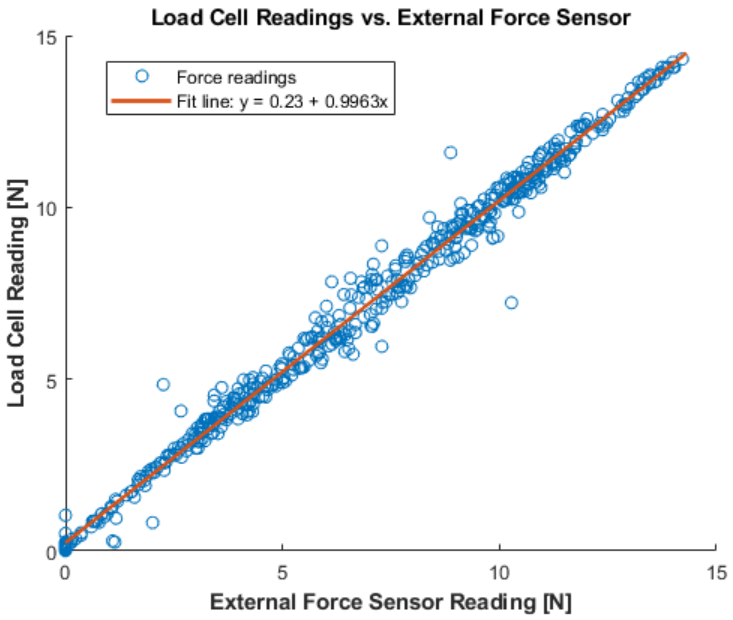


Figure 2-12: Scatter plot and linear regression of force measurement validation test from Figure 2-11. The fit line is of the form $y = ax + b$, where $a = 0.9963 \pm 0.007$ and $b = 0.23\text{N} \pm 0.05\text{N}$ with 95% confidence.

with the force measurement test, the scaling between the sensors is very close to 1 (i.e. as the force increases by 1N in one sensor, the other also increases by close to 1N). The zero offset, however, is farther from zero than during the force measurement test. This result indicates that the load cell was not zero'ed completely before use. Zero-ing the load cell is still an open issue with the force control attachment, and appears to be more difficult and less consistent than with the force measurement attachment.

In a final experiment, the force control system was validated during dynamic operation. A linear stage (FUYU FSL40) was programmed to follow a sinusoidal position profile to simulate disturbances from the operator's hand (such as a tremor). The setup is shown in Figure 2-15. While the stage position oscillates, the contact force on the stage was measured with the external force sensor. Using the MATLAB Curve Fitting tool, a function of the form $F = A\cos(2\pi ft + p) + B$ was fit to the data points after reaching a steady state. Here, f is the known frequency of oscillation (in Hz), F is the contact force, t is time (in sec). A, p, and B are fit parameters corresponding to the amplitude, phase, and constant offset (respectively) of the data.

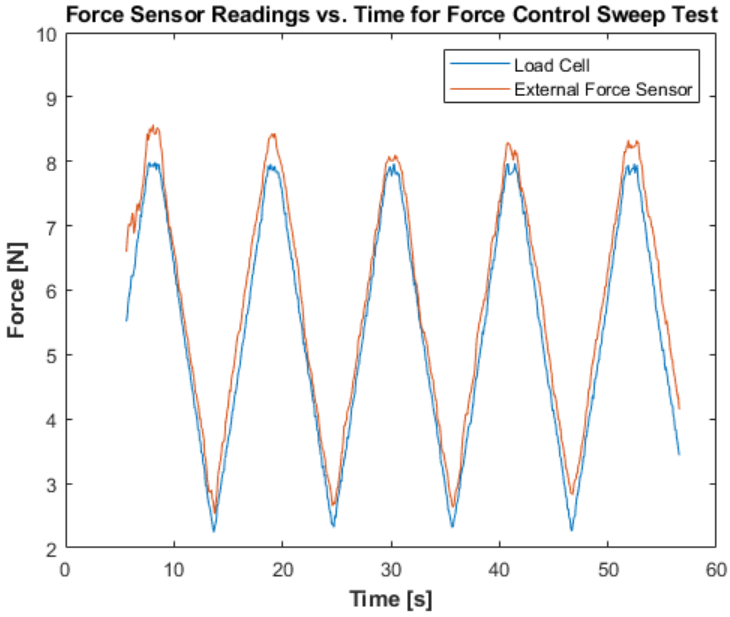


Figure 2-13: Time series data of the force sensors during the slow force sweep test.

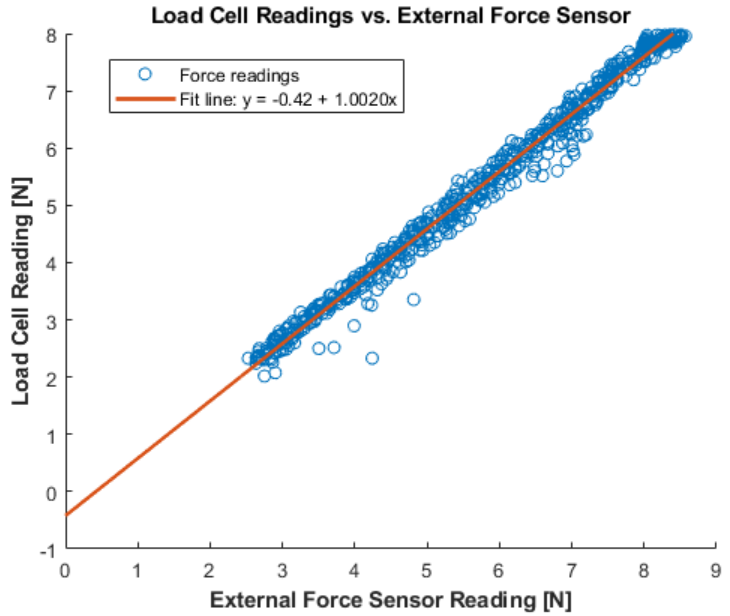


Figure 2-14: Scatter plot and linear regression of force measurement validation test from Figure 2-13. The fit line is of the form $y = ax + b$, where $a = 1.002 \pm 0.008$ and b = -0.42N $\pm$ 0.05N with 95% confidence.

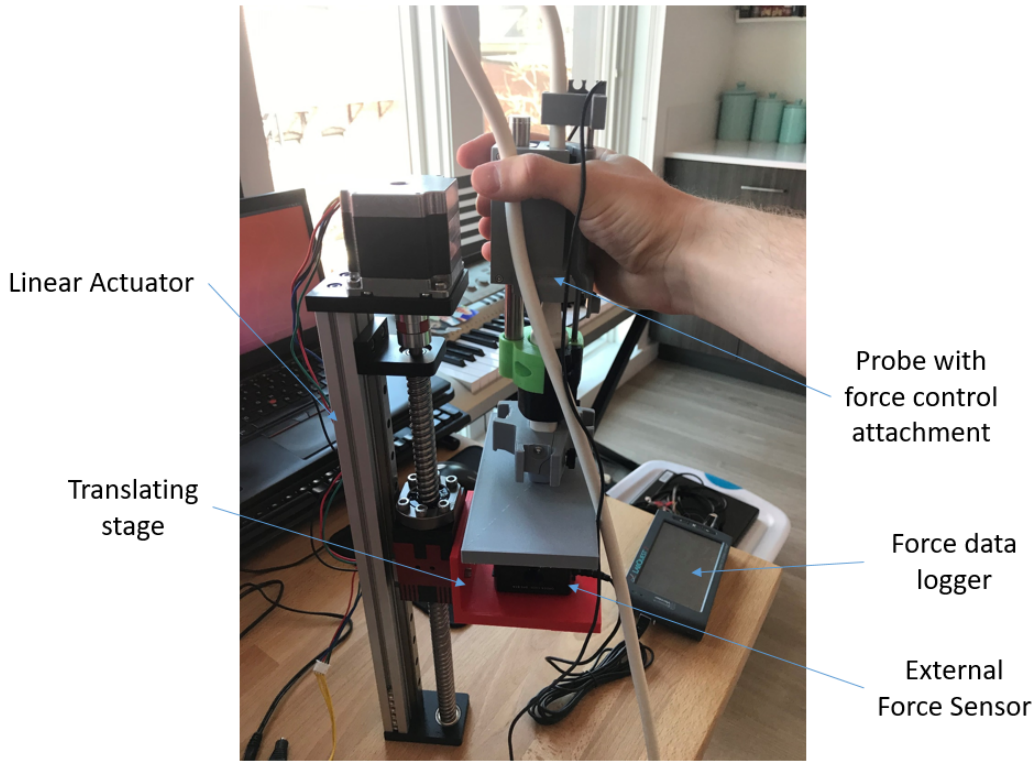


Figure 2-15: Linear stage test setup used for force control dynamic validation test.

The values for fit parameter A were then used to characterize the oscillatory error in the force readings. These measurements were plotted against the Simulink model Bode plot and are shown in Figure 2-16. The model consistently underestimates the force error, but the measurements were of similar order of magnitude to the model at each frequency. This error may be due to friction in the mechanism, which is generally low (as discussed in section 2.2.2) but may be more evident when the total force error is of similar magnitude.

The predicted and measured errors are likely to have little impact on the ultrasound image quality. For example, assuming an extreme situation where a hand oscillates with 0.002m amplitude at 5Hz, the model and experimental data predict an error of approximately 0.4N. Many tests in Chapter 6 were conducted with contact force of approximately 5N, resulting in less than 10% error in a worst-case scenario.

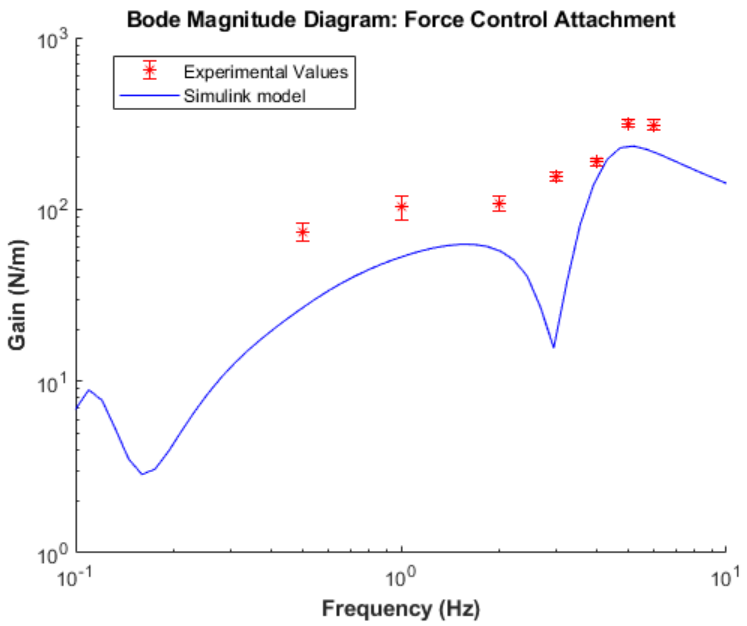


Figure 2-16: Bode magnitude plot relating the input hand disturbance to the contact force fluctuation.

2.5 Future Work

The force attachments can be further refined to reduce errors and make performance more robust. Many components, such as the load cell, were used simply because they were already owned by the lab and were not necessarily optimized for this particular system.

One major improvement that could be made is using a lower capacity load cell and/or increasing the IAA100 amplifier gain. The LSB200 load cell had a capacity of 50 lbs, which is much higher than forces typically seen during ultrasound imaging. Therefore, only a small portion of the total range was used. Furthermore, the full range of the Arduino's ADC could not be used safely.

A separate IMU (or other orientation measuring sensor) could be added to the force coupling modules in order to make them standalone. In the current implementation, the force calibration relies on the pitch and roll measurements from the optical tracking module. Because all attachments were on the probe during experiments, there was no need for a separate sensor. For clinical deployment or further development, it may be useful to allow force coupled ultrasound without the need for the

camera.

For the force control attachment, a simple PID controller was used for ease of implementation and tuning. It is likely that other control schemes could be used to yield better performance, but they were not explored in this thesis. Mechanical improvements could also be made to improve the attachment's performance. For example, a higher quality linear rail could be used that has lower friction and could lower the force error.

Chapter 3

External Mechanical Vibration (EMV) Module

3.1 Hardware Overview

The external mechanical vibration (EMV) module has been developed extensively in previous work by Yang and Chavez [3, 11]. Two "thumper" cases are rigidly attached to an ultrasound probe with a specially designed clamp, as shown in Figure 3-1. Each case contains a voice coil actuator (Bei Kimco LAS04) with a 3D-printed 6mm diameter spherical end effector. The cases are secured to the probe clamp with ball plungers. The end of the case is gel- and debris-proofed using a silicone-molded cover while allowing the end effector to have direct contact with the skin.

The voice coil actuators (VCAs) are driven with a DRV833 motor driver (Pololu) and controlled with a Teensy 3.5 microcontroller. The voice coil actuators have built-in position encoders which are powered by 5V input and provide 0 to 5V output. This output voltage is sent to the microcontroller through a voltage divider to scale the voltage to a 0 to 3.3V range. The 5V power source is provided by an LM2596 DC-DC converter, which scales down the 12V system supply voltage. The motor driver is also supplied 5V power for driving the voice coil actuators. The motor driver is commanded with a PWM signal from the microcontroller. This PWM signal is modulated using a digital position feedback controller, designed by Chavez.

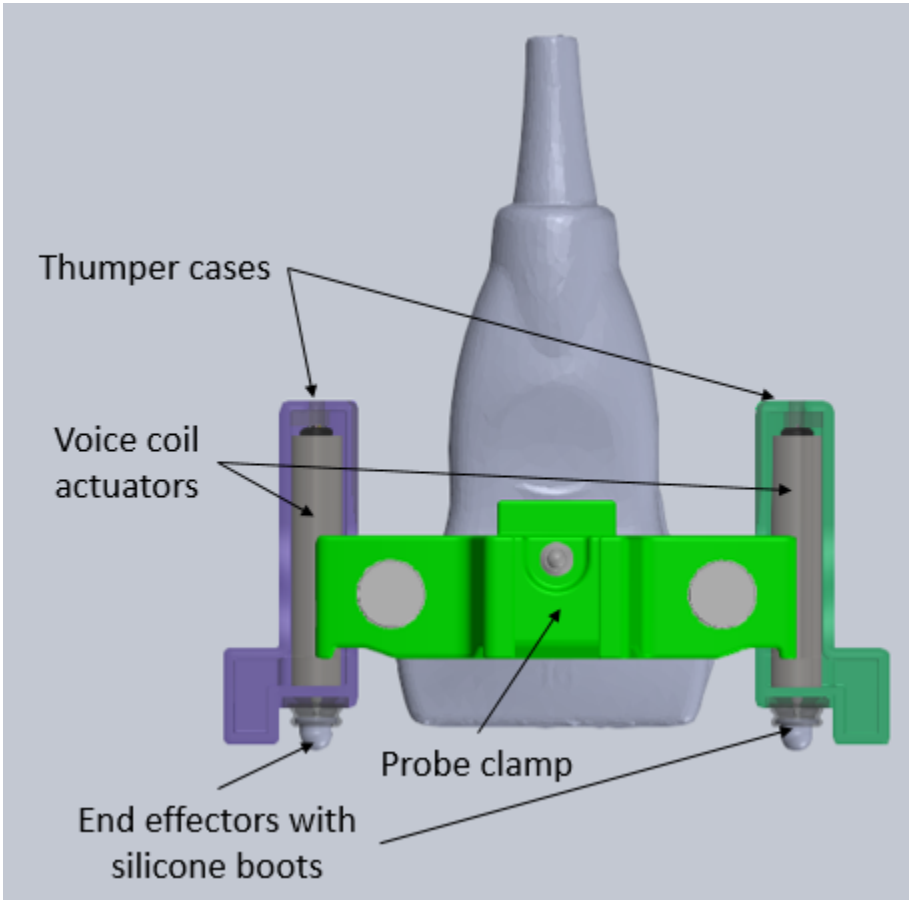


Figure 3-1: CAD rendering of thumper cases.

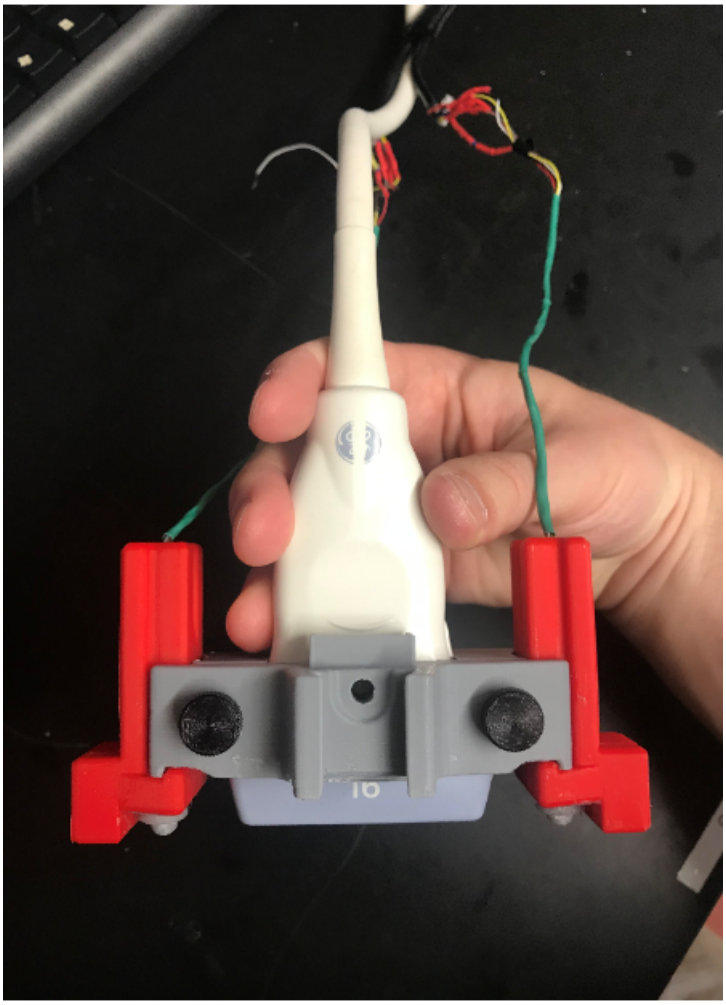


Figure 3-2: Thumper cases and probe clamp attached to probe.

To properly perform shear wave elastography with the EMV module, the VCAs should vibrate closely in phase. The VCA's were driven with separate digital controllers, but each controller was commanded with the same position setpoint. This control structure helps overcome disturbances that affect each VCA separately.

3.2 Synchronization with Ultrasound Machine

In this work, several methods for synchronizing the VCA pulses with the ultrasound machine were considered:

1. Open loop triggering: The EMV module is programmed to pulse the VCA's at a fixed frequency. The times at which the pulses occur are recorded. Synchronization requires either the force coupling module or optical tracking module to synchronize the system's ROS time with the ultrasound machine's time.

2. External trigger: The EMV module reads a signal from a output trigger line from the ultrasound machine to synchronize pulses with the machine's imaging sequence.

In principle, the external vibrations from the VCA's replace the acoustic radiation force (ARF) push traditionally used in commercial elastography systems. When using an ultrasound machine without an elastography imaging mode, method (1) provides a simple way to trigger the VCA's. In the ROS environment, the time of each EMV pulse is recorded and synchronization can occur offline. One advantage of this method is that the ultrasound machine can be set to continuously observe tissue motion (such as in B-mode imaging), and elastography could be performed at a higher frame rate than is possible with method (2).

When using an ultrasound machine with an elastography imaging mode, method (2) provides a way to coordinate the VCA pulses with the machine to achieve real-time imaging. An important note is that the ARF push must be disabled to use this method. The EMV module can be programmed to recognize the ultrasound machine's unique trigger line pattern and trigger a VCA pulse accordingly. The

ultrasound machine's built-in imaging sequence is then used to observe the tissue displacements and reconstruct an elastogram in real-time. While frame rate will still be limited by the ultrasound machine's imaging sequence settings, images can easily be collected in real-time from many different machines with minimal hardware or software modifications.

For the tests detailed in chapter 6, method (1) was used because the available ultrasound machine (GE Logiq E9) could not disable the ARF push without significant software modifications.

3.3 Experimental Validation

The closed-loop dynamic performance of the voice coil actuators is crucial for shear wave elastography. A dynamic tracking test was performed to experimentally measure the system's bandwidth and generate an estimate of its Bode diagram.

The test was conducted by continuously commanding the actuator's position and measuring its actual position over a period of approximately 30 seconds. Traditionally, one could do a "frequency sweep", in which the target position is sinusoidal with a certain frequency. That frequency is slowly increased to measure the system's output at discrete input frequencies. A simpler and faster way of achieving the same result involves commanding a target position generated from a Gaussian random distribution. Deconvolution techniques can then be used to estimate the transfer function relating the input (target position) and output (measured position).

The raw data were imported into MATLAB and processed using the System Identification Toolbox. The toolbox converts time-domain data to frequency-domain data and estimates the system's frequency function. This empirical Bode diagram for the closed-loop thumper system is shown in Figure 3-3. The magnitude part of the Bode diagram reveals that the system bandwidth is approximately 80Hz, where the magnitude (in decibels) becomes slightly negative. At frequencies higher than 80Hz, the amplitude of the output position is less than the amplitude of the target position and fails to follow the target position accurately.

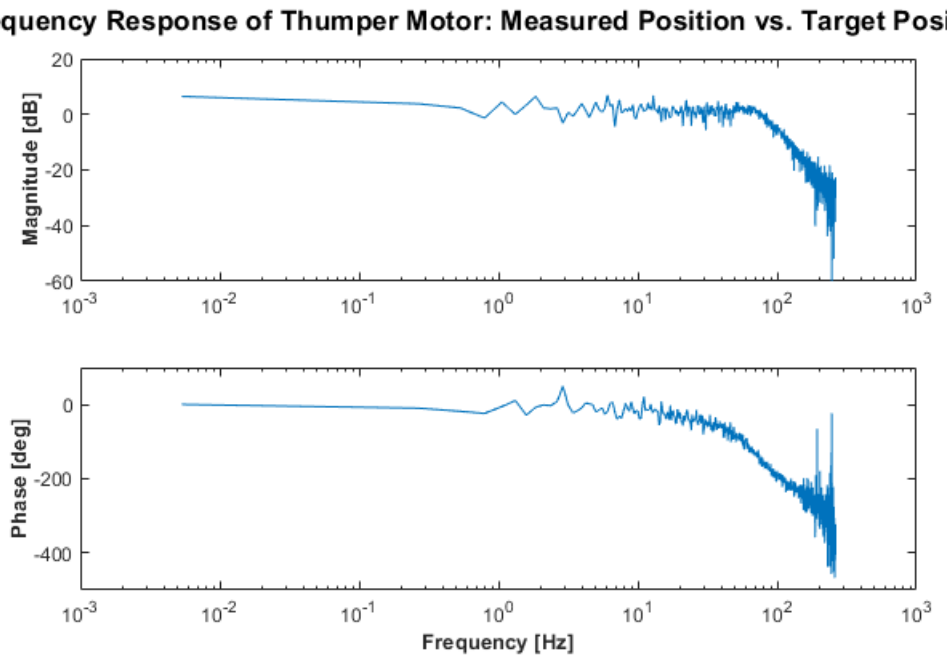


Figure 3-3: Frequency response (Bode) diagram of the closed-loop thumper system. The system input is the target position and the output is the actual (measured) position.

3.4 Future Work

This work focused on developing and validating the EMV hardware designed in work by [3]. The voice coil actuators were repackaged into easy-to-remove, debris-proofed cases for easier deployment to clinical settings. Different synchronization methods were explored and implemented to allow the EMV system to work with different kinds of ultrasound machines. The EMV system was integrated into the ROS environment with the other attachments for easy module integration. Finally, the VCA dynamic performance was characterized to inform its use in further research.

Due to logistical limitations with the available ultrasound machines, reliable data could not be collected to verify that the EMV system produces acceptable elastograms. Future work should focus on adapting the system to work with another ultrasound machine that can either disable ARF pushes or collect fast RF data for offline data analysis.

One interesting EMV concept that may be worthwhile to explore in the future is

using the force control motor for vibrations. Similar work with the PrOSE elastography system showed that EMV-based shear wave elastography is possible by vibrating the entire ultrasound probe [6]. Initial results showed that the predicted shear wave speeds with this system were very similar to those predicted by traditional ARF-based SWE. The PrOSE system enabled much higher frame rates during elastography. It has potential limitations, such as the inability to steer the vibrations in a particular direction. Nonetheless, using the linear motor for both force control and EMV elastography would allow the external hardware to be significantly scaled down for easier use in clinical settings.

Chapter 4

Optical Tracking Module

Volume ultrasound relies on having knowledge of the ultrasound probe's position and orientation (or *pose*), with respect to the tissue, for each image taken. This chapter overviews the optical tracking hardware chosen for pose tracking and the characterization of its performance.

4.1 Hardware Overview

This system implements pose tracking using an optical tracking module; specifically, it uses an Intel RealSense Tracking Camera T265. The module contains two image sensors with fisheye lenses and an IMU unit. The sensor readings are combined through an onboard visual processing unit. This unit performs visual inertial simultaneous localization and mapping (V-SLAM) automatically and outputs, among other variables, the six degrees of freedom of the camera at a rate of 200Hz.

The inclusion of a camera has several advantages for a medical imaging application. First, it eliminates significant drift inherent to IMU's alone. Second, it allows for the camera poses to be *relocalized*. In a medical context, patients may shift around while being imaged, or be scanned in a different room than they were in a previous clinic visit. This particular system does not deal directly with the problem of relocalization, but the onboard camera makes this feature possible in future work.

(a) Side-mounted configuration

(b) Front-mounted configuration

Figure 4-1: Test setup for camera mount repeatability tests. A laser pointer was rigidly attached to the mount in two configurations to create a large lever arm.

4.2 Experimental Evaluation

4.2.1 Repeatability of Mechanical Mount

One major source of non-repeatability in the camera module is the mechanical mount, which slides into one of the slots on the probe clamp and is secured by a spring plunger. To quantify the repeatability of this mounting scheme, a test was performed using a laser pointer to create a large lever arm for measuring angles.

The laser pointer was securely attached to the camera mount part in two different configurations as shown in Figure 4-1. Two configurations are necessary because configuration (a) only measures yaw and pitch, while configuration (b) only measures yaw and roll. One side of probe clamp was securely fixed to a table located approximately 300 inches from a wall. A piece of paper was taped to the wall such that the laser pointer hit the paper when the camera mount was inserted into the probe clamp. For both laser pointer configurations, the camera mount was repeatedly inserted and removed from the probe clamp. After each insertion, the location of the laser pointer was marked on the paper with a small dot.

The horizontal and vertical positions of the dots on the paper were then measured using calipers. One dot was designated as the origin, and all other dots were measured relative to that point. The distances were transformed into angles (relative to the first data point) with:

Angle (degrees)	Pitch	Roll	Yaw
Standard deviation	0.03	0.02	0.03
Range	0.12	0.09	0.10

Table 4.1: Repeatability of the three angular degrees of freedom of the camera mount.

$$\theta = \frac{180}{\pi}\frac{d}{l} \tag{4.1}$$

where θ is the angle (in degrees), d is the measured distance, and l is the lever arm distance.

Repeatability was estimated using both the standard deviation and range of data. Standard deviation gives an "average" variability, while range gives a "worst-case" variability. The results are summarized in Table 4.1.

It should be noted that these tests did not quantify the repeatability of the translational degrees of freedom. Errors in these degrees of freedom were assumed to be small relative to errors introduced by the camera.

4.2.2 Repeatability of T265 Camera

The V-SLAM algorithms used by optical tracking systems can also introduce errors when estimating an object's pose. To quantify this error for the T265, a test was performed in which the camera was repeatedly positioned at two different points using kinematic couplings. Kinematic couplings allow objects to be precisely positioned with high repeatability. In this test, it assumed that any errors in pose introduced by the kinematic coupling mounts are much smaller than the errors introduced by the camera itself.

A picture of the kinematic couplings are shown in figure 4-2. While collecting data on a laptop, the camera is manually moved between the two bases (secured to the table with double-sided tape). The standard deviation and range for each degree of freedom were calculated with the mean-compensated data points. That is, the mean for each location was subtracted from the corresponding readings and all points were used in the calculations. The results are shown in table 4.2.

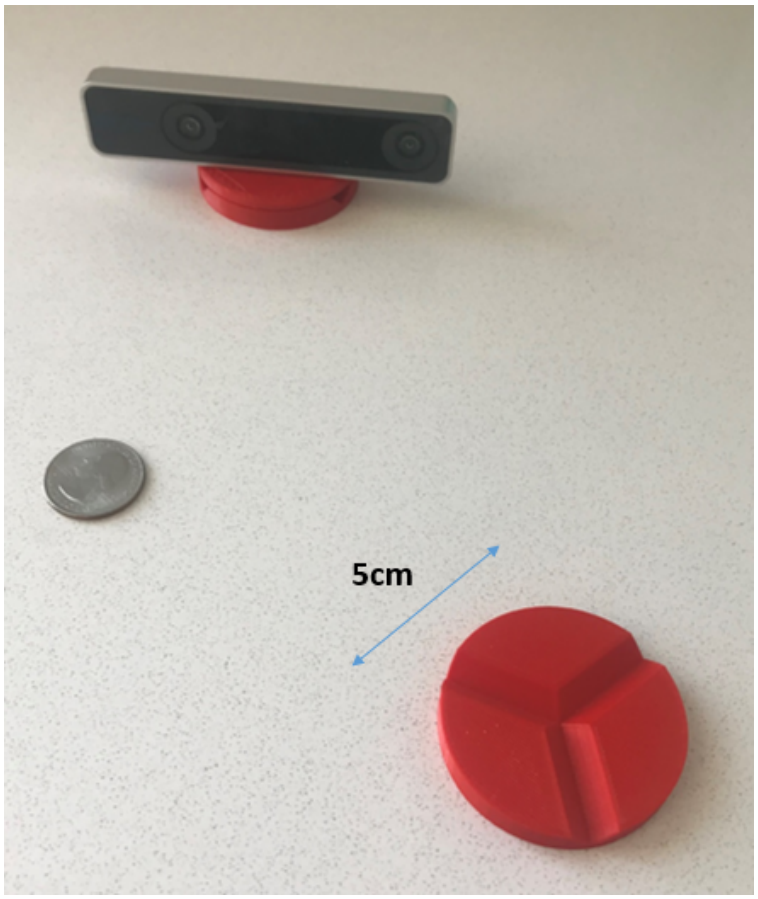


Figure 4-2: Kinematic coupling mounts used to test the T265 camera's measurement repeatability.

	Pitch	**Roll**	**Yaw**	**X**	**Y**	**Z**
Standard Deviation	0.21	0.18	0.14	3.4	3.1	9.0
Range	0.85	0.64	0.50	14.4	10.0	33.3

Table 4.2: Standard deviation of each degree of freedom for the T265 camera, as measured using the kinematic coupling test. The three angles (pitch, roll, and yaw) are measured in degrees, and the three translations (X, Y, and Z) are measured in millimeters.

The standard deviation and range provide two measures of spread for the camera readings. The ranges for the three angles were less than 1 degree, and the standard deviations were less than 0.25 degrees. For the translations, the standard deviations were less than 1cm and the ranges were approximately 3cm or less, with the Z direction having highest variability. Angles are estimated with high precision, which is expected since the accelerometer in the camera's IMU provides a reliable "reference" from gravity. Translations have higher variability since they are impacted more by errors or drift in the sensors.

4.3 Future Work

This work focused on using the Intel RealSense T265 tracking module for pose estimation relative to an arbitrary, fixed coordinate system. For this work, this solution was sufficient because we could better control a patient's motion (ex. asking them to stay as still as possible throughout each scan), and all scans were performed at the same time and location. As discussed in section 1.2.3, tracking the ultrasound probe's pose is only a partial solution.

In clinical settings, it may be necessary to include relocalization capabilities to accommodate environmental changes and patient motion variability. In other words, the tracking module needs to reliably re-register a coordinate system *with respect to the patient's tissue* between scanning sessions. A realistic scenario would be that a patient receives a series of scans on one day, returns to the clinic in a few days, and is scanned again in a different room. Without relocalization, a sonographer may scan a slightly different volume than the original scan.

Work done in [1] attempted to relocalize a tracking module using superficial veins. Future work may focus on adding this capability and making it compatible with the ROS network.

Chapter 5

Module Integration

5.1 ROS Architecture

To manage the processing demands of the multiple attachments and allow them to function independently, the Robot Operating System (ROS) was used for communication. The ROS architecture for this system consists of several sets of computational processors, or nodes. Specific pieces of information, called topics, are communicated between the nodes. Nodes can update (or publish to) topics by sending a message to them, or receive (or subscribe to) topics by receiving the message they contain. Nodes can carry on their individual processes and update their processes as new information is added to the system's topics.

A schematic of the system's ROS architecture is shown in Figure 5-1. All nodes are commanded at a high level by the master node. The master node is programmed in a custom MATLAB application using the ROS Toolbox. The force measurement and force control attachments are controlled by the Arduino Nano node. The thumper attachments are controlled by the Teensy node. Both nodes are programmed using the "rosserial" Arduino library and interact with the master node via USB serial ports. The camera attachment is processed using the ROS wrapper developed by Intel RealSense. The wrapper uses the Intel RealSense SDK to communicate with the camera, then processes the readings from the camera's sensors into ROS topics.

ROS was developed primarily for Linux-based operating systems. The probe at-

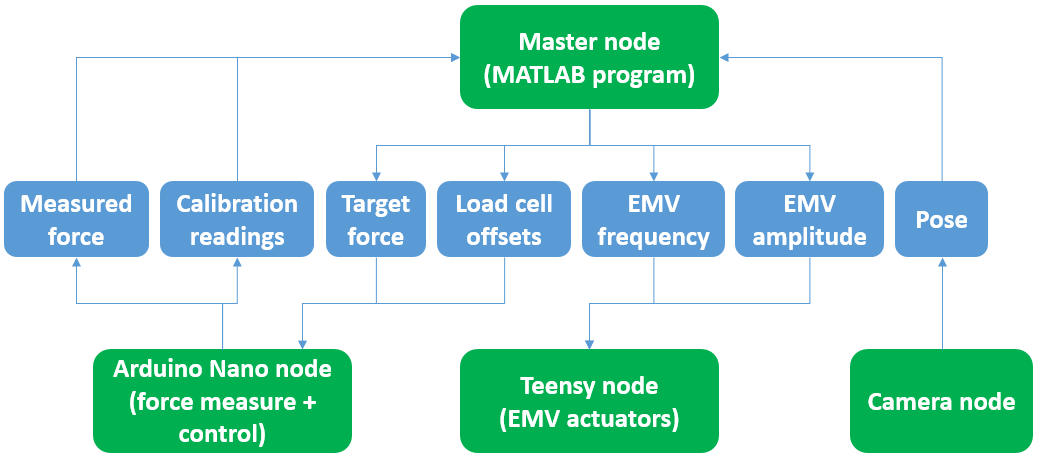


Figure 5-1: Schematic of the system's ROS architecture. The system contains four nodes, including a master node programmed in MATLAB.

tachment system was run using Ubuntu 18.04 and ROS Melodic. It should be noted that Windows 10 has added Linux capabilities using Windows Subsystem for Linux (WSL), and the "rosserial" nodes ran successfully in the WSL architecture. The Intel RealSense node, however, did not.

5.2 Power Management

The system is powered by a single power supply, but individual components may require lower voltages to function accurately and safely. The power supply is rated for 12VDC voltage and up to 5A current. Several components, such as the linear motor controller and the load cell amplifier, operate at the 12V supply voltage. Other components, such as the microcontrollers and voice coil actuators, require 5V or lower.

Figure 5-2 shows a schematic of the power supplied to each electrical component in the system. A DC-DC converter is used to scale down the 12V supply voltage to 5V to power the thumper attachment. The Arduino Nano was powered directly from the 12V supply and receives a steady 5V operating voltage from its internal linear regulator. A regulated 5V operating voltage was important to ensure accurate load cell readings at the ADC.

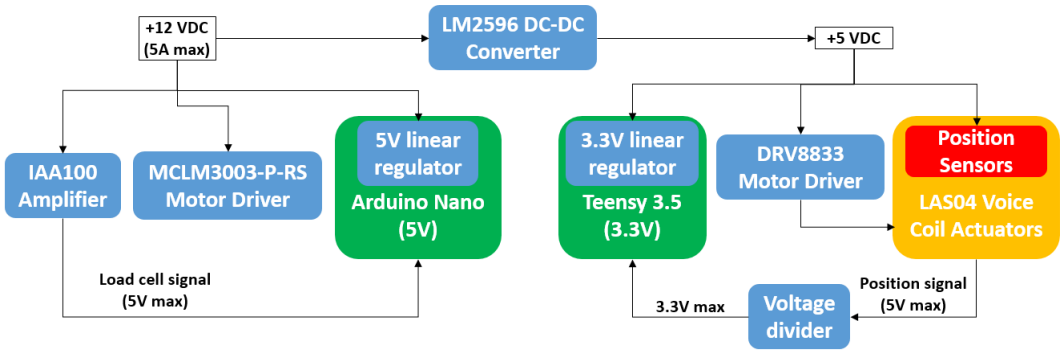


Figure 5-2: Schematic of the system's power management. Power is supplied from a 12VDC, 5A supply and regulated down with various converters.

5.3 Future Work

This thesis focused on integrating the three modules into a single, centralized framework using the Robot Operating System (ROS). By integrating ROS with a custom MATLAB application, the system achieves modularity necessary for a variety of clinical applications. All modules can be enabled, observed, and adjusted in a single location depending on imaging needs. Hardware for all modules was scaled down to the size of a medium-sized breadboard, driven from a single power supply for portability and ease of use, and controlled with inexpensive and open-source microcontrollers.

Future work on this system may involve further software and hardware refinement in preparation for broader clinical deployment. For example, the ROS environment currently requires specific tools found on Ubuntu, but with more work may be easily deployed on other operating systems. Hardware could be consolidated onto a custom PCB for more robust connections and cable management.

Chapter 6

Experimental Validation

This chapter overviews experiments conducted to validate that the hardware developed in previous chapters functions correctly and usefully during ultrasound scans. The results also present new capabilities, such as force-coupled elastogram volumes, that were achieved because modules were used simultaneously.

6.1 Validation with Calibrated Phantom

A series of tests were conducted in which B-mode and elastography images were collected on a CIRS phantom. The experiments allowed for evaluation of the force coupling and camera attachments independently and in combination. The tests were:

1. *Control test*: The probe was equipped with the force measurement and camera attachments. No feedback was provided to the operator.

2. *Full feedback test*: The control test was repeated, but contact force and pose feedback was provided on-screen.

3. *Force control test*: The force measurement attachment was replaced with the force control attachment. Visual feedback was provided, and a force target of 4N was commanded.

Each test consisted of three different sub-tests:

i. *Constant force, constant pose*: The operator finds a location in the phantom which produces clear images. They then attempt to keep the contact force and probe's pose (i.e. position and orientation) as constant as possible.

ii. *Force sweep, constant pose*: The operator keeps the probe's pose constant, but varies the contact force in a "sawtooth" shape. That is, the force is slowly increased, held briefly at a maximum level, then slowly decreased.

iii. *Constant force, varied pose*: The operator keeps the contact force constant, but moves the probe around the phantom to capture many different features.

The EMV thumper attachments were not tested at this time due to hardware limitations. An ultrasound machine was not available which was capable of disabling the acoustic radiation force push used in traditional elastography. Additionally, the system was not set up with an ultrasound imaging system that was capable of ultrafast imaging *and* was calibrated for human use.

6.1.1 Imaging Repeatability and Phantom Properties

Constant Force and Pose Tests

The data from tests 1i, 2i, and 3i were analyzed to explore two main questions related to operator behavior and image quality:

- How do visual feedback and active control affect the precision of contact force and pose during an ultrasound scan?
- How do visual feedback and active control affect image repeatability?

In other words, does visual feedback provide an advantage over having no feedback with respect to the precision of the probe's contact force and position? Does active control provide an advantage over visual feedback in the same way? And if so, do feedback or active control help reduce variability between images?

First, the measured contact force was analyzed directly to compare levels of fluctuation between the three tests. Figure 6-1 shows the time-series force data from

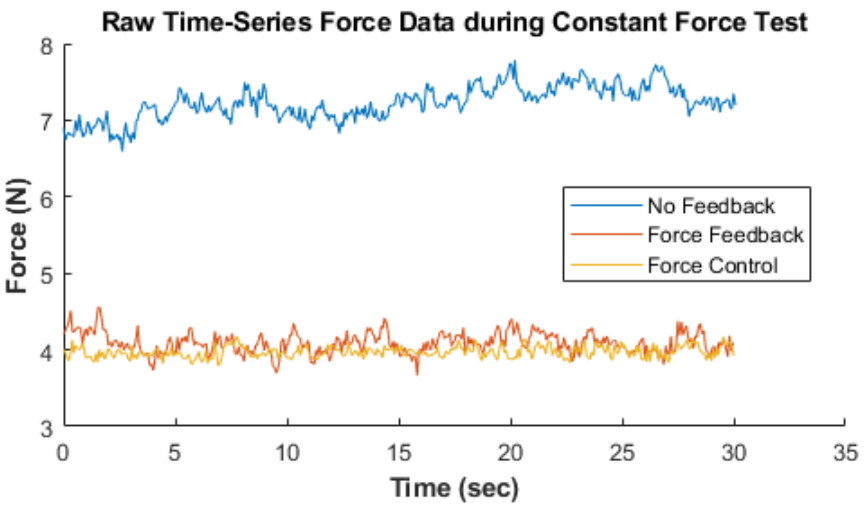


Figure 6-1: Time-series force data from constant force tests.

each of the three tests, beginning at the point when the probe stabilizes after initial contact. Visually, the fluctuations in contact force decrease noticeably going from the no-feedback test to the force-control test.

To examine this observation quantitatively, the standard deviation and its confidence interval were calculated for each time-series data set. The upper and lower bounds of the confidence interval for a standard deviation estimate follow a Chi-square distribution such that:

$$CI_\sigma = \left[\sqrt{\frac{(N-1)S^2}{\chi_{\frac{\alpha}{2},N-1}}}, \sqrt{\frac{(N-1)S^2}{\chi_{1-\frac{\alpha}{2},N-1}}} \right] \tag{6.1}$$

where N is the number of samples, S is the standard deviation of the samples, $\alpha = 0.05$ is the confidence level, and $\chi_{p,n}$ is the value corresponding to the p percentile of a Chi-square distribution with n degrees of freedom.

A comparison of the standard deviation estimates is shown in Figure 6-2. As predicted, the test with no visual feedback resulted in the highest force variability, while the test with active force control resulted in the lowest variability. Because confidence intervals do not overlap, we conclude that the differences between the three tests are statistically significant.

Second, the pose data were analyzed in a similar manner. Figure 6-3 shows the raw data for the "roll" angle of the probe, the angle most likely to cause variation between

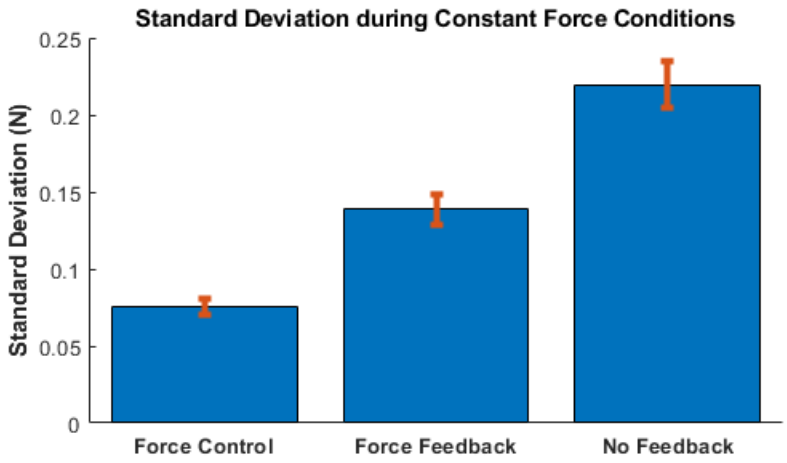


Figure 6-2: Standard deviation (with 95% confidence intervals) in contact force between the three constant force tests.

images. Confidence intervals for the standard deviation of the roll angles in each test are shown in Figure 6-4. There is a statistically significant decrease in the roll angle's variability when feedback is provided. Interestingly, active force control resulted in an *increase* in variability. This increase may be due to vibration disturbances from the linear motor. More tests, perhaps across multiple different users, would need to be conducted to determine if this trend is repeatable.

Finally, the ultrasound images were analyzed without considering the pose and force data. To quantify variability between images, 2D cross correlation (MATLAB function *xcorr2()*) was performed between each image and the first image in the data set. The translation distance (d, measured in pixels) between each set of images was calculated using the combination of horizontal and vertical translation with the highest cross correlation value. Confidence intervals for the standard deviation of d in each test are shown in Figure 6-5. Interestingly, the results indicate that there was no statistical difference in image variability (as measured by the standard deviation of d) between the test with no feedback and the test with feedback. The variability in the force control test was statistically higher than the other two tests. One possible reason is that streaks were present in some images in the force control test, possibly due to electromagnetic interference between the ultrasound probe and the linear motor. However, more tests with multiple users would be needed to validate this trend.

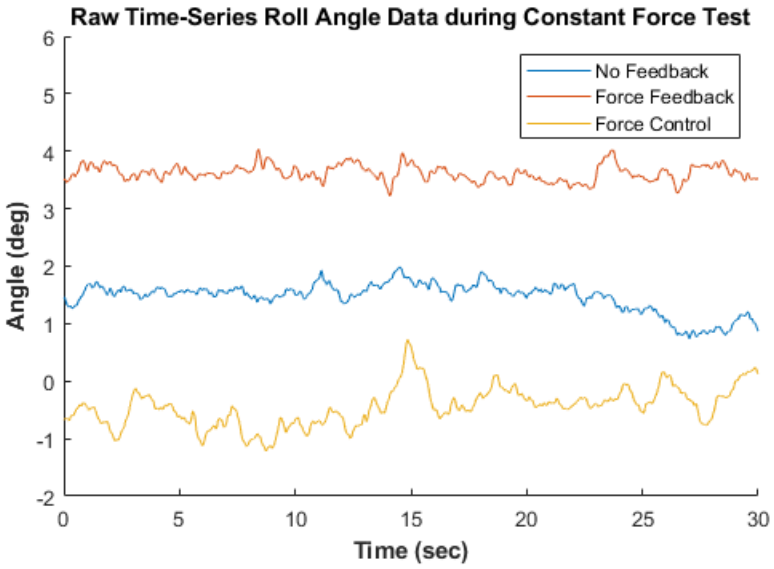


Figure 6-3: Time-series roll angle data from constant force tests.

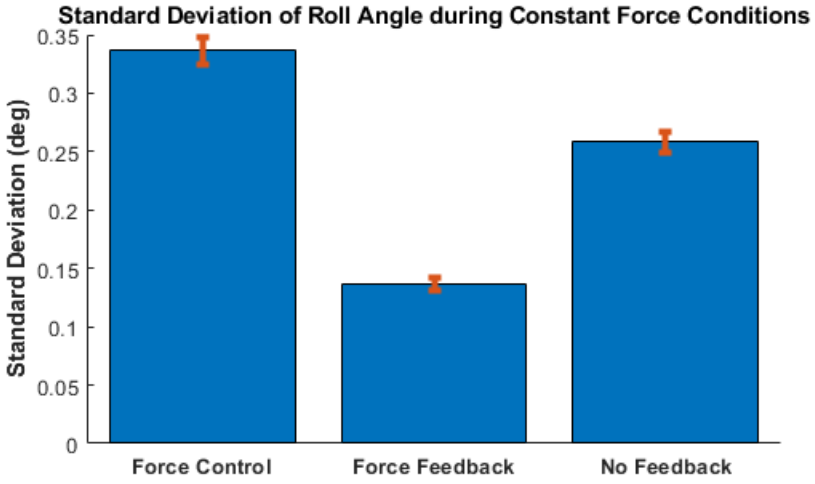


Figure 6-4: Standard deviation (with 95% confidence intervals) in roll angle between the three constant force tests.

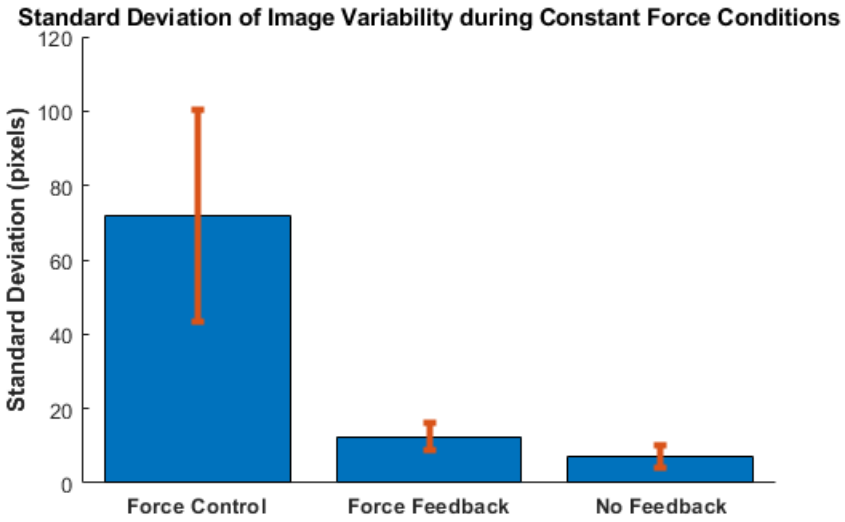


Figure 6-5: Standard deviation (with 95% confidence intervals) in cross correlation distance between the three constant force tests.

Force Sweep Tests

The data from tests 1ii, 2ii, and 3ii were analyzed to explore two main questions related to operator behavior and phantom properties:

- How do visual feedback and active control affect how precisely the operator controls their force sweep?
- How do phantom material properties change as contact force is varied? (i.e. Is phantom material nonlinear?)

First, the measured force data were analyzed to compare how closely they follow the target force sweep pattern. Figure 6-6 shows the raw data from each force sweep test. Visually, the sawtooth pattern becomes more defined with less variation going from no feedback to active force control. For better comparison, one complete force sweep was selected from each test and normalized in time and force. Figure 6-7 shows the normalized force sweep for each test. To quantitatively compare the goodness of fit to a sawtooth pattern, MATLAB's Curve Fitting Toolbox was used to fit the following function to each data set from Figure 6-6:

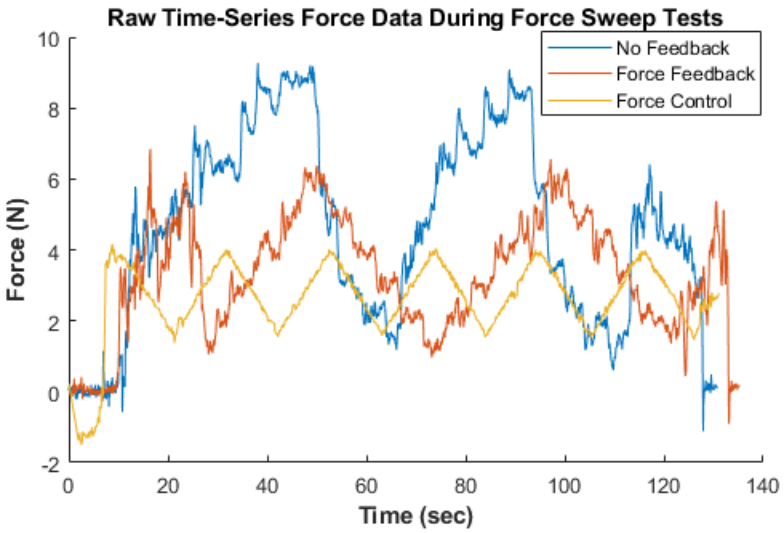
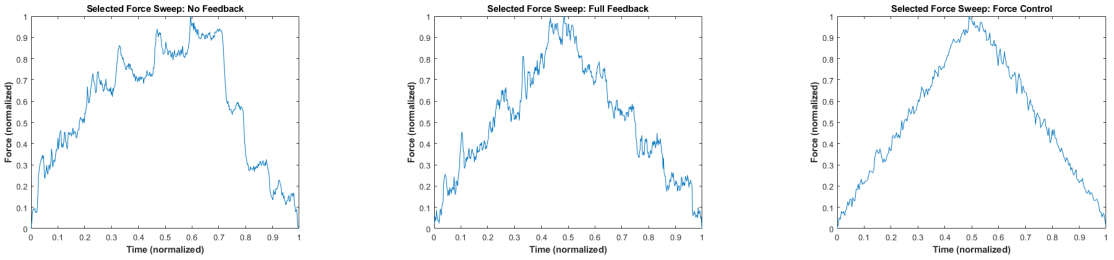


Figure 6-6: Raw Times-Series Force Data from Force Sweep Tests.



(a) No feedback (b) Full visual feedback (c) Active force control

Figure 6-7: Representative sweep from each of the three force sweep tests.

$$F(t) = A\,[sawtooth(2\pi t, 0.5)] + B \tag{6.2}$$

where A and B are fit parameters, $F(t)$ is the normalized force (between 0 to 1), t is normalized time (from 0 to 1), and $sawtooth()$ is MATLAB's built-in sawtooth function. Note that for a perfect sawtooth pattern for this data form would correspond to $A = 0.5$ and $B = 0.5$. Table 6.1 summarizes the curve fitting results. The correlation coefficient R increases towards 1 going from no feedback to force control, indicating that force coupling helps with controlling the force sweep.

Second, the ultrasound images were analyzed to determine how the shear wave speed changes when various forces are applied to the phantom. The RGB pixels of the images were converted to shear wave speed (SWS) using the color scale provided

	A	B	R
No feedback	0.42 ± 0.02	0.57 ± 0.01	0.906
Full visual feedback	0.433 ± 0.009	0.506 ± 0.005	0.971
Force control	0.480 ± 0.006	0.515 ± 0.003	0.996

Table 6.1: Curve fitting results for force sweep tests on phantom. One force sweep from each test was normalized in time and force, then fit to the function $F(t) = A\,[sawtooth(2\pi t, 0.5)] + B$, where $sawtooth()$ is the built-in MATLAB function.

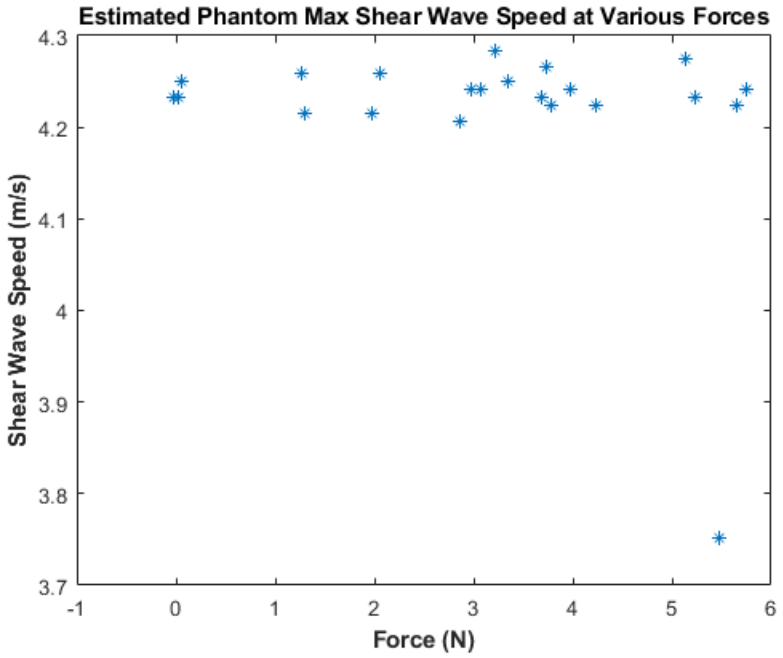


Figure 6-8: Shear wave speed (SWS) in CIRS phantom as a function of contact force.

from the raw elastography images. For each image, the maximum SWS and the corresponding measured force at the time of the image were calculated. Figure 6-8 shows a comparison of SWS as a function of contact force. These data suggest that phantom properties (namely, SWS or elastic modulus) do not vary significantly with contact force. Tests on human or animal tissue would be needed to observe tissue nonlinearity with a combination of force coupling and shear wave elastography.

6.1.2 B-Mode Volume Reconstruction

Test (iii) was used to validate the optical tracking module in a different way. While test (i) collected 2-D ultrasound images under constant pose conditions, test (iii)

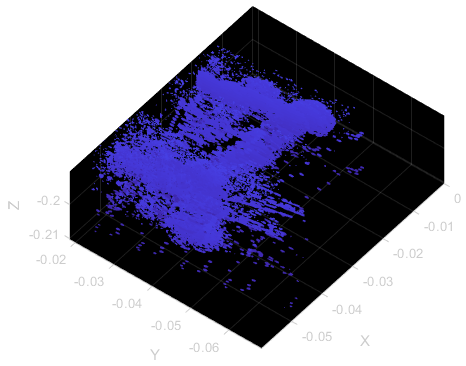


Figure 6-9: Reconstructed ultrasound volume (represented as a point cloud) using 2-D segmented ultrasound images.

collects 2-D images at *different* poses. A combination of the module's poses and the ultrasound images should yield reasonably accurate 3-D reconstructed ultrasound volumes.

Volumes were constructed using code developed by Benjamin [1]. The code was modified slightly to accept the data format of the system detailed in this thesis. The code combines the poses and ultrasound images to create a point cloud, where each point represents a pixel from a segmented 2-D image. Images were segmented in MATLAB using a simple thresholding technique. If the gray-scale pixel was at least 45 (out of 256 possible values), it was considered part of a "feature of interest" and included in the segmented image.

Figure 6-9 shows the result of this reconstruction from test 2iii. The calibrated phantom consists of 3 large diameter, cylindrical reflectors along with a series of small diameter reflectors. Qualitatively, the reconstructed volume closely resembles the predicted shape from the phantom's calibration. Two of the 3 cylindrical reflectors can be clearly identified. The third reflector may have had low reflectivity and was not detected with the simple segmentation method. The volume has notable image noise due to speckling from the segmentation. In the future, a different segmentation technique could be used to reduce this noise. Improved experimental conditions, such as imaging while using more ultrasound gel, could also improve the segmentation technique's ability to register features.

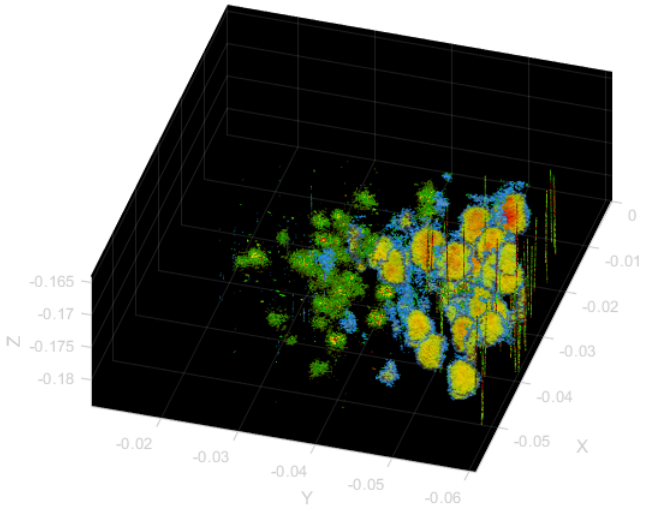


Figure 6-10: Reconstructed ultrasound volume (represented as a colored point cloud) using 2-D elastograms.

6.1.3 Elastogram Volume Reconstruction with Force Coupling

While volume reconstruction from 2-D B-mode images has been done and validated in previous work, the combination of force coupling and optical tracking can enable other types of ultrasound volumes which are more quantitative. Specifically, it was hypothesized that the addition of force coupling to volume reconstruction enables elastogram volumes.

Figure 6-10 an example volume constructed from 2-D elastograms of the CIRS phantom. The volume reconstruction algorithm was modified slightly so the brightness of each point followed a colored scale instead of being binary. That is, the color of each pixel in the 2D elastogram was transformed into a corresponding color in the point cloud.

Note that this volume was constructed under constant force conditions from test 2iii. The CIRS phantom was not an appropriate medium to explore tissue stiffness nonlinearity because analysis done in section 6.1.1 showed that the stiffness of phan-

tom features does not vary with load.

6.2 Validation in Human Studies

Tests with the probe attachments were also performed in preliminary human studies to investigate the following questions:

- Can tissue nonlinearity be predicted using feedback from the attachments' sensors?
- Can force-coupled elastogram volumes be reasonably created using the force-coupling and optical tracking attachments?

A series of tests similar to (and expanding on) the tests performed in section 6.1 were also performed in human subjects. The tests were:

1. *Control test*: The probe was equipped with the force measurement and camera attachments. No feedback was provided to the operator.
2. *Force feedback test*: The control test was repeated, but contact force was provided on-screen.
3. *Full feedback test*: The control test was repeated, but contact force textitand pose feedback was provided on-screen.
4. *EMV module test*: The ultrasound machine was switched to RF data collection mode. The EMV module was activated and the control test was repeated.
5. *All attachments test*: The force control attachment was added to the device. The previous test was repeated, with a force target of 4N for the force control attachment.
6. *Force control test*: The EMV module was disabled and the ultrasound machine was switched back into elastography mode. The previous test was repeated.

As in the phantom experiments from section 6.1, each test consisted of three different scanning exams:

i. *Constant force, constant pose*: The operator finds a location in the phantom which produces clear images. They then attempt to keep the contact force and probe's pose (i.e. position and orientation) as constant as possible.

ii. *Force sweep, constant pose*: The operator keeps the probe's pose constant, but varies the contact force in a "sawtooth" shape. That is, the force is slowly increased, held briefly at a maximum level, then slowly decreased.

iii. *Constant force, varied pose*: The operator keeps the contact force constant, but moves the probe around the phantom to capture many different features.

The data from these tests were then analyzed to experimentally validate the force coupling and optical tracking attachments. Data were not collected with the EMV attachments due to time constraints.

Human studies were conducted in compliance with MIT COUHES Institutional Review Board approved protocol 2012000284. Ultrasound images were collected on the left thyroid of each participant.

6.2.1 Tissue properties

The results from test 3ii were used to examine the nonlinear stiffness characteristics of thyroid tissue. For this analysis, the data from one participant were used to eliminate inter-subject variability.

For each elastogram, the color maps were converted to shear wave speed (SWS) using the color scale provided on the elastogram images. The maximum SWS in each image was recorded. The corresponding force measured by the load cell was recorded. The data are summarized in figure 6-11.

A positive correlation between contact force and SWS was visually apparent, but a linear regression was performed to quantitatively assess the correlation. A bisquare robust linear fit of the form $SWS = a + bF$ was performed, where a and b are fit

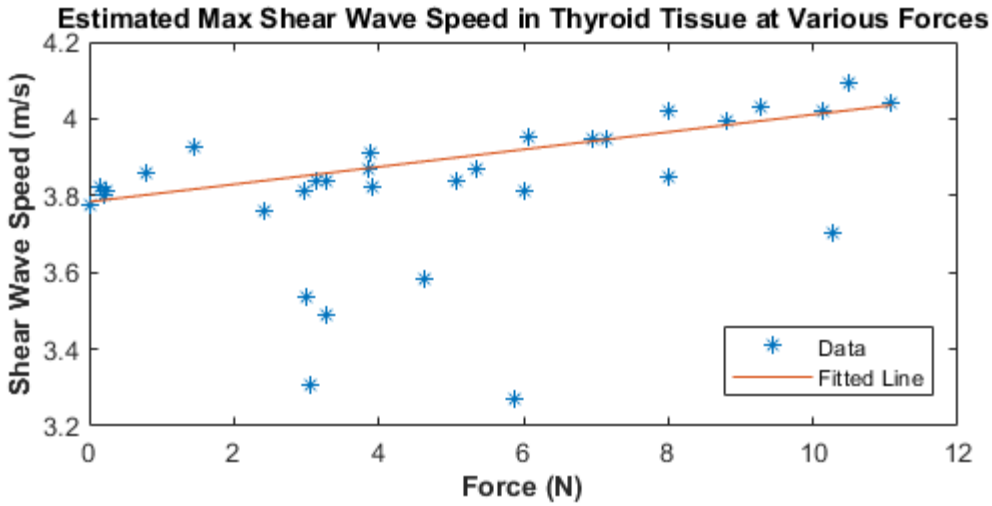


Figure 6-11: Shear wave speed (SWS) in one human participant's thyroid tissue as a function of contact force. The fitted line is of the form $y = a + bx$, where $a = 0.023 \pm 0.009(m/s)/N$ and $b = 3.78 \pm 0.06m/s$ (with 95% confidence).

parameters. A bisquare fit was used because the data showed significant outliers (such as the points below 3.6 m/s) which may affect a traditional least squares regression. The regression yields values of $a = 0.023 \pm 0.009(m/s)/N$ and $b = 3.78 \pm 0.06m/s$ (with 95% confidence). The positive correlation is therefore statistically reasonable and there is evidence to suggest that tissue nonlinear stiffness can be predicted with force coupling.

6.2.2 Force-Coupled Volume Elastogram of Thyroid

As in section 6.1.3, the 2-D elastogram images collected from human participants' thyroids were stitched together to create elastogram volumes. This volume reconstruction serves to validate the simultaneous use of the optical tracking and force-coupling attachments. Alone, the optical tracking enables volume reconstruction of either B-mode or elastogram volumes. Without force measurement or control, however, the quantitative information provided from elastograms are more arbitrary. Therefore, elastogram volumes are a unique possibility for the combination of these two attachments.

Figure 6-12 shows the reconstructed volume elastogram of one human participant.

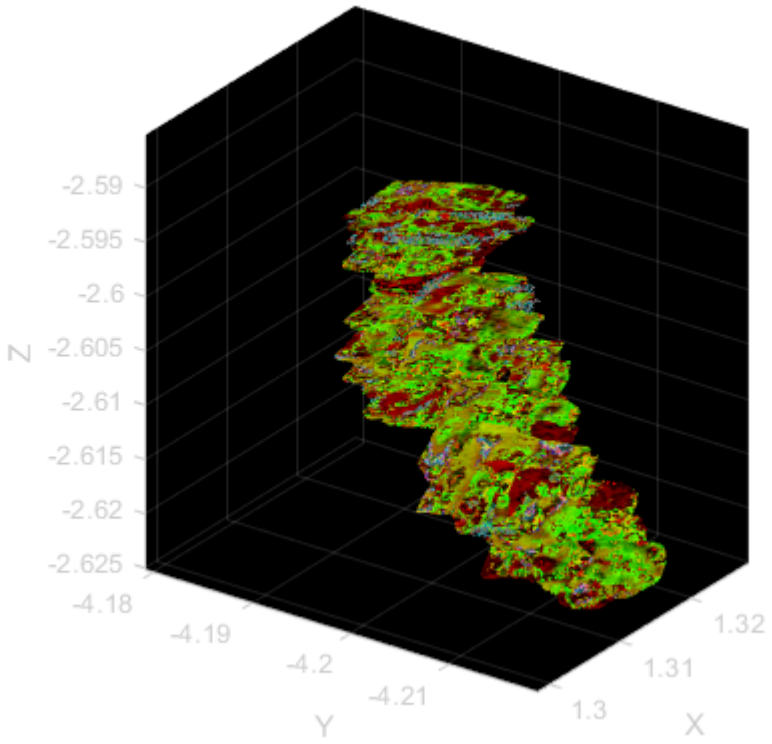


Figure 6-12: Reconstructed elastogram volume of the left thyroid from one human participant. Elastograms were collected at approximately 4N. Regions of red correspond to higher stiffness.

Images were manually segmented in MATLAB. 2-D images were then transformed to the 3D, fixed world coordinate system established by the T265 camera and a point cloud was formed. The origin is therefore arbitrary, but relative position of the points in the point cloud should be accurate.

The shape of the point cloud is consistent with the typical anatomy of one lobe of the thyroid. Furthermore, the size/volume is also consistent, measuring a few centimeters in each direction.

The interesting addition of color from the elastograms provides a new reference for assessing features in a volume. Whereas volume ultrasonography in previous work constructed volumes based on binary segmentation (i.e. points are marked with either a "1" if they are important or a "0" if they are not), volume elastograms allow examination of features *within* a region of interest. The entire volume in figure 6-

12 was segmented to include only the thyroid tissue (instead of muscle, arteries, or trachea, for example). Within the volume, however, one can identify regions of higher stiffness that may be worth investigating further.

6.3 Conclusion

Overall, these results from preliminary human studies have demonstrated some of the capabilities of the force-coupling and optical tracking attachments. Section 6.1 examined operator use of the attachments quantitatively to determine whether variability of contact force and pose could be reduced. Properties of the CIRS phantom were also examined and it was determined that the material did not display enough non-linearity in stiffness to usefully evaluate force-coupling attachments. Preliminary volume reconstruction was also demonstrated in the phantom.

Section 6.2 examined tissue stiffness non-linearity and volume reconstruction in-vivo. Section 6.2.1 showed that force-coupling can reasonably predict a slight tissue stiffness non-linearity. Section 6.2.2 showed a reconstructed volume elastogram as proof-of-concept for the optical tracking and force-coupling attachments.

In thyroid imaging, the volume of the organ can be diagnostically relevant [2]. Volume elastograms would still allow for calculation of organ volume, though this particular indicator was not explored in this work. Future work may focus on enabling this calculation.

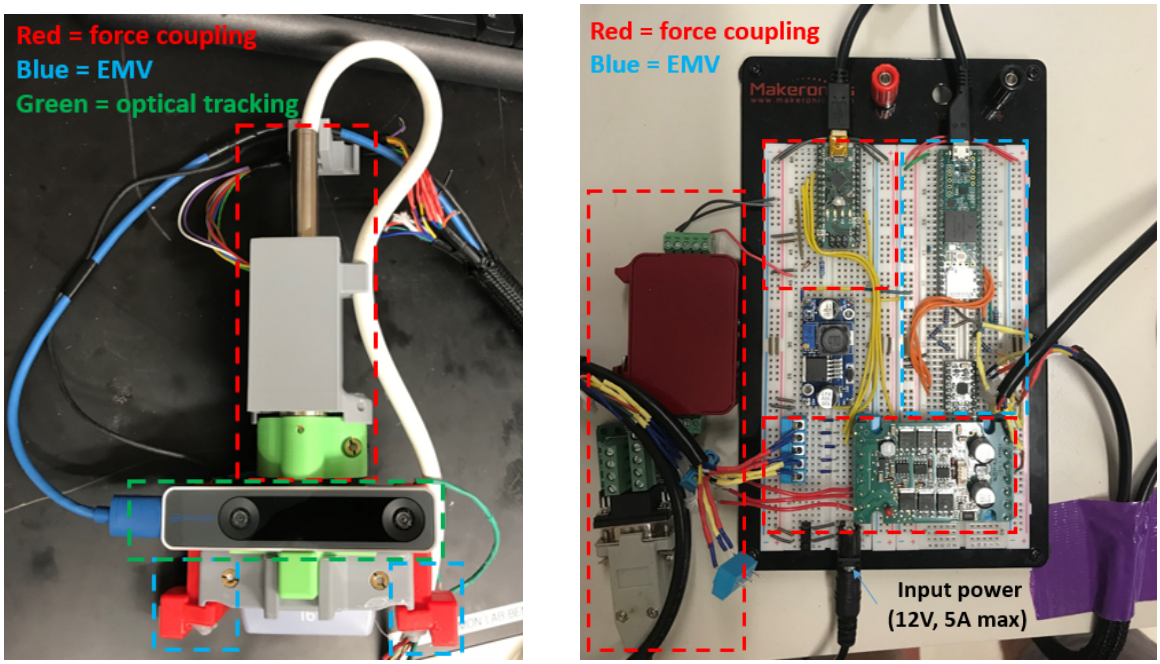
Chapter 7

Conclusions

In this work, the design and validation of several electromechanical attachments for a medical ultrasound probe were detailed. The force coupling attachments, inspired by work from Gilbertson, included a load cell for measuring contact force and a motorized attachment to achieve real-time, active force control. The EMV ("thumper") attachments developed work by Chavez and redesigned the hardware to be more debris-resistant. The voice coil actuator dynamic performance was also characterized. The optical tracking attachment developed work by Benjamin and implemented an easier-to-use, smaller camera module. The repeatability of the camera and the mounting setup were quantified. Figure 7-1 shows a picture of the physical attachments and the breadboarded electronics.

One major advancement in this work was the consolidation and synchronization of electronics for easier data acquisition and clinical deployment. All components (except for the load cell amplifier) fit onto two standard, interlocked breadboards as shown in Figure 7-1b. The microcontrollers are easily connected to a laptop running ROS and collect data through a custom MATLAB application. The ROS network ensures that all readings from the three attachments are synchronized to the same clock.

A novel research outcome from this work was the creation of ultrasound volumes using force-coupled elastograms. While force-coupled ultrasound and volume ultrasound were independently possible in previous work [4, 1], the attachments developed



(a) Attachments with GE 9L ultrasound probe

(b) Consolidated breadboard with all electronic components

Figure 7-1: Pictures of probe attachments system

in this work allow the technologies to be combined. Elastogram volumes are technically possible with only the camera attachment, but force coupling may be necessary in scanning scenarios (such as thyroid imaging) where tissue nonlinearity is significant.

The EMV modules could not be tested extensively at this time due to limitations in ultrasound imaging hardware. At the time of writing, there was not a functioning ultrasound system which could disable the ARF push in elastography mode *and* was calibrated to within FDA limits. In theory, if this data could be collected, the EMV attachments would enable faster and deeper elastography imaging. The volume elastograms could therefore be further augmented with improved spatial sampling. RF data sets were collected during tests in the human studies which used the EMV attachments. Future work may explore whether the raw RF data can be processed into elastograms in response to the external vibrations.

There are several opportunities for improving individual attachments and the overall system to achieve a clinically relevant design. The current design of custom components (such as for the ultrasound probe clamp, EMV cases, and force control parts)

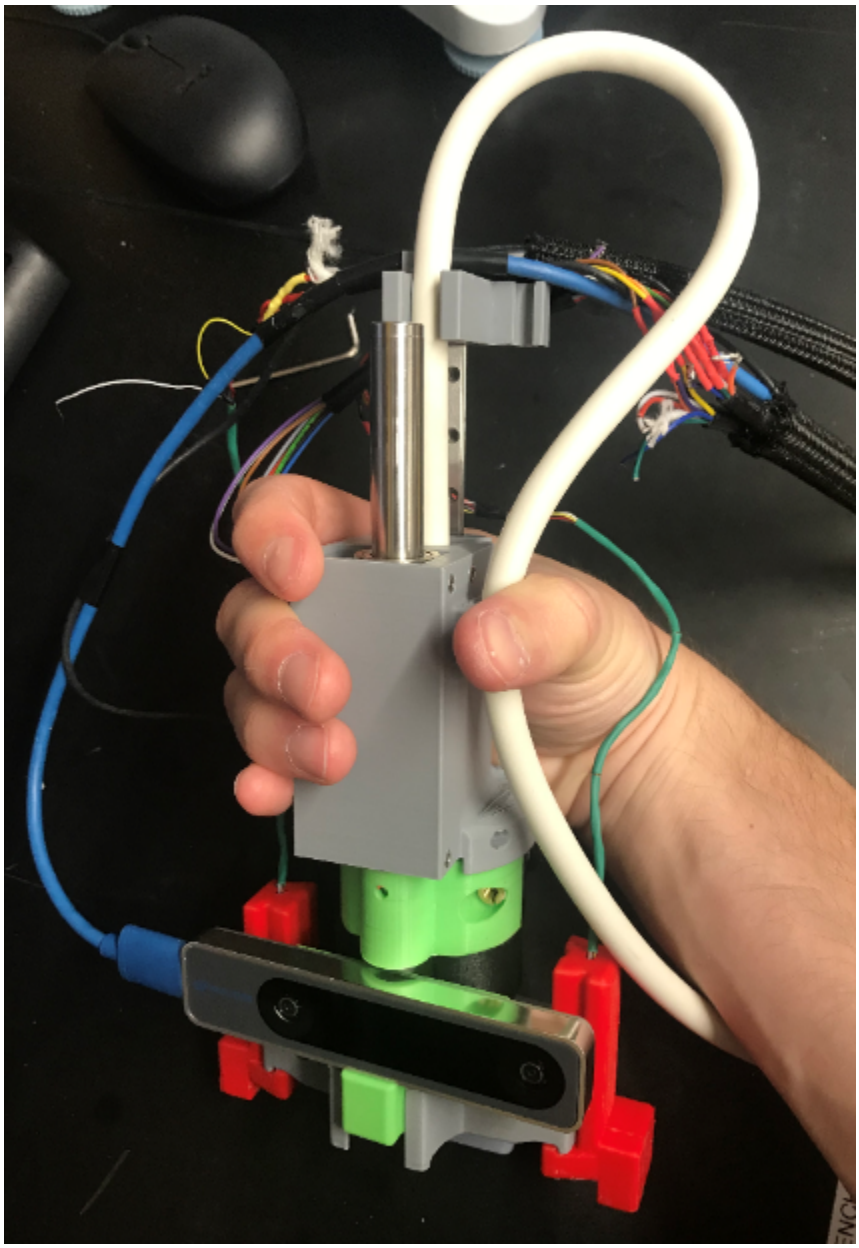


Figure 7-2: All attachments placed on the probe simultaneously and held by the user.

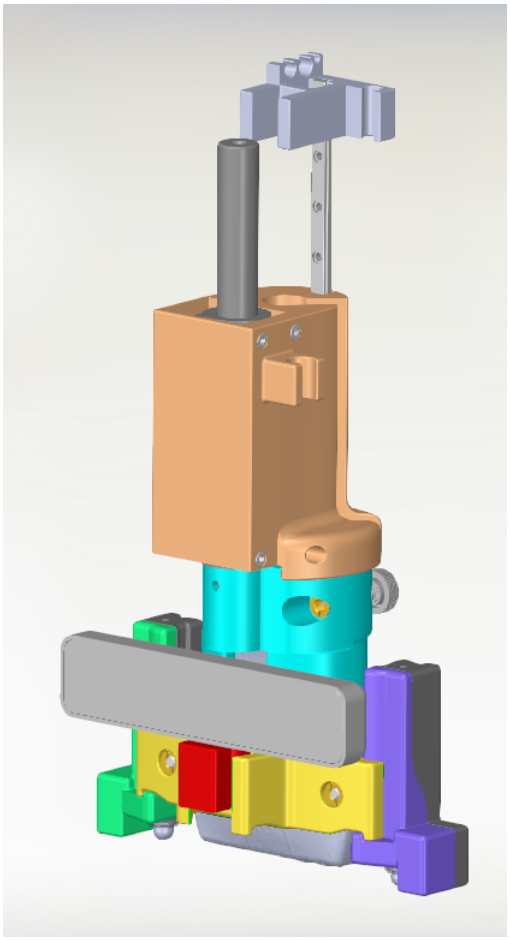


Figure 7-3: CAD assembly of all ultrasound probe attachments with GE 9L probe.

were made as compact as possible while providing flexibility for changes as research progressed. The design may be further miniaturized and made more ergonomic.

Cable management was also a major challenge with this system. First, electromagnetic interference necessitated having power and signal lines housed in different cables. Second, the attachments each needed separated cables so that any combination could be used without requiring unneeded attachments on the probe. Future work may focus on optimizing the cabling to reduce cross-talk while reducing the need for multiple large cables.

Appendix A

Code

A.1 Force Coupling Arduino Code

```
// Script to use Faulhaber LM2070-040-01 motor for force control
//
// Note: Requires rosserial library ***V0.7.9*** or earlier.
// You cannot use V0.9.1 on Arduino Nano as of now.
//
// By: Ryan Koeppen
// Last updated: 3/2/2021

///////////////// Initial Setup ////////////////////////

// Include ROS libraries
#include <ros.h>
#include <std_msgs/Bool.h>
#include <std_msgs/Float32.h>

// User-modified variables
double force_setting = 8.0;     // in N
double defaultPosition = 20.0;  // in mm

// States
boolean calibrating = false;
boolean forceMeasureActive = false;
boolean forceControlActive = false;
boolean positionControlActive = false;

// Motor constants
const double motorStroke = 49.0;
```

```
const double Vmax = 12.0;     // supply voltage to motor (V)
const double posMax = 40.0;   // Motor top limit (mm)


// Set up moving average filters
#include <movingAvg.h>          // https://github.com/JChristensen/movingAvg
movingAvg velocityReads(7);

//// Set up median filters
//#include "MedianFilterLib2.h"
//MedianFilter2<float> medianFilter_force(7);
//MedianFilter2<int> medianFilter_hallA(7);
//MedianFilter2<int> medianFilter_hallB(7);
//MedianFilter2<int> medianFilter_hallC(7);

// Define pins
const int hallA_pin = A0;
const int hallB_pin = A1;
const int hallC_pin = A2;
const int PWMpin = 3;
const int forcePin = A4;

// Initialize global variables

double pos_offset = 0.0;    // For the homing sequence; from tests, ~9.5
double force_offset = 0.0;

double pos = 0.0;
int t = 0;
double velocity = 0.0;
double force = 0.0;

double savedPosition = 0.0;

double pos_last = 0.0;
double pos_last2 = 0.0;
int time_last = 0;
double force_last = 0.0;

double posError_last = 0.0;
double speedErrorWindup = 0.0;
double forceErrorWindup = 0.0;

int tROS;
```

```
//////// ROS Stuff ////////

// Set up ROS node
ros::NodeHandle  nh;

// Set up ROS messages for publishers
std_msgs::Float32 force_measure_msg;
std_msgs::Float32 force_calibrate_msg;

// Functions for subscribers
void forceControlCb( const std_msgs::Bool& msg ){

  if (msg.data) {
    // Perform homing at 35 mm/s
    homingSequence(35.0, defaultPosition);
    savedPosition = defaultPosition;

    forceControlActive = false;
    positionControlActive = true;
  } else {
    forceControlActive = false;
    positionControlActive = false;
  }

}

void forceMeasureCb( const std_msgs::Bool& msg ){
  forceMeasureActive = msg.data;
}

void calibrateCb( const std_msgs::Bool& msg){
  calibrating = msg.data;
}

void forceSettingCb( const std_msgs::Float32& msg ){
  force_setting = msg.data;
}

void forceOffsetCb( const std_msgs::Float32& msg ){
  force_offset = msg.data;
}

// Set up ROS publishers and subscribers
ros::Publisher forceCalibrate("force/calibration", &force_calibrate_msg);
```

```
ros::Publisher forceMeas("force/measured", &force_measure_msg);

ros::Subscriber<std_msgs::Bool> state_forceMeas("force/state/forceMeas", &forceMeasureCb);
ros::Subscriber<std_msgs::Bool> state_forceControl("force/state/forceControl", &forceControlCb);
ros::Subscriber<std_msgs::Bool> calibrateSub("force/state/calibrate", &calibrateCb);
ros::Subscriber<std_msgs::Float32> forceSet("force/setpoint", &forceSettingCb);
ros::Subscriber<std_msgs::Float32> forceOffset("force/offset", &forceOffsetCb);


////////////////////// Main Script ///////////////////////////////


void setup() {
  // Set PWM pin for voltage commanding, set to 0V nominally
  pinMode(PWMpin, OUTPUT);
  delay(1);
  analogWrite(PWMpin, 127);

  // Start moving average objects
  velocityReads.begin();

  // Set PWM frequency on pins D3 and D11 to 31372.55 Hz
  // Faulhaber controller recommends minimum 20 kHz for PWM
  TCCR2B = TCCR2B & B11111000 | B00000001;

  // Start ROS nodes
  nh.getHardware()->setBaud(57600);
  nh.initNode();
  nh.advertise(forceCalibrate);
  nh.advertise(forceMeas);
  nh.subscribe(state_forceMeas);
  nh.subscribe(state_forceControl);
  nh.subscribe(calibrateSub);
  nh.subscribe(forceSet);
  nh.subscribe(forceOffset);
  tROS = millis();

  // Get initial position estimate
  pos_last = readAngle() * (24.0 / (2.0 * PI));
  pos_last2 = pos_last;
  time_last = millis();

}

void loop() {
```

```
// Read position and velocity, apply offsets as needed
readMotorState();


// Perform these steps only if using the force control
// module.

if (forceControlActive || positionControlActive) {

  // Check whether device is out of range.
  // Update state accordingly.
  if ((pos > posMax) && forceControlActive) {
    positionControlActive = true;
    forceControlActive = false;
    savedPosition = posMax;
  }

  // If position controlling, check whether force is
  // high enough to re-engage force control.
  if (positionControlActive) {
    if (force > force_setting + 0.3) {
      positionControlActive = false;
      forceControlActive = true;
    }
  }

}


// Do something, depending on the system's state


if (calibrating) {

  // Calibrate load cell
  calibrateLoadCell();

  // Calibrate position sensor.
  // If force control isn't attached, this step
  // will be nonsense but it won't impact force
  // measurement.
  positionCalibrate();

  // Wrap up the calibration sequence
```

```
    calibrating = false;

  }

  else if (forceMeasureActive) {
    analogWrite(PWMpin, 127);
  }

  else if (forceControlActive) {
    forceControl(force_setting);
  }

  else if (positionControlActive) {
    positionControl(savedPosition);
  }

  else {
    analogWrite(PWMpin, 127);
  }



  // Update variables for next loop
  reassignNextLoop();

  // Publish data to ROS, receive messages
  processROS();

  delay(1);

}


////////////////////////////// Custom functions ////////////////////////////////


void processROS() {

  if ((t - tROS) > 75) {
    // Receive messages about states and pose


    // Publish force data
    force_measure_msg.data = force;
    forceMeas.publish( &force_measure_msg );
```

```
    // Finish up
    nh.spinOnce();
    tROS = t;
  }

}


void positionCalibrate() {

  // Turn motor off if it isn't already
  analogWrite(PWMpin, 127);
  delay(1000);

  // In case hall sensors haven't been
  // read for a while...
  for (int i=0; i<50; i++) {
    readAngle();
    delay(1);
  }

  // Apply offset based on measured stroke length of motor
  // and reset position
  pos_offset = readAngle() * 24.0 / (2.0 * PI);
  pos_last2 = motorStroke;
  pos_last = motorStroke;

  readMotorState();
  reassignNextLoop();

}

void homingSequence(double homingVelocity, double hold_pos) {

  // Move to default position at constant (slow) speed

  // Initial sample of motor position
  readMotorState();

  while (pos > hold_pos) {
    readMotorState();
    speedControl(-homingVelocity);     // speed control AWAY from surface (negative direction)

    // Reassign current position to last position for next loop
```

```
    reassignNextLoop();
  }

}


void calibrateLoadCell() {
  // Send back many measurements without offsets.
  // These measurements will be fed into a linear
  // regression in MATLAB app and used to calculate
  // offset.

  // Take a few samples first to start up the filter

  for (int i=0; i<50; i++) {
    readMotorState();
  }

  // Take measurements
  int tForce = millis();
  uint8_t n = 0;

  while (n < 50) {
    int tCurrent = millis();

    if ((tCurrent - tForce) > 75) {
      tForce = tCurrent;

      force_calibrate_msg.data = readForce();
      forceCalibrate.publish( &force_calibrate_msg );
      n += 1;
    }
  }

}



//////// Control Schemes / Motor Commanding ////////



void positionControl(double pos_des) {

  // Takes in desired position, sends voltage command to motor
```

```
  // Uses PD controller based on Faulhaber software
  double error = pos_des - pos;
  double V = 13.0 * error + 5.0 * (error - posError_last);
  commandMotor(V);

  // Reassign current error to "last" error for next loop
  posError_last = error;

}



void speedControl(double velocity_des) {

  // Takes in desired velocity, sends voltage command to motor
  // Uses PI controller based on Faulhaber software
  double error = velocity_des - velocity;
  speedErrorWindup += error * ((t - time_last)/1000.0);
  double V = 0.08 * error + 0.03 * speedErrorWindup;
  commandMotor(V);

}



void forceControl(double force_des) {

  // Takes in desired position, sends voltage command to motor
  // Uses PD controller based on Faulhaber software

  // Important motor constants:
  // Ke = 9.5   -->  back-emf constant ( V/(m/s) )
  // R = 10.83  -->  winding resistance (Ohms)
  // Kf = 11.64 -->  motor force constant (N/A)

  double error = force_des - force;
  forceErrorWindup += error * ((t - time_last)/1000.0);
  double V = 5.0 * error + 0.5 * forceErrorWindup + (force_des / 11.64) * 10.83 + 9.5 * (velocity / 1000.0);
  commandMotor(V);

}



void commandMotor(double V) {
```

```
  int pwmRaw = mapDouble(V, -Vmax, Vmax, 0.0, 255.0);
  int pwmCommand = constrain(pwmRaw, 0, 255);
  analogWrite(PWMpin, pwmCommand);
}




//////// Reading Motor Variables / State ////////



void readMotorState() {

  pos = readPos() - pos_offset + motorStroke;    // in mm
  t = millis();

  double velocityRaw = (3*pos - 4*pos_last + pos_last2) / (2.0*(t - time_last)/1000.0);  // in mm/s
  double forceRaw = readForce() - force_offset;    // in N

  velocity = velocityReads.reading(velocityRaw) / 1.0;
  force = 0.92*force_last + (1-0.92)*forceRaw;

}


void reassignNextLoop() {

  pos_last2 = pos_last;
  pos_last = pos;
  time_last = t;
  force_last = force;

}



double readForce() {
  // Read analog pin
  int forceBits = analogRead(forcePin);

  // Process raw reading; translate into force measurement
  //double forceRaw = mapDouble(forceBits, 0.0, 1023.0, 0.0, 222.4);
  double forceRaw = 1000.0 * (222.4 / (509.5 * 2.0 * 5.0)) * 5.0 * (forceBits / 1024.0);
  return forceRaw;
```

```
}



double readPos() {

  double angle = readAngle();

  // Need to make sure the angle unwraps continuously
  while ((pos_last + pos_offset - motorStroke)/(24.0 / (2.0 * PI)) - angle > PI) {
    angle += 2*PI;
  }

  while (angle - (pos_last + pos_offset - motorStroke)/(24.0 / (2.0 * PI)) > PI) {
    angle -= 2*PI;
  }

  // Convert to position estimate
  double pos = angle * (24.0 / (2.0 * PI));
  return pos;

}



double readAngle() {

  // Calibrated motor offsets:
  // offsetA = 505
  // offsetB = 503
  // offsetC = 501

  int hallA = analogRead(hallA_pin) - 505;
  int hallB = analogRead(hallB_pin) - 503;
  int hallC = analogRead(hallC_pin) - 501;

  int hall_alpha = (2.0/3.0) * (hallA + (-0.5)*hallB + (-0.5)*hallC);
  int hall_beta = (2.0/3.0) * (sqrt(3.0)/2.0 * hallB + (-sqrt(3.0)/2.0) * hallC);

  double angle = atan2(hall_alpha, hall_beta);

  return angle;

}
```

```
//////// Miscellaneous ////////



double mapDouble(double x, double fromLow, double fromHigh, double toLow, double toHigh) {
  // See documentation for "map" function for implementation formula.
  // Needed to implement on my own because I'm using doubles instead of integers.

  double xMapped = (x - fromLow) * (toHigh - toLow)/(fromHigh - fromLow) + toLow;
  return xMapped;

}
```

A.2 EMV Arduino Code

```
/*
// Created by Hank on 7/24/2018.
 This script controls the VCA in a sine wave.

 Pinouts found in VCA_Plant.cpp

 Modified by Yasmin Spring 2020 to include a digital control scheme, a square wave input, and analog outputs

 Modified by Ryan Spring 2021 to be compatible with ROS environment
 Last updated: 4/12/2021
*/

///////////////// Initial Setup ////////////////////////

// Include ROS and VCA libraries
#include <VCA_Plant.h>
#include <math.h>
#include <movingAvg.h>

#include <ros.h>
#include <std_msgs/Bool.h>
#include <std_msgs/Float32.h>

// define ARF signal pin
const int ARFSignalPin = 24;   // digital read ARF signal pin 24, sent from another arduino that is
                               // dedicated to signal processing. Later this pin will come from the same
                               // Teensy board that is doing motion control
```

```
const int interruptSignalPin = 25;
const int lineTriggerPin = 12;

// User-defined variables
int vibFreq = 10;//[Hz]
double vibAmp = .75; //[mm]

// States
boolean EMVactive = false;

// define our VCA plant
VCA_Plant myVCA;

// define our PID controller for frequency sweep
int loop_period = 1000;    // [microseconds], 1kHz --> 1000us
double Setpoint, InputA, InputB, OutputA, OutputB;

// define our Digital Controller
float ErrorA=0,prevOutputA=0,prevErrorA=0, OutputV_A=0;
float ErrorB=0,prevOutputB=0,prevErrorB=0, OutputV_B=0;

// Define other motor-specific variables
double offSet = 0.5; //[mm] has to be high enough for shaft to have enough range to travel desired amplitude

// Variables for ISR and pulsing
const int num_cycles = 2;
const int cycle_gap_time = 2000;   // gap time between consecutive cycles
const int slack_time = 10;   // slack_time is used to make the vibration complete
int vib_duration;   // total vibration time, include gap
int vib_cycle;      // vibration time for each cycle

// Initialize global variables
double Out_s;
int timer;
int start_time;
int isr_timer;
int tROS;
int tROS_trigger;
int tPulse;

unsigned long lastTrigger = 0;    // Clock time at which last line trigger occurred

// Initialize moving avg filter for trigger
movingAvg triggerPeriod(10);
```

```
//////// ROS Stuff ////////

// Set up ROS node
ros::NodeHandle  nh;

// Set up ROS messages for publishers
std_msgs::Bool pulseOnset_msg;
std_msgs::Float32 triggerTiming_msg;

// Functions for subscribers
void EMVstateCb( const std_msgs::Bool& msg){
  EMVactive = msg.data;
}

void EMVfreqCb( const std_msgs::Float32& msg){
  vibFreq = msg.data;
}

void EMVampCb( const std_msgs::Float32& msg){
  vibAmp = msg.data;
}

// Set up ROS publishers and subscribers
ros::Publisher pulseOnset("EMV/pulse", &pulseOnset_msg);
ros::Publisher triggerTiming("EMV/triggerTiming", &triggerTiming_msg);

ros::Subscriber<std_msgs::Bool> state_EMV("EMV/state", &EMVstateCb);
ros::Subscriber<std_msgs::Float32> EMVfreq("EMV/freq", &EMVfreqCb);
ros::Subscriber<std_msgs::Float32> EMVamp("EMV/amplitude", &EMVampCb);

////////////////////// Main Script ////////////////////////////////


void setup() {
    // Set up VCAs
    myVCA.StopMotor();

    // Start ROS nodes
    nh.getHardware()->setBaud(57600);
    nh.initNode();
    nh.advertise(pulseOnset);
    nh.advertise(triggerTiming);
    nh.subscribe(state_EMV);
    nh.subscribe(EMVfreq);
```

```
    nh.subscribe(EMVamp);

    // Set up ISR
    attachInterrupt(digitalPinToInterrupt(ARFSignalPin), vibrate, RISING);
    pinMode(interruptSignalPin, OUTPUT);
    digitalWrite(interruptSignalPin, LOW);

    // Set up ISR to read US line triggers
    pinMode(lineTriggerPin, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(lineTriggerPin), trigger, RISING);

    // Start moving avg filter
    triggerPeriod.begin();

    // Reset variables for loop
    start_time = micros();
    Setpoint=offSet;
}

void loop() {
    timer = micros();

    if (EMVactive) {
      //Sine Setpoint
      //Setpoint = vibAmp * sin(2.0 * M_PI * vibFreq * (((double) (timer - start_time)) / 1E6)) + offSet;
      Setpoint = offSet;

      //Digital Control
      InputA=myVCA.ReadMotorAPositionMM();
      InputB=myVCA.ReadMotorBPositionMM();

      calculateOutputs();

      myVCA.DriveMotorADuty(OutputA);
      myVCA.DriveMotorBDuty(OutputB);

    } else {

      // Set the output to zero if EMV not active and
      // stop the motor (so it doesn't make sound)
      myVCA.DriveMotorADuty(0.0);
      myVCA.DriveMotorBDuty(0.0);
      myVCA.StopMotor();

    }
```

```
    // Check for updates on ROS network
    if ((timer - tROS) > 75000) {
      nh.spinOnce();
      tROS = timer;
    }

    // Update trigger timing
    if ((timer - tROS_trigger) > 0.5e6) {
      triggerTiming_msg.data = triggerPeriod.getAvg();
      triggerTiming.publish( &triggerTiming_msg );
      tROS_trigger = timer;
    }

    // Vibrate thumpers
    if ((timer - tPulse) > 1e6) {
      digitalWrite(interruptSignalPin, HIGH);
      delayMicroseconds(100);
      digitalWrite(interruptSignalPin, LOW);
      tPulse = timer;
    }

    // Wait for next loop
    while ((micros() - timer) < loop_period);
}

void vibrate() {
    if (EMVactive) {

      // Publish a pulse signal to ROS
      pulseOnset_msg.data = true;
      pulseOnset.publish( &pulseOnset_msg );

      // close interrupts
      cli();

      // Calculate pulse variables
      vib_cycle = 1 * (int)(1.0/(double)vibFreq * 1e6);
      vib_duration = num_cycles * vib_cycle + (num_cycles-1) * cycle_gap_time;

      start_time = micros();
      for (int i = 0; i < num_cycles; i++) {
          while (micros() - start_time < ( (i+1) * vib_cycle + i * cycle_gap_time + slack_time)) {
              isr_timer = micros();
              Setpoint = - vibAmp * cos(2.0 * M_PI * vibFreq * (((double) (isr_timer - start_time))
```

```
                / 1e6)) + vibAmp + offSet;
            InputA = myVCA.ReadMotorAPositionMM();  // directly read position, use direct PID
            InputB = myVCA.ReadMotorBPositionMM();
            calculateOutputs();
            myVCA.DriveMotorADuty(OutputA);
            myVCA.DriveMotorBDuty(OutputB);
            while (micros() - isr_timer < loop_period);
        }
        while (micros() - start_time < ( (i+1) * vib_cycle + (i+1)*cycle_gap_time + slack_time)) {
            isr_timer = micros();
            Setpoint = - vibAmp * cos(2.0 * M_PI * vibFreq * (((double) (vib_cycle)) / 1e6)) +
              vibAmp + offSet;
            InputA = myVCA.ReadMotorAPositionMM();  // directly read position, use direct PID
            InputB = myVCA.ReadMotorBPositionMM();
            calculateOutputs();
            myVCA.DriveMotorADuty(OutputA);
            myVCA.DriveMotorBDuty(OutputB);
            while (micros() - isr_timer < loop_period);
        }
    }

    // enable interrupts again
    sei();

  }
}

void calculateOutputs() {
  ErrorA=Setpoint-InputA;
  ErrorB=Setpoint-InputB;
  OutputV_A=2.4277*ErrorA-1.87248501*prevErrorA+0.7264*prevOutputA;
  OutputV_B=2.4277*ErrorB-1.87248501*prevErrorB+0.7264*prevOutputB;

  prevErrorA=ErrorA;
  prevErrorB=ErrorB;
  prevOutputA=OutputV_A;
  prevOutputB=OutputV_B;

  if (prevOutputA > 3.3){prevOutputA=3.3;}
  if (prevOutputB > 3.3){prevOutputB=3.3;}
  if (prevOutputA <0) {prevOutputA=0;}
  if (prevOutputB <0) {prevOutputB=0;}

  OutputA=OutputV_A/3.3;
  OutputB=OutputV_B/3.3;
```

```
  if (OutputA >1) {OutputA=1;}
  if (OutputB >1) {OutputB=1;}
  if (OutputA <-1) {OutputA =-1;}
  if (OutputB <-1) {OutputB =-1;}
}

void trigger() {
  int val = triggerPeriod.reading(micros() - lastTrigger);
  lastTrigger = micros();
}
```

Bibliography

[1] Alex(Alex Robert) Benjamin. *3D organ property mapping using freehand ultrasound scans*. Thesis, Massachusetts Institute of Technology, 2020. Accepted: 2021-01-05T23:11:37Z Journal Abbreviation: Three dimensional organ property mapping using freehand ultrasound scans.

[2] Chuan-Yu Chang, Yue-Fong Lei, Chin-Hsiao Tseng, and Shyang-Rong Shih. Thyroid Segmentation and Volume Estimation in Ultrasound Images. *IEEE Transactions on Biomedical Engineering*, 57(6):1348–1357, June 2010. Conference Name: IEEE Transactions on Biomedical Engineering.

[3] Yasmin Chavez. *System identification and control of a miniature external mechanical vibration device towards clinical ultrasound shear wave elastography*. Thesis, Massachusetts Institute of Technology, 2020. Accepted: 2020-09-03T17:49:47Z.

[4] Matthew Wright Gilbertson. *Electromechanical systems to enhance the usability and diagnostic capabilities of ultrasound imaging*. Thesis, Massachusetts Institute of Technology, 2014.

[5] Sandrin Laurent, Oudry Jennifer, Bastard Cécile, Fournier Céline, Miette Véronique, and Mueller Sebastian. Non-Invasive Assessment of Liver Fibrosis by Vibration-Controlled Transient Elastography (Fibroscan®). *Liver Biopsy*, September 2011. Publisher: IntechOpen.

[6] Daniel C. Mellema, Pengfei Song, Randall R. Kinnick, Matthew W. Urban, James F. Greenleaf, Armando Manduca, and Shigao Chen. Probe Oscillation Shear Elastography (PROSE): A High Frame-Rate Method for Two-Dimensional Ultrasound Shear Wave Elastography. *IEEE transactions on medical imaging*, 35(9):2098–2106, September 2016.

[7] Arinc Ozturk, Joseph R. Grajo, Michael S. Gee, Alex Benjamin, Rebecca E. Zubajlo, Kai E. Thomenius, Brian W. Anthony, Anthony E. Samir, and Manish Dhyani. Quantitative Hepatic Fat Quantification in Non-alcoholic Fatty Liver Disease Using Ultrasound-Based Techniques: A Review of Literature and Their Diagnostic Performance. *Ultrasound in Medicine & Biology*, 44(12):2461–2475, December 2018.

[8] Arinc Ozturk, Rebecca E. Zubajlo, Manish Dhyani, Joseph R. Grajo, Nathaniel Mercaldo, Brian W. Anthony, and Anthony E. Samir. Variation of Shear Wave Elastography With Preload in the Thyroid: Quantitative Validation. *Journal of Ultrasound in Medicine: Official Journal of the American Institute of Ultrasound in Medicine*, 40(4):779–786, April 2021.

[9] Rosa M.S. Sigrist, Joy Liau, Ahmed El Kaffas, Maria Cristina Chammas, and Juergen K. Willmann. Ultrasound Elastography: Review of Techniques and Clinical Applications. *Theranostics*, 7(5):1303–1329, March 2017.

[10] Vincent Wai-Sun Wong, Julien Vergniol, Grace Lai-Hung Wong, Juliette Foucher, Henry Lik-Yuen Chan, Brigitte Le Bail, Paul Cheung-Lung Choi, Mathurin Kowo, Anthony Wing-Hung Chan, Wassil Merrouche, Joseph Jao-Yiu Sung, and Victor de Lédinghen. Diagnosis of fibrosis and cirrhosis using liver stiffness measurement in nonalcoholic fatty liver disease. *Hepatology*, 51(2):454–462, 2010. _eprint: https://aasldpubs.onlinelibrary.wiley.com/doi/pdf/10.1002/hep.23312.

[11] Heng Yang. *Ultrasound shear wave elastography imaging with external mechanical vibration*. Thesis, Massachusetts Institute of Technology, 2017. Accepted: 2018-02-16T20:04:16Z Journal Abbreviation: Ultrasound SWE imaging with EMV.

[12] Heng Zhao, Pengfei Song, Duane D. Meixner, Randall R. Kinnick, Matthew R. Callstrom, William Sanchez, Matthew W. Urban, Armando Manduca, James F. Greenleaf, and Shigao Chen. External Vibration Multi-directional Ultrasound Shearwave Elastography (EVMUSE): Application in Liver Fibrosis Staging. *IEEE transactions on medical imaging*, 33(11):2140–2148, November 2014.