Final Report

# Transparency Optimization of the Galen Surgical System with a Frequency Domain Admittance Controller Design

Brevin Banks

# Table of Contents:

# 1 Introduction

## 1.1 Background

The Galen Robot is a 5 DOF robot that features a parallel platform or delta stage, roll motion joint, and a tilt motion joint for manipulation in cartesian space with the goal of assisting surgeons through hand over hand integration in the operating room [1]. The Galen Robot has been designed for Physical human-robot interaction (pHRI) to steady the surgeon's hand by reducing tremors and allowing the integration of optimized control for virtual fixtures and potentially automated procedures [1, 2]. The current Johns Hopkins Campus Galen Robot prototype contains an on-board control system that enables the operator to place tools on the end effector of the robot, and, by enabling a foot switch, pass through mode can be used to freely move the tool attached to the robot around in the workspace with limited resistance felt by the user from the robot dynamics.

*Figure 1 The Galen Robotic System with a tool attached to the End Effector and an operator manipulating the system [2]*

However, the current Galen Robot controller has performed with mildly low fidelity regarding the stability and transparency of the control system. That is, while a surgeon operates the robot it has been observed to have large jittery behavior during drilling and surface contact in addition to the fact that the robot feels heavy and difficult to move during pass through mode. This has led to the need for an update to the controller that improves the admittance for the robot to help it feel more transparent and to maintain stability. The exact implications of this admittance, transparency, and stability will be discussed throughout this report. This approach to improving the transparency is in effort to help qualitatively improve the performance of the controller as well as generate verified quantifiable metrics for assessing the controller performance.

Transparency in this project scenario is best described as a form of Mechanical Transparency [3], wherein the resistance or the lack of resistance felt by an operator in a pHRI situation due to the robot motors, inertia, friction, latency, etc. contributes positively to the ability to move freely [4, 5]. In pHRI situations, especially for the Galen Robot, this is desired because the robot should move where the surgeon desires quickly and without exhausting them. Stability is also a cause for concern. When optimizing for transparency, a decrease in stability is almost a direct trade off [5]. The goal of this project then becomes finding a balance of optimized values for controller gains for both transparency and stability. The outcome of this project is an admittance controller that optimizes the transparency of the current Galen robot system while maintaining system stability.

## 1.2 Significance & Clinical Motivation

A robot designed for use in the operating room is expected to have a robust level of stability. This ensures the safety of the patient and the accurate controllability for the surgeon [3, 4]. Removing the jittering experienced during contact and operations will help users to feel more confident in the use of the robot. The ability to achieve the desired positions the user drives towards, and to maintain those positions with a great level of accuracy, can help speed up procedures and make them more accurate, reliable, repeatable, etc. [1,2]. The Galen Robot's use in ENT surgery and in other potential microsurgical areas will prove quite beneficial to a surgeon's workflow with a highly transparent and reliable controller system onboard.
Future applications of this project come as a biproduct of having a robust control system optimized for both stability and transparency. This controller opens the door for the easy integration of virtual fixtures and automated portions of procedures.

## 1.3 Project Goals

The following list of deliverables outlines our project goals. These items formed the basis by which we operate and prioritize our efforts to design the controller.

- **Minimum Deliverables:**
    1. A stable frequency domain admittance controller in MATLAB code for the Galen surgical robot in the AMBF simulation environment
    2. A simulated control system package with optimized transparency containing code files that can be used in AMBF and MATLAB and is transferable to hardware.

- **Expected Deliverables:**

1. An applied control system package to the Mid-level Galen controller that is stable and optimized for transparency including virtual fixtures.
2. A validated objective improvement in transparency according to specified metrics, with identified system dynamics

- **Maximum:**
    1. Implementation of Virtual Fixtures that constrain the robot hardware within desired simulated bounds.

Should the above deliverables be obtained, the implementation of a working admittance controller that can be used in plenty of further applications will be avaliable for use. The robot will feel more transparent, and stability will be guaranteed in cartesian control.

In addition to the maximum deliverable is an extended maximum deliverable where a user study could be performed with actual ENT surgeons to receive qualitative feedback. However, this has only been speculated and deemed outside the finite scope of the project for the time being.

For more details on these deliverables and the project execution plan see TOGAC10_D_ProjectProposal.pdf.

Documentation methods and details for every document used in this project can be found in TOGAC2_D_DocumentControl.xlsx

# 2 Technical Approach

To understand the goal and approach of this project, it is important to understand what stability, transparency, and admittance controllers are. We will first address Admittance. With these design terms investigated, it will become possible to implement them on the Galen robot in simulation and on the real hardware. First, we will attempt the control design on a teleoperated version of the Galen robot in a simulated environment. Second, with the controller design verified in simulation, transferal to the real hardware will be performed and similarly analyzed with real world human interaction. The design documentation for this controller and the subsequent derivations of transfer functions can be found in TOGAC6_CD_ControlSystemsDesign.docx, TOGAC17_MSD_R_StabilityFBandFFApproach.docx, and TOGAC27_CD_TransferFunctionDerivAndStability.docx.

## 2.1 Admittance Control Design

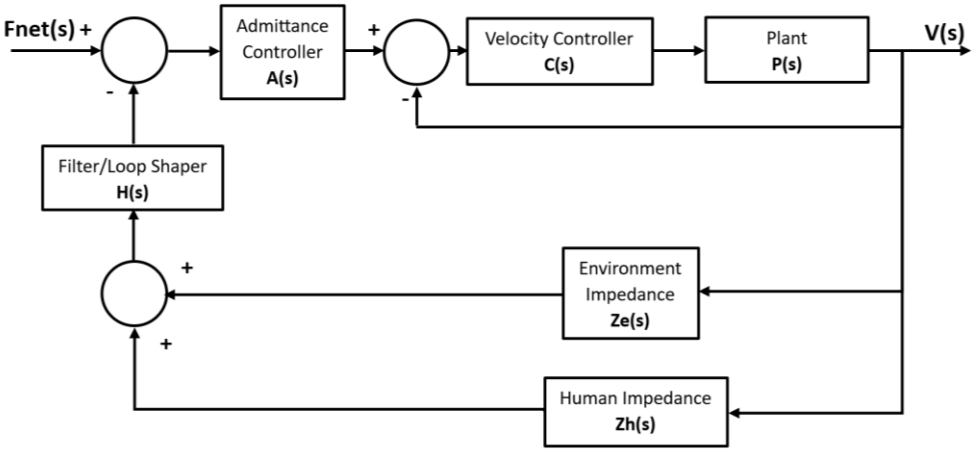


*Figure 2 Basic approach for the frequency-based admittance controller design*

Understanding the above control diagram does require a fundamental background in basic linear control theory. It is requisite that the essentials of a basic feedback controller are well understood since these elements are crucial for the design of an admittance controller. Admittance in this sense is the inverse of impedance. This approach will take a basic velocity control based PID controller, C(s), for an assumed linear system, P(s) and apply an integrator with a gain after the input before the controller to virtually remove system impedance [5, 6]. We'll call this the admittance controller A(s). This will make the system seem 'transparent' because the resistance of the dynamics appears removed, but it is important to know that if the system impedance is completely removed the robot can be easily tripped unstable. It would be essentially like dividing by 0. Moreover, it is not physically possible to compensate for all the inertia and resistance effects of the robot in the real world. There will need to be some level acceptable resistance felt by the operator. All in all, particular care must be taken when tuning the gains of A(s). The goal would be to minimize the impedance and conversely maximize admittance.

The Admittance controller will take the form of a transfer function with 2 tunable gains: a desired mass gain, and a desired damping gain. It will look something like this:

$$Admittance$$
$$A(s) = \frac{1}{m_{ad}s + b_{ad}}$$

These desired mass gains and desired damping gains would be the theoretical values the user would feel while operating the robot. They should in theory be much smaller than the true mass and damping of the robot.

Also note the inclusion of external impedances from the environment and the human arm. Because there are uncertainties in the operation of the robot, we will include these impedances and optimize the transparency for a range of different human and environment impedances. Simply put, based on data in literature, human and environment impedance can be modeled as a mass-spring-damper with each their own m, b, and k variables that can be tuned. Upper bound values for m, b, and k have been obtained for both the human arm (mh = 5kg, bh = 41 Ns/m, kh = 401N/m) and environment and (me = 0kg, be = 0Ns/m, ke = 16599N/m) with the lower bound being equal to 0 for all metrics. For the framework of this impedance approach see [6].
The simulated impedances are applied by using these bounding values in the following form where Zeq = Zh + Ze.

$$Impedance$$
$$Zeq(s) = mhs^2 + bhs + \frac{kh}{s} + \frac{ke}{s}$$

In testing the worst-case impedance values that cause the admittance controller to go unstable, we can observe different situations of the bounding impedance. For instance, we may test the stability of the controller in the case that only human impedance with stiffness is present (i.e. mh = 0, bh = 0, kh = 401, and ke = 0). By investigating the different simulated impedance cases, we can predict what possible values for admittance mass and damping gains would be acceptable in the controller design for all anticipated human-environment interactions. We will see more on this as we observe the testing in simulation.

Notice the plant block, P(s) from figure 2 is supposed to be a linear transfer function that approximately models the real-world dynamics of the robot. Most Robots are highly nonlinear, and their plants are difficult to accurately model. It is valid to be concerned about assuming the plant is linear. To address this concern, we will perform system identification for the dynamics of the Galen robot for both the AMBF (Asynchronous Multi-Body Framework) simulation environment and then the real Galen robot once the simulation for velocity control has shown promising results [4, 6, 7].

Another detail to note about the control architecture is the filter/loop-shaper block, H(s), which may or may not be used in the application of the real controller on the robot. This would be used to smooth the input force from the human and the environment to keep the robot from going unstable from external factors.

## 2.2 Stability

Now we can address what stability in this context means. Stability can be best explained through the example of operation. We see stability in interactions of the system input

(e.g., the operator's hand and the contact between tools and the patient or environment) and the response of the robot dynamics from the hardware. The user inputs some type of desired force, Fnet(s), and then a controller, C(s), interprets this force and outputs it into a form the robot dynamics, also called the plant, P(s), can use to move the hardware to the desired location.

When certain inputs cause the robot to behave erratically, we observe instability in the dynamics of the robot such as uncontrolled shaking or fast, dangerous movements. To ensure that the system is stable we often apply feedback and different elements to our controller, C(s), that will keep the output of the dynamics within safe stability margins. There are many ways to accomplish stability, and the best choice comes down to the market factors chosen to optimize as well as the input types and system to be controlled.

When testing the stability of our controller, we will perform a closed loop pole analysis of the Galen Cartesian system dynamic transfer functions in conjunction with the controller, admittance block, impedances, and filter. The transfer function from the net force input to the velocity output is:

$$TF(s) = \frac{V(s)}{F(s)} = \frac{A(s)C(s)P(s)}{1+C(s)P(s)(A(s)(Zh(s)+Ze(s))+1)}$$

Here one issue is the classification of C(s) and P(s). P(s) is often hard to define and requires rather precise measurements, and calculations, in order to produce an accurate dynamic model, and even with an accurate model it is subject to uncertainties and unknown disturbances. The controller C(s) can be created by the controls designer, but in our case, the Galen robot and the AMBF simulation both have low level PID controllers already onboard. Future work may include refinement of the PID gains for both systems, but our approach will bypass tuning PID gains by lumping C(s) and P(s) together into one transfer function that we obtain from system identification methods. In our system identification we will estimate the order of the transfer function to be 2$^{nd}$ order using MATLAB's tfest tool in the system identification toolbox. This is because the second order system adequately captures the basic performance data for simple movements of our robot, and the stability analysis is straight forward and predictable and have been shown to work in similar applications [5, 6]. The characteristic equation of the identified transfer function, (i.e. the denominator) can be investigated to quantify the resultant system stability. Solving the characteristic equation for its poles and the transfer function for any present zeros can tell us if the robot is expected to go unstable for any potential inputs should any of these solved poles or zeros lie in the right-hand plane.

## 2.3 Transparency

Next let's discuss transparency. Transparency and stability are closely related in terms of the performance of a control system. Transparency is a measure of how easy the system is to manipulate or drive in the desired direction without feeling heavy resistance or drag. Unfortunately, transparency is not as easy as stability in the sense of tuning system parameters and checking closed loop poles to make sure we are in a safe region. Transparency does not have an accompanying concrete mathematical definition nor universally agreed upon optimality preferences. However, transparency can be inferred by observing the predicted impedance during robot performance. We can observe the control system at Zh(s) in figure 2 where we see the human interaction. If we open the loop at this point and investigate the magnitude of the force felt at the end of the loop (from Fnet(s) to the output of Zh(s)) given a range of different anticipated forces (the possible range of human motion) on the input, we can compare the output forces to the controller given different admittance parameters in A(s). This is reflected as the maximum impedance force displayed to the human, Zds, at that point for a given input. The magnitude of this impedance force is given by the following equation.

$$|Zds| = |C(s)P(s)\big(A(s)\big(Zh(s) + Ze(s)\big) + 1\big)|$$

The range of impedances tested will span from forces in the frequency domain starting at 0.01Hz to 20Hz. Observing the sum of these force magnitudes in the range may imply the relative transparency of the system [5, 6]. A low force sum indicates high transparency. Therefore, we will assume that as we tune for stability and change some metrics about the system impedance with our controller, we will be able to optimize the transparency. This approach is known as the design of an admittance controller and the fundamental design our control system will take is heavily inspired by this [5, 6].

With both stability and transparency analyzed for a range of expected admittance mass and damping gains, a map of the acceptable admittance values will be plotted, and the relative transparency of those stable values will be given by an underlaid cost map based on the range of impedances displayed to the human. Using this tool, an admittance control designer can select an admittance mass and damping gain that keep the system stable and transparent and can verify that these gains are stable by using the robot in contact and free motion in simulation or the real robot.

## 2.4 AMBF Simulation

The AMBF simulation allows us to visualize the performance of the robot in simulation. Using a velocity controller in the cartesian space, the robot can be controller used to identify the system in each of the 5 DOF the Galen robot has [7, 8, 9]. For this to be performed it was requisite that the simulated Galen Jacobian be obtained to create the controller, since the present controller was only avaliable for joint space control. The real Galen robot features a mid-level control scheme with servo_cv commands from the

LCSR crtk_commands open-source library which removes the need for the Jacobian application on the real robot [10]. Cartesian control is already present; therefore, the translation from cartesian to joint space method will be a fundamental difference between the simulated controller and the real controller. The forward kinematics and Jacobian for a delta platform and the subsequent joints of the Galen robot from the AMBF environment world to the robot end effector will be found using the methods from [11] in TOGAC40_D_DeltaKin.pdf. The simulated control scheme then features the new control blocks between the admittance controller which outputs the velocity reference given a force and the identified system. Note that the current joint positions, qout, can be read directly from the AMBF simulation to use in finding the instantaneous Jacobian result. The derivation of the Jacobian and the result can be found in TOGAC42_AMBF_FullSymbolicFKderiv.m

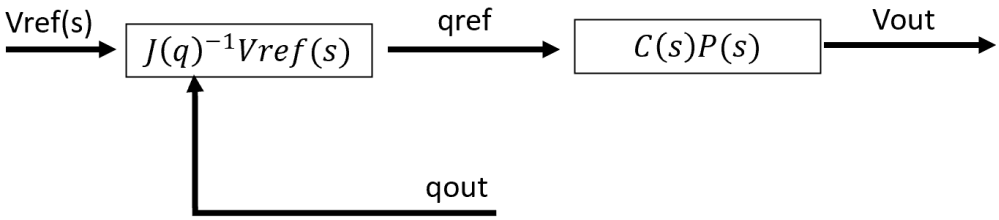


*Figure 3 Cartesian input to joint space style input control for the control of the AMBF Galen robot.*

The above controller will be designed and built in MATLAB and virtual optimization of the controller for transparency and stability will be performed there while controlling the robot in the AMBF simulation. With a functioning and well tested controller in MATLAB and AMBF the controller will be packaged and structured to accept the inputs and outputs of the real Galen Robot hardware be compatible with the python mid-level control commands from the crtk_command library [10].

# 3 Experimental Setup

## 3.1 System Identification

To operate the Galen Robot in AMBF, ROS connections between MATLAB and AMBF need to be generated. We generated an interface that allows us to easily control the robot in simulation with a MATLAB interface, enabling velocity commands and the ability to read the robot sensors. Instructions on how to use this interface and how to setup your OS with ROS and AMBF can be found in TOGAC25_AMBF_IN_AMBFforGalen.docx,

TOGAC29_AMBF_IN_ConnectMatlab2AMBF.docx, and
TOGAC50_AMBF_IN_AdmittanceControlTeleop.docx.

Testing of the controller in the AMBF environment first required the system identification of the Galen robot in each of the cartesian axis. This was done by moving the AMBF Galen robot on a velocity chirp input signal from 0.01 Hz to 5 Hz over the course of 60 seconds.

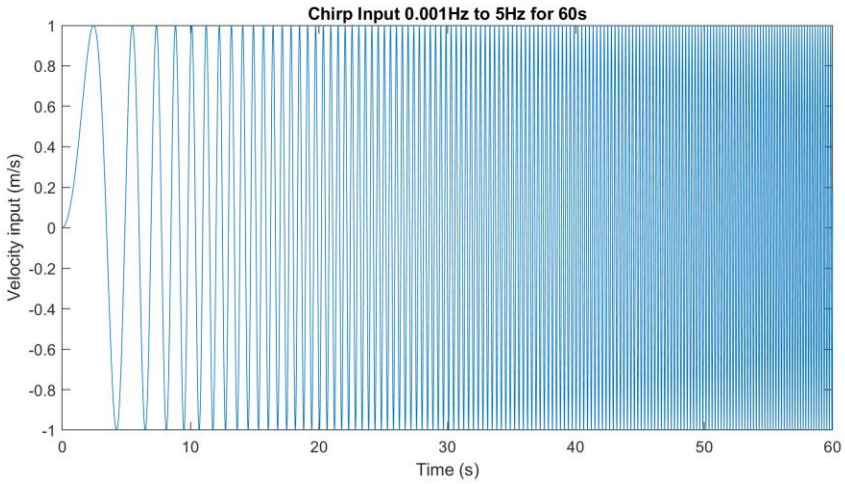


*Figure 4 Reference chirp signal for simulated identification*

Gravity for the robot was turned off during the testing and teleoperation of the robot in AMBF to simulate gravity compensation that is present on the real Galen robot controller. The input and output velocities for a cartesian axis of the Galen robot were recorded and used in a second order estimate with the tfest tool in MATLAB and then the process was repeated for each individual axis. 5 Transfer functions were obtained corresponding to each of the cartesian axis.

## 3.2 Iterative Stability and Transparency Tests

With the system in each DOF obtained, the analysis of the stability and transparency of the simulated Galen robot controller was assessed for the admittance controller. The closed loop poles of the transfer function were checked for stability over a range of admittance mass gains and damping gains. An analysis for the upper bound cases of impedance was found (that is the case where the robot includes human impedance and impedance from contact with the environment.) and an analysis for the lower bound cases of impedance was also obtained (with human impedance, but no environment contact impedance). The range of admittance masses and damping values for the upper bound and lower bound with corresponding impedance cases are found in Appendix A.

The stability analysis was performed, and a cost transparency map was also made using the same magnitude sum and Butterworth frequency method found in [5]. The tools used to perform the iterative stability analysis for the range of admittance gains can be found in TOGAC39_AMBF_CartesianChirpIdentifiedSystemStab.m

## 3.3 Simulated Impedance Verification

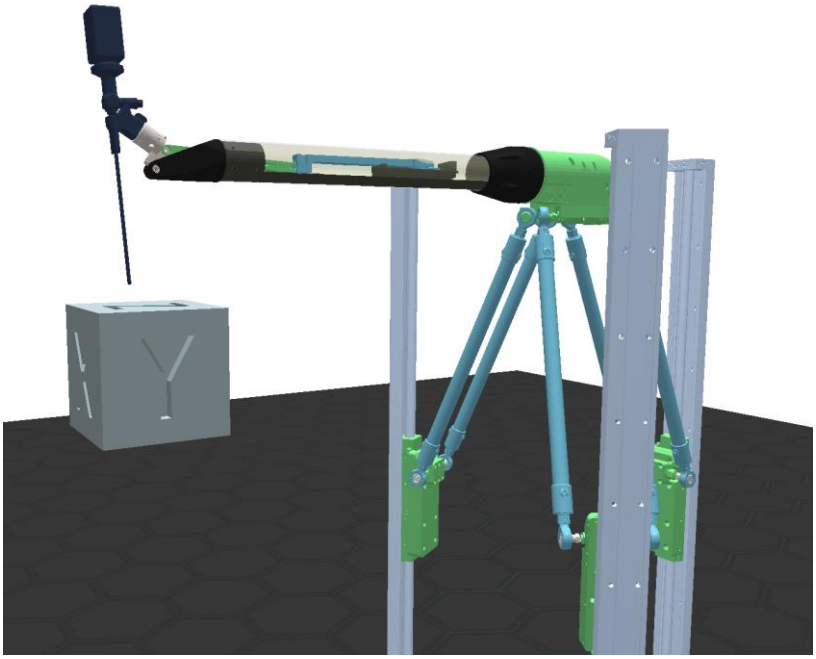

*Figure 5 Galen Robot in AMBF simulation positioned above the contact block.*

With the map generated, a set of values in the expected stable region and expected unstable region of the admittance gains mass and damping were chosen. The robot was then positioned 0.05 m above a solid fixed cube in the AMBF simulation where the robot tip was pointing straight down (along the z axis) and then the robot was given a sigmoid force trajectory of 10N down towards the cube. The input force, input velocity, and output force and velocity of the robot for each joint and DOF were recorded for 10 seconds.

To assist in controlling the robot in simulation a GUI was created to allow for easy positioning of the robot for the test setup. For further investigation of the controller performance, the GUI and AMBF MATLAB scripts were modified to allow the use of a joystick or gamepad for simple teleoperation.

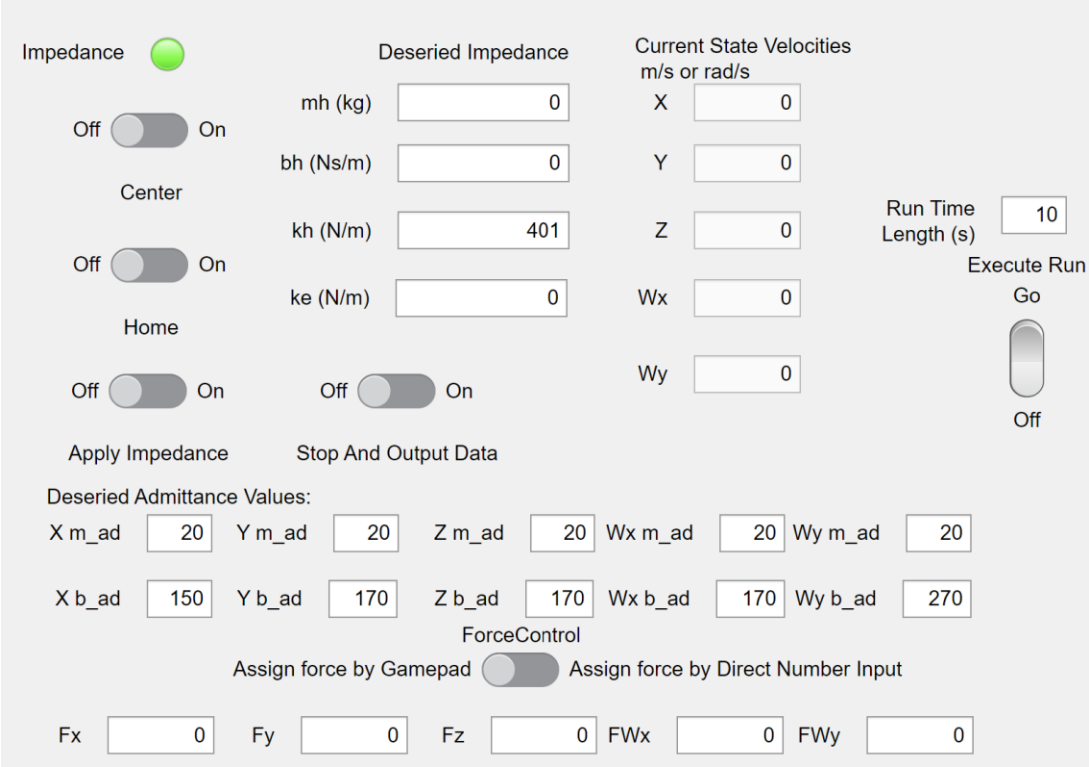


*Figure 6 AMBF Galen Robot Control GUI*

Instructions for how to use this GUI and the gamepad teleoperation can be found in TOGAC50_AMBF_IN_AdmittanceControlTeleop.pdf

The basic control scheme for our operation of the admittance controller takes the following form.

Initialization:

- The Galen interface creates the communication from AMBF to ROS
- The Rate of the control is set to 200 Hz. The operating frequency of the Galen Robot.
- The Galen robot command type is set to velocity control.

Control Loop:

- The time of each iteration is collected.
- The input force is received either teleoperation, GUI input, hard coded commands, or the actual force sensor depending on the desired test.
- If simulating the robot in AMBF the net force is calculated from the difference between the desired input force and the environment contact force.

- On the real system the net force is obtained directly from force sensor measurements.
- The force is applied to the admittance block to receive the velocity reference.
- The velocity reference is given to the servo_cv command on the real robot system. In AMBF the velocity reference is converted to joint velocity references by the Jacobian pseudo inverse.
- The Output velocity is measured and recorded. On the real system filtering of the velocity is necessary because we must take a noisy finite difference approach due to the real sensor only outputting position data.
- In the AMBF simulation the output velocity is used to calculate the predicted impedance force used in the next iteration
- The data is collected for the iteration and the rate of the loop is controlled by ROS.

The frequency-based controls discussed in section 2 and figure 2 are implemented in software by discretizing each respective block transfer function using Tustin's method.

## 3.4 Real Galen Implementation

System Identification was performed for the real Galen robot by applying a similar chirp signal to the cartesian dimensions. We initially began with testing the X, Y, and Z dimensions with a chirp on the range of 0.01Hz to 0.5 Hz over a ten second signal.

With the system Identification and transfer function result, a stability and transparency analysis could be performed to find the admittance gains necessary for the real controller implementation.

With the controller designed in AMBF, the controller mentioned in the previous section was applied to the real Galen robot in the Mock OR in similar manner using the crtk_command library with python mid-level control.
In this application, the user should be able to move the robot by grabbing the end effector or a tool attached to it and sending the control commands by hand. The force sensor present at the end effector will translate the user input into a wrench used in the admittance control scheme to generate a velocity reference.

# 4 Results

## 4.1 System Identification Transfer Functions

The system identification for each DOF resulted in either a first or second order transfer function. Using the tfest MATLAB tool, we were able to initially give the transfer function estimator a second order guess, but if the solver is able to fit a better matching transfer function to the data with a lower order it may opt to do so. In the case of the X and Y directions, the transfer function returned was first order. This significantly reduces the complexity of the control in those directions; however, further investigation of these estimates does show that simulated responses in the X direction only have about 77.93% confidence in this transfer functions ability to simulate the correct trajectory.

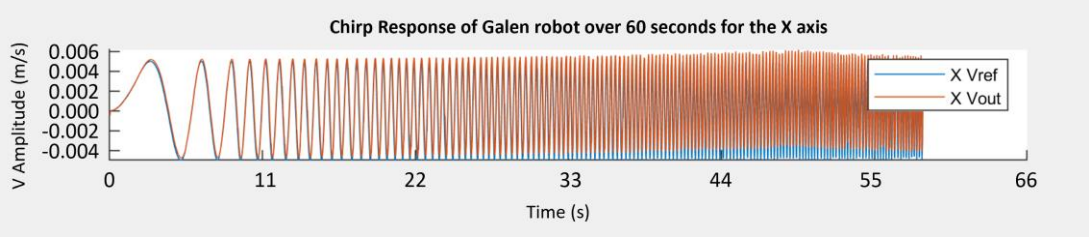


*Figure 7 X axis velocity response for a chirp input in the Galen Robot AMBF simulation*

The output of the chirp response for the X axis can be seen in figure 7. Notice the drifting of the output velocity at higher frequencies. Many factors may contribute to the shifting of the robot such as uncertainties in the model characteristics, noise, sampling time, resonate frequencies, and the PID gains. We observed similar responses in other DOF, but X had the most prominent drift from the reference at higher frequencies. Upon further analysis behind this drifting, it was decided that system identification for these frequencies is acceptable but limiting the frequency of the chirp signal to a much smaller value than 5 Hz may help isolate a more manageable response in future works if the control behaves unexpectedly.

The subsequent transfer function for each DOF and the quality of the transfer function match is found in Table 1.

Table 1. The resultant transfer functions for the system identification for the AMBF Galen Robot.

| **Velocity Component** | **Transfer Function** | **Quality of Match** |
|---|---|---|
| Vx | 132.8/(s + 145.8) | 77.93% |
| Vy | 131.8/(s + 132.4) | 92.15% |
| Vz | 1.973e06/($s^2$ + 1.558e04s + 1.975e06) | 92.58% |

| | | |
|---|---|---|
| Wx | $2.542e05/(s^2 + 153.1s + 2.561e05)$ | 89.39% |
| Wy | $1.587e04/(s^2 + 230.7s + 1.584e04)$ | 91.03% |

Even though the quality of match for X is 77.93%, we observe that the simulated control behaves as expected when sending control inputs along the X direction. The other DOF transfer functions similarly respond as expected with step, sine, and chirp inputs when compared to the real system response.

## 4.2 Stability and Transparency Maps

The following maps were generated from the iterative stability and transparency analysis. The refinement level between iterations was 30 for both the mass gain and damping gain. We divided the max value for the mass gain and damping gain by 30 to determine what value to test for each iteration and obtained plots for each DOF for stability and transparency. The plots for Z are shown here as an example. The other plots and further analysis of these results can be found in TOGAC51_AMBF_R_CartesianSystemIdentification.docx

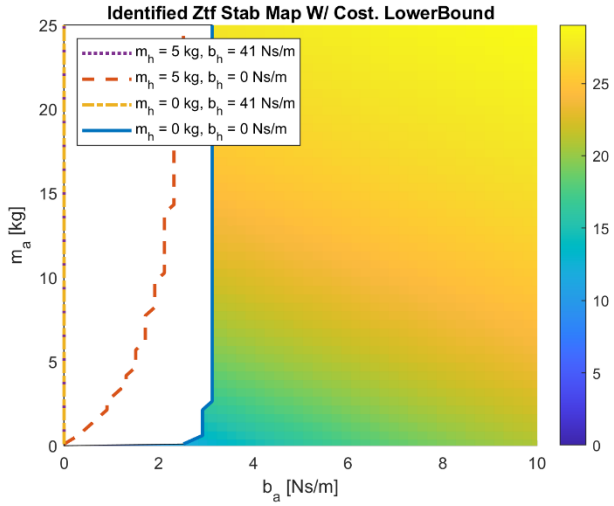


*Figure 8 The lower bound stability and transparency map for the Z axis (no simulated environment contact)*

Figure 8 shows where the robot will be stable and transparent for all the worst cases of human arm contact with the robot. Following along the line that bounds the colored area and the white area, this represents where stability is theoretically lost in the robot response. The area that is colored is stable, and the underlying cost map shows the relative transparency that would be felt by the user upon choosing a set of mass and

damping gains in that region. The bluer the pair of values are, the more transparent the robot should feel. This tool proves extremely useful in the iteration and refinement of choosing admittance gains. Caution should be taken when choosing values too close to the line of stability or too close to 0 as the closed loop stability analysis may not thoroughly capture all elements of robust stability, and the identified system is not a 1:1 match with the real-world dynamics. It is better to be conservative in selection of these gains.

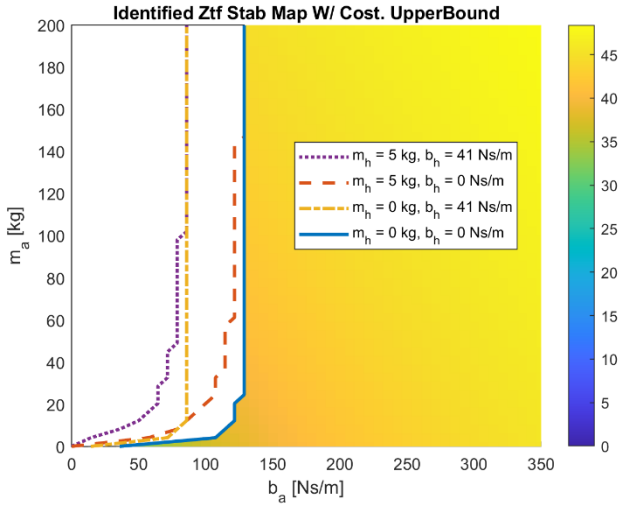


*Figure 9 The upper bound stability and transparency map for the Z axis (with simulated environment contact)*

Similar to figure 8, figure 9 relates the same information with the human arm in contact in addition to theoretical environment contact. The gains in this scenario must be increased to much higher values in order to maintain a strong level of stability. Particularly notice the necessary increase in the damping gain values. While increasing damping will help ensure stability, it will greatly increase the resistance the user will feel [5]. Further, this stability analysis does not account for possible cases where increasing the damping to very high levels could lead to different areas of instability [5,6]. This analysis will not investigate this case as we are interested primarily in finding the smallest most appropriate gains. With both of these figures, selections of gains for the expected use case can be made and tested on the robot in the AMBF simulation.

## 4.3 AMBF Simulated Results

Using figure 9, we selected the following gains for testing the simulated human interaction and environment contact in the AMBF simulation.

Table 2. The selected admittance gains for testing the Galen Robot in AMBF simulation.

| State | Mass gain (kg) | Damping gain (Ns/m) |
|---|---|---|
| Predicted Stable | 20 | 170 |
| Predicted Unstable | 20 | 20 |

The gains were applied to the robot and the force trajectory from 3.3 was applied to the z axis. The following plots show the results from the trajectory assumed from both sets of gains.

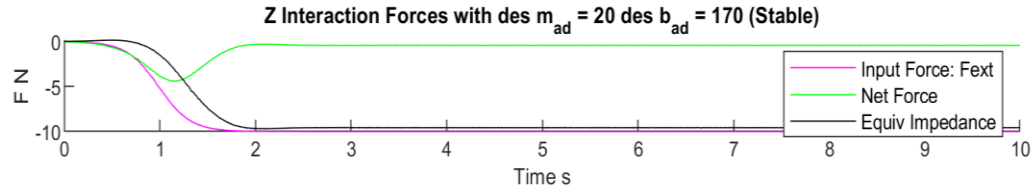


*Figure 10 Stable force trajectory in simulated contact with human and environment for the Galen Robot in AMBF*

For the set of stable gains, the robot moves slightly given a force input. The equivalent impedance force from the resistant human and environment motion then kicks in once it receives the input feedback. The robot does allow slight movement, but ultimately the impedance of the human and environment slows down the velocity and stops the motion. The steady state case has no present velocity or jittering in all axes.

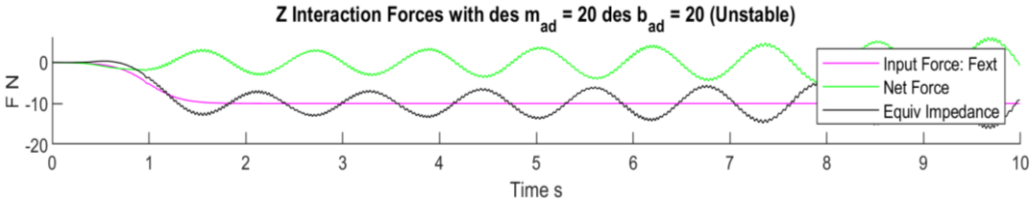


*Figure 11 Unstable force trajectory in simulated contact with human and environment for the Galen Robot in AMBF*

For the set of unstable gains, the robot moves erratically with oscillating movement that increases in amplitude and ultimately goes unstable as expected. Thus, we can assume that by increasing the damping some we could stall the oscillations and increase the stability. More refinement and tuning of these gains could be done, and it would be interesting to see how the results are changed by choosing smaller values for the mass and damping closer to the line of stability.

With the stable gains show to ensure a stable response in all dimensions, the controller can be verified through these contact tests. With this complete the controller can be updated to the python crtk_command environment for controlling the real robot in a similar manner. The next steps would be to repeat the system identification process and execute the same stability tests. Then the gains could be selected and applied to the real admittance controller.

## 4.4 Single Axis Galen Implementation of Admittance Control

Time allotted for the implementation and testing of one single axis of the Galen robot. The Z dimension was chosen and used to evaluate the admittance control's ability to interpret the user's input into a velocity reference.

The following transfer functions for the X, Y, and Z axis were obtained for the real Galen robot system identification. Note that the actual system features a few new errors introducing variables. Two prominent variables that will be addressed here are the delay and the motor backlash.

Table 2. The resultant real Galen system identification transfer functions and estimated delays

| **Velocity Component** | **Transfer Function** | **Estimated Delay (s)** | **Quality of Match** |
|---|---|---|---|
| Vx | $9.883e04/(s^2 + 12.47*s + 1.034e05)$ | 0.115 | 84.23% |
| Vy | 1.607e07/(s2 + 1.971e05*s + 1.655e07) | 0.0599 | 87.13% |
| Vz | 2.075e04/(s2 + 749.4s + 2.098e04) | 0.015 | 92.43% |

The tfest MATLAB estimator was able to estimate the transfer functions in addition to measuring the time delays. This time delay could be caused by several factors. A few of these factors are due to the control processing delay between communication from the mid-level control to low-level to ROS and the actual robot. The delay is also a result of the lagging time in the acceleration or inertia of the robot hardware. This is especially present in the case of motor backlash. When the commanded velocity or present velocity of the robot reaches zero, especially on robot initialization, commands that move the robot out of a stationary state must first overcome the effects of static friction. This is called motor backlash and can cause significant delays and shifting of the output signal.

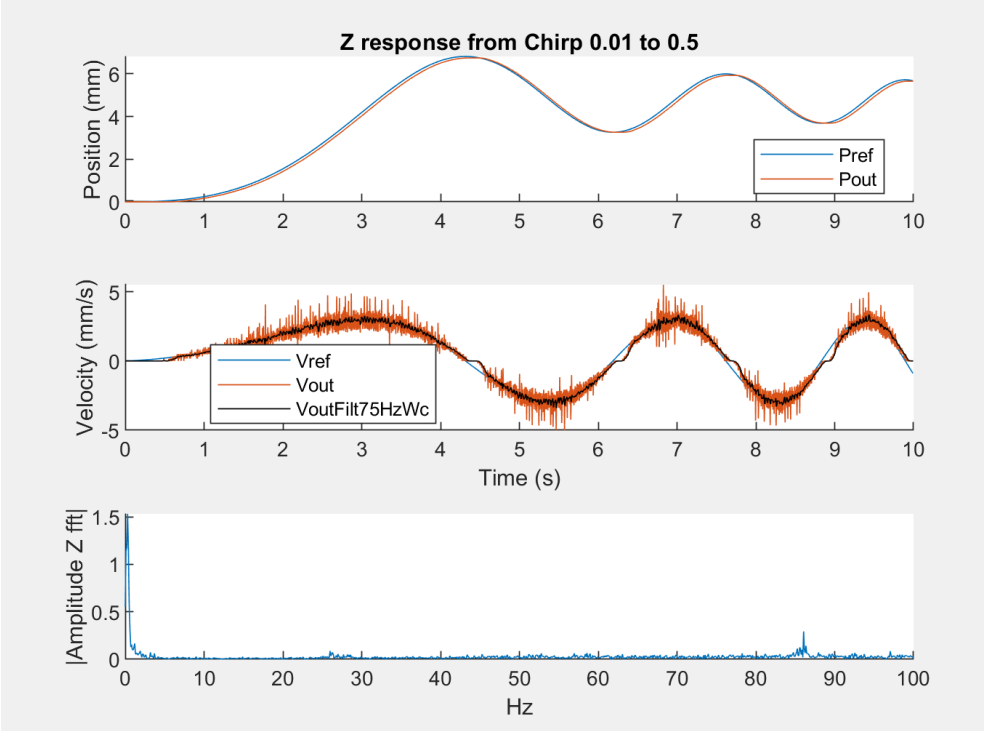


*Figure 12 The output position of the real Galen robot, the velocity, and an fft of the velocity noise*

Observe figure 12 on the velocity input and output plot. The motor backlash delay can be seen at any point where the velocity out reaches zero. Until the robot can break the static friction and overcome the initial motor inertia, the velocity controller must quickly compensate to catch up to the reference.

We also can see that the output velocity of the robot, in orange, is a rather noisy signal. This is due to the fact that we only have access to the position measuring crtk_command tools that observe the motor encoders. To get the velocity of the position signal, we use a finite differentiation method between the current timestep and the previous time step to derive the position.This introduces lots of noise, and in order to utilize a more precise velocity signal for feedback control it is important to filter the data.

We ran an fft on the finite differentiated velocity to observe what appropriate cutoff frequency should be used and deemed 75 Hz was an appropriate frequency to use in a first order low pass filter.

Figure 12 shows an example of the response of the Z axis identification. This axis showed the smallest time delay of the 3 identified systems. We notice that this time delay is large rather large on the X axis and there are many factors still to be investigated that could be causing this. There is much more work to be done on the real system implementation and testing.

For our current approach, we chose to start simply with a single axis where we were most confident to obtain a good admittance response. We found the Z to have the smallest time delay and the stability analysis was deemed the most stable according to the following maps.

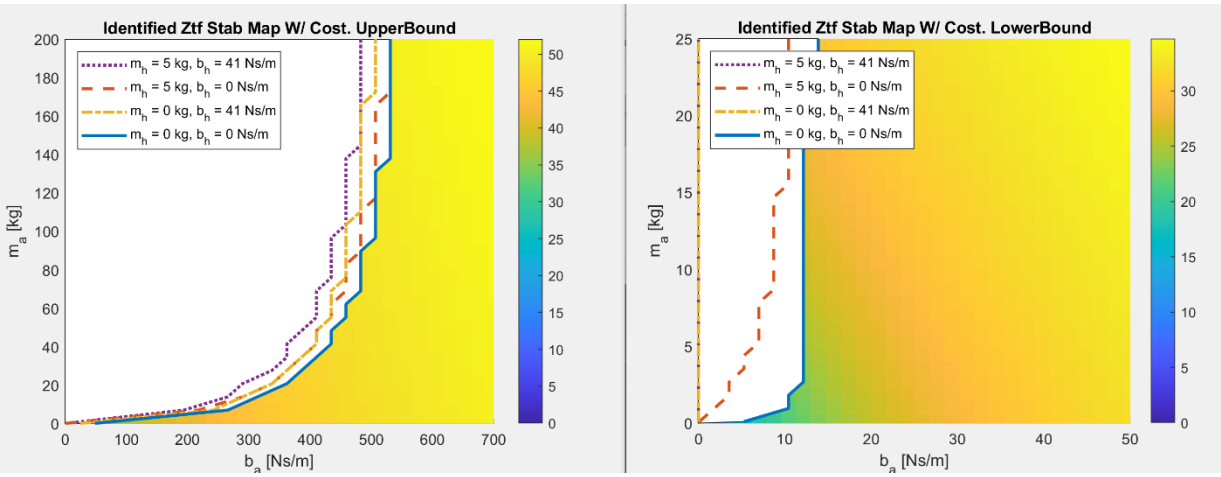


*Figure 13 Upper bound (with human and environment) impedance and Lower bound (only human) impedance stability and transparency maps for the Galen z axis.*

With the above results we were able to select a set of gains for the human only contact impedance case and test them on the real Galen robot.

*Figure 14 User interacting with the Galen robot end effector to apply a wrench*

The admittance gains of 8kg and 20Ns/m were chosen for the mass and damping values respectively. A test was run for 10 seconds with the robot in gravity compensation mode. The user was instructed to wait 5 seconds and then apply an upwards input on the robot end effector. The following plot was obtained for the input reference velocity vs the input user force.

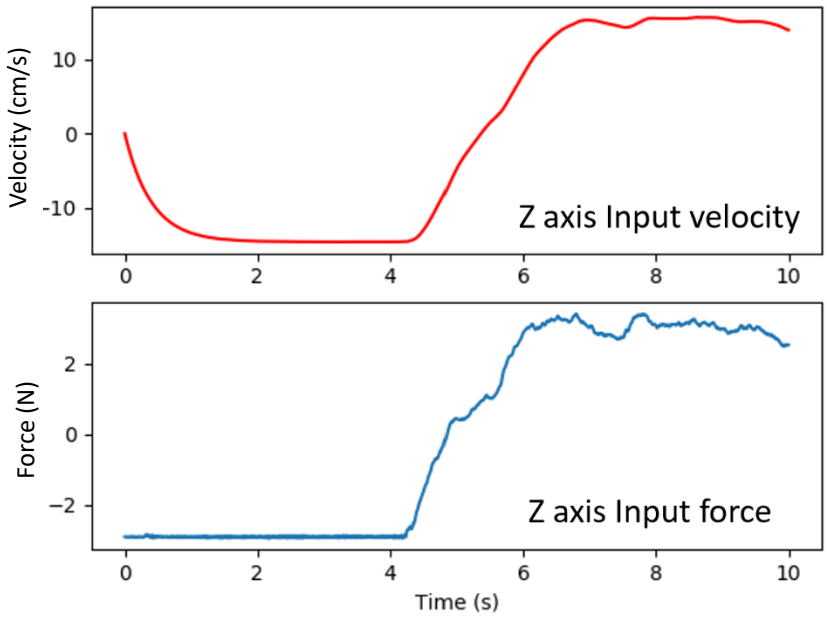


*Figure 15 The recorded input reference velocity as calculated from the admittance controller and the measured true force from the Galen robot end effector sensor.*

As anticipated, the robot tracked the force references and converted the input wrench to a clean and smooth velocity trajectory. Notice that even with the Galen gravity compensation on, during the 0 to 4 second period without human contact, the robot experiences a small downward force. This indicates that the Galen robot needs to be recalibrated or adjusted to better handle gravity compensation.

The present gains for the admittance controller in this Z axis show a good level of admittance that feels smooth, light, and controllable. Qualitatively, compared to Z motions on the current on-board Galen controller, our controller is noticeably better at feeling more transparent. However true testing and evaluation of the controllers will prove if the results truly are more transparent.

# 5 Progress Evaluation

## 5.1 Management Summary

Brevin Banks performed the development of all control related items including:

- Stability Analysis Iteration Tool Development
- Transparency Analysis Tool Development
- Admittance Controller Design and Discretization
- AMBF Galen Forward Kinematics and Jacobian Calculations
- AMBF Galen Robot ADF Model Dynamics and Collision Updates
- AMBF Controller Design and Implementation
- AMBF to MATLAB Communication Interface
- AMBF Teleoperation and GUI
- AMBF System Identification
- AMBF Stability and Transparency Map Generation
- AMBF Testing
- Real Galen System Identification
- Real Galen Stability and Transparency Map Generation
- Real Galen Mid-Level Controller Design and Implementation

## 5.2 Dependencies

There were no dependencies that presented issues for my project. Much of the time and effort towards the project were spent in the AMBF simulation and understanding the theory behind the controller. Access to the software and the environment were quickly obtained.
Complications surrounding the control of the real Galen Robot in the Mock OR did arise. Due to the nature of the Galen prototype, we were hesitant to haphazardly iterate the controller because we wanted to be extremely cautious with our approach. The controller acts rather wildly and unpredictably if incorrect gains, units, or sampling frequencies are used, therefore the scope of this report has focused simply on the execution of the AMBF simulation. Work on the real Robot has been in motion, but the development has been much slower.

Table 3. Project dependencies

| Dependency | Need | Contingency Plan | Planned Deadline | Hard Deadline | Status |
|---|---|---|---|---|---|
| Access to Computer | Need a computer with a Linux platform | Use a lab computer | 8-Feb | 13-Feb | Completed |

| | | | | | |
|---|---|---|---|---|---|
| Access to Galen AMBF Model | File access and editing permissions | Create a basic 3D model and crude dynamics | 14-Feb | 20-Feb | Completed |
| Software Installation MATLAB SIMULINK | License | Use Lab Computer with preinstalled software | 8-Feb | 13-Feb | Completed |
| Software Blender and AMBF Addon | Installation Instructions | Use Lab Computer with preinstalled software | 8-Feb | 13-Feb | Completed |
| Galen Robot and Controller Access | Anton and Adnan complete controller pipeline for Galen task space and joint space control | Go without Galen hardware implementation. Implement on similar robot or move on without implementation | 15-Mar | 15-Apr | Completed |

## 5.3 Adherence to deliverables

Refer back to section 1.3 for the list project deliverables.

All of the minimum deliverables in section 1.3 were completed and delivered in the expected time frame. The controller package, interface instructions, and reports can be found on the project wiki.

The expected deliverables are in progress. System Identification has been done for the X, Y, and Z DOF of the real system, and a prototype controller is on board the Galen Robot that has yet to be thoroughly tested. No work on the virtual fixtures has been attempted yet since the controller is not running. The stability analysis of the identified systems has been done, but they are not yet verified completely by robot applications except for the Z axis.

The maximum deliverables have not been achieved during the duration of this course.

Due to the time spent developing the controller theory, testing in MATLAB and AMBF, and the caution surrounding the real hardware, the maximum and some expected results were not achieved. Priority was given to ensure that the proof of concept behind the

admittance control design was truly feasible. We have been able to show this much as outlined in this report and the others found on the project wiki.

Other key factors outside the scope of our project became important subtasks for each deliverable. These greatly extended the time of development as well. For instance, we were unaware of the rigorous testing and analysis we would need to perform system identification in both simulation and on the real hardware. The execution of this task took longer than expected and was not originally in scope. In addition to the system identification, the maximum deliverable used to contain a user study that was in deliberation for some of the early weeks of the project. The decision to drop the user for the time being freed up time to focus on the development tasks, but we expect to reconsider adding the user study in our maximum deliverables again once the real Galen Robot admittance controller is functional.

With the project in its current state, the software, development, design, and instructions are all well documented and available for review and reference. Future work will be done to complete the outstanding deliverables post spring 2023 semester until the beginning of 2023 summer.

# 6 Conclusion

## 6.1 Discussion

Overall, the project is a success. We were able to accomplish a significant amount of work with only a few hands. I had the help of great mentors who helped streamline the project and direct me to the resources I needed. We succeeded in generating a working Admittance controller in the AMBF simulation that gives us the expected performance we would like with the real hardware. In addition to the controller in simulation, tools to connect the robot to AMBF and teleoperate the system with easily repeatable tools will now allow us to perform many different control tests faster and repeatably.

The control framework presented in the AMBF simulation is applicable to more robots than the Galen Robot. This MATLAB interface can be extended in use for any serial robot or other AMBF robots. The controller dynamics could easily be updated, the stability and transparency maps can similarly be calculated with these tools, and the admittance gains can simply be selected. This will make applying admittance controllers to any form of cobot relativley simple.

While the controller has been verified in contact in the simulation, it still would be nice to have seen this fully implemented on the real hardware. The efforts to do so are still on going, and the results have yet to be seen. We do anticipate the implementation of the controller to be promising as we have observed in the Z axis control on the Galen robot. With the full on board admittance controller repeatable testing of the real hardware with the old controller and our new controller will allow us to capture results that show a measurable improvement.

## 6.2 Next Steps

Several next steps have been discussed throughout this report and will be summarized here again.

A few more steps are necessary to ensure that the AMBF simulation controls well for the robot. It has been shown to work in simulated contact, but it would be very interesting to study the performance of the robot in the AMBF simulation with different forms of contact. If AMBF features tools to control the stiffness, penetration depth, and angle of contact, we could further investigate how the robot performs in contact, that may better predict how it would respond to contact in the real world. This contact could also be done in more rigorous ways by moving the robot in several DOF at different rates and impedance values.

If we are able to more accurately see how the robot interacts with a simulated environment, there are ways we could use the simulated objects in AMBF to create virtual fixtures. We could ‘twin’ the two control systems, to possibly use this controller as a virtual fixture. With objects in contact in AMBF, we could run the real Galen Robot in parallel and observe when they both reach certain positions and velocities. If there is contact in AMBF, we could impose a virtual contact on the real system that would stop the robot and push it back.

This, of course, requires us to first have the controller also working on the actual Galen robot. The next steps in scope of the project completion before summer 2023 are to have the real Galen robot controller running and performing similarly to the AMBF simulation. In turn we should be able to select admittance gains that improve the transparency and stability of the Galen robot.

With the Galen robot controller verified and the other aspects of the simulation environment running as we would like, we will consider performing a user study to asses the qualitative response of the robot. We will have ENT surgeons operate the robot with the previously implemented Galen robot controller and also operate it with our new admittance controller. We will design a series of tests consisting of repeatable and

measurable tasks where we will measure their performance. We will also ask for their feedback on the system.

In conclusion, there is still much work to be done to get the controller working on the real robot. This work will be continued. All the design documents, reports, and software are present on the course website and available for my mentors through the LCSR. Any code written specifically for control of the Galen Robot is on the computer in the MOCK OR. I will continue to work on the project until the beginning of summer as well to complete the controller implementation and publish the work we have done in admittance control development.

# 7 Readings & References


[1] Feng, A. L., Razavi, C. R., Lakshminarayanan, P., Ashai, Z., Olds, K., Balicki, M., Gooi, Z., Day, A. T., Taylor, R. H., & Richmon, J. D. (2017). The Robotic Ent Microsurgery System: A novel robotic platform for Microvascular Surgery. *The Laryngoscope*, *127*(11), 2495–2500. https://doi.org/10.1002/lary.26667

[2] *Democratizing Robotic Surgery and Microsurgery*. Galen Robotics. (2022, November 15). Retrieved February 13, 2023, from https://www.galenrobotics.com/

[3] Alonso, V., & de la Puente, P. (2018). System transparency in shared autonomy: A mini review. *Frontiers in Neurorobotics*, *12*. https://doi.org/10.3389/fnbot.2018.00083

[4] Huang, S. H. (2019). Optimizing for Robot Transparency. *UC Berkeley*. ProQuest ID: Huang_berkeley_0028E_19188. Merritt ID: ark:/13030/m5mq08z2. Retrieved from https://escholarship.org/uc/item/5q20h9cs

[5] Keemink AQ, van der Kooij H, Stienen AH. Admittance control for physical human–robot interaction. The International Journal of Robotics Research. 2018;37(11):1421-1444. doi:10.1177/0278364918768950

[6] Aydin, Y., Sirintuna, D., & Basdogan, C. (2020). Towards collaborative drilling with a Cobot using admittance controller. *Transactions of the Institute of Measurement and Control*, *43*(8), 1760–1773. https://doi.org/10.1177/0142331220934643

[7] Munawar, A., Wang, Y., Gondokaryono, R., & Fischer, G. S. (2019). A real-time dynamic simulator and an associated front-end representation format for simulating complex robots and environments. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. https://doi.org/10.1109/iros40897.2019.8968568

[8] Munawar, A., Wu, J. Y., Fischer, G. S., Taylor, R. H., & Kazanzides, P. (2022). Open simulation environment for learning and practice of robot-assisted surgical suturing. *IEEE Robotics and Automation Letters*, *7*(2), 3843–3850. https://doi.org/10.1109/lra.2022.3146900

[9] Varier, V. M., Rajamani, D. K., Tavakkolmoghaddam, F., Munawar, A., & Fischer, G. S. (2022). AMBF-RL: A real-time simulation based Reinforcement Learning Toolkit for Medical Robotics. *2022 International Symposium on Medical Robotics (ISMR)*. https://doi.org/10.1109/ismr48347.2022.9807609
10.1109/ICORR.2017.8009341.

[10] JHU Laboratory for Computational Sensing and Robotics. GitHub. (n.d.). https://github.com/jhu-lcsr

[11] Williams, R. L. (2016). The Delta Parallel Robot: Kinematics Solutions. Mechanical Engineering: Ohio University, 1–46.

# 8 Appendix A

Table 4. The test range values of mad and bad for the stability and transparency analysis for each DOF

| DOF | Test Case | Range of $m_{ad}$ | Range of $b_{ad}$ | mh (kg) | bh (Ns/m) | kh+ke (N/m) |
|---|---|---|---|---|---|---|
| X | Upper 1 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 41 | 17000 |
| X | Upper 2 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 0 | 17000 |
| X | Upper 3 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 41 | 17000 |
| X | Upper 4 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 0 | 17000 |
| X | Lower 1 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 41 | 401 |
| X | Lower 2 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 0 | 401 |
| X | Lower 3 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 41 | 401 |
| X | Lower 4 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 0 | 401 |
| Y | Upper 1 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 41 | 17000 |
| Y | Upper 2 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 0 | 17000 |
| Y | Upper 3 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 41 | 17000 |
| Y | Upper 4 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 0 | 17000 |
| Y | Lower 1 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 41 | 401 |
| Y | Lower 2 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 0 | 401 |
| Y | Lower 3 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 41 | 401 |
| Y | Lower 4 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 0 | 401 |
| Z | Upper 1 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 41 | 17000 |
| Z | Upper 2 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 0 | 17000 |
| Z | Upper 3 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 41 | 17000 |
| Z | Upper 4 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 0 | 17000 |
| Z | Lower 1 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 41 | 401 |
| Z | Lower 2 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 0 | 401 |
| Z | Lower 3 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 41 | 401 |
| Z | Lower 4 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 0 | 401 |
| Wx | Upper 1 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<200 Ns/m | 5 | 41 | 17000 |
| Wx | Upper 2 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<200 Ns/m | 5 | 0 | 17000 |
| Wx | Upper 3 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<200 Ns/m | 0 | 41 | 17000 |
| Wx | Upper 4 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<200 Ns/m | 0 | 0 | 17000 |
| Wx | Lower 1 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 41 | 401 |
| Wx | Lower 2 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 0 | 401 |
| Wx | Lower 3 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 41 | 401 |
| Wx | Lower 4 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 0 | 401 |
| Wy | Upper 1 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 41 | 17000 |
| Wy | Upper 2 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 5 | 0 | 17000 |
| Wy | Upper 3 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 41 | 17000 |
| Wy | Upper 4 | 0<$m_{ad}$<200 kg | 0<$m_{ad}$<350 Ns/m | 0 | 0 | 17000 |
| Wy | Lower 1 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 41 | 401 |
| Wy | Lower 2 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 5 | 0 | 401 |
| Wy | Lower 3 | 0<$m_{ad}$<25 kg | 0<$m_{ad}$<10 Ns/m | 0 | 41 | 401 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Wy | Lower 4 | $0<m_{ad}<25$ kg | $0<m_{ad}<10$ Ns/m | 0 | 0 | 401 |