Project Proposal

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

Brevin Banks

Table of Contents:

1. Introduction

The novel ENT Microsurgery System from Galen Robotics, which will be addressed further here as the Galen Robot, is a 6 DOF robot that features a parallel platform or delta stage, roll motion joint, a tilt motion joint, and un-actuated tool rotation 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 or similarly implemented environment constraints 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. However, while functional, the control system on the present prototype faces certain stability issues and has not been well optimized for transparency.

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

Transparency in this case 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 or ergonomically [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 the user. Stability is also a cause for concern. When optimizing for transparency, a decrease in stability is almost a direct trade off [5]. This will be addressed in greater detail in the Technical Approach section. On top of the robot feeling heavy due to the lack of transparency, when the current Galen system comes in contact with the environment (e.g., a table) the robot begins to jitter. Similar cases of jittering can be found in niche orientations of the robot with the heavier drill tool attached to the end effector. These current issues highlight the need for a new control system onboard the Galen Robot that co-optimizes transparency and stability.

Therefore, the proposed outcome of this project is a new functional frequency domain-based admittance controller that optimizes the transparency of the current Galen robot system while maintaining system stability. The reasons for this approach with be described in the Technical Approach section.

2. Significance & Clinical Motivation

It almost goes without saying that a robot designed to be used in the operating room should have a rather robust level of stability. All robots that interact with humans are expected to have optimized stability to ensure the safety of not only the patient, but also the surgeon [3, 4]. The Galen Robot itself has a much needed purpose in ENT surgery and in other potential microsurgical areas, and as such improving the performance and reliability of the system obviously would prove beneficial.

On top of stability and safety, optimizing the level of transparency experienced by the surgeon for the robot would elevate the practical use of this robot in surgery. If the movements are clean, quick, and do not cause the operator to strain, surgeries could be performed quicker, more accurately, and the operator will experience less exhaustion overall [1, 2].

Another application of this project comes as a biproduct of having a robust control system optimized for both stability and transparency. The framework will be designed in such a way that virtual fixtures can be easily integrated into the system which can speed up the process or even allow certain activities to be performed autonomously dynamically expanding the use cases of the robot.

3. Project Deliverables

There are 2 minimum deliverables.

The first is a stable frequency domain admittance controller in MATLAB SIMULINK for the Galen surgical robot and the integration of that controller in the AMBF simulation environment. The second is an optimized package for transparency and stability of the controller that is transferable to hardware.

There are 2 expected deliverables.

The first is the application of the control system package to the actual low-level Galen controller that is stable and optimized for transparency including virtual fixtures. The second is a set of data outlining verified objective improvement in transparency according to specified metrics for transparency and stability.

There is a single maximum deliverable.

A completed user study to investigate subjective transparency preference between the current Galen Robot controller and the newly developed controller.

4. Technical Approach

a. Controller Design

To understand the goal and approach of this project, it is important to understand what stability, transparency, and admittance controllers are. 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, Farm(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 (see figure 2). 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 a controller, several factors can be observed and tuned to ensure that at no point in operation of your robot will the controller go unstable. Stability can be observed quantitatively and qualitatively through analysis of things such as bode plots' relative phase and gain margins, left hand vs right hand system poles, and the difference between the number of poles and loops of a Nyquist plot. When a control designer simply tunes system gains, filters, integrators, etc. they can simulate the effects in something such as MATLAB SIMULINK which allows them to quickly see how their changes affect the system performance and stability.

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. Transparency does not have an accompanying concrete mathematical definition nor universally agreed upon optimality preferences. However, because transparency is reflected in system stability and the lack of impedance from the robot during performance, observing impedance in hand with stability may imply the relative transparency of the system [5, 6]. Therefore we will make the assumption 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 as shown below in figure 2 [5, 6].

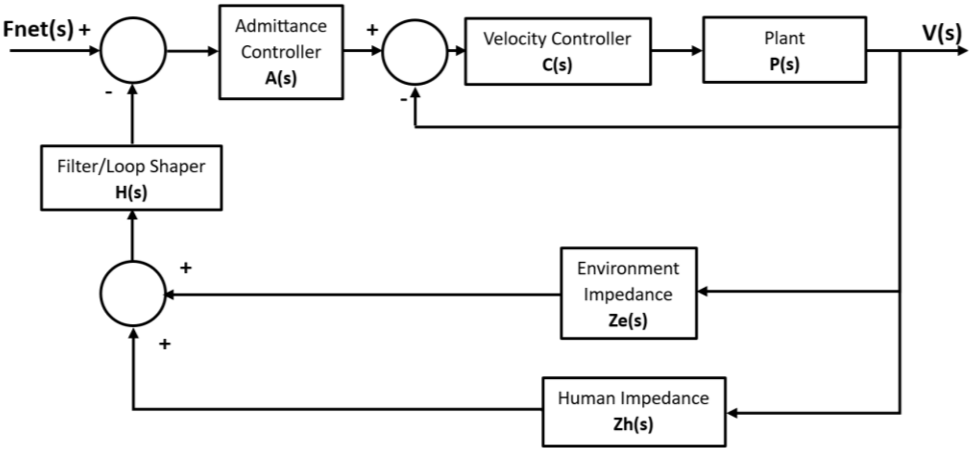


Figure 2 Basic approach for the frequency-based admittance controller design

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 are compensated for, 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 be some level acceptable resistance felt by the operator. All in all, particular care must be taken when tuning the gain of A(s). The goal would be to minimize the impedance and conversely maximize admittance.

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 litterature, 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. See the work of Aydin et.al. (2020) for the framework of this impedance approach [6].

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, once the controller has been verified to work in simulation with a basic mass-spring-damper plant for velocity control (i.e. $P(s) = \frac{ms^2}{ms^2+bs+k}$) we will perform some type of dynamics identification either with linear-time-invariant (LTI) frequency based identification or other acceptable linear approximations [4, 6]. With these linear approximations it will be verified in a sister project that checks on the lower-level Galen controller are already in place to limit inputs and poses of the robot that would allow a linear system to go unstable.

Another detail to note about the control architecture is the filter/loop-shaper block, H(s), which will be used in fine tuning of system poles, phase and gain margins, and filtering if necessary. Ideally, blocks A(s) and H(s) would be the two most critical blocks for tuning regarding the stability and impedance of the system, with the external impedances Zh(s) and Ze(s) present to simply give us an understanding of the affects of interactions on the system. The current control scheme is based in the continuous frequency domain intentionally given that the current Galen controller has latency <10 ms and is assumed to have a fast enough response that a simple discretization of the system post simulation will not significantly affect performance (though this will still be observed).

Lastly, the above controller design in figure 2 will be designed and built in MATLAB and MATLAB SIMULINK, and virtual optimization of the controller for transparency and stability will be performed there. Once the controller has been verified to perform adequately in MATLAB simulation, it will be used in conjunction with the Asynchronous Multi-Body Framework (AMBF) environment [7]. The AMBF environment enables the user to see how a virtual robot can interact with the environment and with the implementation of different controllers a virtual representation of the robot performance will be shown [7 ,8, 9]. Upon tuning the controller in MATLAB and AMBF the controller will be packaged and structured to accept the inputs and outputs of the real Galen Robot hardware. Unfortunately, these inputs and outputs are not yet listed as the Galen Robot hardware integration interface is not yet available. The team is still waiting for Anton Deguet and Adnan Munwar to finish the pipeline.

b. Dynamics Update

A sub task of the controller design is updating the dynamics it will be tested on. For the AMBF simulation it is critical to have a fair model of the Galen Robot available to accurately measure the performance of the controller [8, 9]. Currently a Galen Robot model is available, but the virtual dynamics are not correct. As such the dynamics of the robot need to determined using basic robotics dynamics based on the robot structure and joint type. The forward and inverse kinematics will be determined and a simple low-level controller of the form in figure 3 will be constructed to allow for velocity control of the joint space [6]. The methods for determining the plant dynamics in paragraph 5 of controller design may be used here in this plant if the basic dynamics model does not give acceptable performance.

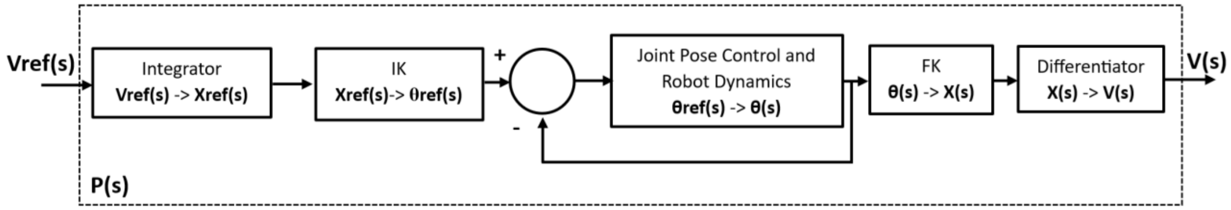


Figure 3 Low level control for the plant containing the dynamics of the actual Galen Robot with velocity control [6].

Note this is a velocity controller, similar to the current controller for the Galen Robot, so it is necessary to integrate the input and use inverse kinematics of the position to get joint angles if the control of the robot should use joint space style control commands on the dynamics. The output will be joint angles that will need to be converted to task space with forward kinematics and differentiated to retrieve velocity.

After the dynamics and controller work synergistically, it will be possible to implement virtual fixtures in the control scheme. Methods for the implementation of these have not yet been discussed by the team, but many acceptable and simple methods exist that are reliable for linear models that can be quickly tested.

c. Low Level Galen Integration

If the hardware and software interfaces are available for integration, then the controller package will be uploaded and the anticipated transparency and stability of the controller will be observed. Testing on the real-world controller would ideally be intermittently and not only after the simulation is fully complete. Due to the time constraints of the project and the nature for the need of the simulated design to match up with the real-world interface design constant testing will be critical. It is expected that there will be much to refine in the controller due to the nature of real-world dynamics, so the team will return and update A(s), H(s), or other necessary metrics until real world stability and transparency are achieved. A simplified design of the entire technical approach leading to the integration on the Galen Robot is outlined in figure 4.

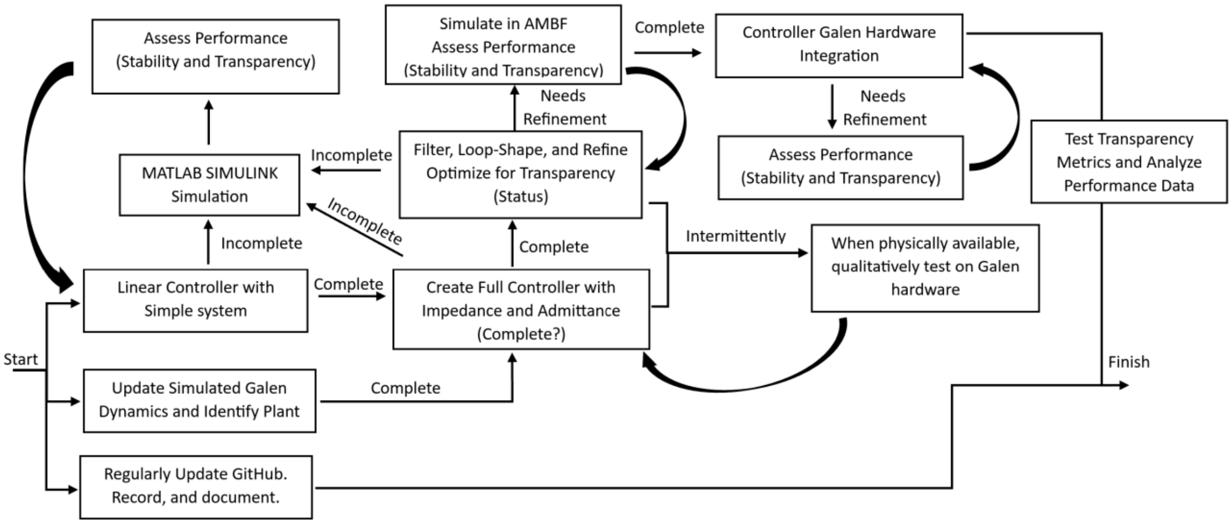


Figure 4 Technical Approach to complete integration on Galen Robot

The code for simulation will be adequately tested in MATLAB and AMBF and intermittently the controller will pushed to the real hardware if available. When completed in simulation, full efforts with the controller will be on the hardware. Simultaneously, necessary dynamics in simulation will be updated as time permits, and documentation for all code updates, software changes, and project changes will be generated and organized. The controller package on the physical hardware will be deemed completed for this project when it is A) proven stable on the physical hardware. B) a clear measurable metric of transparency is obtained and observed on the physical hardware. C) an example of a virtual fixture has been implemented and functions as expected. D) the stability and transparency perform better than the current controller does in its presently unstable conditions. (See testing plan for more details on these completion metrics)

d. User Study

Should time allow, a user study will be carried out where local surgeons (preferably ENT surgeons and those involved in microsurgery) will be recruited to qualitatively report and controller preference. An IRB will be applied for and obtained. Several tests will be designed that have the surgeons run through basic performance tasks (e.g. pick up a small object with a tool on the end effector and place it somewhere, trace a particular path with the robot tip following a virtual fixture) [10]. The surgeon would be running these tasks on the original Galen Robot control system and our new control system. They would then be given surveys on which system they preferred according to several different metrics (relative resistance, ease of use, which felt more comfortable/trustworthy).

This user study would allow future work to reanalyze the effectiveness of a frequency-based admittance controller design and open way to potentially more applications of this control scheme or refinement of the current approach [6, 10].

5. Testing Plan

Testing will be done in 3 ways, each with goal to verify the control works stably and transparently and also provide feedback on how it can be utilized or improved.

The first is a direct assessment of stability. To observe stability the transfer function of the control system will be extracted. Using this for the closed loop case the Root Locus can be obtained. If all the poles of the system have a negative real part that is significantly far away from the origin (i.e. <0.01) then it can be assumed that the system is stable. We can also verify that the open loop control system of C(s)G(s) is stable though the bode plots. Looking at the magnitude and phase for the transfer function we can observe that the phase margin is within 180 degrees and that the gain margin is proportionally positive or large. These qualitatively infer stability, and further analysis can and will be done with Nyquist plots. Examples of bode and root locus plots can be seen in figure 5 [6].

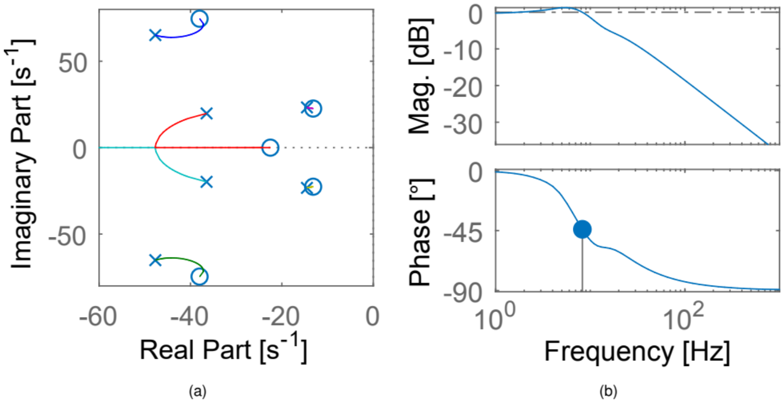


Figure 5 Examples root locus and bode plots from Aydin et.al.(2020). The poles in (a) are all negative in the real part. The bode plot shows a slight positive gain margin, with a phase margin of 135. This example system is stable [6]

The second method for testing our controller will be with the design of a stability map. A stability map considers the bounds of the impedance from the human and environment and their effects on the system before it goes unstable. As we test the controller with different Ze(s) and Zh(s) values we will be able to create a map similar to the one in figure 6 that will relate when the controller will be stable or unstable for the given system [6].

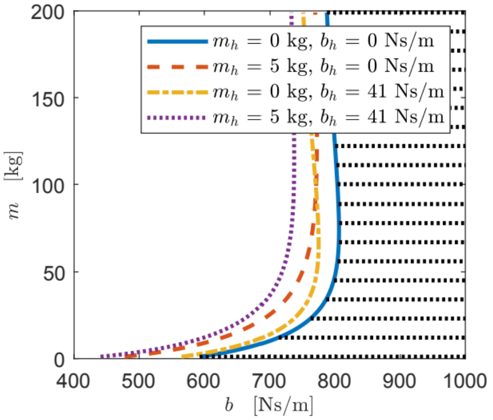


Figure 6 Stability map for Aydin et.al. (2020) pHRI system for given arm and environment impedance metrics. The curves are different boundary combinations of values for Ze(s)+Zh(s). Shaded region shows ensured stability combinations for the bounds of Ze(s) +Zh(s) [6]

Throughout controller development, several maps like this one will be made to show the gradual progression of stability and transparency for the controller.

The final test method will be the user study and the qualitative results of the controller performance that will aid in the direction this control method develops further in future projects.

6. Dependencies

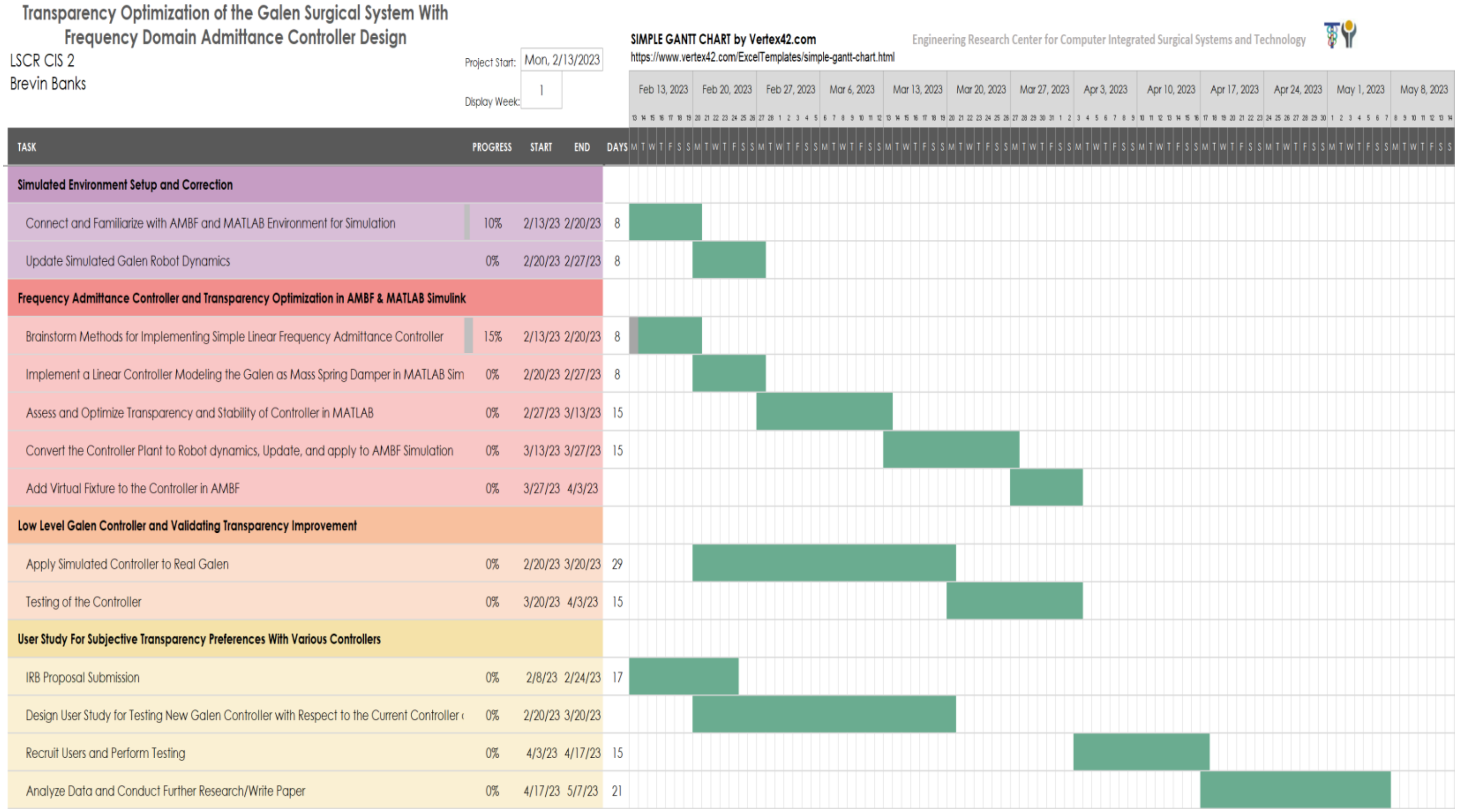
Key project dependencies needed to design, create, and implement the controller. Currently the Galen Robot Hardware access and interface is of concern as mentioned in the Technical Approach 4.a final paragraph. The IRB approval is also currently in the works as the application is in development.

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	License	Use Lab Computer with	8-Feb	13-Feb	Completed

MATLAB SIMULINK		preinstalled software			
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	Waiting
IRB for user study	IRB Approval	Go without the User Study	(Submission 20-Feb) 3-Apr	(Submission 24-Feb) 17-Apr	Waiting

7. Timeline & Tasks

The following Gantt chart organizes the tasks into groups and helps visualize when they are anticipated to be in progress. Tasks are separated into 4 groups (separated by colors) and each group represents different project deliverables. Descriptions of the tasks and details are listed after the chart.



Transparency Optimization of the Galen Surgical System With Frequency Domain Admittance Controller Design

LSCR CIS 2
Brevin Banks

Project Start: Mon, 2/13/2023
Display Week: 1

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

Engineering Research Center for Computer Integrated Surgical Systems and Technology

TASK	PROGRESS	START	END	DAYS
Simulated Environment Setup and Correction				
Connect and Familiarize with AMBF and MATLAB Environment for Simulation	10%	2/13/23	2/20/23	8
Update Simulated Galen Robot Dynamics	0%	2/20/23	2/27/23	8
Frequency Admittance Controller and Transparency Optimization in AMBF & MATLAB Simulink				
Brainstorm Methods for Implementing Simple Linear Frequency Admittance Controller	15%	2/13/23	2/20/23	8
Implement a Linear Controller Modeling the Galen as Mass Spring Damper in MATLAB Sim	0%	2/20/23	2/27/23	8
Assess and Optimize Transparency and Stability of Controller in MATLAB	0%	2/27/23	3/13/23	15
Convert the Controller Plant to Robot dynamics, Update, and apply to AMBF Simulation	0%	3/13/23	3/27/23	15
Add Virtual Fixture to the Controller in AMBF	0%	3/27/23	4/3/23	
Low Level Galen Controller and Validating Transparency Improvement				
Apply Simulated Controller to Real Galen	0%	2/20/23	3/20/23	29
Testing of the Controller	0%	3/20/23	4/3/23	15
User Study For Subjective Transparency Preferences With Various Controllers				
IRB Proposal Submission	0%	2/8/23	2/24/23	17
Design User Study for Testing New Galen Controller with Respect to the Current Controller (	0%	2/20/23	3/20/23	
Recruit Users and Perform Testing	0%	4/3/23	4/17/23	15
Analyze Data and Conduct Further Research/Write Paper	0%	4/17/23	5/7/23	21

Figure 7 Group 1: Purple, Group 2: Red, Group 3: Orange, Group 4: Yellow [11]

Group 1:

- Task 1.1) (2/13 - 2/20) Connect and Familiarize with AMBF and MATLAB Environment for Simulation
  - Install Software, Load in Current Galen Robot
    - Task 1.2) (2/20 - 2/27) Update Simulated Galen Robot Dynamics
      - Observe Real World Dynamics and Add More Realistic Model to Simulation

Group 2:

- Task 2.1) (2/13 – 2/20) Brainstorm Methods for Implementing Simple Admittance Controller
  - Develop Basic Idea of Key Attributes Required for Controller Such as Control Diagram and Key Concepts
    - Task 2.2) (2/20 – 2/27) Implement a Linear Controller in MATLAB Simulink Simulation
      - MATLAB Simulink Simulations for Control Systems using a Primitive Linear Model
        - Task 2.3) (2/27 – 3/13) Assess and Optimize Transparency and Stability of Controller
          - Explore and Adjust Admittance, Frequency Filter, Feed Forward, Nyquist Plot and Frequency Plots
            - Task 2.4) (3/13 – 3/27) Convert the Controller Plant and apply to AMBF Simulation
              - Change the Dynamic Model to be More Realistic of Robot, Perform Simulations in AMBF
                - Task 2.5) (3/27 – 4/3) Add Virtual Fixture to the Controller in AMBF
                  - Apply Virtual Constraints or a Nonlinear Approach to the Control Scheme

Group 3: **(Tasks 3.1 on going during all of tasks in group 2)**

- Task 3.1) (2/20 – 3/20) Apply Simulated Controller to Real Galen Hardware
  - Upload and Update Simulated Controller to be compatible and Safe with Real Hardware
    - Task 3.3) (3/20 – 4/3) Testing of the Controller
      - Debug Issues with the Controller on Hardware, Evaluate the Performance of the Controller on the Galen

Group 4:

- Task 4.1) (2/20 – 3/20) Design User Study Controller and Obtain IRB Approval
  - Design Tests, Data to collect, Number of Necessary Users
    - Task 4.2) (4/3 – 4/17) Recruit Users and Perform Testing
      - Advertise and Find People to Assess Controller Performance
        - Task 4.3) (4/17 – 5/7) Analyze Data and Conduct Further Research/Write Paper
          - Evaluate Collected Data, Significance of Work, and Write-Up Work.

8. Key Milestones

Key milestones are the deliverables associated with the completion of each group from section 7 Timeline & Tasks.
The first key milestone is the completion of Group 1 by 2-27. The associated deliverable is a virtual Galen Robot in a working AMBF environment.

The second key milestone is the completion of Group 2 before 4-3. The associated deliverable is a working control system in AMBF and MATLAB that is transferable to hardware. Group 3 can not fully be completed until Groups 1 and 2 have been completed.

The third key milestone is the completion of Group 3 by 4-3 should Groups 1 and 2 follow through before then. The associated deliverable is a tested and working controller on the physical hardware and data to reinforce that the transparency is verifiably better.

The fourth and final key milestone is the completion of the user study in Group 4 by 5-7. This most likely will not realistically be performed this semester given the work needed to prepare the study and complete the controller, but should group 3 be completed, early work into the summer could consist of the user study. With the user study completed, the associated deliverable would be an academic journal.

9. Team

Student:
Brevin
Roles: Admittance Controller Design, Simulation, Hardware Integration
Mentors:
Ugur Tumerdem, Ph.D.: Primary Lead, Controls Lead
Manish Sahu, Ph.D.: Coding Consultant
Adnan Munawar, Ph.D.: AMBF and Galen Lead
Mohammad Salehizadeh, Ph.D. Galen Consultant
Russell Taylor Ph.D.: General Consultant

10. Management Plan

There are regularly scheduled weekly meetings with Ugur and others who wish to join on Thursdays at 4pm in Hackerman 306.
Brevin will work daily on the project deliverables, regularly updating the wiki and creating documentation on activities.
Weekly correspondence is planned with the current week's deliverables and updates delivered before the weekend by email to the mentor and other necessary updates.
Other optional weekly meetings for Galen Robot Controller team are available but the time is TBD.
All code files and deliverables are to be uploaded to GitHub and the project wiki. This includes MATLAB, C++, Word, Excel, PowerPoint, and PDF files
Code files are regularly pushed and updated to GitHub repository.

11. 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] J. Fong, V. Crocher, Y. Tan, D. Oetomo and I. Mareels, "EMU: A transparent 3D robotic manipulandum for upper-limb rehabilitation," *2017 International Conference on Rehabilitation Robotics (ICORR)*, London, UK, 2017, pp. 771-776, doi: 10.1109/ICORR.2017.8009341.

[11] *Simple gantt chart*. Microsoft 365 Templates. (2022, August 13). Retrieved February 13, 2023, from https://templates.office.com/en-us/simple-gantt-chart-tm16400962