TOGAC6_CD_ControlSystemsDesign.docx

Contents

1. Description 1
2. Requirements for the design 2
  Minimum: (2/13/2023 - 4/3/2023) 2
  Expected: (2/20/2023 - 4/3/2023) 2
  Maximum: (4/3/2023 - 5/7/2023) 2
3. Design Specifications 3
4. Control Diagram 1 4
5. System Identification: 7
6. Control Diagram 2 8
7. Project Organization: 10

1. Description

This Control Design document contains the design specification and approach for our transparency optimized admittance controller for the Galen robot system. Here is listed the **requirements for the design**, the **design specifications**, the **control diagrams**, and the key elements required to produce such a controller.

The diagrams represented here mirror the work of Aydin et al. (2020) from the paper Towards Collaborative Drilling with a Cobot Using Admittance Controller and Stienen et al. (2018) Admittance control for physical human–robot interaction, in that they explicitly show how to design an admittance controller for a cobot. The Galen robot uses the hand over hand interaction system as the robots shown in these papers, and the process in their methods sections for design an admittance controller is nearly identical to how we want to implement our controller. The key differences between our approach and the approach in these papers, however, is that we will perform system identification differently for the Galen robot surgical system itself. We also will be further investigating the more robust stability of the actual system on the real hardware including the investigation of the closed loop pole locations and with disturbance and noise interactions.

2. Requirements for the design

A reminder of the project deliverables:

Minimum: (2/13/2023 - 4/3/2023)

- A stable frequency domain admittance controller in MATLAB code for the Galen surgical robot in the AMBF simulation environment
- 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: **(2/20/2023 - 4/3/2023)**

- An applied control system package to the low-level Galen controller that is stable and optimized for transparency including virtual fixtures.
- A validated objective improvement in transparency according to specified metrics

Maximum: **(4/3/2023 - 5/7/2023)**

- A completed user study to investigate subjective transparency preference with different controllers on the Galen robot

The following are project requirements following the expected deliverables:

1. The controller design is stable in the simulation of AMBF.
2. The controller design has low admittance values.
3. The controller design can handle bound human and environment impedances.
4. The controller design uses velocity control.
5. The controller design has a map for a range of acceptable m_ad and b_ad values.
6. The controller design allows the implementation of virtual fixtures.
7. The controller design has a measurable system for evaluating the transparency.
8. The MATLAB code is transferable and complete.

9. The real controller on the robot is stable.
10. The robot nonlinear system has been identified correctly.
11. The robot system code is transferable and complete.
12. The robot system can be set up and used in a qualitative user study.

3. Design Specifications

1. The Galen robot controller will be controlled at 200Hz.
2. The admittance mass and admittance damping should be as small as possible while maintaining stability.
3. The controller should utilize velocity control commands.
4. The controller should move control all 5 joints.
5. The controller should not crash or move past its joint limits.
6. Control designs should be implemented in MATLAB.

4. Control Diagram 1

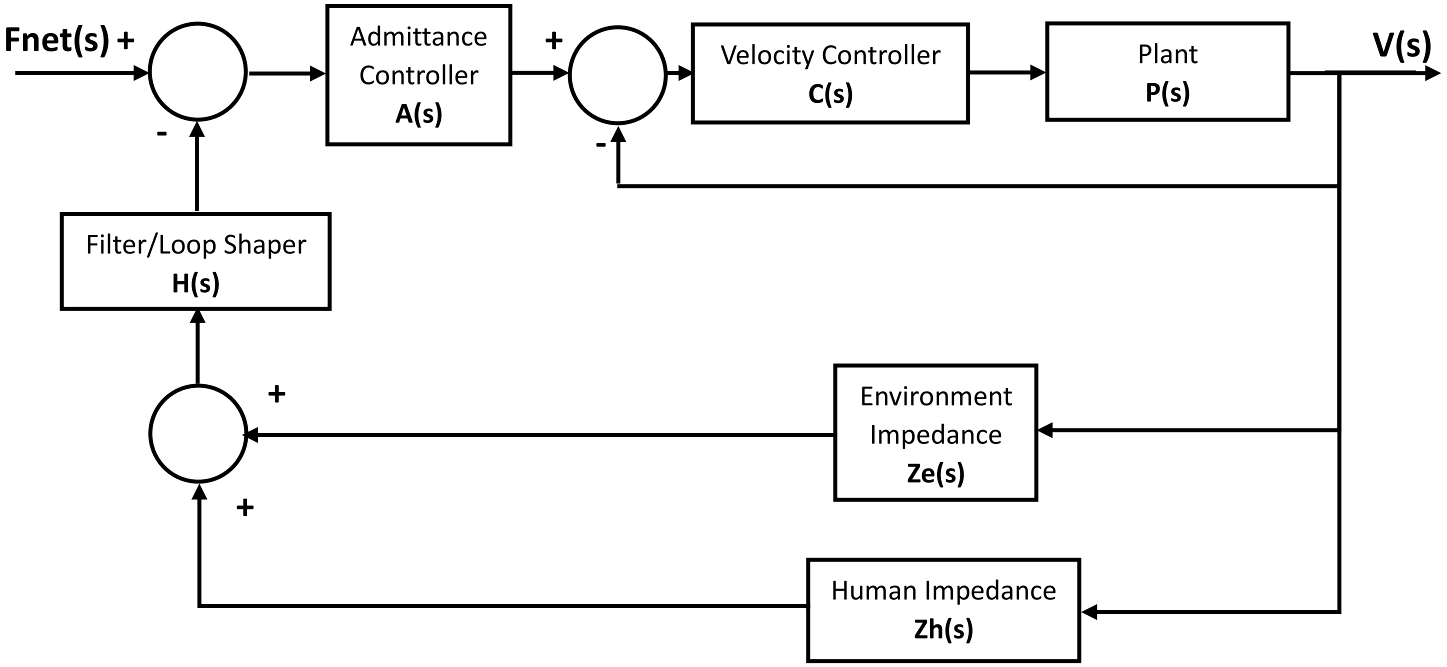

Figure 1. The basic scheme for the design of velocity control for frequency-based admittance design for a controller

The above diagram represents the basic form the feedback controller for the admittance control design should take in its barest form.

The controller contains 6 key blocks that represent the interactions expected from robot use. These will be listed and explained here:

- A(s) – this is the actual admittance portion of the controller and is tunable in that it contains a desired mass and damping that will ‘pseudo’ super impose a mass and damping over the top of the plant P(s) mass and damping. In other words, the mass and damping of the actual plant will feel replaced by a virtual mass and damping assumed by the controller A(s). The operator of the robot holding a tool at the end effector will theoretically feel the mass of the robot and damping of the robot to be at the assumed values put in the A(s) block. The values placed in this block will be chosen such that they keep the robot controller in a safe and stable control band, but also allowing the robot to feel as light and least impeding to the user. It should be taken with caution that lowering the desired mass, m_ad, and desired damping, b_ad, of the system will drive the system closer to instability, since the inertia and damping of the actual system help the system from moving towards unsteady and uncontrollable outputs. Therefore, fine tuning of and investigations of acceptable for m_ad and b_ad values will be a large portion of the work endeavors involved in this project.
- C(s) – this is the PID controller. In our use case, we are attempting to apply a velocity controller to the Galen robot. In this case we only need to at minimum include a proportional, kp, and derivative, kd, term in our control. We can add an integrator, but our initial designs will be made without this integrator so achieve the lowest possible admittance parameters while maintaining stability. Adding the integrator has been shown in the work of Stienen et al. (2018) to remove steady state error but lowers the window of stability. In the case of the real system, depending on the achievable deliverables for the project by May 5th 2023, we will not have access to the low level design of the controller for C(s). Instead we may only have the lumped controller design with the plant P(s) through the system identification process we will perform. Therefore, C(s) and P(s) will become just one singular G(s) = C(s)P(s) that does not have control values we can tune. In this case the only tunable parameters we can play with will be the A(s) values and potentially at loop shaping block or low pass filter on the incoming impedance, H(s).
- P(s) – this is the plant design. This represents the dynamics of the Galen surgical robot system. This plant in reality is a highly nonlinear dynamic system, just as many robotic systems are, and as such it is not fair to say we can assume it is linear as the transfer functions used in the diagrams above would require. Therefore we plan to linearize the real plant through system identification and control the robot using the results of these linearized plants. More details for this process and the goals of system identification can be found in the System Identification section below. For the simulated Galen robot, it is assumed that the models in AMBF ADF file for the Galen are not equal to that of the real system. We will not assume that the simulation mimics close to what we expect to see with the real robot, but the simulation will allow us to test the concepts behind the admittance controller, and they will make the transition from simulation to real hardware seamless. The AMBF simulations will be performed through communication between ROS and MATLAB. Before the simulations are performed in AMBF though, initial testing for completeness and comprehension for the team will be performed in MATLAB and MATLAB Simulink with a simple Mass Spring Damper system with fixed parameters to observe. The work in Aydin et al. (2020) will be

recreated in MATLAB and compared to the results of a simple Mass Spring Damper system to ensure the results agree and are trustworthy for the use in designing of the real controller.

- Ze(s) and Zh(s) – these two blocks represent the environment and human impedance respectively. This means the human interaction force and environment interaction force resistance to the desired input force by the user. These are things caused by the arm stiffness, mass, damping, or the stiffness of the environment. Because of these factors the behavior of the robot can wildly change based on a near infinite, but fortunately bounded, number of dynamic human arm and environment conditions. The values used in testing Ze(s) and Zh(s) will be worst case values that have been identified and implemented in the admittance control design already in the work of Aydin et al. (2020).
- H(s) – as mentioned earlier in the description of C(s), H(s) will be an option block that will be used to improve the performance of the controller after implementation. In simulation, it may not be necessary to filter the incoming impedance as the values will be constant and sampling speeds are assumed fast enough not to introduce a large amount of error, but for the actual robot system filtering will be necessary to keep the impedance inputs bounded and clean. This block may also help in the feedback form of loop shaping in the case we need to change the behavior of the control response in anyway (adding delay, phase margin, increasing gain. etc.)
- Fnet(s) – this is the net input force from the human on the robot end effector as well as the force of whatever tool the human user is interacting with against the environment, patient, table, etc.
- V(s) – this is the output velocity of the controller that the robot is achieving.

The Values used for testing just the simulated Mass Spring Damper are:

```
m = 10; %kg System mass for MSD Ps
b = 5; %Ns/m System damping for MSD Ps
k = 1; %N/m System stiffness for MSD Ps
% the following control values were used from 2/23/2023 to 3/7/2023
P = 900; %Ns/m Proportional gain for controller Cs
D = 10; % Derivative gain for controller Cs
% the following control values were used from 3/7/2023 on for MSD
% simulation
P = 4357.72516429453; %Ns/m Proportional gain for controller Cs
D=10; % Derivative gain for controller Cs
```

For the variables used to test the AMBF simulation for the Galen Robot see the ADF file, galen.yaml file, in the Galen for ambf folder.

Other important design elements to note:

The control design above does not include the need for feedforward gravity compensation. It is assumed that identifying the system for G(s) on the real system will include this, but in design of the C(s) block it may needed. That is, if we reach the point of designing the low level controller, we will need to consider how to compensate for gravity in the real world. We would also need to identify the system in such a way that no on-board controller interferes with the output of that identification. For completeness this approach was considered and the model from Stienen et al. (2018) was followed in structure to add feedforward to the case provided here. See the following documents for further work with the feedforward design.

TOGAC17_MSD_R_StabilityFBandFFApproach.docx

TOGAC27_CD_TransferFunctionDerivAndStability.docx

5. System Identification:

Here the concept and application of system identification for the Galen robot will be explained. System identification is a method that is used to compare input values to output values for a dynamic system. Often this takes the form of approximating transfer function control values to fit the line of the output response given an input response. In our case we will estimate a linear transfer function for the nonlinear Galen system sending different types of inputs to the Galen robot and measuring the resulting output. The MATLAB system Identification toolbox has great tools that can be used to retrieve this transfer function offline with the input and output data.

In our case, we desire to control the Galen robot with velocity control. For this we will send velocity input trajectories to the Galen robot and then measure the velocity output of every joint. We will feed that input and response output to the MATLAB toolbox. The input trajectories will start with small chirp inputs on the range of frequencies expected by a human user. (Bounded approximately by 10^-2 to 10^2 Hz). The magnitude of these inputs will be small enough to keep the Galen from moving more than 5 to 10 centimeters in direction. If the fit of the identified system is poor or the control response has too much error, further work will be done to better identify the system with different input types (step, sine, etc) or different nonlinear analysis identification tools will be used.

6. Control Diagram 2

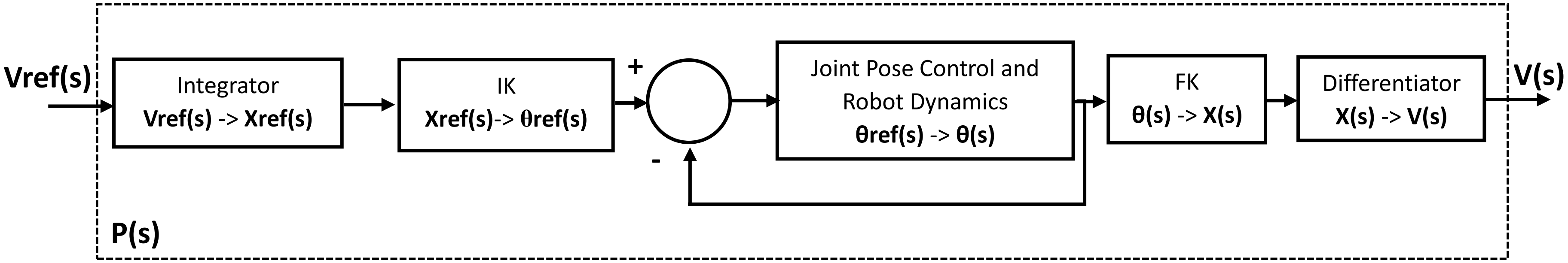

Figure 2. The basic scheme for the internal plant P(s)

Although it is unlikely the project will reach the point during this semester where we design internal workings of the controller, it is important to understand the projections of the project. Much work has been done to already implement these control schemes in python and C++ so they need not be redone (e.g. forward kinematic, Jacobean, basic move commands). But it will be necessary to understand how they work for the integration of the Admittance controller with the pre-existing framework. Most robots in the low-level code use joint commands that are executed by giving direct joint variable commands that move the joints to the desired pose through forward and inverse kinematics coupled by the robot dynamics. This is shown in figure 2.

In the above diagram there are several elements that will be discussed here but may not need to be designed within the scope of our project. Tis P(s) resembles our G(s) and includes the C(s) controller inside it.

Vref(s) – the reference velocity output by the admittance block

Integrator – Converts velocity to position commands.

IK – inverse kinematics that converts position values to robot joint commands.

Joint Pose Control and Robot Dynamics – the controller C(s) and the plant dynamics P(s)

Fk – forward kinematics that turns output joint states to positions.

Differentiator – turns the positions into an output velocity.

V(s) – the output velocity of the robot.

7. Project Organization:

The following is a rough outline of how the design and development of the controller will take place.

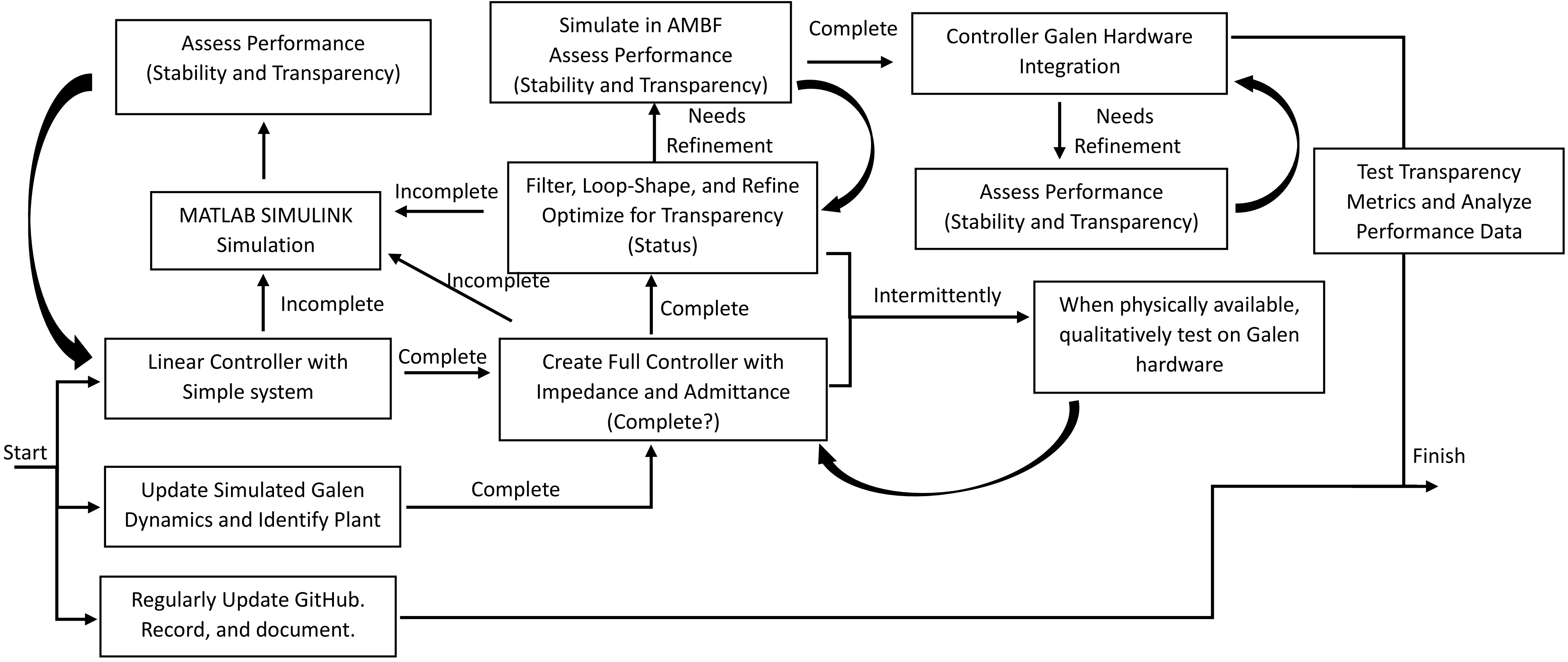

Figure 3. The flow of project work for the Galen robot admittance controller application

References

[1] 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

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