Real-Time Position Control of the Haptic Paddle

Zhubo Zhou
Adaptive Robotic Manipulation Lab
Georgia Institute of Technology
Atlanta, United States
zzhou344@gatech.edu

***Abstract* — This paper describes my research experience as an undergraduate student in Dr. Hammond's Adaptive Robotic Manipulation Lab, working with a pair of single degree-of-freedom (1DOF) force-feedback mechatronic devices, known as haptic paddles. The goal of this project is to build and program a teleoperation suite composed of two identical haptic paddles, one serves as a master, and the other serves as a slave. The master should be able to control movement of the slave, while it should also be able to react accordingly if the slave runs into something (e.g. wall). With the mechanical and electrical system being built and validated in the previous semester, the goal of this semester is to develop algorithms that can allow for real-time position control between the master and the slave. By the end of March 2020, rough position control between the master and the slave has been achieved.**

***Keywords* — master/slave, haptic device, position control, embedded programming, microcontroller**

I. INTRODUCTION

Haptics studies how human use their sense of touch to interact with the environment. Telerobotics studies how robotic devices can be remotely controlled. A combination of the two introduces an innovative way to synchronously control robotic devices while receiving force feedback. However, due to the complexity involved with human operator dynamics, designing stable and immersive haptic devices could be very difficult. Therefore, I was tasked with developing a simple 1DOF haptic workstation as a testbed to evaluate teleoperation control algorithms.

The haptic paddle is an open source 1DOF learning platform for haptic manipulation. It has been used in university curriculums as a teaching tool to provide hands-on learning for students studying system dynamics. It can also be used for research purpose to assist teleoperation algorithm studies. Fig.1 shows the haptic paddle being built in the previous semester. This version of haptic paddle is largely inspired by HapKit3.0 [1] developed by Stanford University in 2015, which is a 3-D printed haptic device. Since HapKit3.0 utilizes a customized microcontroller board with an embedded angle sensor, modifications were made to make the current device compatible with the Arduino platform. As shown in Fig.1, the device is mainly composed of three components: a paddle capable of rotational motion, a DC motor that can drive the paddle via a capstan drive, and a KMA210 angle sensor, which will be used to track angular position of the paddle.

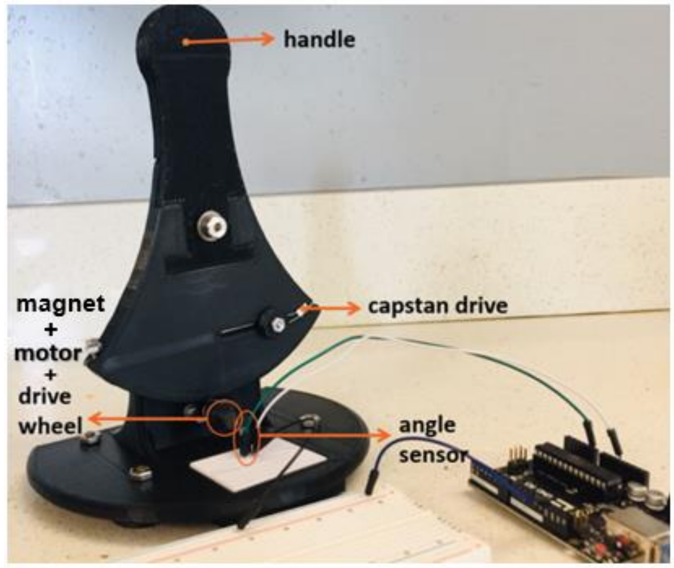

Figure 1. The Haptic Paddle

A. Signal Conditioning

In order to achieve real-time position control between the master and the slave, it is essential to track angular position of the paddle accurately. This is achieved by conditioning raw signals from the KMA210 magnetic angle sensor, which reads angular position of the magnet on the motor drive wheel. Due to limited range of the sensor, noise in the data, and the fact that the motor will go through multiple rotations every time the paddle moves from left to right, few steps need to be performed to get the desired angular position of the paddle.

First, the flip point where angle readings switch from 180 degrees to 0 degrees, or vice versa, needs to be identified, as well as direction of the movement. After that, the accumulated angle can be calculated. Next, the initial angle offset needs to be removed. Lastly, moving-average of sensor readings can be calculated to smooth out noise in the raw data. Due to the dramatic change in readings before and after the flip point, it is a good practice to calculate the moving average after accumulative angle has been obtained.

B. Position Control

In order to program the slave to mimic master's motion, first the algorithm that can allow the slave to reach a pre-specified position needs to be developed. As shown in Fig. 2, a feedback loop with a PD controller is implemented to achieve this goal. Although in theory a PID controller could allow for a faster system response and smaller steady state error, an integral controller was not adopted for that it tends to introduce more instability to the system if not properly tuned. In addition, the simulation result shows that although the PD controller cannot

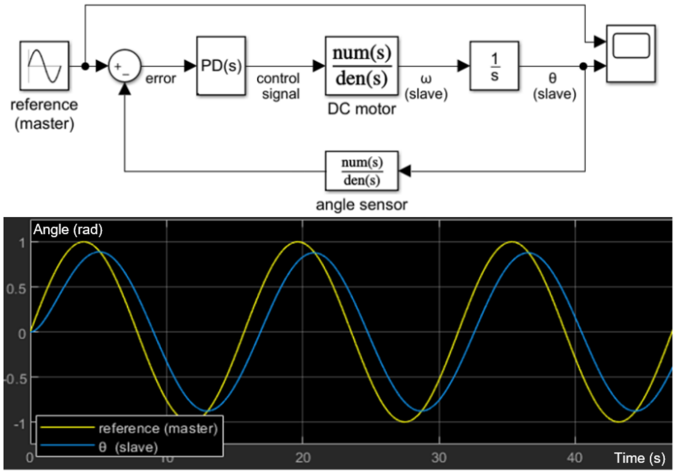

Figure 2. The Controller Schematic and Simulation Result

eliminate steady state error, it can do a relatively good job in terms of tracking the reference signal. However, in practice, if solely using a PD controller, it was found that varying gain values is not enough to stabilize the system. The system tend to oscillate about the setpoint and never want to settle down. In order to solve this problem, I set a threshold for maximum allowed steady state error, and programmed the system to turn off the motor if error (difference between the reference signal and paddle's angular position) is smaller than the threshold. This strategy solved the oscillation problem and allow the system to settle down in a reasonable amount of time, but certainly sacrificed the its steady state accuracy.

C. Synchronous Control between the Master and the Slave

The master and the slave share the same configuration but are actuated via different means. As shown in Fig.3, the master is driven by human hand, while the slave is driven by DC motor. Once the position control algorithm has been proven to work, all that left with is to replace the pre-specified fixed reference with real-time angular position of the master device. A different set of K_p, K_d and threshold value for maximum allowed error were again experimentally determined to allow the slave to move accordingly with the master with minimal discrepancies and oscillations. Starting from setting all the gains to zero, the PD controller is tuned by first increasing K_p until the response to a disturbance is steady oscillation, then increasing K_d until the oscillation goes away, and then repeat the above two steps until increasing the K_d does not stop the oscillation. Arduino's embedded serial plot module is also very useful in terms of visualizing the controller performance.

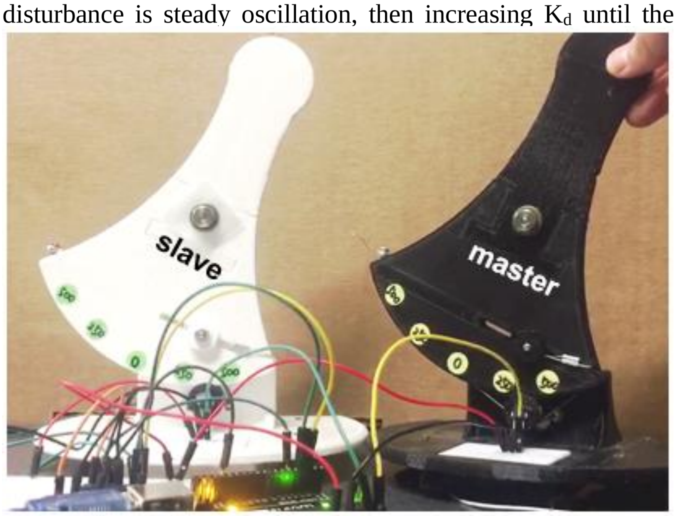

Figure 3. The teleoperation suite composed of two haptic paddles

II. RESULTS AND DISCUSSION

By the end of March 2020, rough position control between the master and the slave has been achieved. Max paddle speed that allows for decent tracking is around 10 degrees per second. Maintaining stability of the system was valued more than minimizing the steady state error.

One intrinsic limitation of the current system is that the angle tracking can easily go wrong at high speeds. If the paddle is running too fast, the controller will not have enough time to process feedback from the sensor properly (e.g. fail to detect the flip point).

This problem was alleviated by limiting maximum speed of the paddle. First, capstan drive was tightened to increase the system's impedance, which can allow for slower paddle motion with same torque input from the motor. Second, torque delivered by the DC motor was reduced by setting an upper limit for duty cycle used to control the motor, as well as reducing the input voltage. Since this part was performed after the COVID-19 outbreak, without access to power supply in the lab, a 3.3V port on Arduino microcontroller is used to power the motor at a relatively low voltage level. In the future, to provide a more stable power supply, a buck-boost converter could be used.

Aside from limiting speed of the paddle, an alternative approach could have been taken to allow for faster controller updates and therefore more accurate angle tracking at high speed is to increase sampling frequency of the angle sensor. Other potential approaches include using a DC motor with an optical encoder, and selecting a stepper motor that can move in a slower and more precise manner.

III. CONCLUSION

In general, the system performance was improved by varying gain values, allowing for larger steady state error, and reducing paddle speed. The paddle speed is reduced by tightening capstan drive and reducing motor speed through lowering supply voltage/ imposing a maximum limit on duty cycle. The system performance can be further improved by choosing a higher sampling frequency, using a more stable power supply, or selecting a different motor. Experimenting with different K_p/K_d values and adding an integral component may also improve the position tracking.

In addition, it was observed that the faster the system responds, the less stable the system tends to be. There is usually a trade-off between the system's stability and response rate.

Further development of the system will focus on increasing repeatability, adding teleoperation functionality, and integrating a load cell into the slave device so that users can feel force encountered by the slave device.

ACKNOWLEDGMENT

Thanks to Bryan Blaise for his devoted guidance. The clarity of his explanation and educational resources he provided always direct me to the correct direction and allowed me to quickly learn about this field.

REFERENCES

[1] Hapkit.stanford.edu. (2015). Hapkit. [Online] Available at: https://hapkit.stanford.edu/.