Improvements on a Langmuir Probe Diagnostics Suite

David S. Zamora

15 September 2019

Abstract

Princeton's Electric Propulsion and Plasma Dynamics Laboratory has a device that's designed to extend a Langmuir Probe into the Insert Region of a Hollow Cathode. This device operates without knowing the position of the probe tip in the insert region. This summer I designed and fabricated a closed feedback system to control the probe's displacement. We weren't able to run the system in an experiment, but other tests show the system works.

Contents

1 Introduction

1.1 Preliminaries

Hall and Ion Thrusters are a popular means of propelling small spacecraft into deep space. However, as missions become longer, it becomes important to determine the lifetime of the components used in these thrusters. One limiting factor in these is the Hollow Cathode. The hollow cathode is a tube made of a material with a low work function, like Lanthanum Hexaboride, that is heated until it starts emitting electrons. As of yet, there is no way to determine the lifetime of this component other than to operate a thruster until failure. Such an experiment would take a long time and be extremely expensive to maintain.

Hence, the Electric Propulsion and Plasma Dynamics Laboratory (EPPDyL) at Princeton University is attempting to develop a theoretical model that can determine the lifetime of a hollow cathode. In 2017, researchers at the EPPDyL developed a mathematical model to determine the attachment length of a hollow cathode [2]. The attachment length is the distance in which the plasma remains so-called "attached" to the cathode in the its Insert Region and is a measure of its lifetime. The Insert Region of the cathode is where the plasma is being generated by the emission of electrons. Though researchers have this model, they do not have the experimental data to validate it. They need a means to determine the plasma temperature, density and potential inside the Insert Region.

Langmuir probes are devices used in experimental plasma physics to measure properties of the plasma like temperature, density, and electric potential. Because they're so simple to construct and implement, they are ubiquitous in all plasma experiments. For this project, we worked with a triple Langmuir Probe, where a pair of tungsten wires is electrically biased with respect to a third tungsten wire that has a floating potential. As the potential across these three wires changes, a current is produced. The graph of current versus voltage is known as an I-V trace; it's from this graph that experimenters deduce the aforementioned properties of the plasma.

1.2 Motivation

Last year, former Princeton undergraduate student Jordan Brown designed a mechanism by which a triple Langmuir Probe could be inserted into the hollow cathode's insert region [1]. This mechanism has one major flaw: it lacks a closed feedback control loop between the position of the probe tip and the displacement of the probe by the stepper motor. In other words, there's no reliable method of determining accurately how far the probe has extended into the hollow cathode. The goal of this project is to develop a closed feedback loop that controls the displacement of the probe based on the position of the probe tip.

1.3 Brown's Diagnostics Suite

A 3D model of the relevant components of Brown's Diagnostics Suite is shown below. The probe is a foot long and the probe tip must travel the 3.43 inches that is the cathode's insert region.

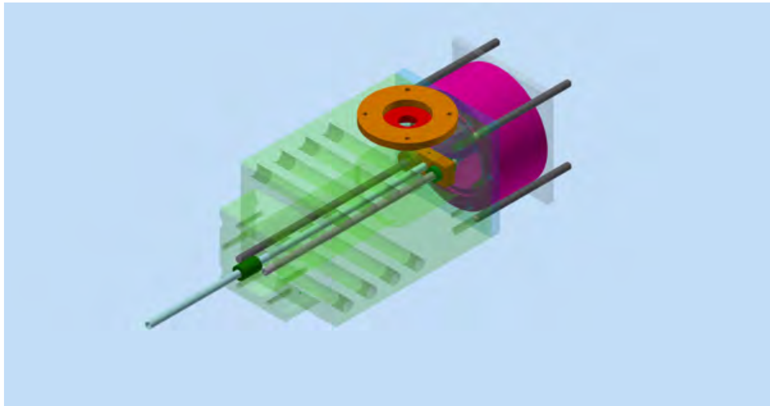

Figure 1: Brown's Diagnostics Suite

2 Electronics

In this section, I'll describe the electronics components necessary to implement our closed feedback loop and how they contribute to the system at large.

2.1 Stepper Motor Driver

The Langmuir Probe is extended into the hollow cathode by means of a stepper motor attached to a lead screw. Stepper motors need stepper motor drivers to properly function, but the Lin Engineering driver that was on Brown's Suite was defective. So to implement our feedback loop, the first thing we needed was a new stepper motor driver.

From one of the MAE Department's teaching laboratories, we acquired an A4988 Stepper Motor Driver by Pololu. We used this driver for three reasons. First, the driver can operate motors from 8-35 VDC and our motor operates at 24 VDC. Second, the driver's maximum output current is 2 A, while our motor needs 0.67 A. Lastly, the driver is easily interfaced with Arduino, a microcontroller we'll describe in Section 2.3.

2.2 RVT-K52-3 Potentiometer

As we discussed in Section 1, the main issue with Brown's suite is the inability to track the probe tip; we need a linear displacement sensor. From the same teaching laboratory, we found a linear potentiometer by Litton Industries designated RVT-K52-3. This potentiometer has a maximum travel of about 4 inches, which is enough for the insert region, and its resistance varies from 30 Ω to 4 kΩ. Litton Industries went defunct in 2001 and, unfortunately, they didn't have specifications online for this sensor nor do they continue to sell it. So, in the following paragraphs I'll describe what we discovered about this sensor and how we implemented it into the final design.

Figure 2: An image of the potentiometer we used. Note the thread on the head of the slider and the 6 wires, these will become relevant later.

The basic principle of the potentiometer is that as the slider moves, the resistance changes and there's a linear relationship between the resistance and the displacement. Using this, we can infer by Ohm's Law, $V = I \times R$, there must be a linear relationship between the voltage and the resistance so long as the current remains constant. This is significant because the Arduino microcontroller's inputs measure voltage. In other words, the relationship between resistance and displacement isn't as important as the relationship between voltage and displacement. To maintain the current constant, we used an inexpensive Constant-Current circuit designed by my graduate student supervisor, Pierre-Yves Taunay.

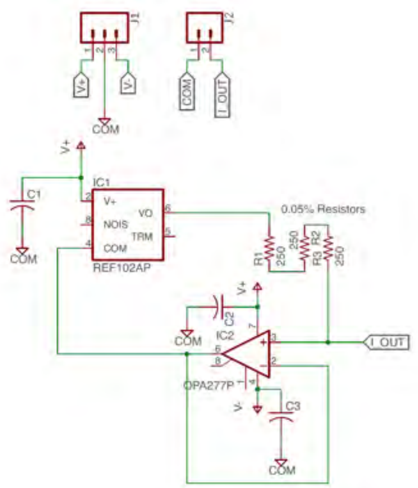

Figure 3: Constant Current circuit designed by Pierre-Yves Taunay. The output current, in our case, would be to the potentiometer.

A schematic of this circuit is shown above. There is one major difference between the circuit in Figure 3 and the final circuit. While experimenting with this circuit, we learned that if the resistance between VO and I_{OUT} is greater than the resistance of the load, the circuit would not maintain a constant current. Therefore, instead of having 750 Ω, we had 10 kΩ between VO and I_{OUT}. The minor difference is we used high-precision electronic components instead of the parts used in this schematic; the error on the final circuit is less than 1%. In conclusion, the final circuit outputs a steady 0.5 mA to the potentiometer, thus we can accurately relate voltage to displacement.

This potentiometer comes with 6 wires, so the next task was to determine what the wires meant. We determined the red, green, and yellow were connected and that the resistance between either the red and green or yellow and green changed with the slider's position. However with the red-green wiring, the resistance started at 4 kΩ while with the yellow-green wiring, the resistance started at 20 Ω. Both of these were measured for zero displacement of the slider. None of the other wires resulted in a change in voltage when the slider was displaced; only the red-green/yellow-green pairings were relevant. Between these two, we arbitrarily chose to measure the change in voltage across the red and green wires.

To calibrate the potentiometer, we connected the red wire to the load of the constant circuit and the green wire to common. Then we powered the circuit and measured the voltage while measuring the distance between the end of the head to the golden piece at the top of the potentiometer. After taking 47 measurements, I plotted the data in the graph below and found the relationship between voltage and displacement.

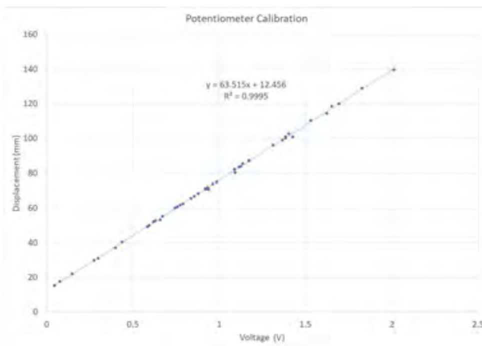

Figure 4: Calibration for the Potentiometer. Note that 15.4 mm corresponds to full retraction or zero diplacement of the head.

2.3 Arduino Uno Rev3

So far in our design, we have a actuator and a feedback mechanism, but no controller to coordinate between these two: enter the Arduino Uno. The Arduino Uno is a microcontroller that can seamlessly integrate sensors and actuators. Besides also being easy to program, it's a device I'm familiar with. The final schematic is shown below. It includes the connections for the driver, stepper motor, potentiometer, and an emergency stop switch. This last switch was added at the request of Pierre-Yves; when the switch is pressed, the motor will immediately start retracting the probe until it's at the home position. The Arduino was powered by a power supply and the stepper motor was powered by a separate power supply.

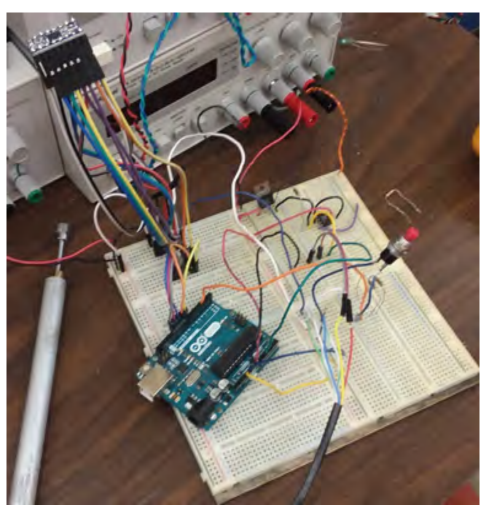

Figure 5: An image of the final electronic setup. In this image we can see the stepper motor, Arduino, potentiometer, constant-circuit current, and the power supplies.

2.3.1 Programming

Below is a flowchart of the control system. The Arduino was programmed according to this flowchart. For the full code see Appendix A.

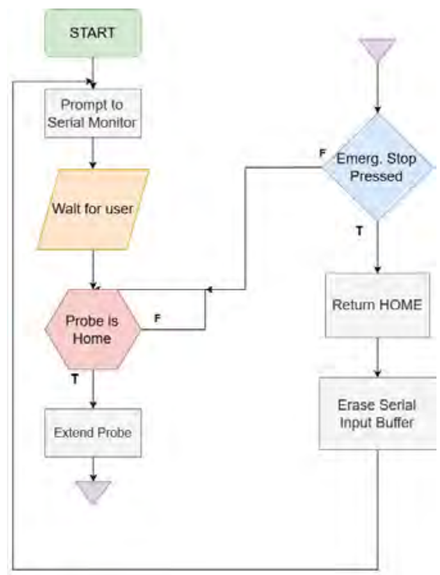

Figure 6: The flowchart for this control system

3 Mechanical Assembly

In this section we'll discuss the mechanical components that were designed and fabricated to enable this control system to work. We mentioned in Section 1.3 that we needed our control system to operate within Brown's Diagnostics Suite. Now that we have the electronics for the control system, in this section we'll show how we integrated the electronics into the current probe setup.

3.1 Adapter

Since the potentiometer needs to track the position of the probe tip, we need it to be attached to the probe somewhere. I used the threaded head of the potentiometer, see Figure 2, to attach it to a subassembly on the diagnostics suite called the Slider Assembly.

Figure 7: An image of the Slider Assembly. The teal rod in the middle is the probe, while the golden brown piece on the left is the flange nut. The flange nut is threaded so that when the lead screw rotates inside of it, the slider assembly moves linearly. The orange piece is called the slider body.

We chose to attach the potentiometer to the slider body because otherwise we wouldn't be able to fit the potentiometer into the space available. The potentiometer is attached to the slider body by means of an adapter that screws onto the 3/8-24 thread of the potentiometer head and connects to a small hole on the long face of the slider body, see Figure 7. Since this adapter would be very close to the insert region, it would need to withstand the same conditions as the slider body. For this reason, we machined it of the same material as the slider body: Alloy 309 Stainless Steel.

Figure 8: An image of the Potentiometer Adapter

3.2 Potentiometer Sealer

We also needed a piece to hold the potentiometer while also maintaining the gas pressure inside the cathode. Our solution is shown below.

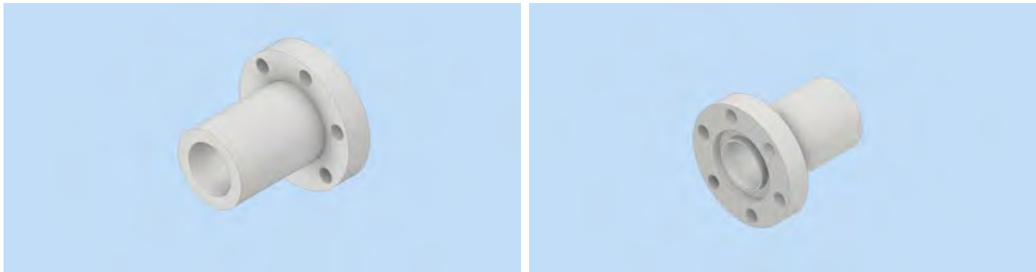

Figure 9: Images of the Potentiometer Sealer. To maintain the gas pressure inside the cathode, we added an O-ring to the sealer. The O-ring prevents leakage in the radial direction. The hole in the middle of the potentiometer is a tight fit, but we also used Torr Seal to prevent leakage in the back.

3.3 Final Assembly

In the preceding sections I described the additions I made to Brown's design to make the control system work. We had to make some other changes so everything could fit properly. Below are images of what the final assembly looked like - certain parts have been deleted from the image for clarity.

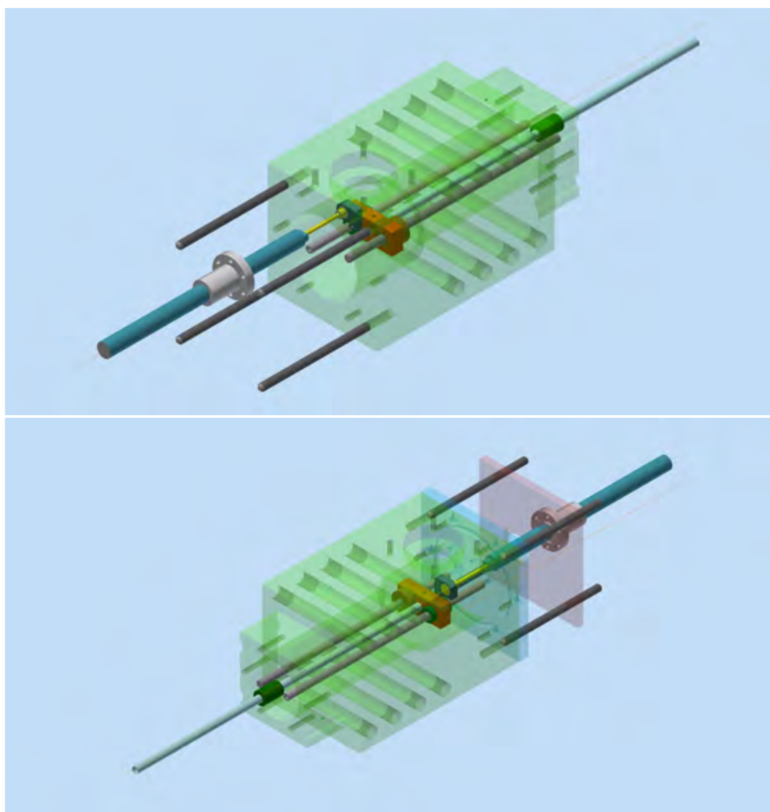

Figure 10: Images of the final assembly.

4 Conclusion

In conclusion, this summer I improved a Langmuir Probe Diagnostics Suite by adding a closed feedback loop. Using an Arduino microcontroller and a linear potentiometer, I was able to control the displacement of the probe. Unfortunately, we weren't able to implement the system during an experiment. However, we tested the system's features both with the Lin Engineering Stepper Motor and with a test motor. The system worked in both cases. This suggests the system should work during an experiment. Besides providing experimental results to validate current models, researchers at EPPDyL hope this system could determine whether there's a position dependence in the insert region.

5 Acknowledgements

I would like to thank Pierre-Yves Taunay for his, seemingly, endless patience with me throughout the summer. Thank you Pierre-Yves answering all of the million questions I asked on a daily basis. Thank you also to Sebastián Rojas-Mata for his guidance of my presentation and for answering general plasma physics questions. I would also like to thank Professor Edgar Choueiri for the opportunity of working in his laboratory. Working here this summer was a truly eye-opening experience and I'm so incredibly grateful to him for having made this possible. And thank you God, who opened this door in the first place.

References

[1] Jordan Brown. *Design and Implementation of an Actuated Probe Suite for an Orificed Hollow Cathode*. Princeton University. 2018.

[2] Christopher Wordingham, Pierre-Yves Taunay, and Edgar Choueiri. “Theoretical Prediction of the Dense-Plasma Attachment Length in an Oriced Hollow Cathode”. In: *International Electric Propulsion Conference* (2017).

Appendix A

```
    /* D.S. Zamora
 * 6/28/19, 7/1/19
 * Have motor move until it’s at the correct distance or it’s hit
 * the emergency stop. This program is called the ’finalSoln’ because
 * it is the final solution for having a close-feedback loop between
 * the extension of the probe and the rotation of the motor. (It’s
 * also the combination of ’soln1’ and ’soln2’.
*/

#include <A4988.h>
#include "A4988.h"
#include <BasicStepperDriver.h>
#include <DRV8825.h>
#include <DRV8834.h>
#include <DRV8880.h>
#include <MultiDriver.h>
#include <SyncDriver.h>

// using a 200-step motor (most common)
#define MOTOR_STEPS 200
// configure the pins connected
#define DIR 8
#define STEP 9
#define MS1 10
#define MS2 11
#define MS3 12

A4988 stepper(MOTOR_STEPS, DIR, STEP, MS1, MS2, MS3);
int emergStop = 3;
int distPotent = A1;

void returnHome() {
  while (analogRead(distPotent) > 20) {
    stepper.rotate(-360);
  }
  stepper.stop();
  return;
}

void setup() {
  // put your setup code here, to run once:
  stepper.begin(90, 1);
  Serial.begin(9600);
  pinMode(emergStop, INPUT);
  analogReference(DEFAULT);
}

void loop() {
  // put your main code here, to run repeatedly:

  Serial.println("Would you like to begin? (Y/N)");
  while (Serial.available() == 0) {}
```

```
  while (analogRead(distPotent) < 200) {    // '200' is arbitrary
    stepper.rotate(360);
    if (digitalRead(emergStop) == 0) {
      break;
    }
  }
  returnHome();
  while (Serial.available() > 0) {
    Serial.read();
  }
}
```