# Repairing and Modifying Translational and Azimuthal Stages for Electric Propulsion Applications

*Author:*
Daniel Card
Research Assistant
dcard@umich.edu

*Mentor:*
Ethan Dale
Ph.D. Candidate
etdale@umich.edu

*Co-Director:*
Benjamin Jorns
Assistant Professor
bjorns@umich.edu

August $16^{th}$, 2019

Summer Undergraduate Research Engineering
University of Michigan - Ann Arbor
Plasmadynamics and Electric Propulsion Laboratory

## 1 Abstract

This research has been conducted to repair and calibrate motion stages for electric propulsion applications. Motion stages are used for moving equipment, typically probes, inside the vacuum chamber. These stages must be accurate as plume profiles have finely resolved spatial features. Through this research, the goal is to be able to improve the reliability, life-span, and accuracy of existing stages at the Plasmadynamics and Electric Propulsion Laboratory. Improving these stages will aid future research, such as characterizing new thrusters with extensive plume mapping, and with non-invasive studies with laser-induced fluorescence that require precise translation of the thruster. To be able to collect repeatable data, our translational and azimuthal stages need to be extremely accurate, precise, and reliable.

The first step was to interface the motion stages with encoders that previously did not have any installed. This will allow redundancy in the stages such that we can eliminate any machined defects and accurately record the position of the stages. Next would be to design a standalone easily-deployed circuit to read the encoder signals. Then from there design an interface for that circuit so that it can be read from a computer. Having this ability would add flexibility to how we collect our data. Finally fix mechanical problems in the high-speed stages to improve repeatability, reliability, and mobility of the stages.

To increase the mobility of the translational stages there are several configurations that we can use - each with their respective advantages and disadvantages. The current problem that the stages are experiencing is loss of motion during long exposure to lower temperatures in a vacuum. It is important that these stages can move unimpeded, as smooth motion results in better data collection. The proposed fix to these long exposure times was to interface heating elements with the bearing carriages. These will mount into the carriage spacers that are in-between the bearing and the carriage top plate. These bearing heaters will heat the greased bearings allowing them to operate in long-duration tests while being exposed to lower temperatures.

Cleaning and re-greasing of the previous stages took approximately a day. Assembling the wiring harness to our high-speed stage was accomplished in two days. The testing and comprehensive debugging of the stages was prolonged taking around two weeks. The next task, designing the circuit and interfacing with Arduino, was a lengthy process taking just shy of a month. Most of the time spent on the circuit was on the testing and debugging of the circuit and program. Lastly, the bearing heaters for the motion stages were designed and attempted to be implemented in about a week.

# 2 Repairing and Modifying Motion Stages

## 2.1 LILAC/HARP Guide Rail Modifications

There are several types of motion stages in use at PEPL. An Azimuthal stage can rotate about the azimuthal axis. This stage often has a long probe/arm attached to it such that it can sweep probes around the front of the thruster. The next type of stage is the translational stage. This, instead of rotating, moves linearly in one-axis. These stages typically have a probe attached to it that allow it to take quick measurements in front of the thruster. These high-speed linear stages are essential to taking measurements directly in front of the thruster. Slower moving stages will have their probes deteriorate from the plasma because long exposure to the plasma will sputter the probes. The Long Iron-less Linear Actuator (*LILAC*) and the High-Speed Axial Reciprocating Probe (*HARP*) are our two linear motor stages.

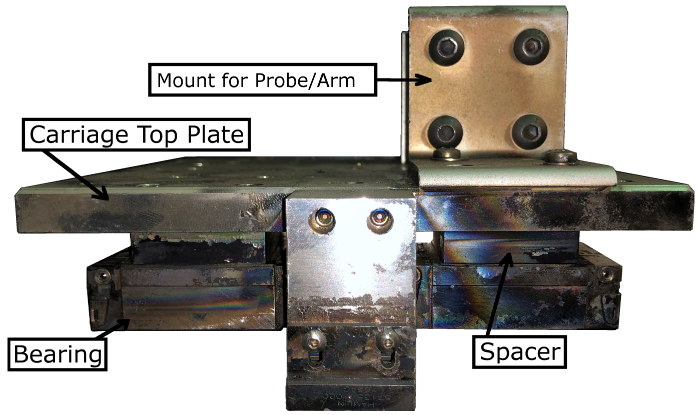


Figure 1: Long Iron-less Linear Actuator

A traditional stepper motor has a rotor composed of coils and a gear-like stator surrounding it. Energizing the coils induces a magnetic field that advances the rotor from one "tooth" to another. A linear motor operates similarly but the stator is "unrolled" into a linear track, and the rotor is a series of coils overlapped into a flat "forcer" that fits inside the stator track. A more in-depth diagram is shown in figure 2.[4]

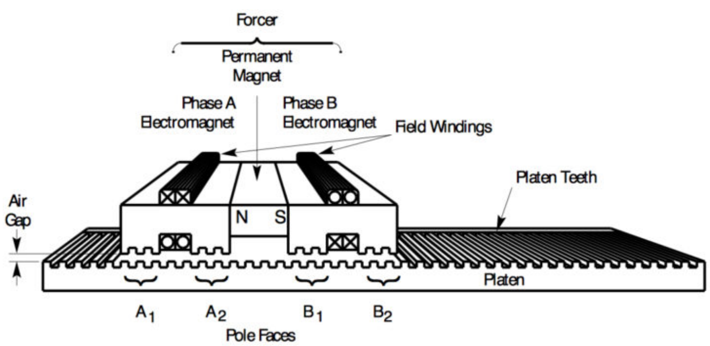


Figure 2: Detailed Diagram of Linear Motor[4]

There remained some issues with our stages that impeded their effectiveness. LILAC first needed to be re-wired and debugged, as it had been pulled from the vacuum chamber and was no longer working correctly. LILAC would also experience a loss of mobility while being exposed to low temperatures during long periods in the Large Vacuum Test Facility (*LVTF*).

The first step to repairing LILAC was to repair the electrical harness and interface it with the feedthroughs available in the control room. Then to make sure that the electrical system has been wired correctly, each wire was checked for continuity with a multimeter. This would ensure that the electrical harness would perform nominally.

Next was to test the LILAC outside of the chamber to check and make sure that it was working nominally. A linear motor must be controlled with a feedback loop so it can be positioned more precisely than the size of the individual magnets in the magnet track. A typical Proportional-Integral-Derivative (*PID*) control scheme is used to ensure that the feedback position, (provided by the encoder) closely tracks the programmed position (where the controller instructs the stage to move). Tuning is done with an automated process implemented in the control software that examines the frequency response of the stage. Tuning must be performed routinely to account for changes in the carriage mass and additional sources of damping[1]. A change is sensed, adaptations are made, and then the process progressively increases the phase margin until the tuning program estimates that the parameters are a bad fit.[6]

## 2.2 Bearing Heating System

The next issue to fix was the bearing carriages seizing during operation in the vacuum chamber. The most likely cause of this is the grease seizing after being exposed to low temperatures for extended periods. These lower temperatures were caused by the new cryogenic pumps that have been installed on LVTF. To fix this loss of mobility, we outfitted the bearing carriage spacers with small heating elements. We are first attempting to heat the bearings with small heating elements used in 3D-printing like the ones shown in figure 3 as they were significantly cheaper than higher grade heating elements. If the low-cost heating elements were incapable of heating the stage, I could opt for the higher grade heating elements, as they had the same dimensions and could easily replace the low-cost heaters. Next was to interface with a thermocouple so that we could observe the temperature of the bearing carriages. This will allow us to optimize the power going into the heating elements, ideally extending their life-span.

Figure 3: 40W 24V FDM Heating Element

The elements being used are shown in figure 3, and are small enough to be interfaced within HARP and LILAC. We chose to use these smaller elements so that we could test if heating the stages made a noticeable difference to the mobility. I estimated the power needed to heat the bearing carriages by using Stephan-Boltzmann formula. The estimated power needed to heat the bearing carriage and spacer to a temperature of 20°C, is approximately 5W.[5] The low-cost heating elements have a power rating of 40W, whereas the higher grade elements have a power rating of 200W, so if needed we could choose to upgrade to these elements if the low-cost alternatives are not suitable for heating the entire stage.

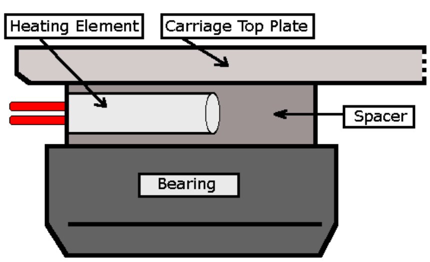


Figure 4: Bearing Heating System

The idea is to take the heating element and install it into the spacer between the bearing carriage and carriage top plate to allow for the heating as shown in figure 4. The prototype bearing heating system is shown in figure 5, with the wires being routed through the carriage top plate. The bearing heaters will be wired in series; this will inform us when a heating element burns out. This will help aid in easier debugging and wiring because if the heaters were wired in parallel there would be no easy way to know which heating element burned out unless each element had its own thermocouple. The wires from the heating elements and the thermocouple will be routed through the pre-existing cable guide. This cable guide will help the stage move unimpeded as having bulky wires will interfere with the motion of the stage.

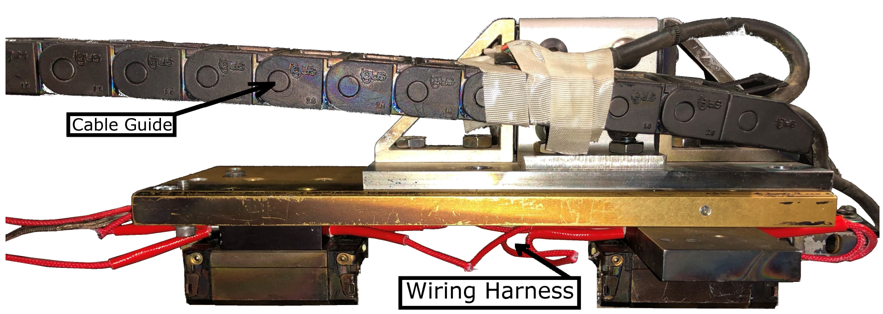


Figure 5: High-Speed Axial Reciprocating Probe With Bearing Heating System

# 3 Encoder Circuit

## 3.1 Interfacing With Integrated Circuits

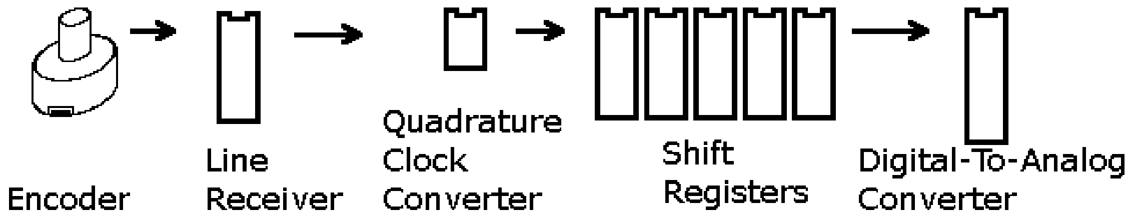


Figure 6: Order of Execution of Integrated Circuits

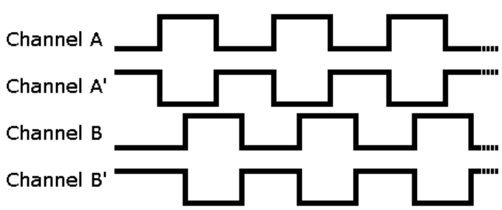


Figure 7: Differential Encoder Signal Output

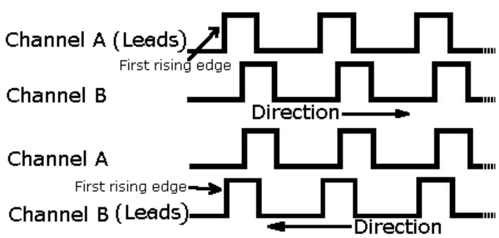


Figure 8: Line Receiver Signal Output

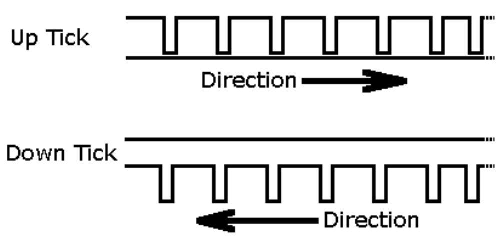


Figure 9: Quadrature Clock Converter Signal Output

By creating an in-house encoder reader we can create a flexible position measuring system that can output data in multiple ways. This creates flexibility in the way that we collect measurements and will help aid with future research. We can also collect position data in real-time, which is most important with the high-speed stages.

The in-house encoder reader works by a series of integrated circuits ($IC$) interfaced with Arduino. The integrated circuits count pulses sent from the encoder and then store it as a binary number held in shift registers. The Arduino is then able to read out these values and calculate the position of the stage from these stored values. Figure 6 shows the order of operation of the integrated circuits, figure 7 through figure 9 show the waveforms of the outputs of each integrated circuit which will be described in more detail in the following paragraphs.

Incremental encoders can be used in position and motor speed feedback applications. An encoder provides these speed and distance feedback in the form of signal pulses, and require minimal hardware providing a simple and reliable solution to find the position of the system. However, these incremental encoders only provide changing information about relative motion, or increments, so the encoder must be interfaced with a reader to calculate its motion and position. We could opt for commercial encoder readers but they typically are bulky, expensive, and have limited adaptability. To provide an inexpensive, flexible encoder reader I built, and designed an in-house encoder reader.[2]

An encoder provides a specific amount of pulses-per-rotation or pulses-per-length. The output channels are phase offset to determine the direction of motion. Figure 7 shows what the output from an encoder looks like. This specific encoder is known as a differential encoder, as it has four signals, two inverted pairs to aid in noise rejection.

The Line receiver conditions the data from the encoder. It takes these four signals and converts them into a single-ended waveform that gets passed onto the next integrated circuit. It executes this task because the next integrated circuit can only compare phases of two signals. Its signal output can be seen in figure 8.

The quadrature clock converter then takes the output from the line receiver and transforms it into pulses. These pulses can be seen as uptick/downticks allowing for both counting up and down the changes in position. The output signals can be seen in figure 9. The pulses are essential for the next integrated circuit to be able to execute its operation.

Shift registers are another type of sequential logic circuit that can be used for the storage or the transfer of binary data. The shift registers take the pulses from the quadrature clock converters and output these as binary counts. Our specific circuit has five shift registers allowing for twenty digits of binary, allowing over one million steps to be counted. Each of these outputs is read into the Arduino where it takes this binary data and performs calculations to compute the position/rotation of the stage.[3]

The Digital to Analog Converter($DAC$) accepts twelve digits of binary and outputs them as an analog signal. This DAC chip will output data from $0V \rightarrow 10V$ with 820 steps per microsecond update rate, allowing us to log this on an oscilloscope to capture the stage's position in real-time.

The purpose of this circuit is to output the data from the encoder in real-time. It is essential that the circuit can output the data in real-time so that we can reduce the spatial uncertainty in high-speed measurements. The Arduino runs at a clock speed of $16MHz$. Our fastest encoders have a pulse rate of $200kHz$, so we need to read encoder signals faster than that. Although the Arduino has a $16MHz$ clock speed, reading and parsing encoder signals with it directly could potentially take milliseconds. Alternatively, an integrated circuit-based reader will update approximately $32MHz$ without latency for parsing.

We can communicate with the Arduino via LabVIEW, which means we can log position slowly but precisely that way. This is an alternative to using the DAC's output and logging with an oscilloscope, which will be fast but relatively imprecise. We could also use the Arduino to output the position of the stage to a small screen, however, this is not used in data collection but is a visual aid in locating the position of the stage.

## 3.2 The Basics of The Program

Essentially, the pulses come in from the Arduino circuit, and the program will then calculate the number of pulses from binary to an integer. From there the program will have the correct number of counts. Then, the program takes this integer number of counts and then multiplies by the normalized parameter to calculate the position or rotation of the stage.

However, once all the shift registers have reached their maximum count, they will all reset or "roll-over" to zero. If the count is already at zero and "goes negative", it will again "roll-over" to the maximum count. So the program has implemented a few cases to determine when this has happened. The program consistently checks for a large jump in values and from there will add or subtract twenty-digits of binary (depending on what direction the jump has occurred). This allows the program to continue counting despite having passed its maximum count. This "roll-over" in counts was not noticeable in lower resolution encoders since the output of counts never exceeded the maximum count, but the higher resolution encoders can easily surpass this maximum count.

By combining the Arduino with the integrated circuits, we will have redundancy in the way that we can collect our data. We can either pull data from the Arduino itself, interface with LabVIEW, or log everything on an oscilloscope; each has its advantages and disadvantages. Every type of data collection is limited to how fast it can output its data. These will be discussed below.

Advantages and Disadvantages to Interfacing LabVIEW, Oscilloscope, and Arduino

<u>LabVIEW</u>

- Easily able to input data and correlate it to the measurements being made
- Interfacing with Arduino, the data interface is limited to the slower clock speed. Whether it be the microsecond execution of the Arduino or the bit rate that LabVIEW can support

<u>Oscilloscope</u>

- Can input data from the integrated circuits in real-time
- Limited to the resolution that the oscilloscope can support

<u>Arduino</u>

- Unable to provide real-time data, has millisecond delays from executing lines of code
- Is able to directly output the position to a small screen

# 4 Summary

In this research, I have been able to repair and update our previous motion stages. I first repaired LILAC's electrical harness and ensured that it would perform nominally inside of the vacuum chamber. I also contributed to debugging LILAC's powering and grounding issues after being installed into the LVTF. I also interfaced incremental encoders to the motion stages that were not previously interfaced with encoders. Additionally, I designed the bearing heaters of the stages to help the performance during long-duration tests.

Furthermore, I assisted with the design of our in-house encoder reader. This provides flexibility to how we collect our data whether that data collection is through Arduino, LabVIEW, or through an oscilloscope. This in-house encoder reader was built with integrated circuits and then interfaced with Arduino to display the position.

# 5 Conclusion and Outlook

## 5.1 Conclusion

Through this research I have contributed towards repairing and updating our existing translational and azimuthal stages. Future research involving in situ probing, as well as non-invasive experiments where the thruster is moved, relies on motion stages to be extremely accurate and precise. Improving these stages will result in more reliable and repeatable data. Encoders will help in collecting much more accurate data by knowing that we have the precise location of a stage rather than its estimated location.

I have also contributed to improving the lifespan of our linear motion stages. By implementing bearing heaters, the lifespan should be extended as there is less wear-and-tear on the guide rails and bearings. This allows for smoother motion of the stages during longer duration experiments. This also makes the hardware more reliable as it can perform nominally longer without the need for maintenance.

## 5.2 Outlook

Going further with this research, it will have a lasting effect on the reliability of our stages and the repeatability of data. Having the ability to collect data in multiple ways will provide flexibility on how we can collect our data. As stated before, we can choose to input our data either through an Arduino, LabVIEW, or an oscilloscope. This allows flexibility in the way that we collect and report our data by choosing the resolution that we collect it. This will allow us to collect data in a way that can better suit the research being conducted. Such as if we are doing quick probe sweeps in front of the thruster we would need to have the data output in real-time to ensure that we can collect reliable data.

The bearing heaters will have the most significant impact on long-duration tests. This is because the long exposure to lower temperatures may be the root cause of the seizing or loss of mobility to the stages. By outfitting the stages with heating elements it may allow us to have the ability to run longer duration tests with the motion stages. By having the ability to run LILAC or HARP for extended periods this could also help aid in erosion testing where we run the thruster for extended periods.

# References


[1] AeroTech. Easytune advanced motion controller autotuning tool, 2019. Last accessed 2 August, 2019.

[2] Dynapar. Incremental encoder overview. Last accessed 3 August, 2019.

[3] ElectronicsTutorials. The shift register. Last accessed 5 August, 2019.

[4] Haydon Kerk Pittman. Linear actuators - stepper motors, 2019. Last accessed 2 August, 2019.

[5] SoftSchools. Stephan-boltzmann law formula. Last accessed 7 August, 2019.

[6] Mike Soper. What is a pid loop?, July 27, 2017. Last accessed 2 August, 2019.