[[1]](#footnote-1)

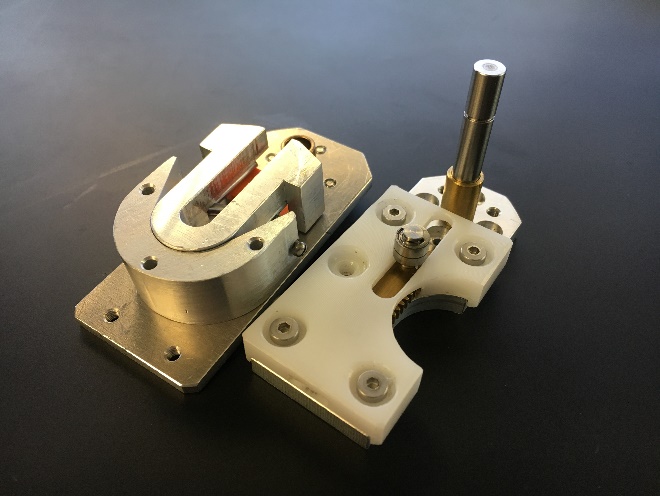


Figure : The two halves of the VSA

Christian Wahrmund is with the Advanced Robotic Manipulators (ARM) lab, Department of Mechanical Engineering, University of Texas at San Antonio UTSA, One Circle, San Antonio, 78249, Texas, US (corresponding author, email: christian.wahrmund@utsa.edu)

Control System Development and Implementation of a Novel High Impedance Variable Stiffness Actuator

Christian Wahrmund

*Abstract*— This document explores the construction of 3 controllers (PID, LQR, and Pole Placement) for a high impedance VSA (Variable Stiffness Actuator). The mechanical design and modeling of the VSA are detailed and a state space model is built for the system. The system is tested experimentally in an uncontrolled state and again under a tuned PI controller. The system and its controllers were implemented in Matlab and Simscape. Performance criteria and initial conditions were set, and the controllers were designed and evaluated based on system step response. Overall, the PI controller exceeded design specifications but with a large steady state error attributed to poor assumptions made in the model and hardware limitations.

# Introduction

## Background

VSA’s build inherent mechanical compliance into electrically or mechanically powered components. They operate by building allowing the end effector of an arm to differ in position from the actuator by a certain degree. This is usually accomplished via mechanical linkages and springs, VSA’s also allow the stiffness of the system to be varied dynamically. They are mostly used in areas where robots and humans work in close interaction, and require another layer of safety. The largest problem facing VSA’s is one of accuracy [1] [2] [3], by building in mechanical compliance the true position of the end effector will always differ from the position of the actuator. The natural frequency of the system will affect accuracy as well. By building in a spring like stiffness to a joint, oscillation can occur during periods of acceleration or deceleration. This VSA mitigates both of these issues by virtue of its high impedance design. Even under rapid movements and changes in position and stiffness the actuator displays a critically damped response without any damping. This greatly simplifies the control scheme as damping does not need to change with stiffness values eliminating an input and output from the system.



Figure : Components of VSA test rig. The blue box on the left houses the Dynamixel MX-64 AR motor used to spin the VSA. The orange shaft on the right connects the VSA to this motor. In the center we see the VSA with the Dynamixel AX-12 motor used to control the stiffness of the actuator. Design was based upon [5].

# System Modeling

## Physical System

Looking to Figure 1 we can see the inner mechanism of the VSA. Torque from the motor is transmitted to the system through the bottom base plate. The pivot point (located in front of the shaft) can slide up and down the U shaped bracket mounted to the top plate. The U bracket is connected to the top plate through a freely rotating shaft. When force from the pivot is applied to the U bracket it torsionally compresses a spring. The reaction force on the pin connecting the U bracket to the top plate is what transmits force out of the system. This can be modeled as a lever with a spring at one end and a rigid mount at the other. As the pivot moves away from the spring and closer to the rigid end, the pivot must exert more force to compress the spring the same amount. This design gives the system an infinite effective range of stiffness. When the pivot is located at the springs location it takes no force to compress the spring. Conversely, when the pivot is located at the U joints shaft no force can be transmitted to the spring.

## System Dynamics

Equation : State space model of open loop control system. Where is the torque produced by the motor, and are the angular position and speed of the motor. and are the angular position and speed of the output link of the VSA relative to the base of the VSA. K and D are the overall stiffness and damping ratio of the system. and are the inertia moments of all rotational mass before and after the spring. , , , ,

The control loop for the Dynamixel motor can be seen in Figure 11. This loop represents all of the internal workings of the Dynamixel motor. It is a fully self-contained unit with built in PID control that directly converts an input position into an output position. We can represent the motor as a collapsed unit, as shown in Figure 12. Looking to this block diagram, we can see that the drive motor has outputs of position and torque. Torque is assumed to be the only input to the VSA. Position is fed to a sum block to output the global position of the end link. This block diagram was the basis for forming the state space equations of the VSA as shown in Equation 1. The state space model takes and input of torque and position from the motor and translates them into an overall output position of the link relative to the initial starting global position.

As the purpose of this VSA is to prove that it can be PID tuned even at extremely low stiffness values, a stiffness value of 10 Nm/rad was held constant for the system during all modeling and evaluation. The moment of inertia was simulated as a half meter and half kilogram long rod with a small sphere at the end of the link. The motor input torque was calculated by subjecting the motor to the inertia of the system and looking at the motors maximum acceleration. The natural damping ratio of the system was found experimentally. These values remained constant across all testing.

# Un-tuned Simulation

## Stability

The root locus and step response of the open loop system described by the state space relation of Equation 1 can be seen in Figure 3 and Figure 4 respectively. This system has poles of 0, -4.4e6 and -.7 +- 2. As all poles lie on the negative x axis and the system step response shows that the system eventually settles, the open loop system as modeled can be assumed stable.

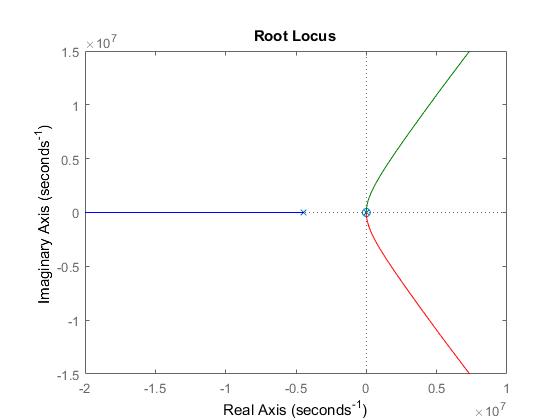


Figure : Root locus of open loop system. All poles lie in the second and third quadrant

## Open Loop System Performance

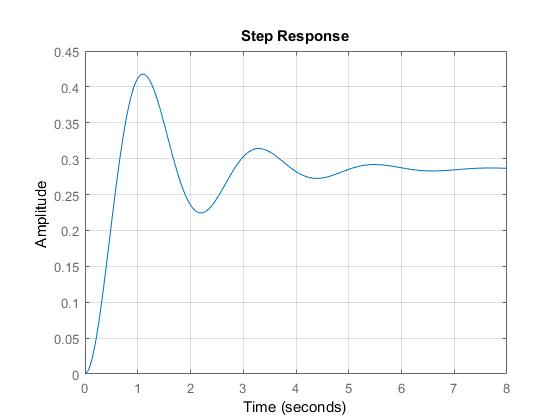


Figure : Open loop step response.

|  |  |
| --- | --- |
| Rise Time | .43 |
| Settling Time | 5.62 |
| % Overshoot | 46.29 |

Table : Open loop performance. The settling time of 5 seconds is falls well outside acceptable parameters.

The open loop step response can be seen in Figure 4. The system, when excited, achieves a relatively fast rise time of .43 seconds. The system has a settling time of 5.6 seconds and an over 45% overshoot. The performance of the open loop system is unacceptable. The system oscillates multiple times before settling, and has a large steady state error of over .7. These dynamics will be improved via closed loop feedback, PID, Pole Placement, and LQR control.

## Closed Loop System Performance

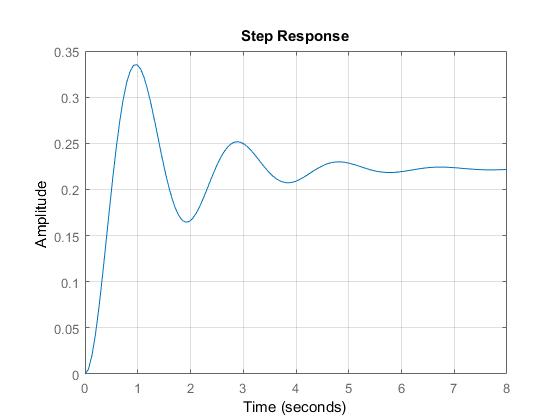


Figure : Closed loop step response.

|  |  |
| --- | --- |
| Rise Time | 0.37 |
| Settling Time | 5.12 |
| % Overshoot | 50.83 |

Table : Closed loop system performance. While the rise and settling times have been slightly cut, the overshoot has actually increased.

Figure 5 and Table 2 show the step response and performance of the closed loop system. While the system performance has improved in some areas, percent overshoot has actually increased. This highlights the need for more advanced methods of controls, simple negative feedback does not properly control this system.

# PI Control

## Design Specifications

Desired specifications for the system are as follows, rise time and settling time should be under 1 second and percent overshoot should be less than 10%. Three methods were used to tune the PID controller. Method one involved manually tuning the system and checking response, method two used a series of loops in Matlab to iterate Kp, Ki, and Kd values from .1 to 20 in steps o f.1, and method 3 involved building the system in Simscape, and using the built in Matlab PID tuner to find values of Kp, Ki, and Kd.

## Hand Tuning

Hand tuning produced poor results. Looking to the root locus of the open loop system (Figure 3), the only line to start searching for values of ideal poles starts at a 46% overshoot. This leaves little to no room for tuning by hand. Without being able to search along the root locus lines for ideal system performance, a second method of finding ideal values was implemented.

## Matlab PID Tuner

The built in Matlab PID tuner was implemented after constructing a full model of the VSA in Simscape (Figure 16). A step signal was input to the system, and the system response was measured and mapped through Matlab.

|  |  |
| --- | --- |
| Rise Time | 0.61 |
| Settling Time | 0.92 |
| % Overshoot | 1.84 |

Table : Adding a tuned PID controller has cut the settling time down below 1 second and the percent overshoot has dropped to below 2 percent. Well within acceptable design parameters.

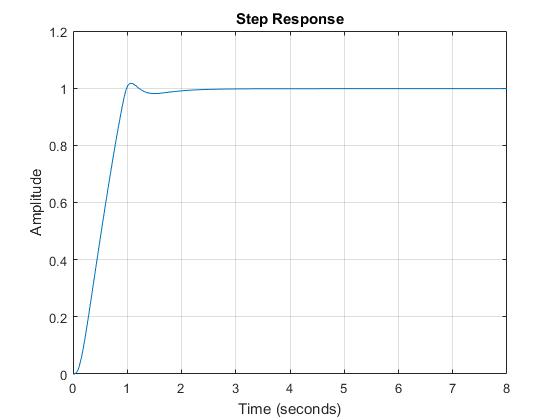


Figure : PID tuned closed loop step response.

|  |  |
| --- | --- |
| P | 9.4 |
| I | 27 |

Figure : Tuned PI Values

Figure 6 and Table 3 show the PID tuned step response and performance values of the system. The system now falls well within acceptable parameters, rise and settling time are below 1 second and percent overshoot has dropped to below 2%. This shows that even under a heavy inertial load with a low overall stiffness and no additional damping the VSA can achieve impressive system performance using only a PI tuner.



Figure : Step response of pole placement controller

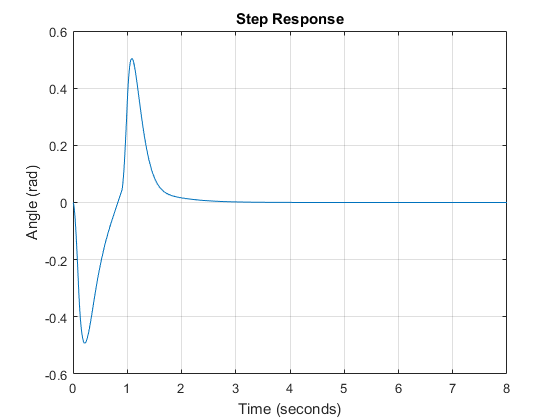


Figure : PID Tuned output angle variance of VSA from motor.

## Pole Placement

Pole placement started with the same initial problem as hand tuning with the PID controller, namely that the root locus of the un-controlled system starts at a point well outside of performance specifications. The solution to this problem was to build a set of random first order controllers in series with an integral controller. The root locus of each pseudo-controller was evaluated until one produced a point to start from. After an acceptable pseudo-controller was found, poles were picked and the negative feedback controller was put into state space form. The characteristic equation of the plant can be found by Equation 2.

Equation : General form of closed loop characteristic equation

After solving this equation form, values of k1 through k6 were found and the desired closed loop characteristic equation (Equation 3) was calculated. This equation was then formed into a new transfer function (Equation 4) for the controlled VSA.

Equation : Desired characteristic equation

|  |  |
| --- | --- |
| Rise Time | 0.11 |
| Settling Time | 3.56 |
| % Overshoot | 9.06 |

Table : Pole Placement controlled system performance.

Equation : Transfer function of pole placement system with controller

Figure 9 and Table 4 show the final step response and performance of the pole placement control system. This system has an impressive rise time, an acceptable percent overshoot, and no steady state error but has a settling time of over 3.5 seconds, well outside of specifications. Overall, the process of picking poles proves to be an unreliable method of controlling this system.

## LQR

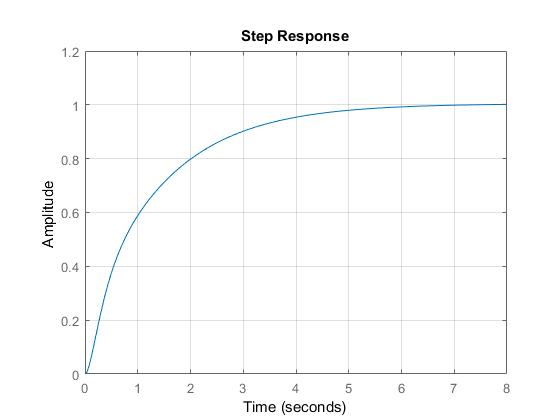


Figure : Step response of final LQR controller

|  |  |
| --- | --- |
| Rise Time | 2.87 |
| Settling Time | 5.29 |
| % Overshoot | 0 |

Table : LQR step response system performance

Equation : The Riccati equation is used to build the LQR controller

Construction of the LQR controller began by converting the already built transfer function of the pole placement controller Figure 4) to a state space model. The Riccati equation (Equation 5) is used in the construction of the LQR controller. The P matrix is solved for after inputting the A matrix of the poll placed system and user selected Q and R matrices. A new step response can be found and evaluated. The process of tuning an LQR controller is an iterative one, an optimized step response and performance evaluation can be found in Figure 10 and Table 5. An integral controller was put in series with the LQR controller to While the optimized LQR controller produces no overshoot it has an unacceptable settling and rise time.

An LQR controller may have applications where energy efficiency and overshoot of the joint takes precedence over settling time. However, for the intended application of this joint, it does not present an acceptable level of performance.

# Test Bed and Methods

## Experimental Setup

After building the controllers the PI system was selected to implement for experimentation due to is ease of use and performance in simulation. A linear quadratic encoder was used to count rotations. Voltage from an external power supply was input to the encoder and readings were collected from an NI myDAQ sampling at 200000 HZ. Using MATLAB, P and I values were written directly to the motor at the start of each test. Testing consisted of setting the VSA to a constant low stiffness and writing a goal position of 800 (roughly .6 rad) to the motor. Immediately after setting a goal position to the motor callbacks to the Dynamixel were used to collect motor position data. Due to hardware limitations only 25-30 motor positions were taken over the roughly 2 second testing period compared to the 600000 samples taken of end effector variance. A spline fill was used to fill in gaps between motor position readings. After allowing the system to come to steady state, data collection was stopped and the motor was returned to its starting angular position. A picture of the experimental setup is shown in Figure 19.

# Results and discussion

## Controller Model Comparison

The open and closed loop step responses of this system show somewhat atypical behavior for an un-damped VSA. While the response is clearly undesirable, the system response shows that the high natural impedance of this VSA offers a base improvement over more conventional VSA’s [3] [4]. It also highlights the need for more advanced methods of control than simply providing negative feedback. Controller design began by construction of a PI controller, (Section IV). Hand tuning was performed but ultimately produced inferior results to Matlab's PID tuner. Matlab's tuner was manipulated to produce a response (Table 3), that would meet and exceed performance specifications.

After construction of the PI controller, a Pole Placement controller was designed and implemented, Poles were picked based on desired second order system response and produced results shown in Table 4. While some characteristics such as rise time improved when compared to the PI controller, overall Percent Overshoot and Settling Time both suffered with Settling Time failing to meet desired system performance.

The final controller constructed for the system was an LQR controller (Section IV.E). The LQR controller produced the only step response with 0 Percent Overshoot (Table 5). While this is an exceedingly desirable quality to have in something designed for high precision control applications, the Rise and Settling Time exceed the desired performance by too large of a value to seriously consider this controller. While the designed LQR controller may have applications in specific circumstances where high energy efficiency and low overshoot dictate selection criterion, it still fails to produce as well rounded of a response as the PI controller does.

## Comparison of Model to Experimental Results

As the P and I values are being written to the Dynamixel motor directly there was a large discrepancy between the optimal model values and the actual tuned values. For example, the MX-64A comes “out of the box” with a P value set to 20, 10 points above the model P value. Setting a P value of lower than 15 (higher than the theoretical optimal) caused the motor to oscillate even when at a standstill. Because of this, a baseline P value of 20 and an I value of 0 was chosen to represent the uncontrolled system shown in Figure 11.

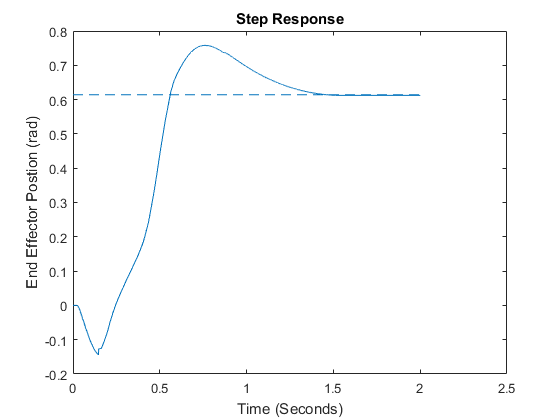


Figure : Step response of P controller with P value of 20.

|  |  |
| --- | --- |
| Rise Time | .60 |
| Settling Time | 1.57 |
| % Overshoot | 23.44 |

Table : Performance characteristics of the uncontrolled system

Figure 12 show the actual end effector position relative to input position. Although this hand tuned model gave the best performance (Table 7) with respect to the performance criteria used to evaluate simulated model performance the steady state error of this system is quite large even when using an integral controller.

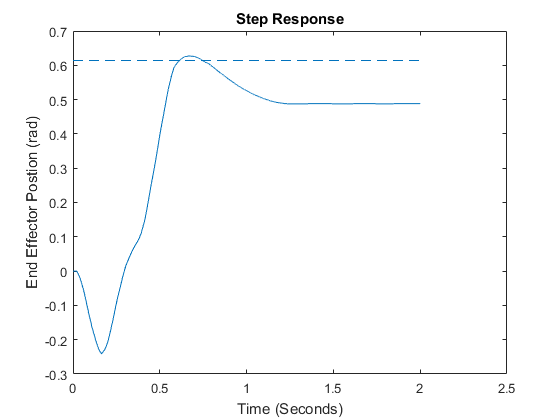


Figure : Experimental step response of PI controller with a P value of 34 and an I value of 4

|  |  |
| --- | --- |
| Rise Time | 0.53 |
| Settling Time | .69 |
| % Overshoot | 2.1 |

Table : Performance characteristics of the PI tuned system

# Lessons Learned

Closer examination of the system during and after testing revealed several likely sources of error.

1. Although the spring is modeled as a solid shaft and inertial load the 3-D printed shaft twists under load acting like a spring to the system.
2. Using the myDAQ prevented the implementation of a true closed loop feedback system. The myDAQ cannot publish values during a data collection session, thus only the internal motor position was used in closed loop and integral calculations.
3. The load used during step testing is likely causing a larger than anticipated friction load on the sliding surfaces of the VSA. This force was assumed to be negligible but observation of the system during testing suggests that this is not the case. This is shown most readily in the low settling time of the actual system, the large friction coefficient takes enough energy out of the system to prevent the oscillation characteristically seen in under damped systems
4. While the torsional spring was modeled as producing the same stiffness values when loaded in either direction the stiffness actually varies by 10% - 15% depending on the direction of travel.

These sources of error will be rectified in future work on this system.

# Closing Remarks

## Conclusion

In this paper three different controllers where evaluated for a use in controlling a novel VSA. The uncontrolled system and PI controlled system where experimentally implemented and tested. After deriving equations for the VSA a state space model and several block diagrams were developed to model the system. After evaluating the open and closed loop step response of the actuator, and setting performance specifications, each controller was designed to produce the most desirable step response. Design and evaluation of each controller was implemented using Matlab, the Matlab control system and PID tuner packages, and Simscape. After modeling the system a test bed was built and the uncontrolled and PI systems were evaluated. Overall these results showed a discrepancy between the model and reality. After examination of the results several sources of error were discovered that likely contributed to the error.

## Acknowledgments

While completion of this project depended on the support of multiple individuals the author would like to extend special thanks towards Dr. Amir Jafari for providing the operating theory for the mechanical design, and Dr. Jaehoon Lee for implementing the mechanical design of the actuator.

# References

|  |  |
| --- | --- |
| [1] | B. Vanderborght, A. Albu-Schaeffer, A. Bicchi and S. e. a. Wolf, "Variable impedance actuators: A review," *Robotics and Autonomous Systems,* pp. 1601-1614, 2013. |
| [2] | L. Visser, R. Carloni and S. Stramigioli, "Energy-Efficient Variable Stiffness Actuators," *IEEE TRANSACTIONS ON ROBOTICS,* vol. 27, no. 5, pp. 865-875, 2011. |
| [3] | W. Wang, X. Fu, Y. Li and C. Yun, "Dynamics Modeling of Variable Stiffness Joint Actuator," in *International Conference on Robotics and Automation*, Anchorage, AK, 2017. |
| [4] | R. Schiavi, G. Grioli and A. Bicchi, "VSA-II: a Novel Prototype of Variable Stiffness Actuator for Safe and Performing Robots Interacting with Humans," in *IEEE International Conference on Robotics and Automation*, Pasadena, CA, 2008 . |
| [5] | A. Jafari, N. Tsagarakis, I. Sardellitti and D. Caldwell, "A New Actuator With Adjustable Stiffness Based on a Variable Ratio Lever Mechanism," *IEEE/ASME TRANSACTIONS ON MECHATRONICS,* 2012. |

# Appendix

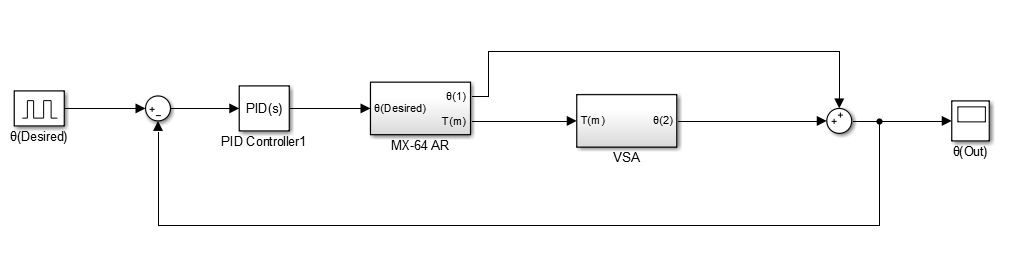
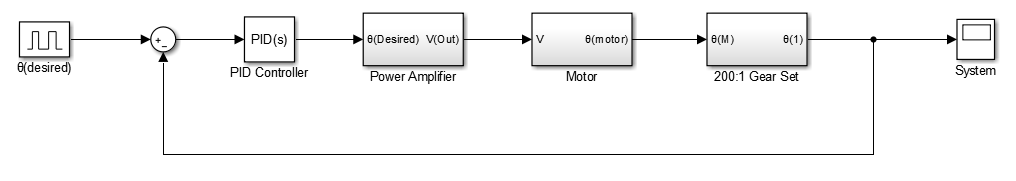


Figure : Dynamixel closed loop control block diagram

Figure : Closed loop control block diagram of system with PID tuner.

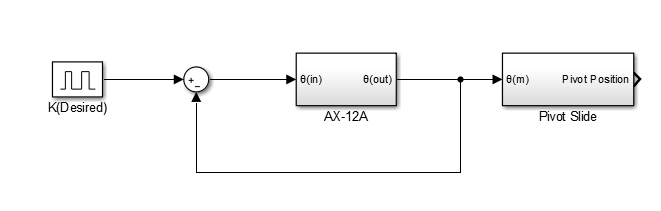


Figure : Closed loop block diagram of stiffness control unit

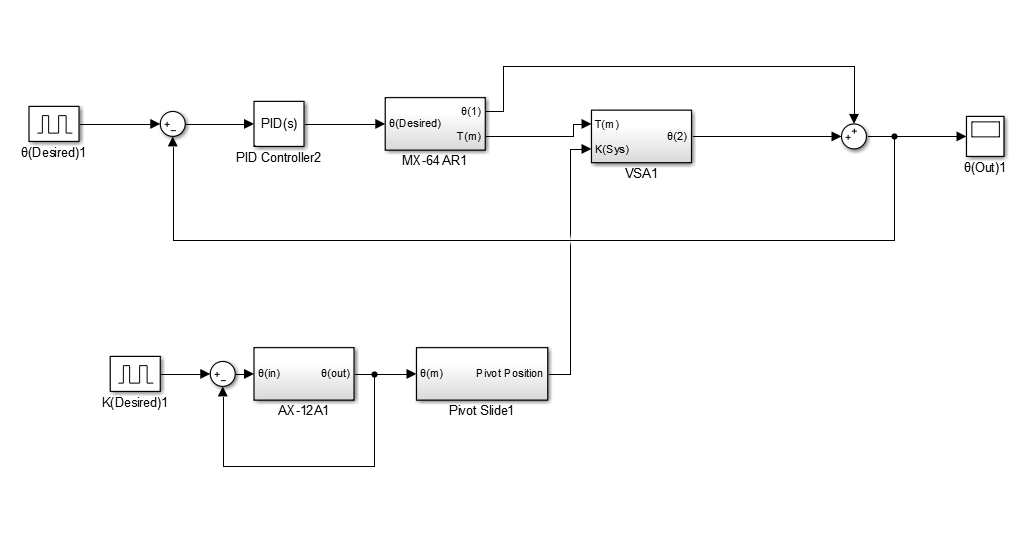


Figure : Simscape model of system with open loop feedback

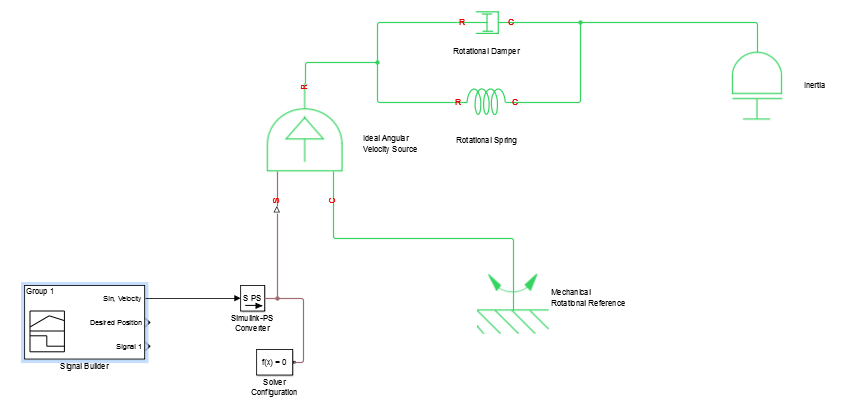


Figure : Control block diagram of system with stiffness input



Figure : Full Simscape model of system, physical sensors, and closed loop feedback control

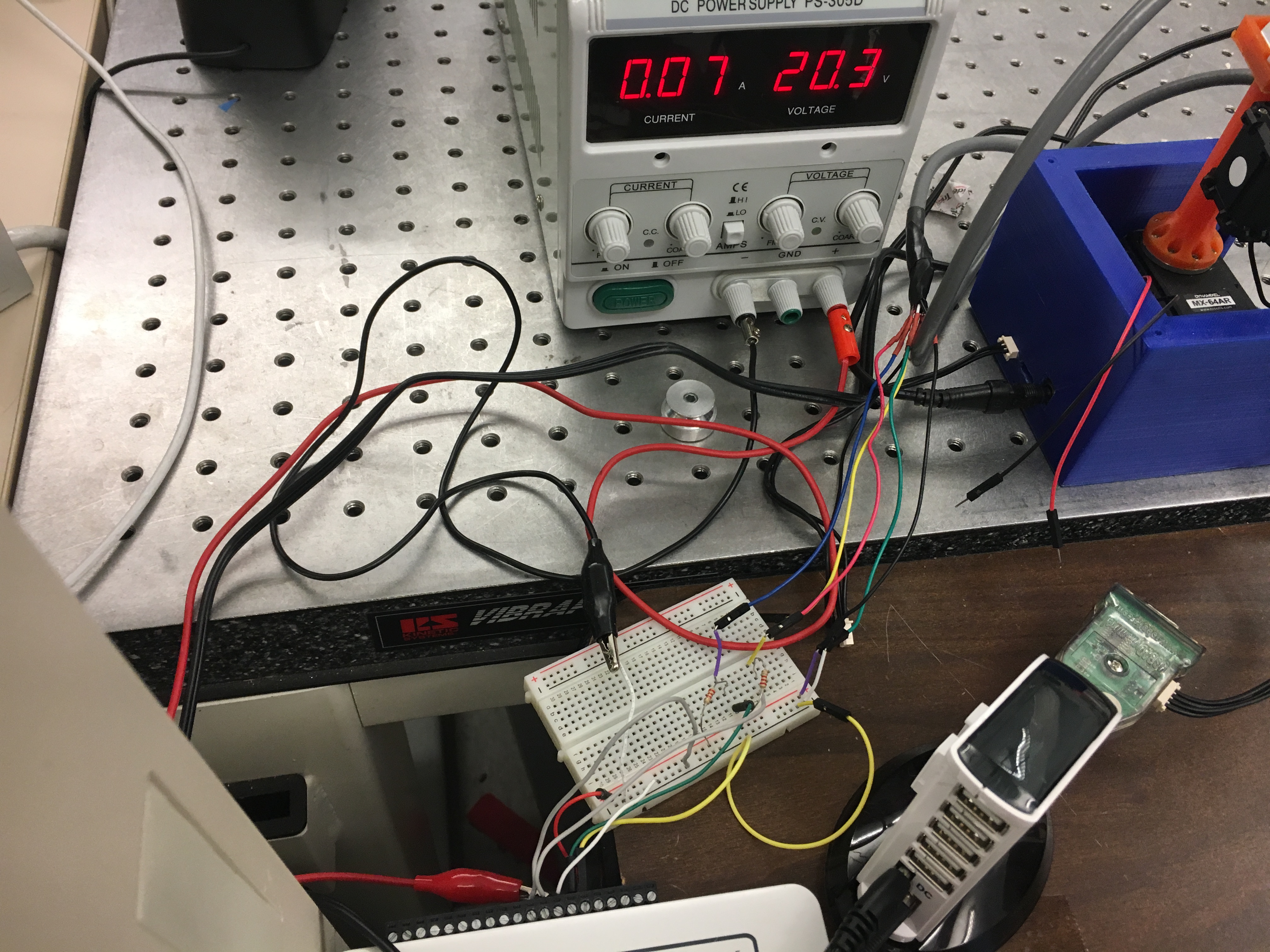


Figure : Photo of test bed setup

1. [↑](#footnote-ref-1)