Designing a PID Control Algorithm for a Lunar Tyrolean Robotic Survey Platform

Ander A Solorzano (ander@solorzano.com)
Rose-Hulman Institute of Technology '13
Carnegie-Mellon University Robotics Institute Summer Scholar Program
Supervisor: Dr. William (Red) Whittaker

ABSTRACT

This paper presents the methodology for designing and simulating a motion control algorithm for a Tyrolean-based descent and traversal robotic platform. A PID controller is developed which can perform a variety of traversal scanning trajectories including pulse and sine wave. The desired trajectory of the robot is first created using MATLAB and then utilized to set the velocities of each of the motors with respect to time. The path planning algorithm is modular and can be easily changed depending on the environmental constraints. The control algorithm deployed on the physical platform will then read the time v. velocity vector for each of the motors and compute the error between the ideal and the measurement to rectify its path trajectory. The parameters of a simulated plant model with a respectable uniform noise were used to calculate the desired PID gains of the motors. Simulation demonstrated that a stable controller with 5.82% overshoot, a settling time of 3.2 seconds, and control effort gain of 0.937 is achievable. The sampling rate of the discretized system done in the simulation is 0.005 seconds. A simulation for various velocities was performed to observe the steady state response of the controller.

***Keywords*—Tyrolean, PID, sky moonlight, moon, simulation, survey, discrete-time**

I. Introduction

First discovered in 2009 by Japan's Kaguya spacecraft and further inspected by NASA's Lunar Reconnaissance Orbiter (LRO), the "moon skylights" pose great interest for scientists since it can potentially serve as the foundation for possible colonization attempts due to natural protection from deadly electromagnetic radiation, meteorite bombardments, and large temperature variations [1][2]. Furthermore, scientists wish to explore these locations since it may consist of a network of underground lava tubes that can contain useful resources for future space exploration and colonization [1]. The mission objective consists of creating a robotic platform, called *Tyrobot*, which will be deployed to the moon in the near future. The robot is tasked with exploring the "moon skylights" or moon pit locations.

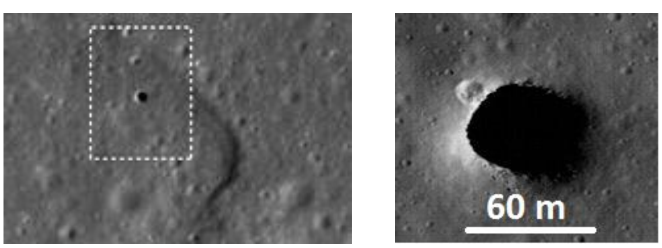

Figure 1. **Snapshot taken of the first moon skylight located on Marius Hills.** This picture was among the first set of photos captured by Japan's Kaguya spacecraft as it orbited the moon. The height of the dashed rectangle is about 1 km in length. The pit is large enough to fit the White House completely inside [1][2].

Figure 2. **A moon pit and the Capitol Building.** This figure helps grasp the scale and magnitude of the moon pit locations recently discovered on the moon.

This paper discusses and explains the control algorithm needed to deploy a robotic platform that can successfully explore and scan these locations in Earth-based test locations that resemble the moon pits. The *Tyrobot* consists of suspended platform that travels along a tightrope that is anchored down across the diameter of the pit. Once the robot is successfully suspended and deployed, it can traverse along the tightrope and lower or raise the carriage as commanded by a pre-determined path trajectory. As the robot moves along its trajectory, the attached sensor package will take thousands of laser scans and video data that can provide useful information about the composition and structure of these locations. Future improvements of the *Tyrobot* may even deploy a rover to autonomously explore the unknown surface and determine whether they consist of a network of underground lava tubes. These robots could be armed with radar-penetrating technologies to provide data and accurate models showing the stability and structural design of the underground lava tubes. Work on underground modeling applications on Earth has already been developed that maps out caves and tunnels using laser-range technologies [10].

Figure 3. **The proposed stages and lunar missions of the *Tyrobot.*** From left to right, the lunar lander will land in a location close to the moon skylight. It will then deploy a rover that will secure anchor points around the perimeter of the skylight and deploy the Tyrobot. The Tyrobot will then take scans of the wall structure and surface to discover information about the

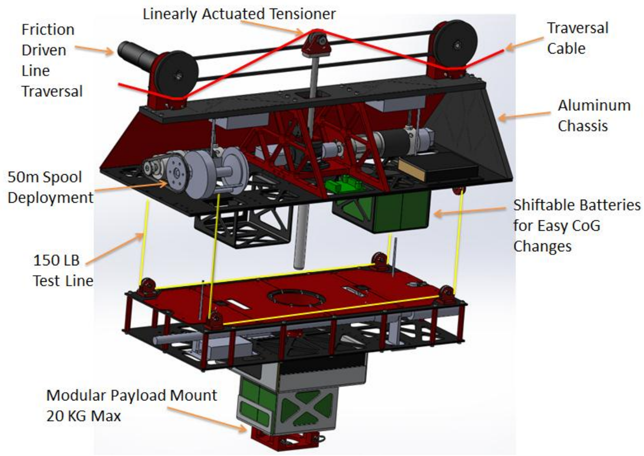

Figure 4. **Tyrobot robotic platform design.** This figure shows the overall design of the Tyrobot consisting of two sections: the carriage (top section) and platform (bottom section). The carriage houses all the actuators and motor controllers. The platform houses the computer and sensor package (not shown). The carriage and the platform communicate over radio signal.

Once the robot is deployed and suspended on the tightrope, the carriage will be able to traverse the width of the pit while lowering or raising the platform for data acquisition. The carriage's sensor packet, named *Ferret*, contains a LIDAR, an IMU, a camera, and a tracking prism. These sensors are used to capture video data about the pit and provide a 3D model. The tracking prism and IMU can provide the position and velocity of the platform as it moves. Sensor synchronization is provided via clock distribution system to match sensor scans, line tension, and position as measured by the ground survey system. The *Ferret* firmly attaches to the bottom of the platform (not shown) and remains static through the run.

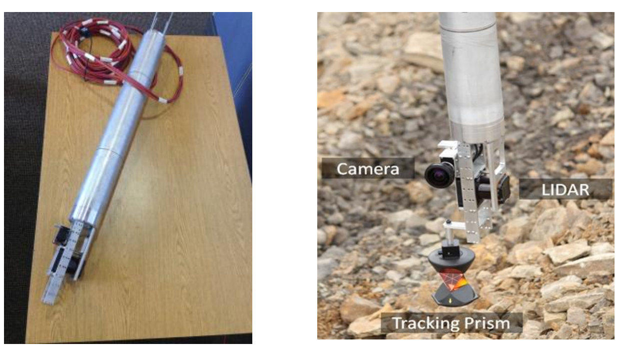

Figure 5. **Ferret sensor package.** This figure shows the sensor package that it is firmly attached to the bottom of the platform. This device is responsible for acquiring video data showing stratigraphic sequence of a rock wall and mapping a 3D model of a pit.

This paper focuses on the design, initial tests, and simulation of an algorithm that will control the traversal (x-axis) and winch (z-axis) motors via a PID closed-loop controller. Although other common controllers were also considered, like LQR and Ackermann pole-placement, a PID controller was implemented for this application to show proof of concept and due to the robustness and reliability that it can offer in uncertain domains. For future designs of the robot, a more complex and adaptive controller could be implemented once enough realistic field data is acquired.

Since this work consists of an innovative application, development of an effective control algorithm was based on several earth-based applications, like the control algorithm of a hoist, the control algorithm of the ACROBOTER, and the control theory of CMU's Ballbot, and mobile control lectures from MIT and Brown University [3-8]. Perhaps the most relevant work done that resembles a line survey platform consist of the *Expliner* robot [11][12]. This robot is used to perform tests and inspections on high-voltage lines while traversing along the lines. Several ideas and concepts concerning the mechanical design and control architecture were considered for the *Tyrobot.*

II. PROCEDURE

A. *Plant modeling via simulated data and initial PID testing*

The first step was to acquire some simulated data from a virtual motor since the actual device and unknown system was not constructed at the time. The simulated plant model acquired sets the motor velocities to a reference value and includes some simulated noise. The model records the Linux CPU time versus simulated motor velocity in ticks. A uniform 10% noise was added in the simulation.

In experimentation, the motor load should be incorporated in a physical test and measured by running in open-loop while creating a data log over time as velocities change to acquire a closer estimation of the motors by including inertia values, load of the motors, power consumption, and power output. The change in inertia and load can then be used to create a more accurate description of the plant which will then yield a more stable controller.

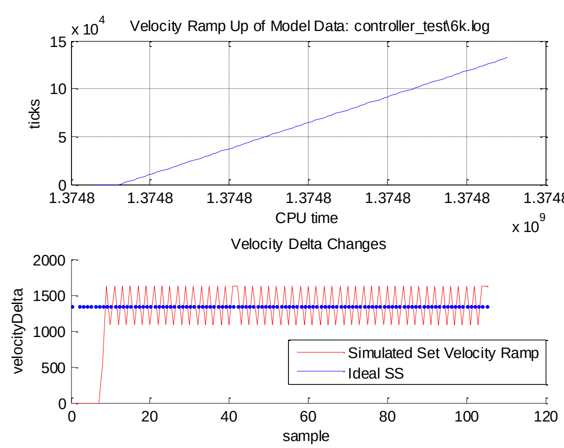

Figure 6. **Simulated plant model for PID tuning.** The figure on the top plots the accumulation of ticks as a function of CPU time. The figure on the bottom shows the change in tick position (i.e. velocity).

After acquiring a simulated velocity log file for 4 different settings, the corresponding differences of the motor velocity (in ticks) were plotted as shown in Figure 6. The ideal steady-state value (Ideal SS) can then be computed after isolating the noise. This ideal reference value is then used to come up with the desired motor inputs and to compute the controller gains.

B. *Designing and tuning a PID controller for Tyrolean traversal*

The design and tuning of the PID controller was done using MATLAB software tools. Using Simulink and Sisotool utilities, a PID controller was designed and tuned for the simulated plant models that would minimize the error or difference between the actual value and the command value [5][6]. At the time, several physical constraints of the robot were estimated as shown below.

Known system requirements for *Tyrobot*:

- Max traversal speed: 20 cm/s
- Max raise/lower speed (i.e. winch motor): 20 cm/s
- Operational time: 2.5 hours
- Max slope: 20 degrees
- Max payload power draw: 50 W
- Max deployment depth: 50 m
- Max payload mass: 15 kg
- Max carriage mass: 25 kg

Types of operation expected for the *Tyrobot*:

- Data acquisition while traversing tightrope;
- Raise/lower platform and traverse to follow pre-planned trajectory as defined by ground operator;
- Allow radio control communication for emergency retrieval;
- Computer communicate wirelessly to carriage compartment containing motor controllers;
- Increase angle wrap of tightrope to prevent slip during operation;
- Device subjected to pendulum motion and environmental changes such as wind;

The constraints and controller requirements taken into consideration consisted of having a percent overshoot (P.O.) of less than 15% and a rise time of less than 5 seconds. In an ideal scenario, these constraints would correlate to having a control architecture that achieves stability within 5 seconds. Since the robot's max speed is a slow 20 cm/s, achieving steady-state operation in this time makes the robot more reliable and robust to changes caused from abruptly changing speeds. However, due to the nature of the simulate log file for the motors that was used to create the controller, physical system limitations are not being taken into account. This includes the control effort, given the max speed, that the motor can physically attain and stall torque power when the motors need to hold the rotational speed of the platform at zero. The controller was discretized using zero-order hold and a sampling time selected consisted of 0.005 seconds was selected.

The controller takes in desired and simulated velocity commands and sets the gain for the unknown system to achieve those values within the specified constraints. In the physical implementation, position and velocity feedback is provided via encoders attached to the motors. Using the clock synchronization from the computer, the control algorithm deployed on the survey platform will receive a list of velocity commands with a time stamp attached to it. It will then verify that the actual velocity commands are matching the reference values and corresponding timestamps.

	Run 1	Run 2	Run 3	Run 4
Sampling Time (s)	0.010	0.010	0.005	0.005
Filter Coefficient (N)	2.430	2.430	2.430	2.430
Rise Time (s)	1.360	0.670	1.030	0.340
Settling Time (s)	4.210	2.110	3.200	1.060
P.O. (%)	5.880	6.040	5.820	6.040
Controller Effort	0.709	1.400	0.937	2.830
Kp	1.101	2.197	1.451	4.383
Ki	0.010	0.851	0.017	0.154
Kd	-0.519	-0.519	-0.519	-0.519

Table 1. **PID configurations with varying controller constraints.** The controller gains were computed for various controller requirements including rise time, P.O., and sampling time.

Each of the controller configurations shown above were modeled and tuned to determine a feasible result for the simulated plant. Configuration 3 (i.e. Run 3) was chosen due to its settling time, overshoot, and control effort gains (highlighted). The process for selecting these parameters relied on the control system requirements and the type of application being developed. In a physical scenario, the controller's max effort gain is estimated to be around 0.93 thus allowing a gain of 0.07 (i.e. 7%) in case more effort is needed to achieve settling motion. Since fast and abrupt changes to the velocities are detrimental to the *Tyrobot*'s operation, a settling time of 3.20 seconds is acceptable. Once the controller gains were set, the controller was tested for various inputs to observe its stability and settling time. Also, only PID configurations were tested (i.e. P,D, PI, PD not considered) since the P and D controller affect the rise time and P.O. while the I controller helps with the steady-state settling time [5][6][9].

C. *Path making program for Tyrolean survey*

To take data of area of interest (e.g. a moon pit or an Earth-based test location), a path planner algorithm was created that generates a velocity command file containing time, traversal motor velocity, and winch motor velocity vectors. Although physical data was not available at the time of this work, the paths created are vertical in path to minimize the work of the winch motor due to gravity.

The program that makes the desired Tyrolean trajectory (pulse wave or sine wave) is modular and can be easily changed to accommodate for the test site. In particular, the width and height of the pit location (in meters), an offset distance (to account for unexpected irregularities in the site and hanging objects from the Tyrobot), and the period of the wave can be edited and entered as needed. Once the pit settings are measured and defined, the user needs to specify a run time (in seconds) for the data gathering trajectory. The top left coordinate of the pit is defined as the origin (0,0) while the winch motion of travel goes along the negative z-axis.

The functions used to make the paths are parameterized with respect to time so a velocity-time command file can be created. The algorithm does not take into account the slack of the traversal rope. It assumes an ideal case scenario where the traversal rope is perfectly tensioned. Thus the physical experimentation will be initially affected by these unaccounted forces. These repercussions should be dealt when data becomes available.

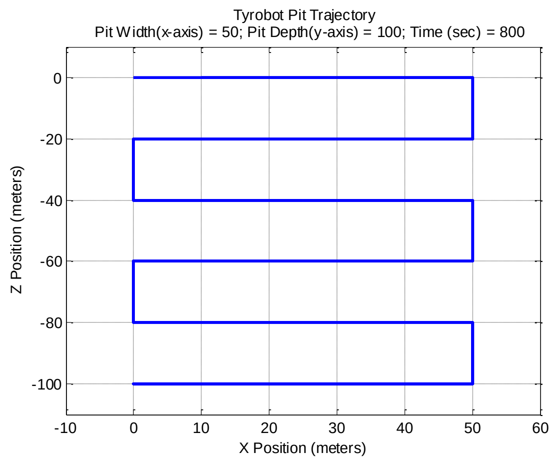

Figure 7. **Pulse wave Tyrolean trajectory.** The user can select a step-like trajectory for data gathering once the pit dimensions and run time are properly tested. Here the robot starts at the origin and proceeds along the blue line. Once it gets to the bottom, the Tyrobot does a second sweep of the bottom before proceeding upwards if time remains.

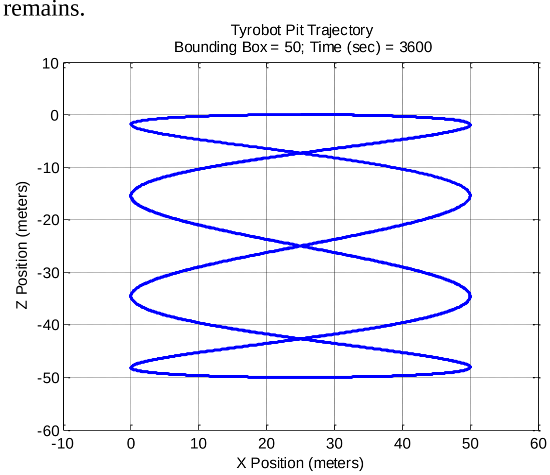

Figure 8. **Sine wave Tyrolean trajectory.** The user can select a sine-like trajectory for data gathering once a bounding box dimension of the test area (in meters) and the run time are specified. Here the Tyrobot starts at the top middle point and proceeds along the blue line. If time remains, the robot just keeps on repeating its path.

From the generated path files, the position of each of the motors can be known independently and plotted versus time. Although not currently designed for use, another file containing the position and time of the traversal and winch motion is also created by this program. This can later be tested against encoder feedback and data from the tracking prism to obtain the error of the trajectory and refine the control algorithm to take into account such errors. From the position-time command file and plot, the derivative is then taken to compute and output the velocity-time command file for the traversal and winch motors.

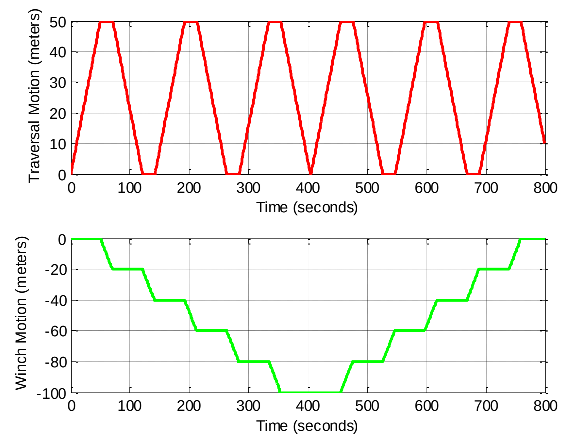

Figure 9. **Position versus time plot for the traversal and winch motions when using a step wave.**

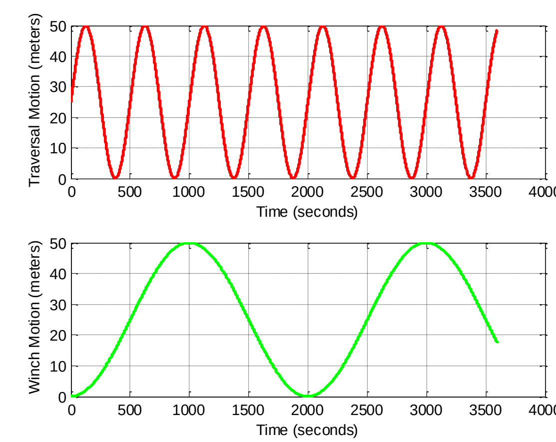

Figure 10. **Position versus time plot for the traversal and winch motions when using a sine wave.**

From the parameterized traversal and winch motion commands, the positions of the motors with respect to time can be known. In the figures above, the red plots represent the traversal positioning while the green represent the winch positioning through the timed run for data acquisition. It is important to keep in mind that the step wave increments or decrements discretely which in turn affects the velocity commands by changing the commands abruptly rather than gradually. Thus the sine wave poses a trajectory that can be stable due to its gradual motion.

Looking at the velocity plots (Figure 11 and Figure 12), one can predict how the changing of speeds will alter the swinging motion of the platform especially when the platform is at its lowest points. Simply changing the gain of the speeds and having the platform operate very slowly (e.g. max motor velocity for traversal and winch is 10%) can dampen the swinging motion of the platform especially as it reaches its lowest descent points. To achieve higher stability and minimize the oscillations due to swinging motion, data from entire testing scenario that incorporates the pendulum motions, the slack of the traversal rope, and the external forces acting on the Tyrobot can be modeled to tune the control algorithm.

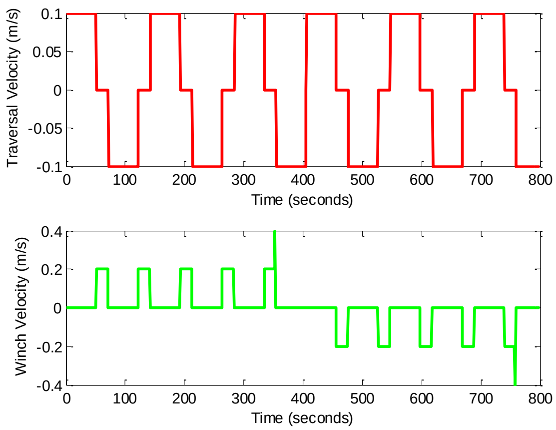

Figure 11. **Velocity versus time plot for the traversal and winch motions when using a step wave.** Using a limiter of 10% output, the max traversal velocity is 0.1 m/s (system requirement). The winch was capped at 20% output.

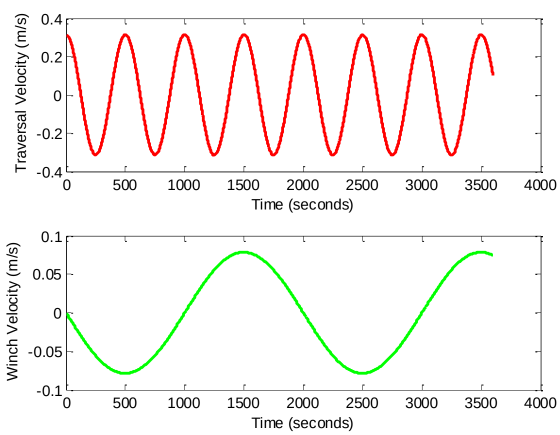

Figure 12. **Velocity versus time plot for the traversal and winch motions when using a sine wave.**

III. RESULTS

A. *PID simulation results for Tyrolean velocities*

After obtaining the desired controller configurations and controller gains from some desired system requirements, the controller was tested for different speeds to achieve its steady-state response. The simulation shows that the stability of the scenario is achievable in less than 5 seconds with less than 10% overshoot. Figure 13 shows the simulated tests done in MATLAB.

The next step would be to test the controller against the physical hardware in a real testing scenario to observe a more adequate system response. Due to lack of raw data and unexpected forces not taken into account when performing the simulation, it is expected that the controller might need tuning after several dry runs. The control effort yielded by the controller maintains the motor under 100% of maximum output.

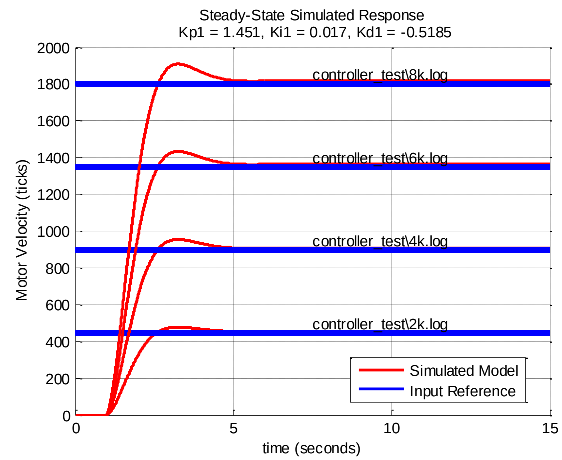

Figure 13. **PID controller steady-state response.** This plot shows the working PID controller done in simulation for several velocity reference inputs.

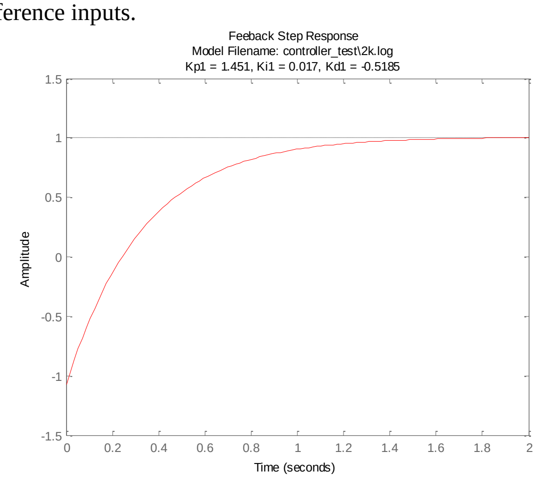

Figure 14 **Feedback step response plot.** This shows the feedback step response of the simulated controller.

IV. FUTURE WORK

Future work will focus on tuning and perfecting the controller once actual test data becomes available as scenarios of the real surveying conditions are performed.

In the short term, possibilities for tuning the controller include gathering time data logs with motor velocity and position values as reported by the encoders. From there a more accurate description of the plant can be correctly modeled and a controller can thus be designed that also includes acceleration and jerk. By limiting the acceleration, the stability and robustness of the control architecture can be improved. Also, the corresponding velocities of the motors, in either cm/s or m/s, have to be measured against ticks/s when the motor is under load conditions. From there a more accurate representation of the path and velocity command file can be created.

In the long term, a more accurate projection of the trajectory can be obtained that takes into account the swinging motion of the pendulum as a function of distance away from the carriage. The errors and differences of the path trajectory due to the slack of Tyrolean rope can also be calculated and accounted for. In addition, IMU data (accelerometer and gyroscope) from the sensor package located in the platform can be used to minimize the damping of pendulum, thus reducing uncertainties in the system.

V. SUMMARY

My work consisted in creating and simulating a control algorithm for a Tyrolean survey robotic platform that will be launched to the moon. Simulation validates that stability of the system is achievable in 3.2 seconds with an overshoot of 5.82% and control effort gain of 0.930. Tests of the physical device are needed to perfect the control algorithm for smooth data acquisition. Future work will focus on perfecting the algorithm by implementing unaccounted variables.

VI. REFERENCES

[1] B. Handwerk. (2009, 10, 26). *First Moon "Skylight" Found – Could House Lunar Base?* [Online]. Available: http://news.nationalgeographic.com/news/2009/10/091026-moon-skylight-lunar-base.html

[2] D. Coulter. (2010, 7, 12). *Down the Lunar Rabbit-Hole.* [Online]. Available: http://science.nasa.gov/science-news/science-at-nasa/2010/12jul_rabbithole/

[3] W. Su. "A Model Reference-Based Adaptive PID Controller for Robot Motion Control of Not Explicitly Known Systems." Vol. 12. No 3. September 2007. Int'l Journal of Intelligent Control and Systems

[4] G. Stepan *et al*. "ACROBOTER: a ceiling based crawling, hoisting and swinging service robot platform," Budapest Univ. of Tech. and Econ.

[5] C. Jenkins. "Lecture 4: Control Theory and Robot Dynamics" in *CS148-Building Intelligent Robots.* Brown Univ.

[6] C. Batten. "Control for Mobile Robots" in *Maslab IAP Robotics.* Jan. 2005.

[7] U. Nagarajan, G. Kantor, and R. Hollis. "Integrated Planning and Control for Graceful Navigation of Shape-Accelerated Underactuated Balancing Mobile Robots." May 2012. IEEE ICRA. CMU Press.

[8] U. Nagarajan, G. Kantor, and R. Hollis. "Trajectory Planning and Control of an Underactuated Dynamically Stable Single Spherical Wheeled Mobile Robot." May 2009. IEEE ICRA. Kobe Int'l Conference Center. CMU Press.

[9] R. Throne, *Linear Control Systems*, Rose-Hulman Institute of Technology, Chapter 9, 10, 11, 14.

[10] U. Wong, A. Morris, C. Lea, J. Lee, C. Whittaker, B. Garney, W. Whittaker. *Comparative Evaluation of Range Sensing Technologies for Underground Void Modeling.* In Proc. Intelligent Robotics and Systems (IROS), 2011.

[11] P. Debenest, M. Guarnieri, K. Takita, *et al*. "Expliner – Robot for Inspection of Transmission Lines." May 2008. IEEE Int'l Conf. on Robotics and Automation, pp. 3978-3984.

[12] P. Debenest, M. Guarnieri, K. Takita, *et al*. "Expliner – Toward a Practical Robot for Inspection of High-Voltage Lines." May 2008.