

Automatic Control of a Soft Trunk Robot Actuated by Strings

Jacob Trivisonno

*Intelligent Control and Robotics Lab
The University of Rhode Island
jtriv135@uri.edu*

Xiaotian Chen

*Intelligent Control and Robotics Lab
The University of Rhode Island
xiaotian_chen@uri.edu*

Cameron Amaral

*Intelligent Control and Robotics Lab
The University of Rhode Island
cama083001@uri.edu*

Emadodin Jandaghi

*Intelligent Control and Robotics Lab
The University of Rhode Island
emadjandaghi@uri.edu*

Leonardo Garofalo

*Intelligent Control and Robotics Lab
The University of Rhode Island
lgarofalo@uri.edu*

Chengzhi Yuan

*Intelligent Control and Robotics Lab
The University of Rhode Island
cyuan@uri.edu*

Abstract—A soft robot is a robot made of soft materials and is capable of doing complex tasks. The soft robot that was created in this experiment is a soft trunk robot actuated by strings. The purpose of this robot is to be able to move around small spaces easily and be able to perform complex tasks. For example, positioning a tip mounted sensor to gather data. From the results in the experiment, the robot can move to a target position when given a set of coordinates by using a P-controller.

Keywords—P-controller, Soft Robot, Actuation

I. INTRODUCTION

Soft robotics is a growing field in robotics. To be able to achieve a “soft” robot, it needs to be constructed with soft materials, such as silicone. There needs to be as little rigid structures used as possible. Soft structures make the robot as flexible as possible, which can be exceptionally convenient for completing certain tasks, but this introduces many problems with it. One major problem is to accurately control the soft robotic system. While there are multiple ways to control the soft robot, each method comes with its own advantages and disadvantages. The different methods of actuation are electric-driven, magnetic-driven, pressure difference, and displacement difference [1].

The first method of actuation is electric-driven. Electric energy can be used to deform soft materials. One type of system used are dielectric elastomers, which react when electricity is applied to them. Dielectric elastomers have a very quick response time; however, they need extremely high voltages to work. Furthermore, some more advantages of dielectric elastomers are that they possess high forces and large strains [2]. Another example of an electric-driven material are piezoelectric materials. Piezoelectric materials produce a voltage when mechanical forces are applied to them. A couple of example materials that have been experimented on are quartz, topaz, and salt. All of these crystals produced a surface charge, and the discovery was called piezoelectricity [3]. The major problem with these materials is that they are dense and stiff. However, to avoid this problem, they can be incorporated into soft materials through geometric patterning [1].

The next method of actuation is magnetic-driven. Magnetic-driven actuation is similar to electric-driven actuation. When certain soft materials respond to a magnetic field, their

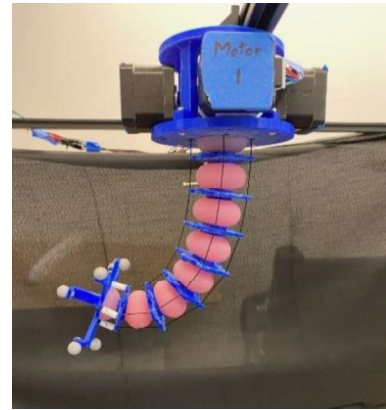


Fig 1: Soft trunk robot actuated by strings

dimensions and stiffness change with the magnetic field. Additionally, a couple advantages are that magnetic fields can go through a broad range of materials, their spatial gradients can be created over the space, and compared to other kinds of actuation, a soft material's response to a change in magnetic field is rather quick. Furthermore, these materials must possess a low elastic modulus and be mechanically strong, with significant toughness and tear resistance. Magnetic actuation has been demonstrated to be especially useful with certain tasks, such as invasive surgery [4].

The following method of actuation is a pressure difference actuation. This type of actuation uses pneumatic or hydraulic systems to control the soft robot. Pneumatic and hydraulic systems are actuated by the pressure of a gas or liquid to perform their tasks. These types of systems have many strengths when used underwater compared to other forms of actuation. Additionally, hydraulic actuators have a high-power density, making them suitable for applications requiring a lot of force and torque. Pneumatic actuators have less power and more complicated control because of the compressibility of the gas [1].

The last method of actuation mentioned was a displacement difference. This was the main method used in the experiment. There are four wires placed from the base of the robot to the tip of the robot and these wires were actuated by four stepper motors. These wires are held in place by retainers placed in

between sections of the soft robot. There are multiple ways string can be used. They can either be pulled, like in this experiment, or twisted. The strings are supposed to represent artificial muscles. Also, strings have a high strength to weight ratio and have low mass. Using strings and stepper motors as actuators have a high potential for soft robotics in the future [5].

Compared to the other actuation methods mentioned beforehand, pulling strings for actuation allows for more flexibility, and more lifelike control. Additionally, the use of steppers allows for angle control which results in reliable positioning. Furthermore, an advantage with strings is that they can be any size or length. Nylon string is used in this experiment with a 0.8 mm diameter and the length can be changed for any specific case. No particular length was decided for this specific experiment. In addition, strings can be used as actuation in other applications to represent artificial muscles. In other experiments, they were used to represent trunks, arms, or hands [5]. However, in this experiment no particular ideas were in mind when creating the robot. First, we created a very simple prototype for the trunk. The sections going down the trunk were disk-like, and it was relatively short. As more iterations were made to the design, each led to an improvement which resulted in the current rendition. The sections were no longer disk-like, they were rounded off to be more spherical, and the sections decrease in diameter going down the trunk. Additionally, the hardness of the silicone was adjusted to provide a more solid body and the length was increased to as long as the bed of the 3D printer can make. All of this was done to increase the stability and functionality of the soft robot, along with making the design of the robot look more refined.

Now, there are several types of control strategies. A couple of methods for controlling soft robots are using the Finite Element method and Proportional Integral Derivative control [1]. The rest of this paper will be structured as follows. Section II will be about controlling the soft robot. Section III will be about the control methods of our experiment. Section IV will discuss the results of the experiment. Finally, section V will be the conclusion.

II. BACKGROUND AND RELATED WORK

There have been many developments in the soft robotics field, and they all focus on construction, control and applications of the soft robot. This section will have a brief summary of other works and different controllers.

Soft robots are made entirely of materials that can deform into any shape and the design of them are supposed to resemble living organisms. Some examples of a design for soft robots are an elephant trunk or the tentacle of an octopus. This is to make robots more natural and to stray away from the common rigid bodies for robots [6]. The current goal of soft robotics is to create effective control methods. However, soft robots are very hard to predict because they have infinite degrees of freedom and are extremely nonlinear. An accurate analytical model needs to be created to control the robot and make use of its malleable nature.

Soft robots have many control methods for their complex structure. Controllers such as FEM and PID are widely used in the world of robotics, and they have their own advantages and disadvantages compared to other control methods. For a simple

controller, a PID controller should be used, and it is used in this experiment. Finally, accurate analytical models for soft robots are extremely difficult to create, so most control methods use model free control methods.

The Finite Element Method (FEM) is a popular option for controlling a soft robot. FEM eliminates the need for an explicit analytical model and offers a practical method for handling nonlinearities in soft robotics. Some of the reasons why FEM is highly used in soft robotics is that it can handle large deformations from the robot. It can also be used to anticipate performance and assess the capabilities and limitations of soft actuator designs with varying inputs. Additionally, the use of finite element modeling can help us better understand the stress concentration and strain distribution in soft robots [7]. In a similar experiment, Wu et al., conducted an experiment with a soft trunk robot with FEM control. The robot was difficult to control, and to remove the problem, they used a FEM simulator to control the robot, called SOFA. With the help of SOFA and designing a PID-like controller, they were finally able to achieve their goal of controlling the robot with FEM-based gain-scheduling control [8].

This experiment was conducted using PID control. PID control or Proportional Integral Derivative control, is also a very common use of control for robots. It is widely used because it is extremely simple and it has intuitive tuning processes. However, PID control has many flaws compared to other control methods, but it is still a viable control method [9]. Another good point about the PID controller is that not every component needs to be present. Other versions of the controller consist of the P-controller, PI-controller or PD-controller. The main type of control used for this experiment is P-controller. The P-controller only uses the proportional action of the system. This topic will be further explained in the next section of the paper.

III. P AND PID CONTROLLER

This section will explain the formulation of P and PID controllers in the experiment, and the use of piecewise functions to accommodate multidirectional movement based on current position.

A. Equations

PID bases its computation off the error in the system: e , and three gains: k_P , k_I , and k_D . k_P is the proportional action of the controller, k_I is the integral action of the controller and k_D is the derivation action of the controller. These gains when combined with their error components create the following equation:

$$U(t) = k_P e(t) + k_I \int e(t) dt + k_D \frac{de(t)}{dt} \quad (1)$$

This experiment only uses P-controller so k_I , and k_D are equal to zero in this case.

In this experiment four motors are used, and each motor has its own controller equation. Since there are four motors, there are four equations to find U from specific k values. The above-mentioned equations are:

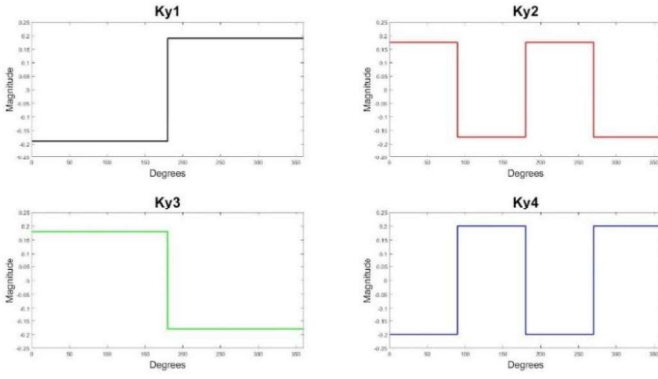


Fig 2: Piecewise K_y functions

$$U_a = k_x(x_o - x) + k_y(y_o - y) + k_z(z_o - z) \quad (2)$$

where the variable U is the number of ticks for the motor to move and each k is a constant in terms of $\frac{\text{ticks}}{\text{mm}}$. One tick is equal to 1.8 degrees of rotation for the stepper motor.

The k values that provide the most ideal results with the error converging to zero and the fastest settling times are:

$$k = \begin{bmatrix} 0 & -0.190 & 0.205 \\ -0.170 & -0.175 & 0 \\ 0 & -0.180 & -0.200 \\ 0.195 & -0.200 & 0 \end{bmatrix}$$

The magnitudes for k were found through many trials and the signs of k were determined based on what axis the motor is on. The sign of k ensures that the robot pulls in the correct direction. For example, motor two is on the negative x axis so it has a negative value of k_x . Also, some k values are zero because the motor is perpendicular to that axis, therefore they have no effect on that direction. Using motor two as an example $k_z = 0$ due to the motor lying on the x axis. With the motor positioned on the x axis, regardless of the motor pulling or releasing the string, it cannot make the robot move along the z axis, therefore it has no effect on the z direction.

B. Piecewise Functions

In order for the robot to bend in the correct direction conditions are required for the values of k_y based on the motor's location in the x-z plane. As shown in figure 2, the k_y values are either positive or negative depending on the quadrant in the x-z plane that the robot is located. Using motor two as an example, if the robot is bent in the positive x and positive z direction then k_{y2} is positive, however if the robot is bent in the negative x and positive z direction then k_{y2} is negative.

There is one more condition for the robot and it ensures that it can properly move vertically when returning to the origin. The condition states that when the robot is within x: (-5, 5) mm and z: (-5, 5) mm, all k_y values are negative. Once met, if there is an error in the y direction, the k_y values cause each motor to pull the robot in their respective directions. When all motors pull, they counteract each other resulting in vertical motion.

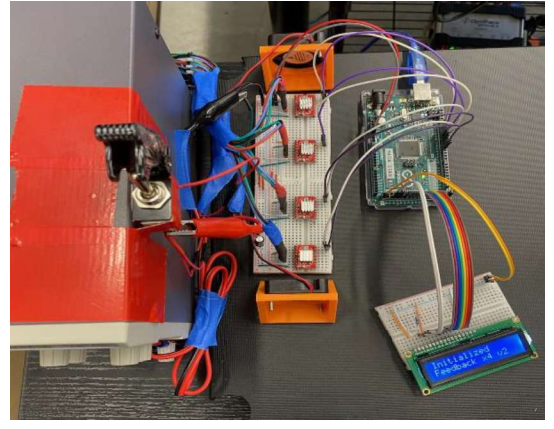


Fig 3: Electronics that control motors



Fig 4: Top (left) and bottom (right) part of the mold and silicone used (Ecoflex 00-50)

IV. EXPERIMENTAL DATA

The purpose of this section is to present data of the soft trunk robot moving to certain coordinate points. Additionally, a summary on the fabrication and actuation of the robot.

C. Fabrication and Actuation

When designing the shape of the trunk, an array of "flattened spheres" that taper off towards the tip was chosen. The section at the base of the robot started at 1.5 inches in diameter, which tapered off to 1 inch in diameter at the tip, using a quadratic pattern for the most ideal results.

To achieve the properties of a soft robot, the trunk of the robot was created by casting silicone in a mold. 3D models of both halves of the mold, as shown in figure 4, were modeled in SOLIDWORKS and 3D printed using polylactic acid filament, also known as PLA. The two halves of the mold are identical except for the fill holes which were located in the top mold. To cast the trunk, Ecoflex 00-50 A and B are poured into a mixing chamber in equal parts and mixed together thoroughly. Finally, the solution is poured into the mold and left to cure for three hours.

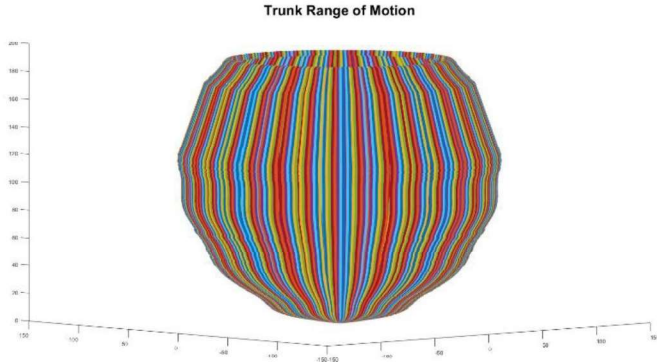


Fig 5: Estimated range of motion of the trunk

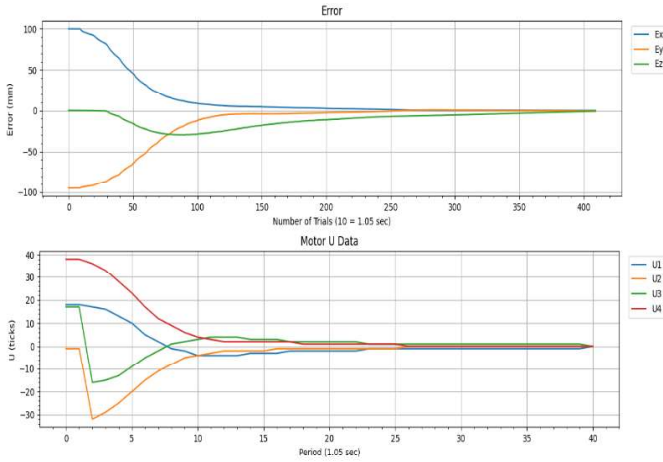


Fig 7: The target position for the second test was (100, -95, 0) mm.

In order to link the motion of the strings to the trunk, retainers were needed to hold the string around the trunk. The design of the retainers followed the taper of the trunk sections. They were modeled and printed in halves, along the same manner as the molds, and are connected via small pins. Additionally, the base of the robot that holds the four stepper motors was printed allowing for the robot to be hung from an elevated support.

The robot is mounted hanging in the positive y axis with motors one through four being mounted on the: positive z axis, negative x axis, negative z axis, and positive x axis, respectively. To move the trunk, each stepper motor has a spool of nylon string, which is threaded through the retainers and attached at the tip. When the stepper motors are powered, the robot moves by either pulling or releasing the strings. Each motor is wired to an Arduino Mega 2560 R3 board through an A4988 motor driver chip which supplies the motors with the 12 volts needed to move, based on signals from the Arduino. A switch is also wired between the driver chips and the power supply to function as an emergency shut off switch. Two fans are wired directly to the power supply to allow for proper airflow over the driver chips. Also, an LCD screen is wired to the Arduino Mega board which displays the U values sent over

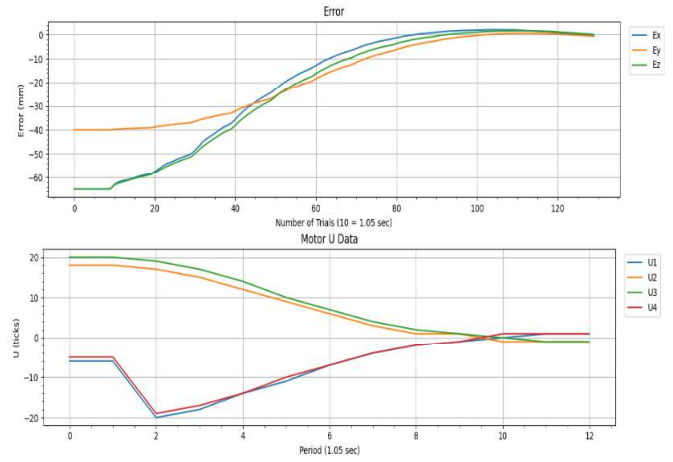


Fig 6: The target position for the first test was (-65, -40, -65) mm.

serial which allows for easier debugging. Finally, the Arduino board is connected to the computer via USB.

Furthermore, to get the positional data of the soft robot, eight cameras are encircling the soft robot from below mounted to a frame and they send the robots positional data to the computer through the Motive software created by OptiTrack. The cameras use an infrared detection system, which emit infrared light that is then reflected off objects. The cameras then detect how much light is reflected back to determine the position of the object. A total of six reflective balls were attached at the end in an asymmetric arrangement to provide accurate positioning and determine orientation.

An approximated usable range of the robot can be visualized by figure 5. Using graphed positional data of the robot revolved around the y axis, its range of motion can be visualized. The graph is relatively vase shaped due to the mechanics of the trunk. As the strings pull the robot it bends outwards increasing in radius and height until it reaches a maximum and the radius starts to decrease with the height still increasing. According to tests done in the lab, the robot has the ability to reach its base which is about 180 mm above the tip, and it has a maximum outwards range of about 100 mm from the origin when in full extension.

D. Results and Discussion

For all trials displayed, the robot was initially located at the origin position, which is determined by the location of the tip at the time the robot is given its first target. The robot has the functionality to go to multiple unique target locations in a row while keeping the same origin position, however for the following trials only one target was given per trial. The results provided will show a graph of the number of ticks for each movement, and the error as the robot approached its target.

Figure 6 shows the data from the first trial. The target for this trial was (-65, -40, -65) mm and the final position the robot was able to reach with the given coordinates was (64.580, -39.319, -65.091) mm, giving an error of (-0.420, -0.681, -0.091) mm. The y curve was nearly critically damped and had no overshoot however, the x and z curves had slight overshoot. All

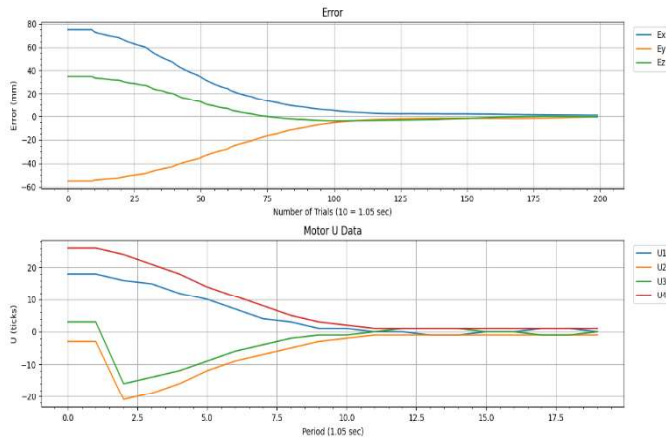


Fig 8: The target position for the third test was (75, -55, 35) mm.

coordinates settled in around the same amount of time which was about 10.5 seconds. For the final errors in the system, the z positional error was exceptionally low, with the x and y positional errors within the error margins of 1 mm. When compared to the trials to come, this trial had a quite fast settling time, and an impressively low error in the z direction. The fast settling time is due to the target x and z values having the same magnitude causing the target to lie directly between two motors, making targets like this easy for the controller.

For the second trial, figure 7 shows the data for a target of (100, -95, 0) mm. The final position achieved from the given coordinates were (100.079, -94.486, 0.895) mm, giving an error of (-0.079, -0.514, -0.895) mm. The x and y curves are both nearly critically damped with slight differences in their curves. The z curve is different from the other curves due to a unique situation that occurs along the axes. The robot cannot be exactly on an axis so it will determine what quadrant it is in, then the corresponding motors will pull to reduce the error in x and y. Once the x and y have a relatively small error and the z has a comparably larger error, the controller compensates by pulling the z back in line. The slowest coordinate in this trial determines the total settling time which was z which has a settling time of about 37 seconds. This is very slow due to x and y having a settling time of about 26 seconds, and the other trials both having a significantly faster settling time. The final errors for y and z were acceptable being within the 1 mm margin, and the error for x was exceptionally small.

For the third trial with data shown in figure 8, a target of (75, -55, 35) mm was chosen. The final position the robot was able to achieve with the given coordinates was (74.087, -54.533, 35.587) mm giving an error of (0.913, -0.467, -0.587) mm. Y and z are both damped effectively with y being slightly underdamped and z being slightly overdamped. Due to the system prioritizing the line $x = z$ at first, the x component was underdamped and took the longest to reach a steady state. X determines the settling time of the system due to it having the largest settling time of about 18 seconds while the y and z components had a slightly faster settling time of 16 seconds. This is a fairly good result being between the previous trials and is a relatively average result when compared to all other targets.

All the final errors in the system are also average with all components being within the error margin of 1 mm.

Most of the results were within the margin of error of 1 mm, however some results were exceedingly better than others. The first and second trials were able to produce some results with an error of less than 0.1 mm. The settling times of the trials were relatively inconsistent with the first trial having a very fast settling time of under 11 seconds and the second trial having a very poor settling time of over 35 seconds. With the target of the third trial being what most use cases would be, having no correlation between the x, y, and z coordinates, the expected settling time of this system should be around 18 seconds. This shows that targets around 45, 135, 225, and 315 degrees will have the fastest settling times and they will increase for targets around 90, 180, 270, and 360 degrees which will have the slowest settling times. If integral and derivative components were added the systems error would be further reduced and the settling time would improve in addition to reducing any overshoot behavior in the system [9].

V. CONCLUSION

In this paper, a P-controller was used to move a soft trunk robot actuated by stepper motors and wires. We were able to automatically move the robot to certain target coordinates, and mitigate error with an acceptable average settling time in most use cases. In the future, an integral and derivative component can be added to improve the system performance. As time goes on, different materials, and different control methods will be created that will increase the potential for soft robots and someday, they will be able to near perfectly imitate or even outperform the capabilities of living things.

VI. ACKNOWLEDGEMENT

This work is supported by National Science Foundation under Grant CMMI-1929729.

REFERENCES

- [1] Wang, Jue, and Alex Chortos. "Control Strategies for Soft Robot Systems." *Advanced Intelligent Systems* 4.5 (2022): 2100165.
- [2] Ackerman, Evan. "Electric and Magnetic Fields Drive Soft, Flexible Robots." *IEEE Spectrum*, IEEE Spectrum, 24 June 2021.
- [3] Sohn, J. W., and S. B. Choi. "Various robots made from piezoelectric materials and electroactive polymers: a review." *Int. J. Mech. Syst. Eng.* 3 (2017): 122.
- [4] Chung, Hyun - Joong, Andrew M. Parsons, and Lelin Zheng. "Magnetically controlled soft robotics utilizing elastomers and gels in actuation: A review." *Advanced Intelligent Systems* 3.3 (2021): 2000186.
- [5] Bombara, David, et al. "A twisted string actuator-driven soft robotic manipulator." *IFAC-PapersOnLine* 54.20 (2021): 141-146.
- [6] Della Santina, Cosimo, Christian Duriez, and Daniela Rus. "Model based control of soft robots: A survey of the state of the art and open challenges." *arXiv preprint arXiv:2110.01358* (2021).
- [7] Xavier, Matheus S., Andrew J. Fleming, and Yuen K. Yong. "Finite element modeling of soft fluidic actuators: Overview and recent developments." *Advanced Intelligent Systems* 3.2 (2021): 2000187.
- [8] K. Wu and G. Zheng, "FEM-Based Gain-Scheduling Control of a Soft Trunk Robot," in *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3081-3088, April 2021, doi: 10.1109/LRA.2021.3061311.
- [9] "PID Control," *Autonomous Robots Lab*, <https://www.autonomousrobotslab.com/pid-control.htm>