PROGRESS REPORT FOR ENGR 190: SEGWAY PROJECT

Our assignment was to design, build, and program a Segway robot that would be able to balance itself and operate without outside interference or assistance. This project has presented us with many challenges, from testing and choosing a design to writing a correctly functioning program in MatLab.

We first started out with a design found on the internet called the NXTWay (cite) but soon found that the NXTWay design was not going to work for us. Next we decided to modify the design to try and lower the center of gravity and point of rotation of the axis by moving the wheels to the outside rather than underneath the robot and also moving the NXT down as low as we could put it. After continued testing we found that lowering the center of gravity made it so that our hardware and software could not successfully run in the manner that was needed to achieve balance and stability. So, we then decided to go with the Segway design found on a link from the LEGO Mindstorms webpage (cite), which consisted of a robot with a much higher center of gravity and an attached "rider". This "rider" helped to balance the vehicle and could possibly be mechanized at a future point once the balance programs had been finalized. We have also decided to add a second Light Sensor to help with obtaining accurate readings. After running tests on the robot using both light sensors we determined that in order to make a more precise reading from the second sensor we would have to modify the design again to allow for a lower mounting point for the second light sensor (Note: both light sensors are mounted on the bottom of the robot, one in the front and one in the rear of the chassis). And now both of the sensors are at approximately the same height from the surface. This allows for more consistent readings. Since we have had the best results by far we have decided to stick with this design for the remainder of the project.

Our design was not the only thing to undergo many changes. We have constantly been changing and rewriting our program for the robot. Due to the constantly fluctuating light in the project lab we have to continuously change our "target" for the light readings and then fix the other variables that depend on the "target". Also, we have opted to use a free whiteboard as our test sheet because it is smooth and one continuous color, unlike the test pad. A significant change to the program occurred when we decided to use "direct motor command" instead of "NXTMotor" because it offers a faster response and quicker activation of the motors. We have also replaced the "GetLight" commands in the program loop with a set variable "current" so that the light sensor will not a return more than one reading per loop. This allows the robot to act more accurately and with less confusion. Also, upon adding the second light sensor we have began using more complex commands such as "If-Or" instead of just "If" statements, which allows for more flexibility and for the reading of multiple sensors at one time.

We have not fixed all of the bugs, but the problems yet to be solved are few though important. A difficult challenge yet to be overcome is finding the balance between reactivity and power. We've written programs that have the necessary fast reaction time, but lack the power to keep the robot upright. We have also written programs that allow for the necessary power for the corrections, but it does not have the quick reaction time. To solve this we are working on splicing these two programs together, though so far we have been unsuccessful.

WORKS CITED

DESIGN 1:

NXTWay → http://philohome.com/nxtway/nxtway.htm

DESIGN 2:

Segway w/ Rider → http://www.nxtprograms.com/NXT2/segway/steps.html