WobbleBot

Team 25 – Marc Backas, Phillip Lovetere, and Mingrui Zhou
ECE 445 Design Document – Fall 2019
TA: David Null

1 Introduction

1.1 Objective

Problem: It is a challenge to build a robot that is capable of balancing atop a cylinder. Many people studying robotics or controls would be interested in experimenting and improving upon a robotic platform like this, and it does not currently exist.

Solution: We will build such a robot in an affordable way, using hardware and software that enables precise realization of the control algorithm chosen by the researcher. The robot will measure data, process it, and actuate its motor so that it will stay on top of the cylinder. Once this project is complete, it will be an accessible development platform for those studying robotics and controls.

1.2 Background

People have been designing these kinds of robots for years, using different kinds of sensors, algorithms, and actuators to make them better and better. A good example of such a robot is our very own Professor Dan Block's Segbot [1]. Today, fast processors and algorithms exist along with precision sensors and actuators, making these kind of robots possible.

This robot is an example of a dynamic, unstable system. In order to create and control such a dynamic system, many design choices must be made. First, the system itself must be designed and manufactured in a way that is predictable and controllable. An accurate system model must be developed so that a good controller can be designed. The controller is then installed into the system, used to precisely and predictably steer the unstable system towards stability, as fast as possible. This particular robot was chosen for the complexity of the robot's dynamics. It will take intelligent system design, accurate system modeling, and excellent controller design to create a robot that is capable of high performance at the task of balancing on top of a cylinder.

1.3 High-level Requirements:

- The robot must be able to recover from an initial tilt angle of 40 degrees from vertical
- The robot must cost less than $200 to build
- The robot will be able to move a distance of 3 feet along the ground in less than 3 seconds. The robot and cylinder must be stationary at both the beginning and end of the movement.

2 Design

2.1 Physical Design:

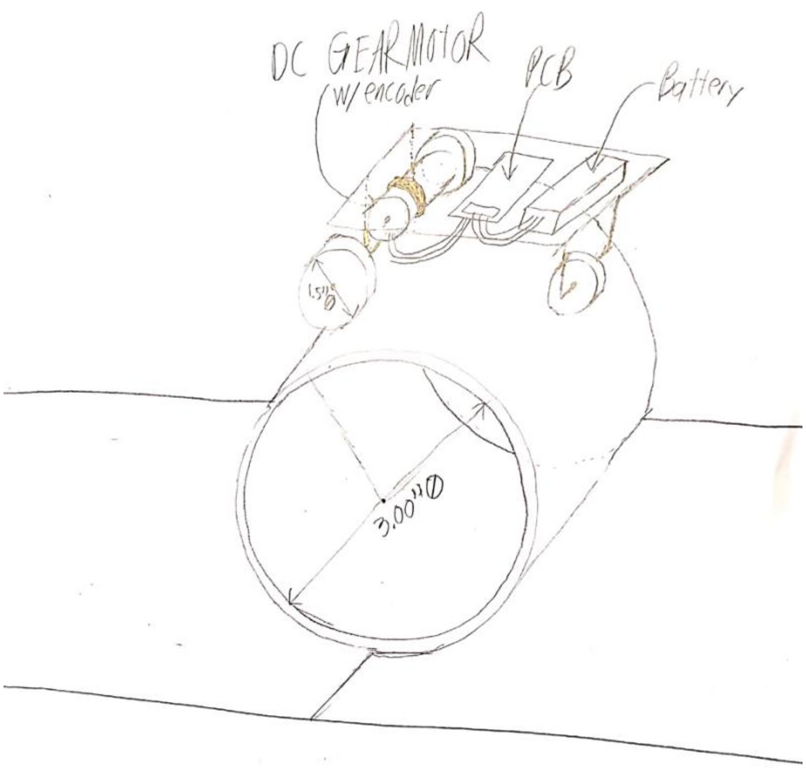

Figure 1: Physical Design Sketch

This sketch represents the physical design of the robot. It consists of three wheels attached to a platform holding the PCB and (possibly) a battery. One of the three wheels is driven by a DC gear motor. The robot rests atop a hollow cylinder.

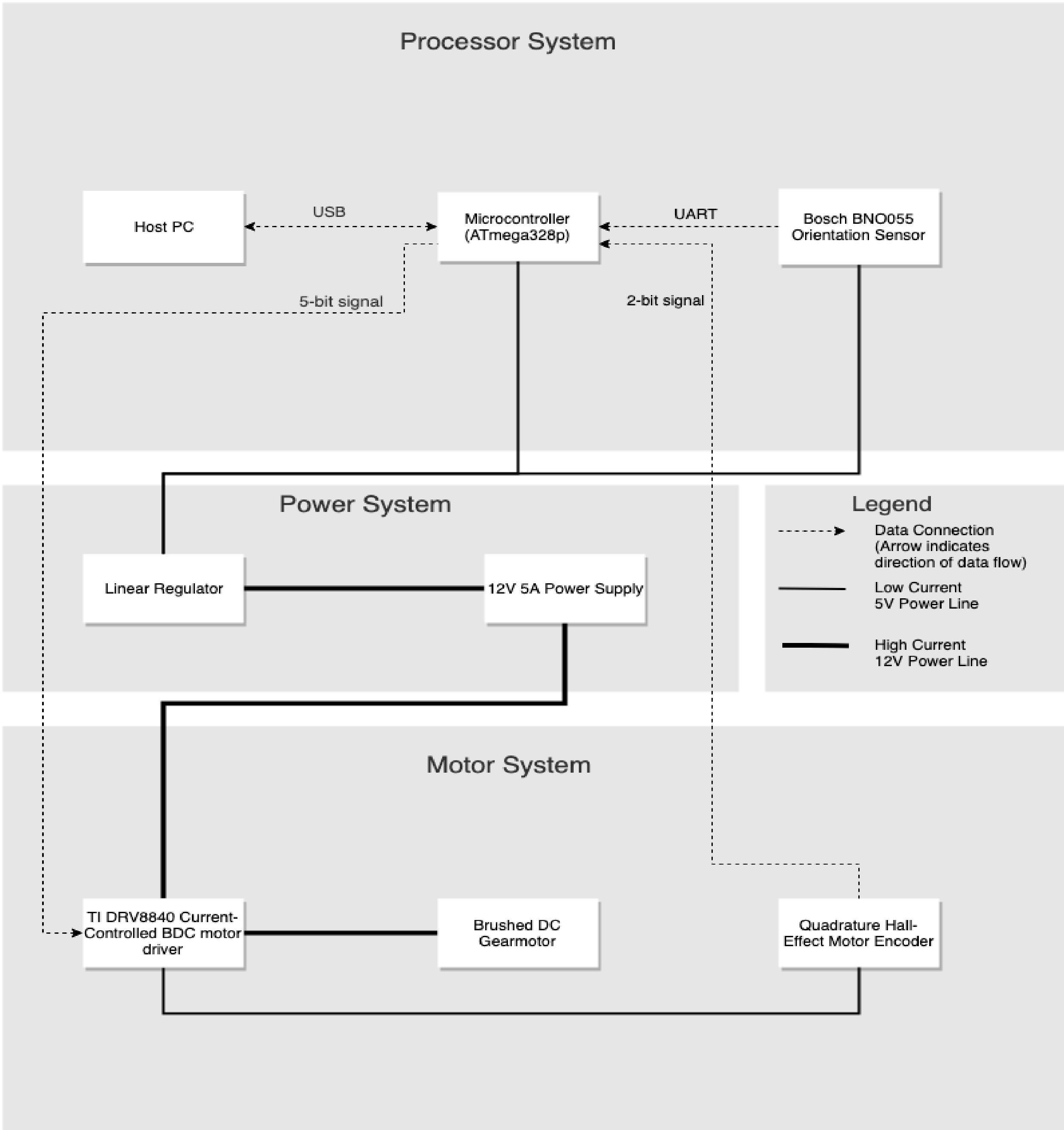

Figure 2: Block Diagram

The design illustrated will satisfy the high level requirements. The fast, high accuracy orientation sensor and motor encoder, paired with the fast processor and torque controlled motor system will be able to return the robot to stability from a 40 degree tilt angle, and it will be able to drive the robot quickly from one spot to another. All the components involved will have a sum cost of less than 200 USD.

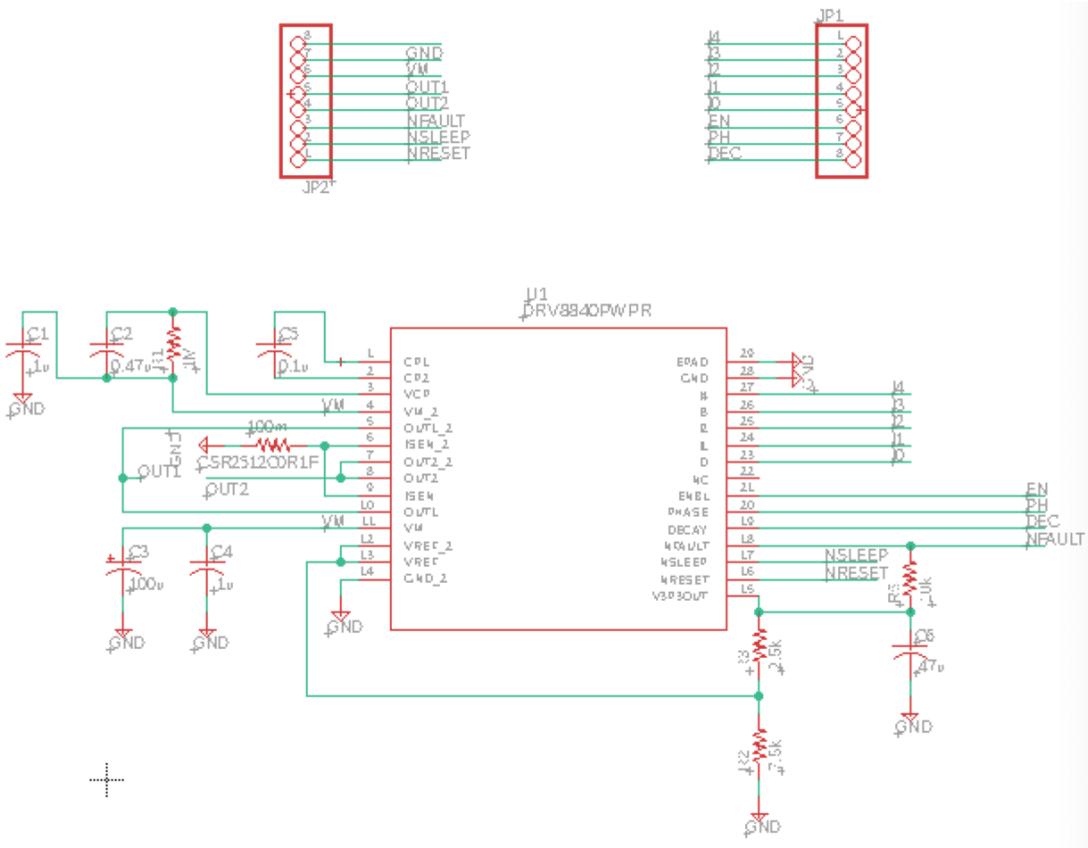

Figure 3: Motor Driver Schematic

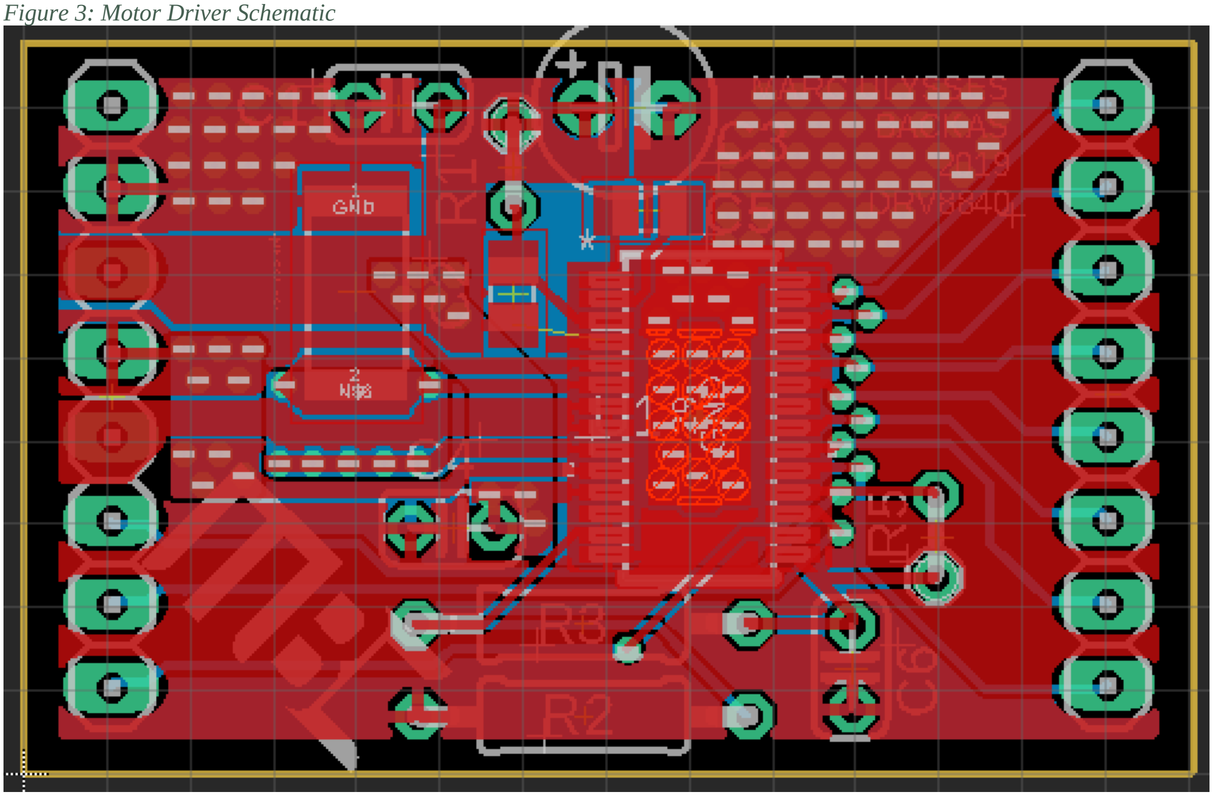

Figure 4: Motor Driver Layout

2.1 Power System

This system will deliver power to all other blocks in our design so that the robot remains functional. The system consists of a 12V 5A power supply and a 5V linear regulator.

2.1.1 Wall Power Supply

Supplies the power needed by all other blocks. Once it is connected to the PCB via a barrel connector, it is directly connected to the motor driver and the linear regulator through copper traces on the PCB.

Requirements	Verification
1. Must provide 5A of current at 12V continuously without overheating 2. Output voltage must remain within 5% of 12V	1,2. Plug in power supply to wall power and attach resistive load such that current draw is 5A. Ensure voltage remains between 11.5V and 12.5V

2.1.2 Linear Regulator(LM1117-5.0/NOPB)

Sources the stable 5V signal needed to power the MCU and orientation sensor.

Requirements	Verification
1. Must provide 30mA continuously while staying cooler than 125 degrees Celsius. 2. Output voltage must remain between 4.5 and 5.5V	1,2. Attach power supply voltage at input of regulator, attach resistive load such that current draw is 30mA. Measure voltage and ensure it is between 4.5 and 5.5V. Measure temperature using infrared thermometer, ensure it is below 125 degrees Celsius

2.2 Motor System

This system will exert the necessary torque, calculated by the processor system, on the drive wheel to balance the robot.

2.2.1 Motor (ROBOT ZONE 638260)

Converts current output by motor driver into torque output to the drive wheel. The torque applied will drive the robot toward stability atop the cylinder.

Requirements	Verification
1. Must rotate at a max speed of 153rpm 2. Must provide at least 1.86 in-lbs. of torque when stalled 3. Must provide torque output proportional to the applied current 4. Must not exceed 60 degrees Celsius while operating under load	1. Connect 12V power supply to motor terminals, measure speed from encoder signals with arduino 2. Connect 1 ft. lever arm to motor shaft, measure force at end of lever with digital scale 3. Apply different currents from 0 to 5 volts and measure stalled torque. Plot data and ensure the correlation is linear. 4. Apply 12V to stalled motor, measure temperature using infrared thermometer

2.2.2 Motor Driver (TI-DRV8840)

Converts the digital control signal from the MCU into a precise current used to drive the motor.

Requirements	Verification
1. Must output at least 5A at 12V continuously while staying below 160 degrees Celsius 2. Must output current proportional to control signal (coefficient TBD) through motor load	1. Connect 12V power supply to driver, set current limit to 5A, load with motor, and measure current and voltage with DMM, and the temperature with an infrared thermometer. 2. Specify all 32 current levels using Arduino, and measure output current through motor using DMM. Plot the data and find the line of best fit. Ensure the maximum error from the line is less than 5% of expected

2.2.3 Motor Encoders (included with motor)

Reports the shaft position of the motor to MCU. Necessary for MCU to compute proper control signal.

Requirements	Verification
1. Must report shaft position within 5% accuracy at all times 2. Must report shaft position at least 100 times per second	1. Use Arduino to count pulses from encoder. Allow motor shaft to rotate 10 full times. Ensure count number is 1973. 2. Use Arduino to poll the encoder count number every 10ms. Set a GPIO pin high while the processor is idling, and low while it is busy, monitor the pin using an oscilloscope and ensure the code runs in less than 10ms each cycle.

2.3 Processor System

2.3.1 Microcontroller (ATmega328P)

The microcontroller (or MCU) will run the control algorithm, processing data prom the sensors and generating a control signal for the motor driver.

Requirements	Verification
1. Must process sensor data and output control signal at 100Hz 2. Must communicate with orientation sensor over UART at 100Hz	1. Use MCU to set a GPIO pin high while the processor is idling, and low while it is executing control signal calculation. Monitor the pin using an oscilloscope and ensure the duty cycle of the pin output is always greater than 0. 2. Connect MCU to orientation sensor over UART and poll data from it every 10ms, ensuring the code runs smoothly using same method.

2.3.2 Orientation Sensor (Adafruit BNO055 absolute orientation sensor)

Measures the tilt angle of the robot in the world reference frame, reporting it to the MCU. The goal of the robot is to drive this angle to zero degrees from vertical

Requirements	Verification
1. Must report tilt angle with less than 3% accuracy at all times 2. Must report tilt angle at least 100 times per second	1. Build a test platform with attached inclinometer to measure actual tilt. Record tilt angle from sensor. Record data pairs into a table. Ensure the maximum error is less than 3% for all angles. 2. Program MCU to poll the sensor at 100Hz, ensure new data is available from the sensor at each poling time

Tolerance Analysis

A crucial component of our robot is the controller running on the microcontroller. It's job is to monitor the sensors of the robot and calculate the correct control signal to deliver to the motor driver over each sampling period.

Let us define our system. One way to do this is with a state-space model:

$$\dot{X} = AX + BU$$

The vector X contains the state variables: $\dot{\theta},\ \theta,\ \dot{\phi},\ \phi$, where $\theta, \dot{\theta}$ represent the angular position and velocity of the robot with respect to vertical, and $\phi, \dot{\phi}$ represent the angular position and velocity of the cylinder the robot rests upon. We want to predict, and ultimately control the robot's accelerations $\ddot{\theta},\ \ddot{\phi}$, in order to keep the robot balanced. The 'A' matrix relates the robot's state variables to the derivatives of those state variables. The 'B' matrix relates the input control effort to the state variables' derivatives. To solve for these matrices, we use Lagrange's equations of the second kind:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) = \frac{\partial L}{\partial q_j}$$

Where $L = T - V, and\ T = \frac{1}{2}\sum m_j v_j^2\ , and\ V = g\sum m_j h_j$. Once these equations are solved, we have the A and B matrices representing the dynamics of the real physical system. Since our system is discrete, and not continuous as our derivation suggests, we need to convert our continuous model into a discrete one, which looks like:

$$x[k+1] = A_d x[k] + B_d u[k]$$

Using a zero-order hold technique for both sampling and giving control signals, the discretized Ad and Bd model parameters reside in the equation:

$$\begin{bmatrix} Ad & Bd \\ 0 & 1 \end{bmatrix} = e^{T*\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}}$$

Where T is the sampling period. We will use these to obtain a controller for the system. We define our input $U = -KX$, where K is our controller vector. The whole system then becomes:

$$\dot{X} = (A - BK)X$$

The eigenvalues of the matrix, $A - BK$, are the poles of the system. Our choice of the controller K determines these poles. In order for the system to be dynamically stable, all the poles (eigenvalues) must have large, negative real parts, and small imaginary components. To find the eigenvalues of the system, we find the roots of the characteristic equation, or the values of s that satisfy $det[sI - (A - BK)] = 0$. By defining our desired characteristic equation as

$$(s - p_1)(s - p_2)(s - p_3)(s - p_4)$$

where $p_{1,2,3,4}$ are the desired poles of the closed loop system, we can then solve for the parameters in K, and derive our controller. We will now test our controller in simulation.

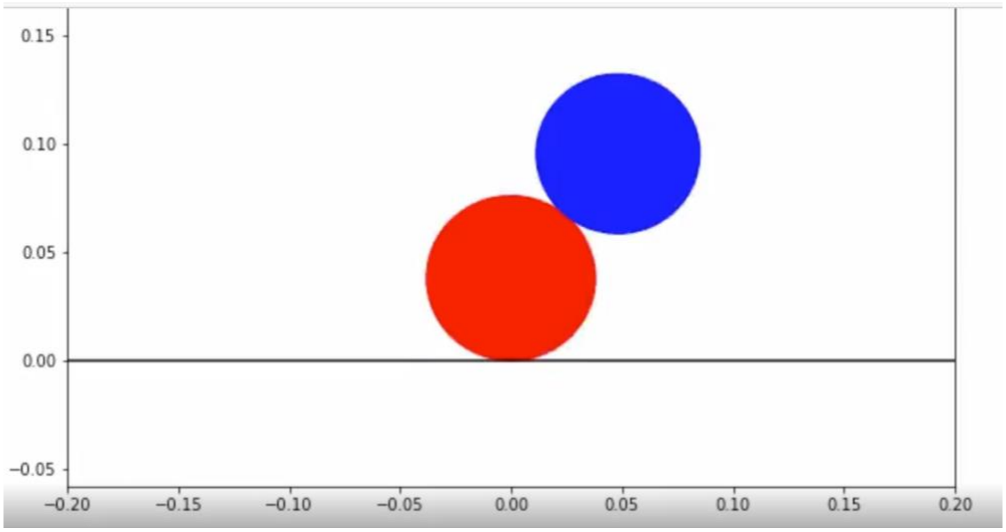

Figure 5: simulated robot

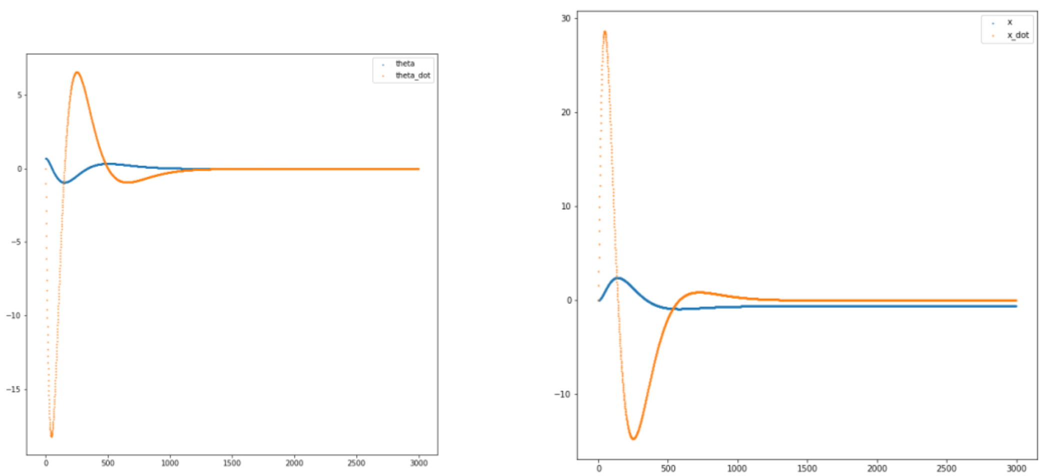

Figure 6 and 7 robot and cylinder angular position and velocity w/o sensor error

*Figure 7: same data with .1*pi constant tilt sensor error*

From the simulated results, it is clear the control algorithm designed is tolerant to significant errors in sensor data.

Cost Analysis:

Based on a 100k salary per group member and 8-week time window in which the project is performed, plus an estimated upper limit of $200 for every component making up the robot, this project will cost:

(100,000 * 3 group members) / ((12 months/year)/(2 months/project)) = $50,000

- $200 for the robot

= $50,200

The estimated upper-limit of $200 is based on the cost of the $50 motor and $30 orientation sensor(the two most expensive components present), combined with a fairly wide buffer for the unforseen(i.e. something breaking, etc.).

Schedule

Week	Marc	Phillip	Mingrui
7-Oct	Build motor driver boards	Work on main board schematic and layout	Create the initial object of Robot and cylinder.
14-Oct	Build and test motor system, submit early bird pcb	Orientation sensor testing and characterization	Add into movement, player can move around, robot and cylinder can be touched and pick up.
21-Oct	Debug motor system	main board soldering and testing	Connection system goes in, player can hold robot and can keep robot on the cylinder for sure (with physical consideration)
28-Oct	connect motor system with python and log data for testing	continue main board work	Add physic formula into the robot and cylinder, making this play like in real life
4-Nov	final pcb round	main board debug	help with susbsytem debugging
11-Nov	Assemble robot with machine shop	orienation sensor interface debug	analyze/debug software timing requirements
18-Nov	mock demo	motor driver board interface debug	
25-Nov	Fall break	fall break	
2-Dec	Final demo debug	Data logging testing and debug	
9-Dec	Final presentation	final paper	

3 Safety and Ethics

Our robot contains hard, possibly fast moving parts which could cause injury to a person in the event of a collision. This event would go against IEEE #9 “to avoid injuring others.”[2] To avoid this, people will be kept clear of the robot’s trajectory while it is powered on, and a group member will be prepared to disconnect power rapidly if the robot begins to behave dangerously.

Our project has the potential of furthering others' knowledge in the area of control systems and robotics, but only if it is made available for use. This concerns IEEE code of ethics #5, “to improve the understanding by individuals and society of the capabilities...of conventional and emerging technologies, including intelligent systems.”[2] To abide by this, our hardware and software designs will be open source, freely available for duplication by anyone interested in the topic.

Our project team consists of three students who wish to gain valuable technical experience. Although there are many individual areas of work to be done, it is possible for one or more group members to take responsibility for the project’s completion away from the others, denying them of the experience they wish to gain. This would go against the IEEE code of ethics #10, which requires its members to “assist colleagues and co-workers in their professional development.”[2] To prevent this, our group will ensure the work to be done is shared equally, and allocated according to each member’s overall career development goals.

References

[1]U. Control Systems Instructional Laboratory at University of Illinois, "UIUC Segbot - Home", *Coecsl.ece.illinois.edu*, 2019. [Online]. Available: http://coecsl.ece.illinois.edu/segbot/segbot.html. [Accessed: 20- Sep- 2019].

[2] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 9-19 Sep-2019]