The Creation of a Self-Balancing Security Robot

Neeloy Chakraborty, Edward Fitzpatrick, Amir Megherhi

Pittsburgh Science and Technology Academy

Author Note

Neeloy Chakraborty, Student, Pittsburgh Science and Technology Academy

Edward Fitzpatrick, Student, Pittsburgh Science and Technology Academy

Amir Megherhi, Student, Pittsburgh Science and Technology Academy

This research was conducted under the guidance of Dr. George Kantor, Robotics Institute, Carnegie Mellon University

This research was supported in part by the Pittsburgh Promise Scholarship

Abstract

The purpose of this project is to create a self-balancing robot that can roam the hallways of a school and identify people who shouldn't be on campus. The basic premise of the project is that while the robot would be roaming the hallways, it would be using face recognition to see whether a certain person or group of people are not in the database of students, teachers, or staff. If it does find someone, then an alarm would be triggered allowing for appropriate security to come to the scene and establish safety on the premises. This is being accomplished through three main production steps: (1) Hardware aspect of building frame and hooking up electronics, (2) Programming self-balancing code of for wheels of robot, (3) Programming face detection and face recognition code through OpenCV. As there is a growing reliance on human security and technology, mistakes can occur. This robot would ensure a safer environment on campus.

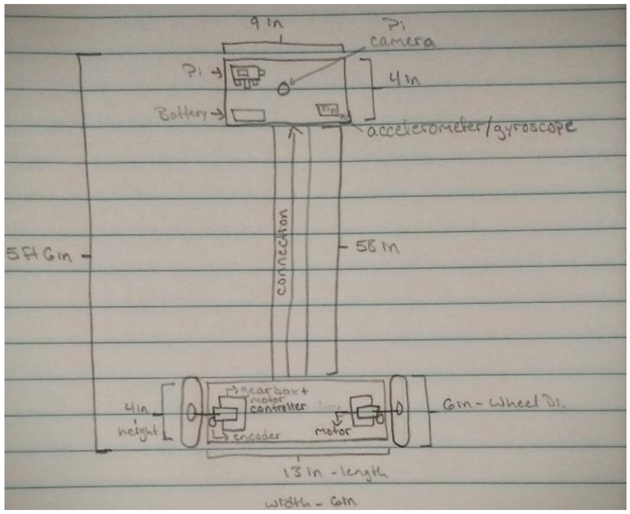
The Creation of a Self-Balancing Security Robot

Introduction:

Security is a vital necessity to maintain safety in society. With the world's growing reliance on technology, new systems must be instituted to protect the populous. Currently college and high school campuses rely heavily on traditional methods of security such as metal detectors and human security guards which are prone to mistakes and may have drastic impacts on people's lives. The solution is robotics, a growing field of engineering that has countless societal applications. The project's topics consist of three separate processes including hardware/material collection, electrical components, and face detection/recognition algorithms. Combining computer vision with segway versatility enables the creation of an autonomous human interactive security robot. To build a complete and fully functioning human interactive, self-balancing robot, we must first research and prototype the components we will be working with.

Hardware:

Hardware, electronics, and software all go hand in hand in the process of constructing a robot. The frame, including the materials of which it is composed of, can be assembled using plywood, aluminum, or even 3D printed pieces. Each material has their own strength, flexibility, and durability. Plywood is the weakest of the stated materials with a max tensile strength 31 MPa and little elasticity with a shear modulus of 0.62 GPa. Plywood can be used as the base of the robot, carrying the electronics and connecting the wheels. This would provide the robot with a sturdy yet light base because Plywood has a density of 615 kg/m^3. The robots electronics will need to be protected. This is where aluminum is used. Aluminum has a large maximum tensile strength of 90 MPa and its shape can be easily manipulated, when compared to plywood, because of its shear modulus of 69 GPa (Materials, 2004).



Initial Overall Sketch

Engineering requires in depth planning before any physical building can take place. One important method required was CAD (Computer Aided Design), a computer based program that allows the user to create three dimensional model of design. This project utilizes the CAD program Autodesk Fusion 360 to envision the prototype for the bottom box. Crafting the parts of the bottom box used the function of Full Spectrum laser cutter from the CMU robotics institute. With the CAD models, the laser cutter was able to precisely slice parts of plywood into sections of the robot that are put together to form the chassis (Fusion 360, n.d.).

Electronics:

Robot software and electronics can be based on a multitude of microcontrollers (small computers that rely on a single integrated circuit). Possibilities for microcontrollers in this project include the Raspberry Pi or the Arduino, both with their own forms of implementation. The Raspberry Pi is much more complex than the Arduino with the capability to run its own

Operating System on it, such as Linux. For that reason, the Pi is able to run other software like OpenCV (discussed later) directly on the chip. In contrast, a positive option to the Arduino is that any program written on the computer can be run instantaneously when powered up, which is a process that must be separately coded into the computer (Justo, 2015). Other electronics that will be used include the MPU-6050 gyroscope and accelerometer, L293D motor controller, Raspberry Pi Camera module, and other miscellaneous power supply sources. All of these devices can easily be interfaced to the Pi computer through programming libraries.

Balancing, accomplished through writing a PID, is an integral part of any robot. A PID is an algorithm used to fix balance errors on a machine by correcting its proportional overshoot. An acronym for Proportional, Integral, and Derivative, each singular portion works together to accomplish one goal, complete balance. A PID algorithm uses a control loop feedback mechanism to constantly search for error values. Using a set value to compare against calculated values constantly being reported back, a PID attempts to fix the error over time by changing the constant variable (RCModelReviews, 2016).

An undesired value, a value other than the set constant value, is returned when inertia on the mass of the robot pushes it out of proportion. When this occurs, the PID uses its proportional program to fix it by adding the opposite and bringing the robot to a state of balance and equilibrium. For example, if a plane is meant to fly straight but is tilting left, the proportional knows that the plane must tilt left to bring the forces acting on the plane to a state of equilibrium, and vice-versa if the plane is initially tilting too far towards the left. After the PID corrects this error, the data is returned as integral information that will be used to aid the proportional in correcting new current problems that arise and the derivative to predict future problems and fix them ahead of time (RCModelReviews, 2016).

Computer Vision:

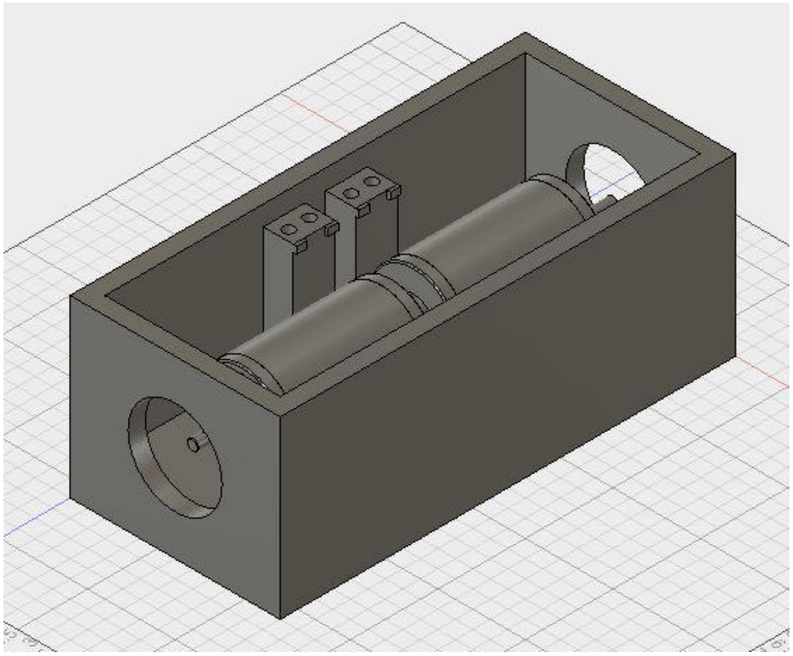
Computer vision is a growing field of computer science that utilizes neural network training to allow a machine to “see” or understand its surroundings. Several applications of this field includes face recognition/detection, optical character recognition, and GPS mapping. As the computer vision field becomes more advanced, it allows for computers to understand their surroundings more accurately and efficiently. This project implements computer vision in cohesion with the Raspberry Pi’s Camera Module to recognize people’s faces with the application of security (Lowe, 2015).

There are several different algorithms that have varying results in the face recognition field including EigenFace, FisherFace, and Local Binary Pattern Histogram (LBPH). EigenFaces and FisherFaces are similar in that they both complete a training set of images through Principal Component Analysis at once. This means that the EigenFaces and FisherFaces analyze a person’s facial features, such as their eyes and nose bridge, in a series of images where their expressions differ. This provides the programs with a database of different faces and expressions it can use to identify a certain person. LBPH actually trains each training image separately from others. This allows for prominent features to be found easier per face being recognized. The EigenFace and FisherFace algorithms differ in that FisherFace is capable of handling greater insensitivity to light due to linear separation (Markusˇ).

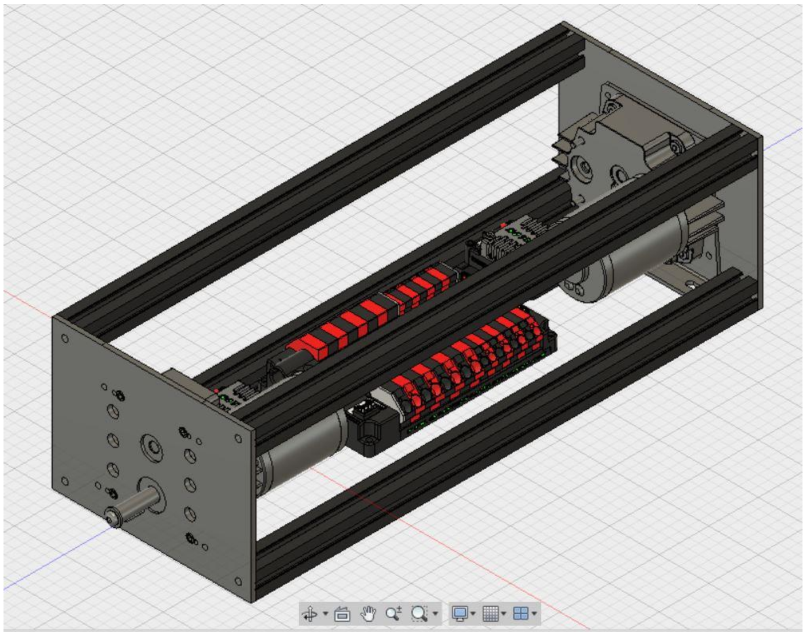
Materials and Methods:

This robot will consist of several electronic components collaborating together in order to give the output of a balancing robot. The main “brain” or computer of the robot is the Raspberry Pi B. The Pi controls all of the components of the robot, including the motor controllers and alarm system, through its Ground Power Input Output (GPIO) system (Justo, 2015). The motor controllers used are the L293D that control DC motors through Pulse Width Modulation (PWM). This means that the motor controller gives a positive and negative pulse back to back continuously in order to turn the motor. The PWM controller fails in its inability to tell the Pi the motor's position; therefore, an encoder must be used on the motor to identify its position at all times (Hirzel, n.d.). These motors are connected to rubber treaded wheels through a live-axle setup. The MPU-6050 accelerometer and gyroscope is used to know what angle and acceleration the robot is moving at, which is helpful in PID calculations (MPU-6000 and MPU-6050, 2013). And finally, the Pi Camera Module is the "eye" of the robot, and allows the Pi to undergo OpenCV face recognition. The Pi is also connected to an external alarm system through the GPIO, and goes off whenever a face is not recognized (Getting Started With PiCamera, n.d.).

The process of creating this self-balancing security robot is split up into several subsections including individual unit testing, 3D modeling of the outer body, creating one or multiple prototype robots, programming of movement and face recognition, and testing in several scenarios. Early on in the development of the robot, all electronics must be unit tested in order to make sure that the limits and the capabilities of each component are known, and each component is capable of working under normal stress conditions. This individual unit testing first occurs with a power supply connected to each component separately, and later as a whole to ensure efficiency (What is Component testing?, n.d.). After testing components, making a 3D sketch of the robot in a Computer Aided Design program allows for easier computing in equations that later solve for mass, torque, and overall balance capabilities of the robot (Fusion 360, n.d.). The creation of the draft robot and programming its movement go hand in hand, as the movement must be tweaked according to the possible PID instability problems faced (RCModelReviews, 2016).



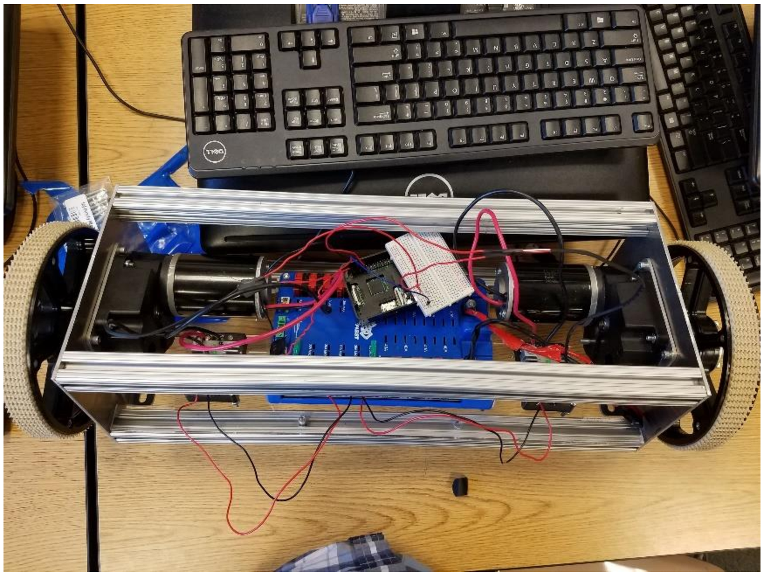
First Draft CAD



Current CAD

Conclusion:

We have made significant strides in the production of our robot, although we did not necessarily reach the end goal we were hoping to achieve. The main aspects of the project that have been produced successfully are the sturdy frame, python code for movement of motors via PWM and talon sr motor controllers, as well as a partial understanding of the creation of a face recognition software for identifying specific people. As the project has come to a close for this year, and another candidate has come forward to continue the production of the robot, a documentation page has been created to give her a solid foundation for continuing the project.



Current Robot Design

Bibliography

Fusion 360. (n.d.). Retrieved from Autodesk: http://www.autodesk.com/products/fusion-360/students-teachers-educators

Getting Started With PiCamera. (n.d.). Retrieved from RaspberryPi: https://www.raspberrypi.org/learning/getting-started-with-picamera/

Hirzel, T. (n.d.). *PWM*. Retrieved from Arduino: https://www.arduino.cc/en/Tutorial/PWM

Justo, P. D. (2015, December 4). *Raspberry Pi or Arduino Uno? One Simple Rule to Choose the Right Board*. Retrieved from Makezine: http://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/

Lowe, D. (2015). *The Computer Vision Industry*. Retrieved from UBC CS: http://www.cs.ubc.ca/~lowe/vision.html

Markusˇ, N. (n.d.). Overview of algorithms for face detection and. 5.

Materials. (2004, June 17). Retrieved from homepages: http://homepages.which.net/~paul.hills/Materials/MaterialsBody.html

MPU-6000 and MPU-6050. (2013, August 19). Retrieved from IvenSense: https://www.cdiweb.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf

RCModelReviews (Director). (2016). *What PIDs do and how they do it* [Motion Picture].

What is Component testing? (n.d.). Retrieved from ISTQB EXAM CERTIFICATION: http://istqbexamcertification.com/what-is-component-testing/