NannyBot for Robots Developing to Walk

ECE 445 Design Document

Chulwon Choi, Alejandro Diaz De Argandona Araujo, Karthikeyan Sundaram

TA: Vassily Petrov

Group 38

1. Introduction
1.1 Problem and Solution Overview
1.1.1 Background
Ever since robots were invented, humanoid robots have always been an interest to people. Setting aside childhood romances of giant humanoid robot fighting monsters, bipedal robots interest people for how useful they could be. While less stable than a wheeled robot, bipedal robots can theoretically traverse in any way that humans can. Many companies have developed humanoid robots, such as Honda's Asimo [2], Sony's Qrio [3], Robotis's Robotis OP2 [4]. Bipedal robots have the potential to can climb stairs, step over objects, and maneuver on uneven terrain, allowing them to be useful in places that are not already paved flat by human hands. However, these useful robot have a fatal flaw that people have been working on. That is, they are hard to balance, and tend to trip. Engineers can't really program every movement of the robot to ensure that it never falls. As a solution to this, engineers began using reinforcement learning to teach robots how to walk. In the recent years, many companies have turned to this method to teach robots how to walk. The reinforcement learning allows the robot to attempt to walk, fail, and learn from the failures, and try again until they find a way to learn how to walk without falling. This brings us to the problem we wish to address. In order to learn to walk through reinforcement learning, the robots will inevitably fall many times. These falls mean that the robots could get damaged, and that would be disastrous financially, as the robots are expensive. The problem becomes worse when working with larger robots, as large robots have a heavier body and therefore fall harder, making it more likely to become damaged. Also, every time a robot falls, a human has to pick it up, and make it stand again. This can become a tedious task, and can be dangerous when working with big robots. People have been searching for a solution to this problem. Some found ways to minimize fall damage [5]. Some, like the Robotis OP2, gave the robot the ability to stand up by itself after it falls. Professor Kim Joohyung installed machines that would be tied to the robot, which would reel the robot in when the robot falls. However, these solutions do not address the core problem of the robot falling in the first place, and installation of a machine restricts the areas the robot can be tested in.

1.1.2 Objective
Professor Kim Joohyung pitched the idea to build a Nannybot to solve the problem of robots falling as they learn to walk. Our goal is to build a robot to aid robots that are learning to walk. The Nannybot is to be a wheeled one for stability. We will build a robot that can follow the walking robot closely, and prevent the complete fall of the walking robot. The walking robot would need to fall to a certain degree, enough so that the robot realizes that it has fallen, but not enough to actually hurt the robot.

After the robot falls, the Nannybot will lift the robot up and bring it back to the starting place. The Nannybot we build is for robots with the maximum size 30cm*30cm*50cm. Our Nannybot is to be a prototype for Nannybots for larger robots, as if our Nannybot works well, Professor Kim Joohyung intends to try creating a Nannybot for larger robots. As it is a prototype, our Nannybot will be first controlled by a user by a joystick and a button. The Nannybot will have L-Foam Neoprene with Nylon on one side, made by Macro International. The foam 6.5mm thick, and is used to create continuum keyboards. This foam will be covering the interiors of the robot, where the walking robot is exposed to, so that the walking robot will not be damaged upon accidently hitting the Nannybot.

1.2 Visual Aid

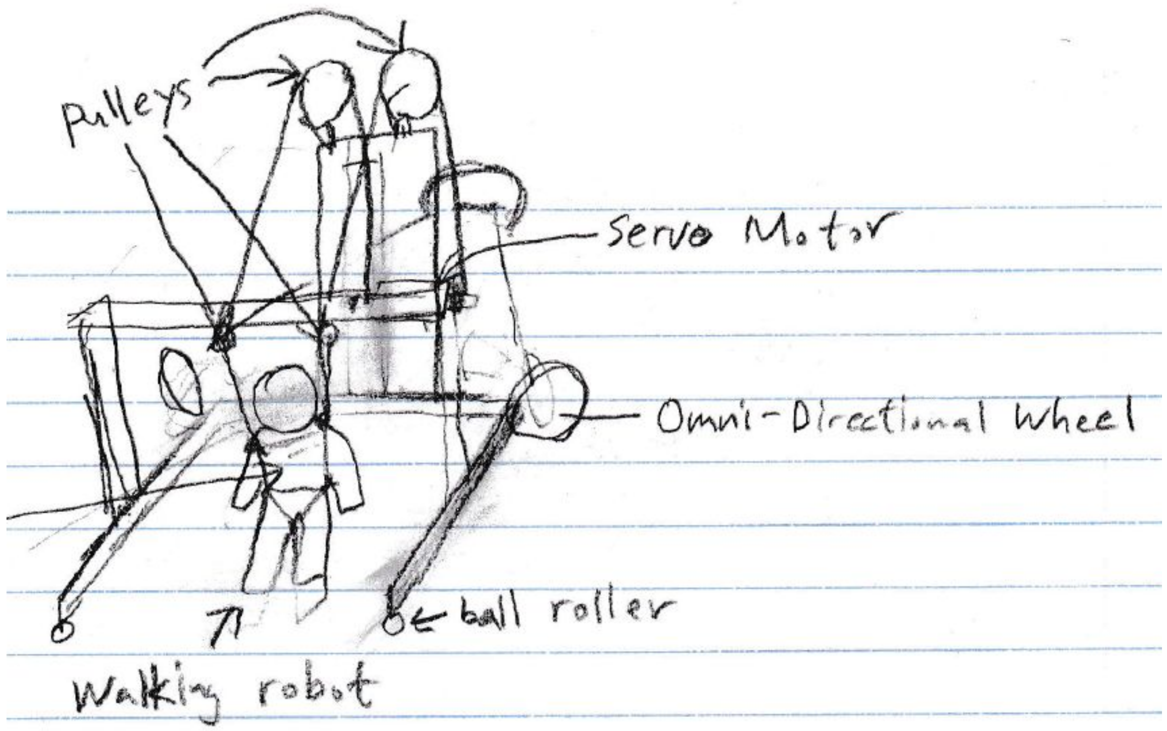


Figure 1.1

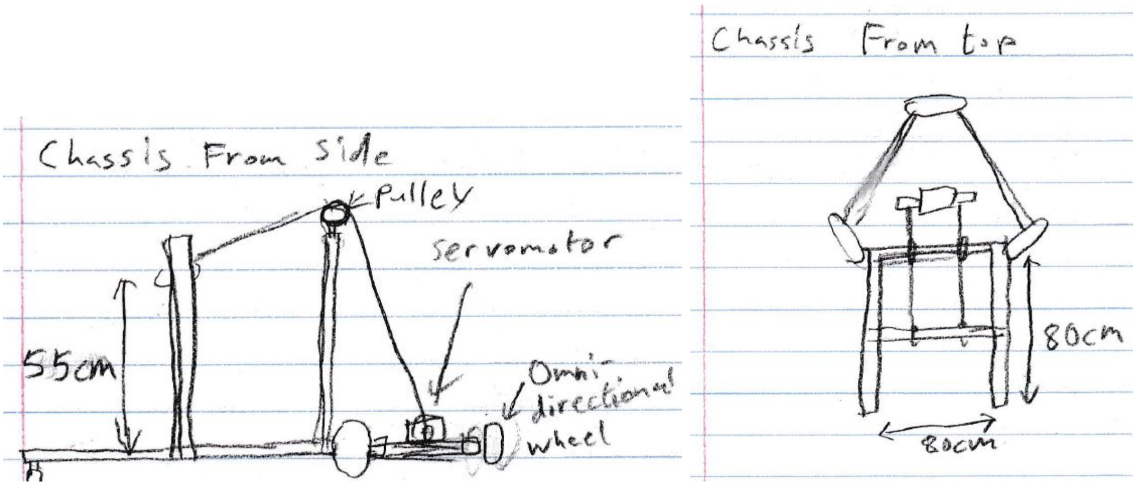


Figure 1.2

The figure 1 shows the physical design of the Nannybot from various viewpoints. The NannyBot is composed of a Chassis, a pickup system, and a pair of legs that stretch out to the sides of the walking robot. The legs help distribute weight when the walking robot is lifted, and also act as a place to put supports for the pulleys that pull the walking robot up. They are 80cm apart so that the walking robot has room to move and doesn't hit the NannyBot. The Chassis has 3 omni-directional wheels and has a triangular formation that allows it to travel in all directions. A stepper motor is used to pull the strings laid over the pulleys to lift the walking robot. It is placed on the Chassis, so that the center of mass of the NannyBot stays low. There will be foam on all surfaces that the walking robot could possibly touch so that the robot will be protected if it accidently hits the Nannybot.

1.3 High-level requirements list:

- NannyBot must be able to follow a Robotis OP2 under the user's control, which has a velocity of 24cm/sec.
- NannyBot must be able to lift a Robotis OP2, which weighs 3kg.
- NannyBot must be able to carry the Robotis OP2 and travel at least 4m.

2. Design

2.1 Design Block

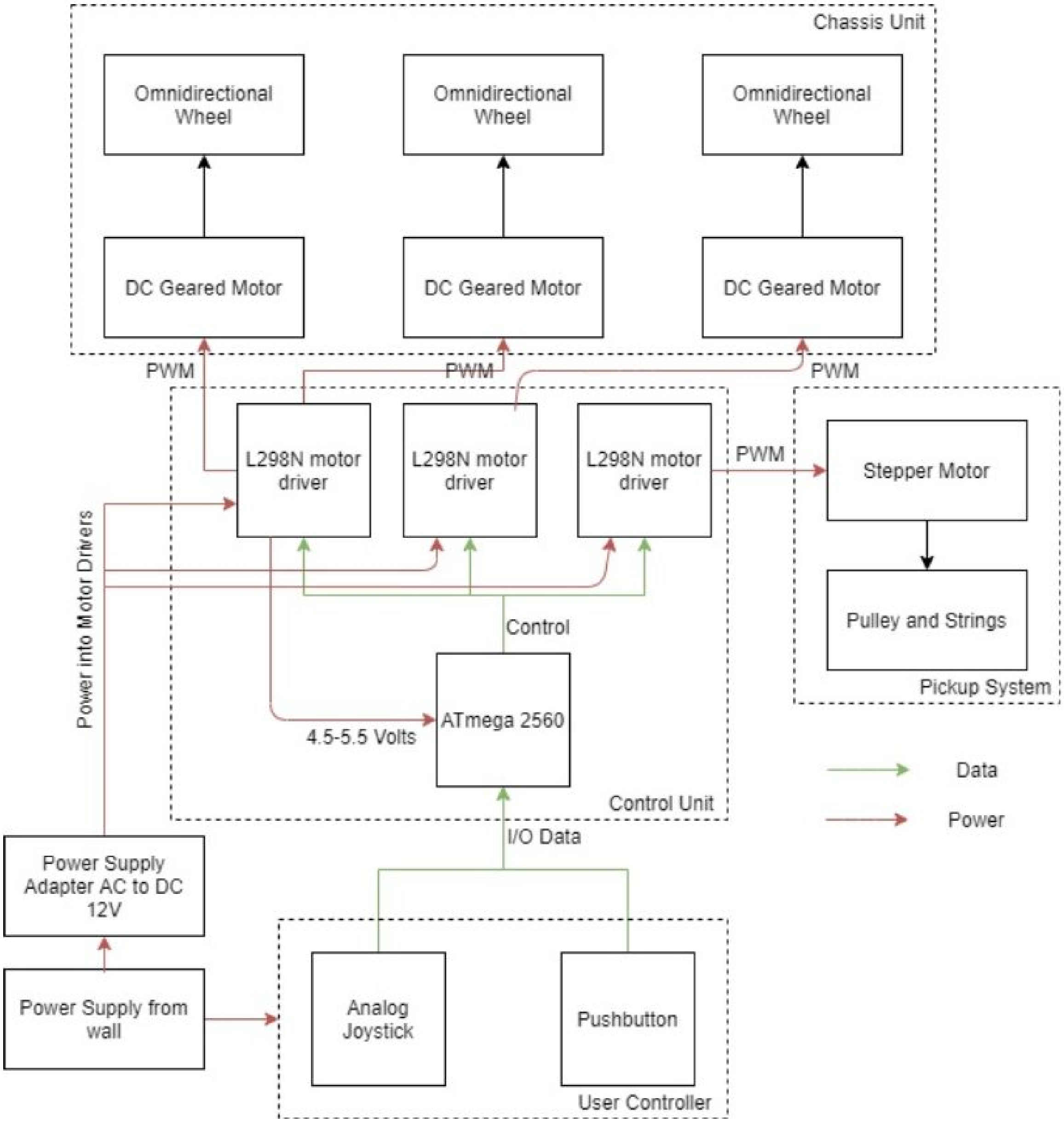


Figure 2

We will use a microcontroller to send PWM signals to the 3 DC Geared motors and 1 Stepper motor. The microcontroller will be connected to a Joystick and 2 buttons. The signals sent to the motors will be determined by the signal from the joystick and the buttons. The PWM signals will tell the motors how fast they need to turn, and in what direction they must turn.

2.2 Functional Overview and Block Requirements

2.2.1 User Controller

The User controller is used by the user to transmit commands to the microcontroller. It consists of an analog joystick and a button.

2.2.1.1 Analog Joystick

An analog joystick receives commands from the user and sends signals to the microcontroller accordingly. The analog joystick sends different signals depending on how the user manipulates it, and this signal is used to make the Chassis move in different directions.

Requirements	Verification
The analog joystick delivers different signals when it is pushed in different directions.	Measure the outputs from the analog joysticks when it is on and observe if they are different for different directions.

2.2.1.2 Push Button

A push button receives the command from the user to send a signal to the microcontroller telling it to use the stepper motor to pick up the robot.

Requirements	Verification
The push button sends a signal to the microcontroller when it is pushed.	Measure the output from the push button when it is on and try pushing it to observe the change.

2.2.2 Control Unit

The control unit receives signals from the user control and sends signals to other parts. It is powered by a cable from the wall. It consists of a microcontroller and two motor drivers. The microcontroller receives commands from the user control, and sends appropriate PWM signals to the motors in the Chassis unit and the stepper motor in the pickup unit.

2.2.2.1 ATmega 2560 Microcontroller

For this project we are using 3 DC geared motors and 1 stepper motor. In order to control the direction and speed of the motors, we need a microcontroller that has at least 2 PWM pins for each motor. This leads to a total of 8 PWM pins. Further, we are trying to integrate a User's controller for this project, we would need at least 2 Analog Input pins for the analog joystick and we also need at least 1 Digital I/O pin for the pushbutton. After taking these requirements into consideration, we decided to make use of the ATmega 2560 microcontroller. This microcontroller has 15 PWM pins, 54 Digital I/O pins and 16 Analog Input pins.

Requirements	Verification
1. 8 PWM pins must be available to send signals to motor drivers. 2. 2 Analog pins to take inputs from the analog joystick 3. 1 Digital I/O pin to take an input from the pushbutton	1. Verify that the pins work by testing motors through the microcontroller

2.2.2.2 L298N Motor Driver

Most microcontrollers, including the ATmega 2560, do not have the capabilities to provide the power that is needed to drive the motors. Further, they are not safeguarded by the back emf that could be created during the functioning of the motors. Therefore, it is necessary to include a motor driver between the microcontroller and the motor. The L298N motor driver has a power rating of 25W and this would be sufficient to drive the motors. There are 3 L298N Motor Drivers. One can drive 2 DC Geared motors, but the stepper motor requires a whole L298N Motor driver to itself.

Requirements	Verification
1. It should be drive a 3V-12V gear motor	1. Verify that the motors operate at differents speeds

2.2.3 Chassis Unit

The chassis is where the DC geared motors and the wheels are located. Our Nannybot has a triangular chassis instead of a square or rectangular one, as having a wheel on the side of the walking robot could lead to the robot hitting the wheel, and having less wheel and motor means less weight. The chassis uses omni-directional wheels and is capable of movements in all directions.

2.2.3.1 DC Geared Motor

Three motors will receive signals from the control unit and be used to turn the 3 omni-directional wheels to move the Chassis in any direction. The gear assembly will allow us to reduce speed and increase torque of the wheels.

Requirements	Verification
1. Each motor must have at least 28rpm or higher. 2. The 3 motors must be able to make the Nannybot travel at least 4m while carrying a Robotis OP2 which weighs 3kg. 3. The 3 motors must be able to spin around the walking robot, with the robot as the center of the turn. The robot makes a full revolution in around 8 seconds.	1. Apply tape to the motor, so the tape rotates when the motor is run, but such that the tape doesn't stick to anything. Run the motor near a microphone, record, and have the tape hit something as it rotates so it makes sounds. Using sound recording software like audacity, it should be possible to count how often the sound is made per minute by analyzing the sound data. If the count is over 28 in a minute, it passes. 2. Have the Nannybot travel 4m with a 4kg barbel on it and see if it can travel the distance. 3. Put an object on the ground, and have the Nannybot spin around it once. Time the spin, and check the distance from the Nannybot to the object for both the chassis and the legs.

2.2.3.2 Omni-Directional Wheels

Three Omni-directional wheels will be used to allow movement in all directions. They are spun by the DC Gear motors.

Requirements	Verification
The omni-directional wheels must be able to travel in all directions.	A. Have the chassis travel front and back B. Have the chassis travel left and right C. Have the chassis spin on the spot

2.2.4 Pickup system

The pickup system receives signals from the control unit and uses the stepper motor to pull up the walking robot by the use of strings attached to the stepper motor and pulleys.

2.2.4.1 stepper Motor

A stepper motor will be installed on the Chassis, and connected to 2 strings. These strings each go through a set of pulleys to and are connected to the walking robot's harness. The motor reels in the strings to lift the walking robot up when the signal is given from the control unit. 2 strings were used so that the walking robot doesn't spin in the air when it is lifted.

Requirements	Verification
1. The stepper motor must be able to pull 5kg +/- 0.5kg of weight off the ground by at least 2cm.	1. Purchase the stepper motor with the right specs and test it by having a 5kg barbel strapped onto a string connected to the stepper motor, and have the stepper motor pull on it to check if it lifts off the ground for at least 2cm.

2.2.4.2 Pulleys

4 pulleys will be used to lay the strings from the stepper motor to the robot above the robot's head. There are 2 sets of 2 pulleys, and there is a string going through each set. The 2 strings bring the walking robot up

Requirements	Verification
1. The pulleys much not impede the string movement. 2. The pulleys must be strong enough to withstand 5kg +/-0.5kg of weight.	1. Lay a string across the coil, with a weight attached to it. Pull the string to lift the weight, and see how well the pulley turns. 2. Repeat the above procedure with 5kg barbel as the weight.

2.2.4.3 Strings

2 strings are used to connect the stepper motor and the walking robot. Using pulleys, the strings one end of the strings are attached to the walking robot's harness from above. When the stepper motor turns, the strings are reeled in, thereby lifting the robot via the harness and strings.

Requirements	Verification
1. Strings must be long enough to reach the robot, and have enough slack to let the robot lean 30 degrees. 2. String must be strong enough to withstand at least 5kg of weight.	1. Measure the string. If the string is shorter than required, get a new string, as it is preferred to have a longer string, which we can cut to adjust the length. 2. Tie a 5kg to a string, and pull it up. Confirm it does not snap.

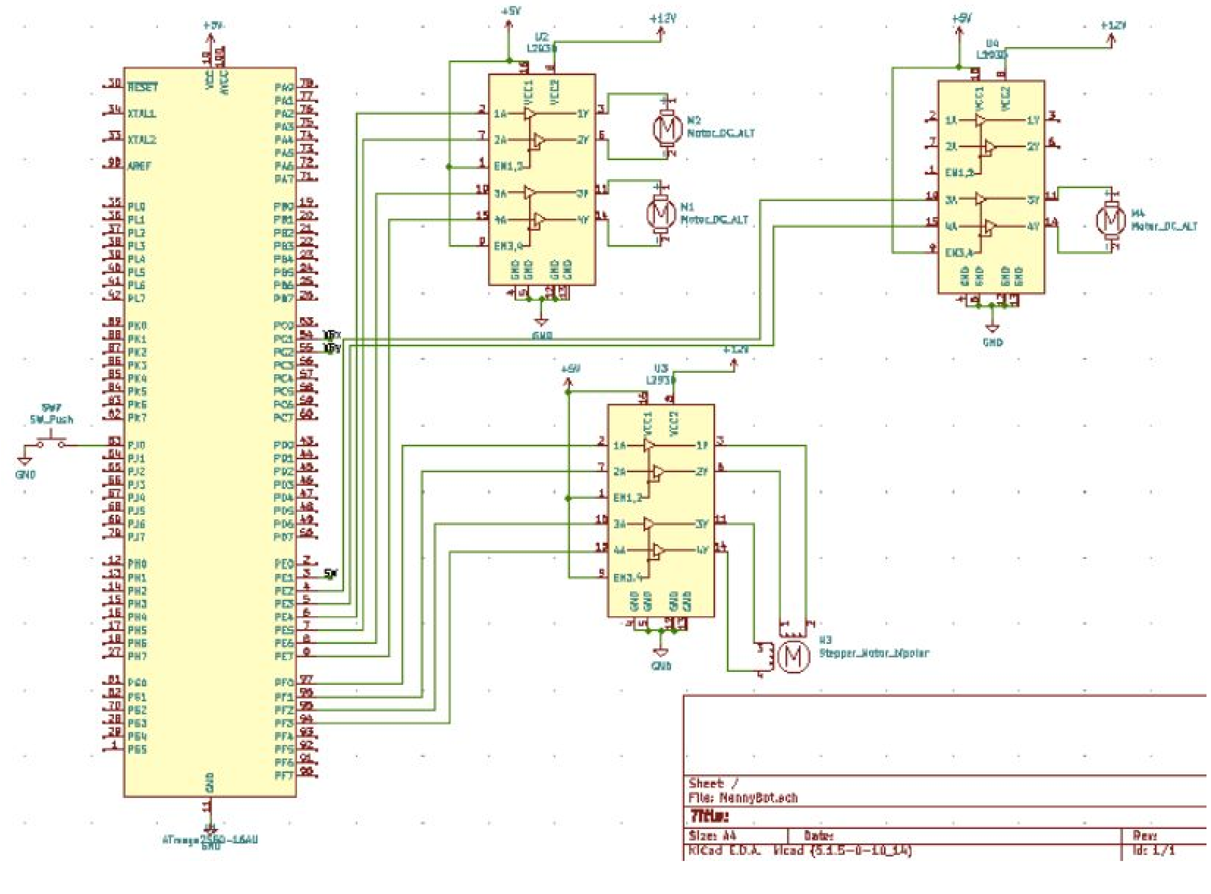
2.3 Calculations

For a continuous stepper motor to lift a load of 5kg with a stepper arm length of 1cm, the motor should have a stall torque of at least 5kg/cm. This would be sufficient to lift the Robotis OP2 which weighs 3kg.

The DC gear motors should be able to provide enough speed to the NannyBot so that the NannyBot can follow the walking robot. When choosing the motor, the radius of the omniwheels must be taken into consideration. In our case, the smallest possible radius that we would be using is 5.08cm. Since the wheels have to cover a larger distance in a shorter time when the walking robot is rotating about a point, this situation would be considered when choosing the motor with the required RPMs.

The OP2 rotates at 8RPM. The radius of the circle that the NannyBot's outer wheel would circumscribe is 40cm+69.28cm = 109.28cm. This means that we would need the wheel to move at 8*2π*109.28 = 91.55cm/s. Therefore the motor would have to provide an angular velocity of $\frac{91.55}{5.08*2\pi}$ =27.4RPM.

2.5 Circuit Schematic



3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

Having 3 members in the group, assuming that one person gets paid $30/hour, planning to work about 15 hours/week, and considering there are 16 weeks in the semester, we get:

$$3\ teammates\ \times\ \$30/h\ \times 15h/week\ \times\ 16\ weeks\ =\ \$21,600$$

3.1.2 Parts

Part	Individual Cost ($)	Total Cost ($) (shipping fee and tax included)
3 12V 30RPM DC Gear Motor	16.88	50.64
3 5 100mm Omni-directional wheels	20.16	77.45
Arduino Mega	38.50	38.50
3 6mm Mounting Hubs	8.46	25.38
3 L298N Motor Drivers	6.89	20.67
Stepper Motor	24.16	24.16
Power Supply adapter	18.99	18.99
Power Supply splitter	7.66	7.66
Total	133.7	263.45

These are the bare minimum required, and additional costs are expected as we must buy an extra motor and extra wheel for the machine shop, as well as potentially add sensors to get it autonomous.

3.2 Schedule

Week	Task	Alejandro Diaz	Chulwon Choi	Karthikeyan Sundaram
02/23/20	Design Document	Cost and Schedule, Risk Analysis,	Problem and Solution	Block Diagram, Subsystems,

		Discussion of Ethics and Safety, Citations	Overview, Visual Aid, High-Level requirements	Tolerance Analysis
03/01/20	Design Review	Prepare Design Review	Order components	Research and select components
03/08/20	Early bird PCBway	Prepare scheme for the machine shop	Design PCB	Design PCB
03/22/20	Machine shop revision	Program microcontroller	Study datasheet for microcontroller and components	Assemble PCB
03/29/20	Individual progress reports	Research and purchase sensors	Test sensor accuracy	Assemble sensors
04/05/20	Final Round PCBway	Test microcontroller	Test wheels and chassis unit	Test pulling system
04/12/20	Mock demo	Prepare mock demonstration	Test accuracy of the motors	Test individual components
04/19/20	Demonstration	Run tests on final project as a whole	Run tests on final project as a whole	Run tests on final project as a whole
04/26/20	Presentation	Fix bugs	Fix bugs	Fix bugs
05/03/20	Final Papers	Finalize presentation and final papers	Finalize presentation and final papers	Finalize presentation and final papers

3.3 Risk Analysis

Since we are going to design two modules for our project, the movable wheeled structure that follows the walking robot and the stepper motor that pulls the strings to lift the walking robot, one of the main problems could be to effectively integrate both of them into a single structure.

For the wheeled structure, one of the main problems we will have to address is to be able to follow the walking robot without interfering with its regular walk. Firstly, we are going to move the NannyBot with human interaction by using a remote control with a joystick. In the future, if we accomplish our basic goals, we will try to introduce sensors to the NannyBot so that it can be autonomous and that human interaction is no longer needed.

However, the main problem we have to face is to lift the walking robot effectively without harming it or interfering with its walking. For that reason, we will design a specific harness for the walking robot that is attached to two lifting points that will be placed on the shoulders of the robot. By doing that, we can create a universal NannyBot that can work with different walkingbots and the only thing we will need to change is the specific harness for the walking robot.

All these problems represent the challenges and risks to the successful completion of the project. If at some point we are stuck with some of those problems, we would try to focus on the main goals: to lift the walking robot with the help of the NannyBot.

4. Discussion of Ethics and Safety

As engineers from the University of Illinois at Urbana-Champaign, we are committed to create a project that is aligned with the values of the IEEE [6] and ACM Code of Ethics [7]. For that reason, these are our main ethical statements:

Since we are working with a moving robot that can hit someone, we will work in a safe environment in which we can avoid harm to any of the group members or other people while working. We will use a small fence during the testing of the motors and wheels to protect all the members from the possible acceleration of the walking robot. The fence will also be covered by a thin layer of neoprene, which is soft enough to provide protection to the robots in case they hit the fence.

We all have completed the laboratory training and we are aware of the risks that working in a project with machines and electricity imply. When working on the chassis and implementing all the modules given by the machine shop, we will make sure that there are no risks for any of the members by using the lab equipment safely and avoiding electrical shocks and burns.

We believe that working in an engineering project involves two risks, plagiarism and misinformation. We are going to respect the ideas and inventions of others. Whenever we use an idea that does not belong to us, we will mention the original author with a proper citation. Also, we are going to be honest about our work and results. If we do not get the expected results on a specific test, we will try to find out why and report the issue. We want to be engineers working with integrity and strong values because we know that our work affects people’s lives.

We want to maintain high standards of professional competence, conduct, and ethical practice. We will learn the technical knowledge required to develop the project and we will upgrade our skills through independent study, attending conferences with professors, and asking experts on the matter.

5 Citations

[1] Bipedal Robot Learns To Run
Will Knight -
https://www.newscientist.com/article/dn6809-bipedal-robot-learns-to-run/

[2] History Of Asimo
https://asimo.honda.com/asimo-history/

[3] Qrio
IEEE Spectrum - https://robots.ieee.org/robots/qrio/

[4] Robotis Op2 [us]
Heirlab - http://www.robotis.us/robotis-op2-us/

[5] A Minimized Falling Damage Method For Humanoid Robots - Qingqing Li, Xuechao Chen, Yuhang Zhou, Zhangguo Yu, Weimin Zhang, Qiang Huang, 2017
Qingqing Li-Xuechao Chen-
Yuhang Zhou-Zhangguo Yu-Weimin Zhang-Qiang Huang -
https://journals.sagepub.com/doi/full/10.1177/1729881417728016

[6] IEEE Code of Ethics

https://www.ieee.org/about/corporate/governance/p7-8.html

[7] ACM Code of Ethics

https://www.acm.org/code-of-ethics