Balancing Robot

William Brown, Eric Moseman, Corey Van Sickle
Ohio Northern University, Ada, Ohio 45810
Email: w-brown.5@onu.edu, e-moseman@onu.edu, or c-vansickle@onu.edu

Abstract
The purpose of this design project was to make an autonomous robot that can maintain its balance on two coaxial wheels. This design was a scaled down version of a personal transportation device. One goal of this design was to minimize the cost of building the full scale personal transportation device in order to be competitive in the market while maintaining similar performance to what is already available.

The input to make this robot move is to lean the top of the robot forward or backward (as if the rider was leaning forward or backward). The base of the robot was designed to be motorized and move to balance the mass above it. The balancing was intended to be controlled with an integrated system actuating two separate motors (one for each wheel) based on the angle that the top of the robot is leaning from vertical, as measured with respect to gravity by an inclinometer.

The robot was designed and built with lightweight materials in order to minimize the torque that the motors would need to balance the robot. In order to control the robot, several components such as two motors, two batteries, an inclinometer, a micro-controller, and two H-bridges were needed.

Calculations were performed to determine the torque required to balance the robot. The components were purchased using these design calculations and the criteria provided. In order to make sure the robot meets all of the requirements and constraints it will need to go through rigorous testing and fine tuning once the control system has been developed and implemented.

A control system utilizing all of these components needs to be developed to stabilize and balance the robot. A fuzzy logic program will be used to balance the robot. Two different programs will be written to control the system. The first will simply balance the robot, and the second will allow the robot to turn. These fuzzy logic programs will be tested in MATLAB, with the aid of transfer function analysis, to properly tune their parameters for optimal performance before they are used with the prototype.

Introduction
Many people and companies all over the world are turning to personal transportation devices to aid in everyday operations due to the ever growing price of fuel. Because of this, personal transportation is a fast growing market as the demand for these devices increases and the price of electronics and technology continuously decreases. While the demand for personal transportation increases, the price tag of these devices remains quite high.
One of the most recognized names in the personal transportation industry is Segway™. The engineers at Segway™ were one of the first to design a personal transportation device that balances and propels itself. Seen in Figure 1, the Segway PT™ model has two coaxial wheels that drive the device forward or backward as a result of the direction the rider is leaning. Steering of the device is controlled by tilting the handle bars to the right or to the left.

Figure 1: Segway PT™

Segway’s™ line of personal transportation vehicles is very popular among professions such as security, parcel delivery, and tour guide providers. These provide a relaxing method of traversing a great deal of ground with little effort, but carry a heavy price tag, starting at just over $6200[^5].

Another example of a personal transportation device is the Enicycle™. Shown in Figure 2, the Enicycle™ is simply a unicycle that is propelled by a motor inside the hub of the wheel based on the leaning of the rider.

Figure 2: The Enicycle™

To steer the device, the rider presses down on either foot peg. This tilts the wheel which creates an uneven pressure distribution across its contact area, forcing the Enicycle™ to alter its path. Unlike the Segway™, a rider of the Enicycle™ will have to learn to balance himself left and right because of its single wheel design. According to the designer, however, this can be done in as little as thirty minutes. This version of personal transportation also carries a high price with it, just around $4700[^4].
While devices like these are highly sought after, the high cost involved with purchasing them will continue to prevent the market from expanding to its full potential. Lowering the cost of producing these devices will encourage new buyers into the market. Both of the personal transportation devices mentioned use an expensive array of several accelerometers to determine the angle that the rider is leaning, which then sends a signal to the motor. While this is a viable and accurate way to measure the angle of tilt, simpler solutions exist. When it comes to technology, simpler usually means cheaper too.

The goal of this project is to design and develop a balancing robot to act as a personal transportation vehicle. The cost to produce this robot is desired to be much less than what is currently available on the market. The performance of this device should be similar to these other products that were previously mentioned. It was determined that the design is to have two coaxial wheels like the Segway™ in order to prevent the need to balance in multiple dimensions. The design of this device used DC electric motors powered by rechargeable batteries. The robot will need to be controlled with an embedded control system in order to allow it to balance without being connected to anything else. While budget constraints prevent the construction of a full scale prototype, a scaled down version will be built to verify the design. A mass at the top of the robot will simulate a rider for this prototype.

While there are definitely multiple ways to develop this kind of system, the following questions were asked to determine how the designed system would operate:

- What is the desired angle at which the pendulum is to lean?
- How will the system know how far the pendulum is currently leaning?
- How will the system use that information to send a signal to the motor(s)?

The answer to the first question depends on what kind of system is being developed. If a remote controlled robot is being developed, this desired angle may vary depending on the user’s input to the controller. If a personal transportation device is being designed, the answer to this question is undoubtedly zero degrees when measured with respect to gravity. This way the system will try to stand the pendulum upright constantly, meaning that while the user is leaning forward, the robot continues moving forward.

There are a variety of answers to the second question including, but not limited to, an inclinometer, an array of accelerometers, and an encoder. All of these are viable ways of obtaining the angle at which the pendulum is leaning at the current moment. When turning is incorporated, a potentiometer will be used to measure the rotation of a center shaft. The measured angle will be used to control how fast the motors spin when turning.

The third question is not as easily answered. As mentioned, the system needs to not only send a signal to the motor(s), but stabilize the pendulum as well. The most basic method to control the system is with a Proportional Integral Derivative controller (PID). This method uses transfer function analysis to determine exactly how to force the steady state error to zero with an acceptable response time. The more involved method of control is with the use of a Linear Quadratic Regulator (LQR). LQR utilizes a state-space representation of system to more easily deal with multiple inputs and multiple outputs. While a PID is sufficient to maintain balance in a single direction, it does not easily allow for an additional input such as the desire to turn the
device in the case of the personal transportation vehicle. The LQR will be able to use the inclinometer and a potentiometer together, meaning that the robot can be given the ability to turn and balance at the same time. The goal for the project is to develop both types of systems.

Theory

The design of this project is based on the inverted pendulum problem. This classic control systems application involves an upright pendulum which is the plant of the system, and a base underneath it that moves forward and backward to keep the pendulum balanced which serves as the controller. A successful inverted pendulum control system will determine how far the pendulum is leaning and convert that information into a signal that drives the motor(s) to keep the base directly underneath the pendulum. The difficulty surrounding this application is the unstable nature of the plant. For any given input, the pendulum will fall over unless properly controlled. Beginning with the PID method of control, a mathematical equation was found to properly describe the system’s motion in response to an input. This representation is known as the transfer function of the system’s plant. In this situation, the input in question is the force applied by the motors, which is equivalent to the torque provided by the motors multiplied by the radius of the wheel. The output under consideration is the angle that the pendulum is leaning from vertical. This equation was derived by developing the two equations of motion needed to fully describe the system, one for the pendulum and the other for the base. These equations were then linearized, combined into one equation, and its Laplace transform was taken. Equation 1 becomes the plant transfer function of the system (the variable $q$ was used to simplify the answer) [1]:

$$\frac{G(s)}{P(s)} = \frac{\Phi(s)}{P(s)} = \frac{\frac{mL}{q}S}{s^3 + \frac{b(l+ml^2)}{q}s^2 - (M+m)mgLs - \frac{bmg}{q}} \tag{1}$$

Let: $q = (M + m)(l + ml^2) - (ml)^2$

where $\Phi(s)$ is the angle the pendulum is leaning from vertical, $P(s)$ is the force applied to the base by the motors, $m$ is the mass of the pendulum, $M$ is the mass of the base, $L$ is half the length of the pendulum, $b$ is the damping coefficient of friction, $g$ is the acceleration due to gravity, and $I$ is the mass moment of inertia of the pendulum. Figure 3 shows how this transfer function relates to the input-output relationship of the overall system. The controller $R(s)$ in this diagram will have to reduce the steady state error to zero in order for the system to be stable and be able to balance itself. The value $k$ is simply the feedback gain which can be used to decrease the response time if necessary.
While a controller for this plant has not yet been developed for the purposes of this project, some time was spent to prove this system could be theoretically stabilized using a PID controller. MatLAB was used to observe the impulse response of this transfer function both with and without a PID\(^2\). For the purposes of this observation, reasonable assumptions were made for each parameter in the equation and can be found in Table 1.

**Table 1: Reasonable assumptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g )</td>
<td>Acceleration due to gravity</td>
<td>9.81</td>
<td>m/s/s</td>
</tr>
<tr>
<td>( m )</td>
<td>Mass of the pendulum</td>
<td>2.25</td>
<td>kg</td>
</tr>
<tr>
<td>( M )</td>
<td>Mass of the base</td>
<td>4.5</td>
<td>kg</td>
</tr>
<tr>
<td>( l )</td>
<td>1/2 Length of the pendulum</td>
<td>0.13</td>
<td>m</td>
</tr>
<tr>
<td>( b )</td>
<td>Damping Coef. of friction</td>
<td>0.1</td>
<td>N/m/s</td>
</tr>
</tbody>
</table>

Using MatLAB’s **impulse** function, the impulse response of the plant was plotted in Figure 4. This is indicative to how the system would react if given any input while standing straight up.
This showed that for any given input the error in the system approaches infinity, meaning that the robot will fall over if uncontrolled. This proved that the inverted pendulum system is indeed unstable and provides some validity to the developed plant transfer function.

This same function was looped through a basic PID in an attempt to prove that it could be stabilized. Through several iterations of fine tuning the PID variables, a stable combination was found and plotted in Figure 5.

**Figure 5: Impulse response of the plant when controlled with a PID where $P = 222$, $I = 3$, and $D = 20$**
The plot shows that the pendulum returns to an angle of zero in less than half a second. While the PID constants will need to be adjusted properly once the robot is built and the parameters can be definitely described, this plot proved that the system can be stabilized with a steady state error of zero and a relatively fast settling time.

In order to gain a better understanding of how a control system is to be implemented, the first programming method will utilize fuzzy logic. Fuzzy logic will allow for the ease of programming while still maintaining a continuous input-output relationship. In a program that simply balances the robot the fuzzy logic will use three different memberships. For this system, the angle of tilt as well as the angular velocity of the pendulum will be used to determine the proper motor output. This provides an equivalent to a PD controller, which takes into account position and its derivative.

In addition to balancing the robot, the LQR will incorporate an open loop control for the turning of the robot. By rotating the center shaft, where the seat would be located, the user will turn the device. The user is responsible in this system for providing feedback. As the robot is turning, the user can decide whether or not he is turning too far or too fast and adjust accordingly.

A similar type of system can also be created using fuzzy logic. This can be done by adding an input relationship for the turning control and by separating the motor output membership into two different memberships (one for each motor). This program would obviously require a much larger rule set, which is why it will be created once the balancing program has been written.

**Implementation**

A block diagram with each component of the design was created and can be seen in Figure 6.

![Figure 6: Block Diagram of Balancing Robot](image-url)
be required to generate to keep the robot upright. This center of mass, \( x_{\text{center}} \), was found using Equation 2:

\[
x_{\text{center}} = \frac{m_{\text{components}}x_{\text{components}} + m_{\text{top load}}x_{\text{top load}}}{m_{\text{components}} + m_{\text{top load}}}
\]

where \( m_{\text{components}} \) is the mass of the components, \( x_{\text{components}} \) is the distance from the pinned location to the components, \( m_{\text{top load}} \) is the mass of the load at the top of the pendulum, and \( x_{\text{top load}} \) is the distance from the pinned location to the location of the load on top. The weight of each of the robot’s components was also needed to determine the overall weight of the robot. This equivalent force that was a result of the robot’s weight was calculated using Equation 3:

\[
F_{\text{equiv.}} = m_{\text{components}} \cdot g + m_{\text{top load}} \cdot g
\]

where \( F_{\text{equiv.}} \) is the summed forces of the pendulum and \( g \) is gravity. The height was dependent on the highest part of the frame that would support the load. The calculations that were done provided the torque required to rotate the robot from a horizontal position based on an assumed weight, providing an adequate safety factor for the design. This torque \( T \) was found with Equation 4:

\[
T = F_{\text{equiv.}} \cdot x_{\text{center}}
\]

These calculations determined that the robot would need 67.5 inch-pounds of torque (33.75 for each wheel). Motors that provided a torque of 3.82 inch-pounds and gearboxes with a gear ratio of 16:1 were purchased, which gave a total torque of 61.1 inch-pounds for each wheel. This design allowed a factor of safety of 1.8 in addition to the safety factor that the calculations provided.

An Arduino™ was purchased to be used as the micro-controller in the block diagram. This is the component that controls the balancing of the robot. The Arduino™ micro-controller was chosen because it is relatively cheap and easy to code (the programming language it uses is based of C/C++). It also contains necessary hardware such as a 10 bit analog to digital converter, which is needed to convert analog data from the sensor without significant loss in precision, and hardware that makes pulse width modulation (PWM) easier, which is useful when controlling the amount of power to the motors. The program that is needed to control the balancing will first have to read what the angle of the device is from the sensor (an inclinometer). The software will also need to incorporate the control system described elsewhere and convert the calculated value of the force into the equivalent voltage required by the motors. It will then need to send the necessary voltage to the motors via the H-bridge. A basic picture of how this will work can be seen in Figure 8.
The inclinometer in the diagram is what the robot will use determine how close it is to being balanced. Using an inclinometer required the use of only one sensor to calculate the angle the pendulum is leaning from vertical. An inclinometer measures the incline or the angle with respect to gravity, so this type of sensor is exactly what is needed in order for the robot to recognize when it is falling over. The inclinometer that was chosen has several useful features. The output of the sensor is an analog signal between 0 and 5 volts, which is easily interpreted by the microcontroller. The sensor also has a sampling frequency of 100 Hz, which is fast enough that the robot can’t move very far between data samples. The batteries in the diagram provide power to the motors, which need 12-18 volts, and the inclinometer, which needs 10-30 volts. The batteries that were chosen for this project are 14.8V lithium polymer batteries. They were chosen not only because their voltage is in the needed range, but also because they can quickly charge/discharge and they were provided by the university.

In the block diagram the H-bridges were used to properly interface the micro-controller to the motors. This is necessary because the microcontroller can only supply 5 volts and the motors require 12 volts. H-bridges already on a chip were very expensive for high current applications, so one was designed instead. The H-bridge that was designed uses four transistors and four optocouplers to control and isolate the voltage from the batteries. Figure 9 shows the design of the H-bridge in further detail.

Figure 8: Outline of Balancing Robot Program
All of the components in the system’s block diagram needed to be housed in a frame. The frame’s design had many constraints due to all the components of the robot. The frame needed to be lightweight, to reduce the amount of torque needed by the motors; it also needed to be durable to prevent damage resulting from repeated falls that are inevitable when testing the robot. The ability to quickly assemble the parts and easy modification of the design were also important. After many different considerations, half inch PVC pipe seemed the most appropriate option for the frame. In order to hold all of the components to the round PVC pipe, a solid base to mount things on was needed. It was determined that Plexiglas™ would be a prudent choice because it is lightweight and relatively easy to shape into the needed plates. To connect all of the Plexiglas™ and most of the components together, screws were used to ensure all of the parts were fitted securely. Velcro™ was used to hold the batteries in place while allowing them to be removed easily for charging if necessary. The 3D model of the robot, developed in Pro Engineer, can be seen in Figure 7.

Figure 9: H-bridge Design

Figure 7: Engineering Drawing of Balancing Robot
Current Status
Many design plans and calculations have been done in order to determine the best parts and the best design for the balancing robot. There was a significant amount of communication to suppliers in order to select all of the best components needed to make the robot. The main accomplishment of the robot now is that all of the parts are ordered and that it is mostly assembled. The frame was constructed. The Plexiglas™ plates, motors, batteries, wheels, and the inclinometer were all attached to the frame. The cost of the components used in the prototype are listed in Table 2.

Table 2: Cost of Robot Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost (U.S. Dollars)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclinometer</td>
<td>260</td>
</tr>
<tr>
<td>Motors and Gearboxes</td>
<td>130</td>
</tr>
<tr>
<td>PVC Pipe and Fittings</td>
<td>50</td>
</tr>
<tr>
<td>Arduino™ Uno</td>
<td>30</td>
</tr>
<tr>
<td>H-bridge components</td>
<td>25</td>
</tr>
<tr>
<td>Plexiglas™, wheels, batteries</td>
<td>Donated</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>495</strong></td>
</tr>
</tbody>
</table>

For the electronics side of the balancing robot the Arduino™ board has been selected to help control all of the sensors and electric components. Research was done on how to build an H-bridge, to help control the speed of the motors. After the research the parts list was created and the components were purchased. A potentiometer was selected that would give the robot the ability to turn once the state-space control system has been developed. The batteries have been tested with the motors that were purchased and they provided sufficient power for two to three minutes of nonstop maximum speed. The balancing robot will not require the motors to rotate at their maximum speed, so this test indicated that the batteries will allow the motors to operate for an acceptable amount of time. The inclinometer was wired to the microprocessor worked as expected. The current stage of the balancing robot can be seen in Figure 10.

![Figure 10: Current Configuration of Balancing Robot](image)
Future Work and Testing Plan
There are several stages of this robot that still need to be completed. The most important stage left is to develop the software that will run the motors to the required torque in order to stay balanced. This will first be based on the transfer function analysis and PID control. It will need to be determined how to relate the inclinometer sensor to the control system in order to actuate the motors. The software will then be adapted to the state space control system to incorporate turning which will require the use of the potentiometer. The H-bridge to help control the motors also needs to be fine-tuned.

The control systems that are developed will first be tested in MATLAB to ensure they are accurate and stable. This will allow each system to be tuned accordingly without risking damage to the robot or its components. After a control system is embedded onto the robot, it will be further tuned to balance the robot with minimal oscillation and the shortest possible response time. This will be done with fuzzy logic for both control systems. Once the robot is optimally tuned, the usefulness of the design will be evaluated by adding an additional load on the top to simulate it carrying a rider. This information can then be used to determine how well a full scale robot of this design could perform when compared to other personal transportation vehicles.

Conclusion
Although it is not yet completed, this project is on a path to achieving its goal of designing and constructing a scaled down prototype of a personal transportation vehicle that can autonomously maintain its balance. The project is also on track to minimize the cost of building the full scale personal transportation device in order to be competitive in the market while maintaining similar performance to what is already available.

The robot was designed and built with lightweight materials in order to minimize the torque that the motors would need to balance the robot. In order to control the system, several components such as two motors, two batteries, an inclinometer, a micro-controller, and two H-bridges were purchased and assembled onto the frame that was constructed. A control system utilizing all of these components needs to be developed to stabilize and balance the robot. A fuzzy logic program will be used to balance the robot. Two different programs will be written to control the system. The first will simply balance the robot, and the second will allow the robot to turn. These fuzzy logic programs will be tested in MATLAB, with the aid of transfer function analysis, to properly tune their parameters for optimal performance before they are used with the prototype.

Once the robot is controlled and properly tested to ensure that it has the capability to function as a personal transportation vehicle, the design’s performance and cost will be compared to similar devices being sold on the market already.
Bibliography


