Self-Balancing Robot Mechatronics: Proposed Control System Simulation and Analysis

Kevin Zhang and Margaret Crawford
Quantitative Engineering Analysis II
Olin College of Engineering Fall 2016
Needham, MA
email: kevin.zhang@students.olin.edu
margaret.crawford@students.olin.edu

I. Introduction

Within the realm of technology, robotics and automation are becoming ever more prevalent and integrated with everyday life. From smart homes to voice activation to human interaction, artificial intelligence is advancing to improve people's lives, both in efficiency and in convenience. One such automation of particular interest is the self-balancing robot. A self-balancing robot is an automated machine that can be given items to hold or carry without falling over and dropping them, thus making them helpful assistants to give users an extra hand. Such robots generally stand upright on only two wheels for an agile movement range and cost efficiency, and they generally hold a flat surface to carry objects. Figure 1 below depicts what a self-balancing robot might look like if manufactured into consumer production. They can be massive to help carry concrete blocks around a construction site, or they can be small to help bring a water bottle across a table. Self-balancing robots are being developed to increase the convenience of carrying around items from place to place.

Fig. 1: An example of a self-balancing robot.

The critical component of a self-balancing robot is that is must be able to balance itself on two wheels, else all other functionality of the robot falls apart. Furthermore, such a robot must not only be able to balance itself and any items placed upon it, but must also be fast and efficient in doing so, thus maximizing convenience for the user. In order to optimize the balancing mechanism of these robots, we endeavored to develop a mechatronic control system that allows a self-balancing robot to balance itself as quickly as possible.

Our report will focus on understanding the mechanics behind a self-balancing robot and developing a control system to manipulate it towards the desired behavior of balancing itself upright. The first section will examine the system and abstract it to a simpler model. The next section will attempt to derive a set of governing equations to capture the behavior of the system. Then we will discuss and validate our governing equations using limiting cases, analytical analysis, and empirical simulation. Finally, after we have a comprehensive understanding of the system and are confident in our ability to define its behavior, the last section will propose a control system and discuss its performance.

II. System and Abstraction

In general, a self-balancing robot consists of two wheels attached to an axle with motors. From the axle a platform will extend vertically to be able to hold objects. When the platform tilts, the forces on it are no longer aligned in the vertical axis, which creates a moment about the platform's center of mass. Because of this, a righting moment must exist to prevent the platform from falling over. The motor will exert a moment on the wheel and an equal and opposite moment onto the platform. This mechanism allows the platform to right itself to vertical if it is tilting. The interaction also lends itself to real world physics: if the platform is tilting forward, then the wheel will move forward to keep it upright. The forward moment that drove the wheel creates a backwards moment on the platform, which brings it back into balance. In this system, we will define x as the horizontal distance the robot has traveled from the origin with the motor driving the wheel, which directly correlates to the angular acceleration of the wheel. We will also define θ, which is the angle of the platform when measured from the horizontal. The desired angle of the platform is thus

$\pi/2$. Figure 2 depicts what the self-balancing robot would like as a simplified model. Note that Figure 1 shows a robot head, arms, and other additional features, whereas the model limits the robot to just its essentials for functionality to reduce the system down to a simpler form that can be more easily handled without losing accuracy of the real system.

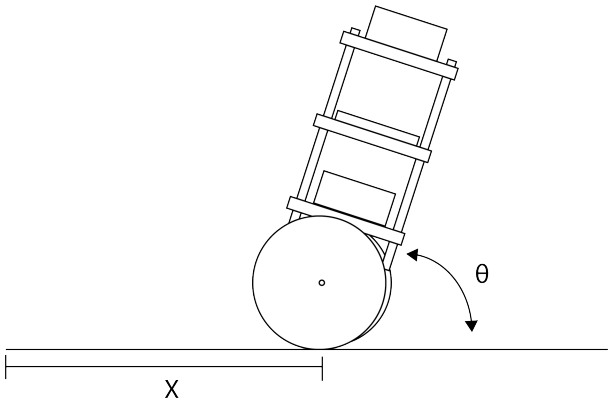
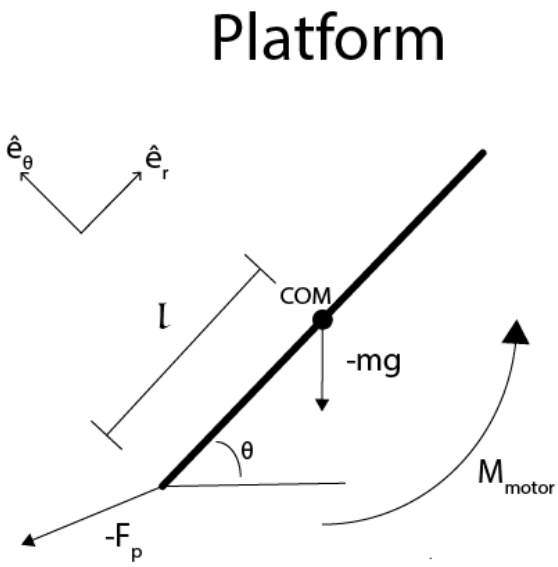


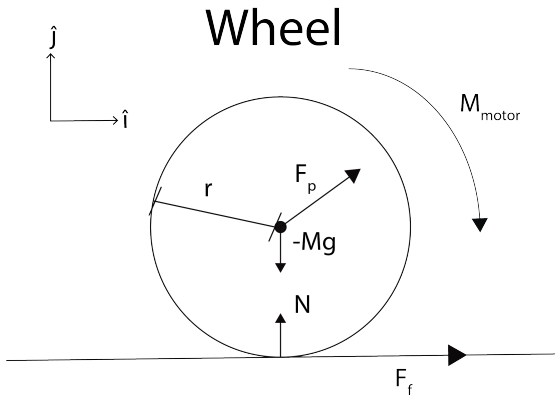
Fig. 2: Abstraction of a self-balancing robot. x is defined as the horizontal position of the robot, and θ is the angle of the platform defined from the horizontal. The desired location of the platform is $\pi/2$

To abstract the system further, we simplify this system down to a two-dimensional plane, where we look at one wheel attached to the platform in a side view, as shown in Figure 3. Rotations are oriented according to the right hand rule, and the current scenario is that the robot is uncontrolled, but is moving with a constant moment coming from the motor, making the entire system move towards the right.

Figure 3a shows the free body diagram of the platform. The platform is assumed to be a rod of uniform density. The center of mass of the platform is assumed to be at the middle point of the platform. θ is defined as the angle of the platform from the horizontal, and the platform is currently having a moment applied to it from the motor. The platform has a force of gravity at its center of mass, as well as a pin force from where it is attached to the axle with the motors. The pin force is a constraint force that can be in any direction, so it is arbitrarily set to be facing away from the platform. The length of the platform is $2l$, with l representing the length of the platform from the axle to its center of mass. The platform is also defined according to a fixed reference frame, with the platform in the direction of $\hat{e}_r$, and its rotation occurring in the $\hat{e}_\theta$ direction, with $\hat{e}_z$ coming out of the page. In the current abstraction, if the robot continues on, then it will speed off towards the right, and the platform will rotate counterclockwise.



(a) FBD of the platform. The platform is assumed to be a rod of uniform density, and exists in a fixed reference frame of $\hat{e}_r$, $\hat{e}_\theta$, and $\hat{e}_z$. l is defined as the length from the axle to the center of mass, and θ is the angle of the platform from the horizontal. The platform has a gravity force from the center of mass which is defined at the middle of the rod. The motor creates a moment about the platform, and a pin force acts as a constraint force from the axle.



(b) FBD of the wheel. The wheel is assumed to be a thick disk of uniform density, and exists in the inertial reference frame. The wheel has an outer radius of r. It contains a pin force to counterbalance that of the platform, a gravity force at its center of mass accompanied by a normal force, and a frictional force exerted on the ground when being driven by the motor. The motor creates a moment on the wheel.

Fig. 3: Free body diagrams of the platform and wheel. In the current scenario, the robot is uncontrolled and moving towards the right at a constant acceleration.

Figure 3b shows the free body diagram of the wheel. The wheel is assumed to be a thick disk of uniform density. The wheel has outer radius r. It currently has a moment applied to it, which is the equal and opposite moment applied to the platform. The wheel exerts a frictional force onto the ground when being driven by the motor, and it also has a gravity force and a pin force, similar to that of the platform. Because the wheel is touching the ground, a normal force serves to counteract gravity. The wheel is also defined according to an inertial reference frame, with the horizontal axis moving in the

$\hat{i}$ direction and the vertical axis aligned with the $\hat{j}$ direction, with $\hat{z}$ coming out of the page. In the current abstraction, the wheel will continue to rotate and move towards the right. In the interaction of the two reference frames, the fixed reference frame of the platform is considered a rotating reference frame in context of the inertial reference frame, meaning that as the platform rotates about the axle of the wheels, the fixed reference frame moves with it.

These will be the abstractions and assumptions used to understand the behavior of the system and create a mechatronics control mechanism.

III. Derivation of Governing Equations

Now that we have our system properly defined and abstracted, we move on to deriving the governing equations of the system. In order to derive the equations of motion, we will first find the translational interactions in the system, and then find the rotational interactions in the system. Our state variables for the system are the angle θ, its derivative $\dot{\theta}$, the horizontal position x and the horizontal velocity $\dot{x}$. We aim to define everything in our system in terms of these quantities as well as $\ddot{x}$ and $\ddot{\theta}$ for our final equations.

To find the translational interactions, we will be using Newton's second law:

$$\sum F = ma \tag{1}$$

And to find the rotational interactions, we will use the rotational version of Newton's second law:

$$\sum \tau = I\alpha \tag{2}$$

where I is the moment of inertia and α is the angular acceleration. We have 6 unknowns, the pin force components F_{px} and F_{py}, the accelerations $\ddot{x}$ and $\ddot{\theta}$, and the forces F_f and N. We then know that we will need 6 equations of motion for the forces and moments on the platform and the wheel.

We will first be deriving the platform's equations. Our strategy will the be to first find the right hand side of the equation, and then equate what we find with the left hand side of the equation. To find the right hand side of Newton's second law, we just need to derive what the acceleration of the platform is, because the masses are known. To find the acceleration of the platform, we will be using the principles of rotating reference frames, and break those down into components along the $\hat{j}$ and $\hat{i}$ axes. We can start by writing our platform mathematically:

$$\vec{r}_{P/02} = l\hat{e}_r \tag{3}$$

where l is the length of the platform from the axle to the center of mass.

Using the principles of rotating reference frames, we know that:

$$\vec{v}_p = \frac{d}{dt}\big|_{01}\vec{r}_{02/01} + \frac{d}{dt}|\vec{r}_{P/02} \tag{4}$$

We can then take the derivative of the velocity equation to find the acceleration of the platform:

$$\vec{a}_p = \frac{d^2}{dt^2}\big|_{01}\vec{r}_{02/01} + \frac{d^2}{dt^2}\big|_{02}\vec{r}_{P/02} + \frac{d}{dt}|^{01}_{01}\vec{\omega}^{02} \times \vec{r}_{P/02} + {}^{01}\vec{\omega}^{02} \times \frac{d}{dt}_{02}\vec{r}_{P/02} + {}^{01}\vec{\omega}^{02} \times ({}^{01}\vec{\omega}^{02} \times \vec{r}_{P/02}) \tag{5}$$

Representing this in terms of our axes defined above, we get:

$$\vec{a}_p = \ddot{x}\hat{i} + 0 + \ddot{\theta}\hat{e}_z \times l\hat{e}_r + 0 + -\dot{\theta}\hat{e}_z \times (-\dot{\theta}\hat{e}_z \times l\hat{e}_r) \tag{6}$$

which simplifies down to:

$$\vec{a}_p = \ddot{x}\hat{i} + l\ddot{\theta}\hat{e}_\theta - l\dot{\theta}^2\hat{e}_r \tag{7}$$

Then, when we express the rotational coordinates in terms of their inertial components, $\hat{e}_\theta = -sin\theta\hat{i} + cos\theta\hat{j}$ and $\hat{e}_r = cos\theta\hat{i} + sin\theta\hat{j}$, so:

$$\vec{a}_p = \ddot{x}\hat{i} - l\ddot{\theta}sin\theta\hat{i} + l\ddot{\theta}cos\theta\hat{j} - l\dot{\theta}^2cos\theta\hat{i} - l\dot{\theta}^2sin\theta\hat{j} \tag{8}$$

Equation 8 is then the acceleration of the platform.

The rest of the calculations are quite simple. For the moment of the platform, I is a known quantity, and the angular acceleration is merely $-\ddot{\theta}$. When deriving the wheel's equations, the horizontal acceleration of the wheel is $\ddot{x}$, and the vertical acceleration is 0 because the wheel doesn't bounce. For the moment of the wheel, I is also known, and using the relationship between angular and linear velocity, the angular acceleration of the wheel in terms of $\ddot{x}$ is $-\ddot{x}/r$. Note the negative signs because the acceleration of the center of mass is opposite the angular acceleration.

The left hand side is the sum of the forces in an inertial reference frame, which can be determined from the free body diagrams. When we combine our two sets of equations for forces, we get the six equations below. Note that N does not give particularly relevant insight to the system, and thus Equation 14 will not be utilized in this system after we define it.

Platform:

$$\sum F_{xplatform} = -F_{px} = m\ddot{x} - ml\ddot{\theta}sin\theta - ml\dot{\theta}^2cos\theta \tag{9}$$

$$\sum F_{yplatform} = -mg - F_{py} = ml\ddot{\theta}cos\theta - ml\dot{\theta}^2sin\theta \tag{10}$$

$$\sum M_{platform} = \vec{m}_{motor} - lF_{p\theta} = -I_{platform}\ddot{\theta} \tag{11}$$

where

$$F_{p\theta} = -F_{px}sin\theta + F_{py}cos\theta \tag{12}$$

Wheel:

$$\sum F_{xwheel} = F_f + F_{px} = M\ddot{x} \tag{13}$$

$$\sum F_{ywheel} = N - Mg + F_{py} = 0 \tag{14}$$

$$\sum M_{wheel} = rF_f - \vec{m}_{motor} = -I_{wheel}\frac{\ddot{x}}{r} \tag{15}$$

To simplify these equations into functions of our state variables, we use substitution for the unknowns that aren't state variables, F_{px}, Fpy and F_f. By substituting Equation 13 into Equation 9 and then substituting Equation 15 into the result, this creates the equation:

$$\frac{\vec{m}_{motor} - I_w\frac{\ddot{x}}{r}}{r} = (M+m)\ddot{x} - ml\ddot{\theta}sin\theta - ml\dot{\theta}^2cos\theta \tag{16}$$

This allows us to express $\ddot{x}, \ddot{\theta}, \theta$ and $\dot{\theta}$ only in terms of each other and constants. It is not a linear equation, because there is a term that depends on both $cos\theta$ and $\dot{\theta}^2$, and one that depends on $sin\theta$ and $\ddot{\theta}$. This will prove to be a problem for analysis later, because we can't take the Laplace transform of it in this form. Notice that the x acceleration does not depend on x position or velocity. Rather, $\ddot{x}$ is dependent on the angular acceleration of the platform. This makes sense because the moment of one object will generate an equal and opposite moment on the other object. Also as expected, the moment from the motor affects both $\ddot{x}$ and $\ddot{\theta}$.

We can also derive another such equation by defining F_{px} as $F_pcos(\theta)$ and F_{py} as $F_psin(\theta)$. We can then use these two definitions in Equation 9 and 10 and combine the two together to arrive at:

$$gcos\theta = \ddot{x}sin\theta - l\ddot{\theta} \tag{17}$$

We are now able to express $\theta, \ddot{\theta}$ and $\ddot{x}$ in terms of each other. This is another nonlinear equation, because of the $\ddot{x}sin\theta$ term. Note once again that $\ddot{x}$ is dependent on $\ddot{\theta}$, which this time is dependent upon gravity, which is to be expected, since the platform is subject to falling down rotating around the axle.

We have now derived our governing equations as well as equivalent versions that only contain state variables and known quantities, and it seems upon first look that these equations relate the behaviors of the system quite accurately.

IV. The Uncontrolled System

Now that we have our governing equations, we will analytically analyze our system to see if the equations can correctly predict behaviors that align with natural physics in the uncontrolled state. We will use various different mathematical tools to analyze these equations, and also validate our analysis with empirical simulations.

A. Limiting Cases

We will begin by looking at limiting cases to gain more insight and validation for our governing equations. If our system behaves as we predict with our governing equations under extreme conditions, then we have a strong case for believing they will work under normal ones. We will explore the case where the wheel's mass is extremely large and the platform's mass is really small, and the case where the wheel's mass is really small and the platform's mass is extremely large. We will be using Equation 16 to examine these cases:

$$\frac{\vec{m}_{motor} - I_w\frac{\ddot{x}}{r}}{r} = (M+m)\ddot{x} - ml\ddot{\theta}sin\theta - ml\dot{\theta}^2cos\theta \tag{18}$$

1) Large wheel, small platform: In the case that the wheel's mass is extremely large and the platform's mass is negligible, we can make the prediction that since the wheel is so large and the platform is so small, all movement in the system will be dictated by the wheel, meaning that any small movements by the wheel lead to a massive response from the platform. Mathematically speaking, we can then approximate the platform's mass to be 0, which changes the equation to simplify down to:

$$\frac{\vec{m}_{motor} - I_w\frac{\ddot{x}}{r}}{r} = (M)\ddot{x} \tag{19}$$

This equation then says that the net moment from the motor affects mainly the wheel's movement, and the platform's forces still exists but pales in comparison to the forces of the wheel. This makes sense because the wheel is so large that it becomes the dominant force in the system, and the platform becomes completely dependent on the movement of the wheel from the motor. Because the platform is so small, it's forces can't override the forces of the wheel, thus meaning that only the wheel's acceleration is directly associated with the motor's moment and the platform's acceleration is becomes completely directed by the wheel, not the motor. This is reflected in Equation 19.

Using MATLAB to simulate this condition, these observations become clear. Figure 4 shows a simulation of the uncontrolled system with no moment from the motor when the wheel is 20 times larger than the platform. All simulations have a starting angle of $2\pi/3$. The graph depicts the movement of the center of masses of the platform and wheel in the x direction. Note that since there is no moment from the motor, the wheel does not move. But because the platform is tilted to begin with, it begins to fall. Since this falling motion creates a moment, one would expect that the wheel would rotate a bit because of the moment generated by the platform. But in this simulation, the mass of the wheel is so great that the platform can't even make it flinch, which is why the center of mass of the wheel doesn't really move while the platform's center of mass falls to the ground. In this simulation it appears that the system acted like an inverted pendulum, which makes sense if the wheel doesn't move. This simulation then aligns with the prediction from the equations of motion that the forces of

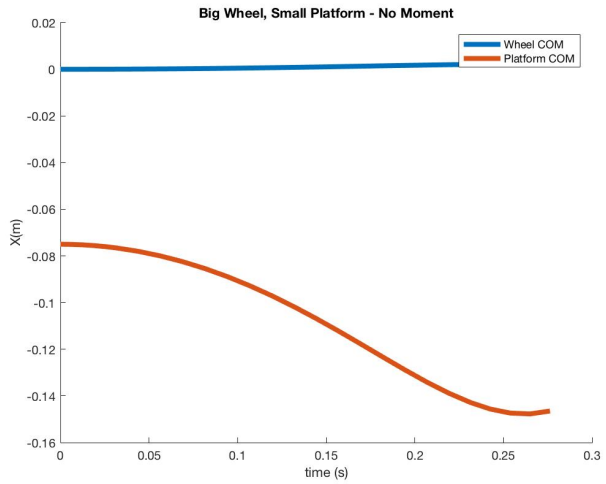


Fig. 4: A graph showing the center of masses of the platform and the wheel in a simulation of a very large wheel and a small platform, with no motor. As can be seen, because there is no motor, the wheel does not move, and the platform can't affect the wheel through its moment while it's falling because it's too light compared to the wheel. This makes sense with our predicted behavior.

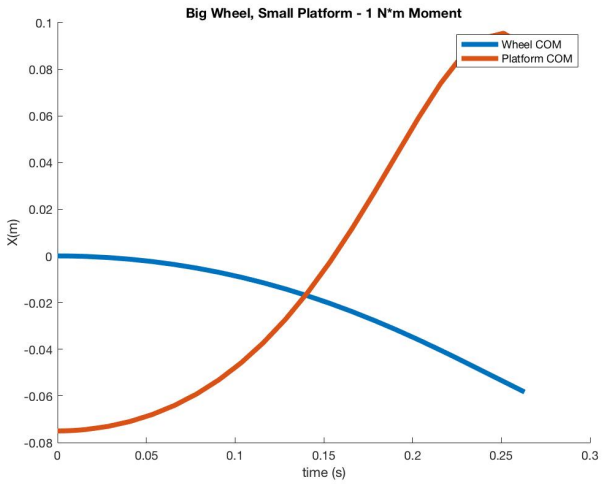


Fig. 5: A graph showing the center of masses of the platform and the wheel in a simulation of a very large wheel and a small platform, with a motor moment of 1 Nm. As can be seen, because there is a moment, the wheel moves slightly, and because of Newton's second law and the law of conservation of angular momentum, the platform rapidly swings around. This makes sense with our predicted behavior.

the platform cannot affect the forces of the wheel when the wheel's mass is much larger because the wheel is the dominant force in the system.

Figure 5 shows a simulation under similar conditions as the one above now with the motor driving the wheel at 1 Nm. As can be seen, with the motor now driving the wheel, the wheel's center of mass now moves a little bit in the x direction. However, the platform's center of mass explodes in the x direction while the wheel barely moves. This makes sense and once again aligns itself with our predictions because the wheel's large mass makes it the driving force that affects the platform instead of the motor, and any small change in the wheel leads to a huge reaction from the platform due to the fact that they have equal and opposite moments acting upon them. According to Newton's second law in the rotational space defined above in Equation 2 and the law of conservation of angular momentum, the wheel's large mass means that any small angular acceleration leads to a huge moment, and this huge moment is reversed onto the platform. Since the platform's mass is so small, that means it'll receive a massive angular acceleration from any moments by the wheel. Thus the wheel can barely move while the platform skyrockets. The simulation then agrees with the predicted behavior of the governing equations.

For this limit case, it appears that our equations have accurately captured the behavior of the system.

2) Small wheel, large platform: The next limit case we will examine is the case where the wheel's mass is negligible and the platform's mass is extremely large. We can make the prediction that since the platform is so large and the wheel is so small, virtually all movement in the system will be directed by the angular acceleration of the platform's angle. The overall behavior can then be tied to the platform's movement. We can then approximate the wheel's mass to be 0, which changes the equation to simplify down to:

$$\frac{\vec{m}_{motor} - I_w \frac{\ddot{x}}{r}}{r} = (m)\ddot{x} - ml\ddot{\theta}sin\theta - ml\dot{\theta}^2 cos\theta \qquad (20)$$

This then shows that the net moment of the motor this time will affect mainly the angular movement of the platform, with a bit of translational movement being exhibited because the platform is linked to the wheel which touches the ground and is affected by the reverse moment from the platform. The platform now becomes the dominant force, and any small changes in the platform create massive responses from the wheel. This behavior is expected because of Newton's Second Law and the law of conservation of angular momentum. Its rotation about the axle, and thus the centrifugal and Coriolis forces, become more dominant in the system. Moments coming from the motor are now more associated with the platform's rotation than with actually moving the robot, meaning that more moment will be needed to achieve the same effect as under normal conditions.

Simulations were done in various conditions to show the behavior of the system in this limit case. Figure 6 first shows a simulation done where the platform was 20 times heavier than the wheel, and the motor had no moment, so the wheel was not driving. As can be seen, the center of mass of the platform barely moves, because it's falling straight down. This makes sense because of Newton's Second Law in Equation

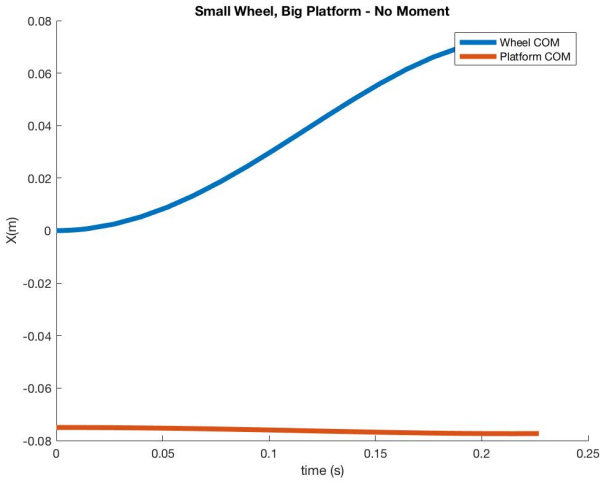


Fig. 6: A graph showing the center of masses of the platform and the wheel in a simulation of a very small wheel and a large platform, with no motor moment. As can be seen, because there is no moment, the wheel shouldn't move, but because of Newton's Second Law and the law of conservation of angular momentum, the rotational motion generated by the platform while it's falling translates onto the wheel, causing it to roll. This makes sense with our predicted behavior.

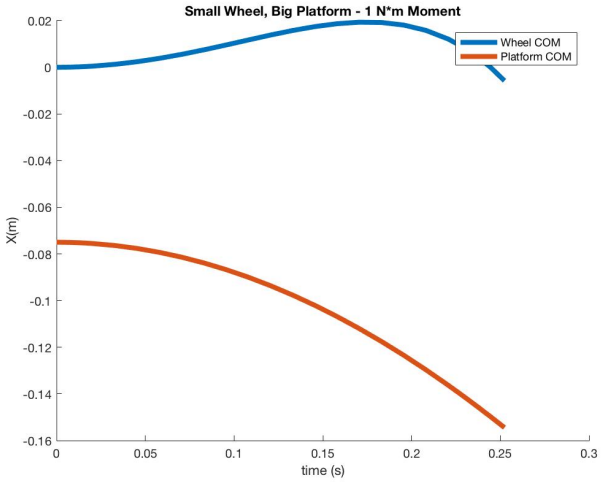


Fig. 7: A graph showing the center of masses of the platform and the wheel in a simulation of a very small wheel and a large platform, with a motor moment of 1 Nm. As can be seen, because there is a moment, the wheel attempts to move one way, but because of the Newton's second law and the law of conservation of angular momentum, the moment from the falling platform brings the wheel back the other way. This makes sense with our predicted behavior.

2 and the law of conservation of angular momentum. The platform's large mass creates a massive moment that must be matched by the wheel, and because the wheel's mass is so small, it's angular acceleration increases dramatically to equate the moment, thus making it roll very far before the platform hits the ground. This is reflected in the graph, as the curve's derivative becomes increasingly positive as the platform falls faster and then slows down when the platform hits the ground. This then proves the model's predictions, in that because the platform is now the dominant force, any small changes in the platform lead to massive responses from the wheel.

Another simulation was done with similar conditions as the one above with now a motor moment of 1 Nm. As can be seen, the wheel attempts to move in a certain direction, but because of the platform's direction of fall and its weight, the wheel was actually forced back the other way, thus the concave and convex curves in the wheel's movement. This is true because the moment driving the wheel is not enough to overcome the moment generated by the falling platform. According to Newton's Second Law in Equation 2 and the law of conservation of angular momentum, the small moment and small mass of the wheel create a small angular acceleration, which is translated to a small moment onto the platform. The massive weight of the platform overcomes this tiny moment and continues to fall in its current direction, and the moment spins the wheel in the opposite direction it was intending to go. This then further proves the model's predictions, in that Newton's second law and the law of conservation of angular momentum assert that a much larger moment is now needed to achieve the same effect as under normal conditions.

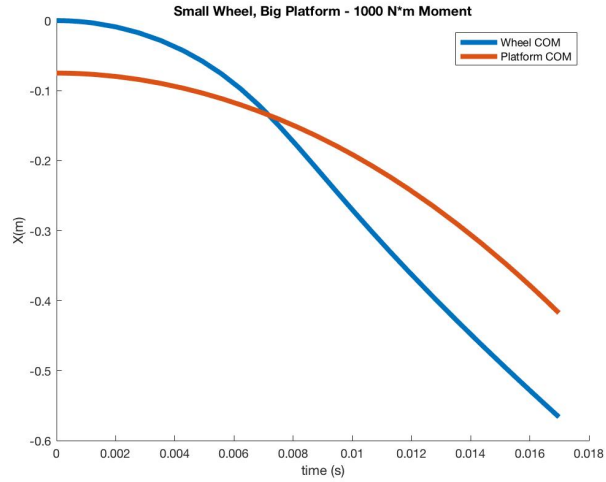


Fig. 8: A graph showing the center of masses of the platform and the wheel in a simulation of a very small wheel and a large platform, with a motor moment of 1000 Nm. Because the moment is so outlandishly high, the wheel can successfully execute Newton's second law and the law of conservation of angular momentum to overcome the moment from the falling platform, thus forcing it to fall the other way while the wheel moves in the direction of the moment from the motor. This makes sense with our predicted behavior.

Another simulation was done with similar conditions as

the one above with now a motor moment of 1000 Nm, a ridiculously and barely realistic value. This time around, the wheel now has the angular acceleration to overcome the moment generated by the falling platform, so the platform now falls accordingly to the wheel. It is only at this ridiculously massive moment coming from the motor that the system can finally behave normally, once again abiding by Newton's Second Law in Equation 2 and the law of conservation of angular momentum, and proving our predictions correct in that when the platform is much larger than the wheel, the moment driving the wheel must also be much larger for it direct the platform's motion.

With both of these limit cases, it appears that our model quite accurately predicts the behavior of the system at both ends of the conditional spectrum.

B. Equilibrium Points and Stability

A final way of analyzing our equations is by finding their equilibrium points and determining their stability. By finding the stability of the system, we can validate that our model is correct and behaves as it would in the real world. We predict that the system is unstable in the uncontrolled state, meaning that since there's nothing acting on the platform to keep it upright, any tilt to the platform will cause it to fall down. In other words, the system in the uncontrolled state will always move away from equilibrium. To prove this, we will find the equilibrium points of the system and analyze them to determine the stability of the system. To find the equilibrium points of ordinary differential equations, we must first linearlize our equations because as mentioned before, non-linear equations are different to handle. Using Taylor series, we can find that:

$$sin\theta = 1 \tag{21}$$

$$cos\theta = \frac{\pi}{2} - \theta \tag{22}$$

for values close to $\frac{\pi}{2}$, which represents the vertical position of the platform. To simplify further, we assume a new variable u which is the complementary of θ, and assume that $\dot{\theta}$ is sufficiently small that it's negligible, so:

$$u = \frac{\pi}{2} - \theta \tag{23}$$

$$\dot{\theta} \approx 0 \tag{24}$$

Using these approximations, we can substitute them into Equation 16 and 17 to produce our linearlized equations:

$$gu = \ddot{x} + l\ddot{u} \tag{25}$$

$$m_{motor} - \frac{I_w \ddot{x}}{r} = (M+m)r\ddot{u} + mlr\ddot{u} \tag{26}$$

Solving the first equation for $\ddot{x}$, substituting it into the second equation, and rearranging the terms, we can find a ordinary, linear, second order differential equation:

$$m_{motor} = -(\frac{I_w}{r}l + Mrl)\ddot{u} + (\frac{I_w}{r}g + Mrg + mrg)u \tag{27}$$

Now that we have a linear differential equation, we can now find its equilibrium points. To find the equilibrium points of a second order differential equation, we will be utilizing the characteristic equation, $r^2 + ar + b = 0$. For any value of c, an equilibrium of a linear second-order ordinary differential equation with constant coefficients $x"(t) + ax'(t) + bx(t) = c$ with $b0$ is stable if and only if the real parts of both roots of the characteristic equation $r^2 + ar + b = 0$ are negative, or, equivalently, if and only if $a > 0$ and $b > 0$. The characteristic equation for our differential equation is:

$$r^2 + b = m_{motor} \tag{28}$$

where:

$$b = -\frac{\frac{I_w}{r}g + Mrg + mrg}{\frac{I_w}{r}l + Mrl} \tag{29}$$

Notice that since all the terms that comprise b are positive, thus making b an overall negative number. That means that in our characteristic equation, b is negative, which violates the above principle of stability. Solving for the roots of the characteristic equation finds that:

$$r = \pm\sqrt{4b}/2 \tag{30}$$

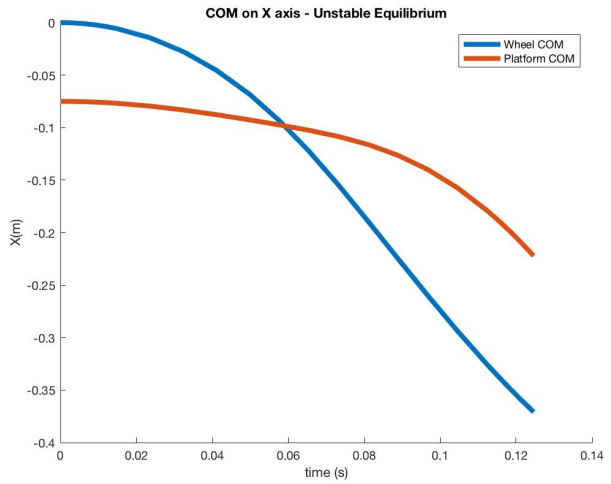
To determine how this plays out with our system, we would take the current equation back into the time domain for easier analysis. Given the roots of the system, r_1 and r_2, the solution to the system is of the general form:

$$y(t) = c_1 e^{r_1 t} + c_2 e^{r_2 t} \tag{31}$$

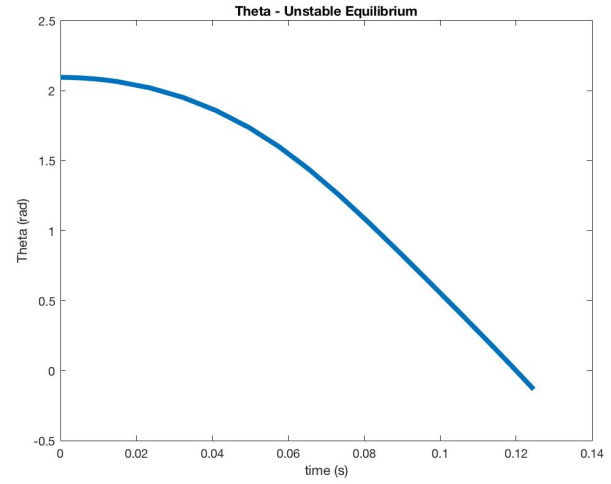
Notice that in the time domain, the roots becomes part of the exponentials, meaning that in this specific case, one exponential function will be a negative exponential, and the other will be a positive exponential. The positive exponential will cause the function to blow up, preventing it from every reaching equilibrium and showing that the uncontrolled system is unstable. This makes sense because in the uncontrolled system the platform has no mechanism to keep it upright against gravity. So long as the platform is tilted even a little bit at rest, it will fall. Thus the system is naturally unstable because the angle of the robot will never reach and stay at equilibrium.

Simulations in MATLAB further validate this predicted behavior.

Figures 9 shows a simulation done with the uncontrolled system where the robot is moving at a constant moment of 1 Nm on the motor. Figure 9a shows the center of mass of the platform and wheel moving along the x axis. Notice that the two center of masses never align together and continue to move in random trajectories that depict no pattern. This means that the two center of masses are moving away from



(a) A graph showing the X axis of the center of masses for the platform and the wheel in the uncontrolled system where the motor is moving at a constant 1 Nm. Notice that the two center of masses never align in equilibrium, and they actually cross over each other because the platform flipped to the other side of the wheel during movement. This shows that the system is unstable.



(b) A graph showing theta vs time in the uncontrolled system where the motor is moving at a constant 1 Nm. As you can see, the theta never reaches equilibrium and continues to drop until it hits the ground.

Fig. 9: Graphs of the uncontrolled system, proving that the system is naturally unstable

equilibrium. Figure 9b further proves this point by showing the angle of the platform over time. As can be seen, the angle of the platform never reaches equilibrium and continues to swing around until it hits the ground, at which the simulation terminates. In this manner, the system is blowing up and never moving towards a consistent value, proving that the system is indeed unstable.

Based on our evaluation for extreme conditions, we believe that our governing equations accurately describe the system of interest. When the platform is much heavier than the wheel, the center of mass of the platform does not translate, and the platform's angular acceleration becomes the dominant force. When the wheel is much heavier, the center of the wheel does not translate, and the wheel's translational motion becomes the dominant force. This analytical analysis proves to align with both natural physics logic and with empirical simulations. In addition, our analysis of the equilibrium of the system shows that the system is naturally unstable, meaning when in the uncontrolled state, will always move away from equilibrium. This also lends itself to natural physics and empirical simulations.

This shows that our understanding of the uncontrolled system has been analytically and empirically validated.

V. The Controlled System

With a comprehensive understanding and strong credibility for our uncontrolled system model, we now move onto proposing a control mechanism to manipulate the system towards a desired behavior.

A. Introducing PID Control

Our proposed control mechanism will be PID control. When we introduce control into the system, we have an closed-loop transfer function that includes the PID control function $G_A(s)$, and the transfer function for the uncontrolled system $G(s)$. As shown in Figure 10, the input to the system is a setpoint that is used to determine error between the current theta of the platform at that instance in time and the desired status of the system, which goes through a PID transfer function to determine the moment for the motor needed to right the system towards the desired behavior. Then, the second transfer function outputs an angle θ based on that moment. The angle is cycled back into the loop and then subtracted from the setpoint to get the new error.

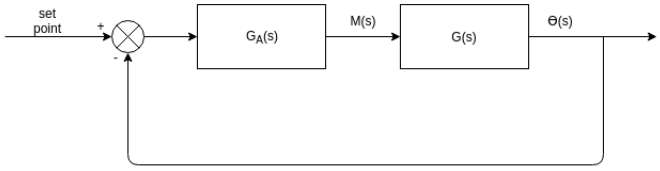


Fig. 10: Block diagram for controlled system. It is a closed feedback loop that constantly updates the needed moment to arrive at the set angle.

To create our control mechanism, we need to get the transfer function for both the uncontrolled system and the control system. We can do this by taking our governing equations, combining them and taking the Laplace transform to create one transfer function with motor moment as input and angle as output, then combining the transfer function with the PID transfer function to create our control mechanism. We will use the linearlized equations we derived before:

$$gu = \ddot{x} + l\ddot{u} \tag{32}$$

$$m_{motor} - \frac{I_w \ddot{x}}{r} = (M+m)r\ddot{u} + mlr\ddot{u} \tag{33}$$

For the uncontrolled system, we take the Laplace transform of the linearlized equations to get

$$gU(s) = s^2X(s) + ls^2U(s) \tag{34}$$

$$M(s) - \frac{I_w s^2}{r}X(s) = (M+m)rs^2X(s) + mlrs^2U(s) \tag{35}$$

and then we solve Equation 32 for $\ddot{x}$ and substitute into Equation 33 to get:

$$\frac{U(s)}{M(s)} = \frac{r}{(I_w g + (M+m)r^2 g) - (I_w l + Mlr^2)s^2} \tag{36}$$

Equation 36 is the transfer function for the uncontrolled system, with motor moment as input and angle, currently defined as the complementary of θ, as output.

Next, we will find the control transfer function using the general form for a closed-loop control transfer function:

$$\frac{G_A(s)G(s)}{1 + G_A(s)G(s)} \tag{37}$$

where $G_A(s)$ is the control transfer function and $G(s)$ is the uncontrolled transfer function. For a PID control system, the transfer function is:

$$G_A(s) = K_p + Kds + \frac{K_i}{s} \tag{38}$$

However, this equation, when combined with our uncontrolled transfer function, creates a third order denominator, which makes finding the roots much harder. So, we decided to solve for PD control and experimentally find I, which takes the form:

$$G_A(s) = K_p + K_d s \tag{39}$$

Thus, the control transfer function is found by combining Equation 36 with Equation 39 using Equation 37, which can be simplified to:

$$\frac{(K_p + K_d s)r}{-(I_w l + Mlr^2)s^2 + K_d rs + (I_w g + (M+m)r^2 g + K_p)} \tag{40}$$

Equation 40 is our overall transfer function for the controlled system. Looking at this function, it appears that K_d is known as the damping constant because it's associated with the s term, which makes sense because it handles the derivative of the control and thus slows down or speed up convergence. K_p is a constant, which means it directly affects how fast the system converges. Most of the terms are known quantities, with K_d and K_p being the only two unknown. This makes sense because we must tune our PID control using K_d and K_p to achieve various states of stability and various speeds of convergence on equilibrium. Using this function, we can now find the parameters for K_d and K_p that will lead to the fastest time needed for a robot to balance itself.

To analyze the times of convergence for balance, we must find the poles of our control transfer function. We can use the standard quadratic equation on the denominator of Equation 40 to find:

$$s = \frac{-K_d r \pm \sqrt{P}}{-2(I_w l + Mlr^2)} \tag{41}$$

where

$$P = K_d^2 r^2 + 4(I_w l + Mlr^2)(I_w g + (M+m)r^2 g + K_p r) \tag{42}$$

Setting P equal to zero, we can find the relationship between K_D and K_P. At this point we substitute actual values for parameters to simplify our constants. The actual values were experimentally calculated from real parts that would've constituted an actual self-balancing robot. After multiplying out our values, we arrive at the equation:

$$-462.63K_D^2 = K_P \tag{43}$$

Equation 43 tells us that the K_D will be significantly smaller than the K_P, which lends itself to the idea that in a self-balancing robot system, what's needed is extra proportional gain to quickly right the platform before it falls over rather than try to dampen the righting motion, which makes sense because of the time frame from the vertical position to before the platform hits the ground. The equation also tells us that K_D and K_P will be negative values in order for the system to be stable. While normally, it is believed that PID values should be positive, we believe that our derivation led to this conclusion because of the way we defined our system.

We now have created a mechatronics control system and have the ability to find the optimal configurations for it to right a self-balancing robot in the fastest possible time.

B. Evaluation of the Proposed Control System

Now that we have found our PID control system, we will evaluate its performance in manipulating the system towards the desired behavior of keeping the platform in the vertical position. We will investigate three different cases to determine the accuracy of our proposed control mechanism.

1) Unstable Control: First we will evaluate whether our control mechanism can properly fail to control our system. Our control mechanism should fail if even with its implementation the system is still an unstable system that can't converge to an equilibrium. To analyze this, we will find values of K_d and K_p that would perpetuate an unstable system, and validate the predicted behavior through simulation. To find values for PID that would further destabilize the system, we will look to find values that will result in a function in the time domain that blows up. This specifically means that the poles of the controlled transfer function will not all be negative, thus creating a positive exponential in the time domain that explodes the system. This can be very easily done by choosing a K_d value that is positive, which given Equation 41 would create at least one pole that is positive, assuming that K_p is not comparatively larger than K_d. Thus unstable control can be achieved if

$$K_d > 0 \tag{44}$$

$$K_d r > \sqrt{P} \tag{45}$$

Analytically, this makes sense because if a pole is positive then that means a positive exponential will exist in the time domain. That positive exponential will then proceed to blow up the system.

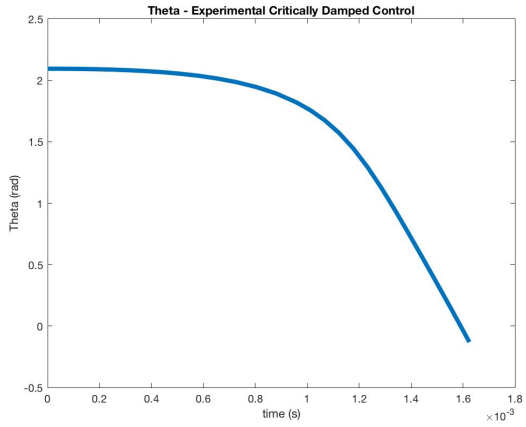


Fig. 11: A graph depicting a controlled system simulation through the angle of the platform over time given the PID values of K_p = -100, K_i = -.01, and Kd = 4. With a positive K_d, it is very evident that the angle of the platform does not move towards equilibrium. This proves the idea that a positive K_d should create an unstable control mechanism.

Empirically, this also makes sense. Figure 11 shows a simulation of the controlled system using the PID values K_p = -100, K_i = -.01, and Kd = 4. As can be seen, a positive K_d value blows up the system, as the angle of the platform never converges towards the vertical equilibrium point and continues to swing around until it hits the ground. This then validates the idea that our control mechanism can properly fail given the right conditions.

2) Underdamped Control: Next, we will look at Underdamped Control, meaning that the control mechanism will create an actual controlled system where the platform converges to the vertical, but it is a slow and generally oscillatory convergence because the control wasn't precise enough with its calculations and thus undershoots the equilibrium, forcing it to take longer to reach convergence. Most PID control values fall into either this category or Overdamped Control, which is when the control system overshoots the equilibrium point and thus also takes longer to reach convergence. Since both types of control exhibit relatively similar behavior of sub-optimally being able to control the system, we will merely look at Underdamped Control. Underdamping occurs when the PID values create imaginary parts to the poles, that is, in Equation 41 the poles become imaginary when

$$P < 0 \tag{46}$$

This occurs when the characteristic roots are not real. In terms of finding values for K_d and K_p that work for this type of system, the following condition must be satisfied:

$$K_d^2 r^2 < 4(I_w l + Mlr^2)(I_w g + (M+m)r^2 g + K_p r) \tag{47}$$

This makes sense because it means that the damping constant, K_d, is really small compared to the rest of the control system, in particular K_p, which would create underdamping. But given what we found in the unstable control section, K_d must be negative such that the system actually converges and reaches an equilibrium. Thus as long as K_d is a small negative value, the control system will be underdamped.

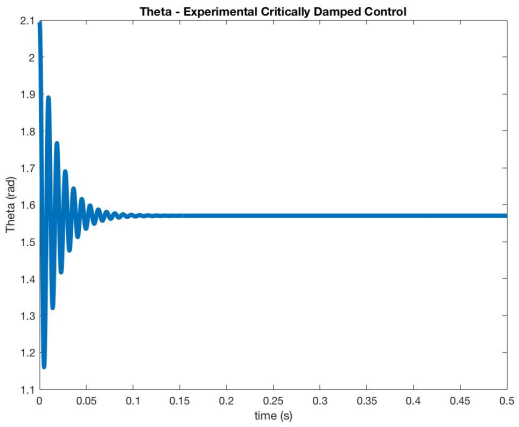


Fig. 12: A graph depicting a controlled system simulation through the angle of the platform over time given the PID values of K_p = -500, K_i = -.01, and Kd = -0.1. With a small negative K_d, the system inefficiently oscillates towards the vertical equilibrium point, proving that the system is underdamped.

Simulations also agree with the predicted behaviors. Figure 12 shows a simulation of the controlled system using PID values of K_p = -500, K_i = -.01, and Kd = -0.1. Notice that the angle of the platform oscillates and slowly moves towards $\pi/2$. This behavior matches the analytical evaluation of an underdamped system. With K_d being -0.1, the poles of the system contain imaginary parts. The fact that K_d is comparatively small to K_p slows down convergence because the system undershoots, and the imaginary parts of the poles create sines and cosines in the time domain due to conversions of i, which explains the oscillatory behavior. This then empirically validates the underdamped control system, as the analytical behavior matches that of simulation.

3) Critically Damped Control: Finally, we will analyze the critically damped control system, defined as the control system with parameters such that it creates the behavior marked by fastest convergence, thus creating the optimal control mechanism. Due to recent findings with our simulations and experimentation, we will first present what was found by the team and then discuss possible areas of improvement to the

optimized control system that were discovered near the end of this project.

A critically damped system is one where the system converges towards equilibrium the fastest. In other words, the angle of the platform neither undershoots nor overshoots the equilibrium point, instead hitting it straight on and reducing any lag time. This is defined as when the poles of the characteristic equation for the control transfer function are the same and as negative as possible. Using Equation 41, this means that for a critically damped system:

$$P = 0 \tag{48}$$

As long as P becomes 0 and K_d is as negative as possible, then the system becomes critically damped. This is true because qualitatively speaking, when P is 0, there is only one root to the characteristic equation. This then means that the same exponential power will appear in the time domain function. In comparison, if P was not 0, then the time domain function would result in two different exponentials whose powers would be one higher and one lower than the powers of the exponentials when P is 0. In time domain function, higher power exponentials always dominate, thus meaning that a system where P is not 0 will always converge more slowly than a system where P is 0 because of a higher power exponential which converges more slowly to an equilibrium will always exist in the former. Then to maximize the speed of convergence, we find the K_d value that is as negative as possible and still satisfies the above condition. This will lead to the critically damped system with our control mechanism. In terms of the what this should look like, rather than spinning out of control and blowing up or oscillating back and forth before reaching a steady state, a critically damped robot should have its platform smoothly reach the vertical equilibrium point without any oscillation, thus converging and reaching stability as fast as possible.

The simulation of our critically damped system brings up some small questions about the credibility of our system. Figure 13 shows the simulation that was performed using the value for PID derived from our control transfer function. As can be seen, when using PID values of K_p = -462, K_i = -.01, and Kd = -.9979, which were found from our mathematical calculations and should represent the critically damped control system, the system is actually overdamped, as shown by the small overshoot and then immediate convergence in the graph. This means that the angle of the platform potentially had too much power and not enough damping force when it was trying to right itself, thus overshooting by a small margin.

Upon further research and experimentation, it was discovered that the critically damped system does not actually occur when P = 0 and the roots of the characteristic equation are the same and as negative as possible. The reason lies in Equation 40, which contains an additional s term. Our current system assumes that the fastest convergence occurs when the two roots coincide, but this mainly works for first order differential systems. Ours is second order. What we found is actually

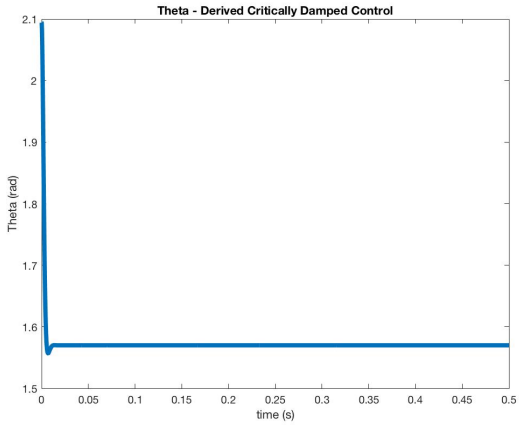


Fig. 13: A graph depicting a controlled system simulation through the angle of the platform over time given the PID values of K_p = -462, K_i = -.01, and Kd = -.9979. As can be seen, using the derived PID values that should create a critically damped system actually result in a slightly overdamped system since there is the very noticeable bump in the curve of the angle of the platform over time.

called the 'breakaway' point of the root locus problem and represents a point where the poles move from being on the real axis to being complex, but it doesn't necessarily represent the fastest approach to equilibrium. It appears that in order to truly find the most optimal control system, certain additional mathematical steps must be taken in regards to the numerator which includes the s term. The s term is a byproduct of having a second order system but still significantly impacts how it behaves, however this was ignored in our analysis. Due to time constraints and lack of easily obtainable resources, we will not be pursuing this extra venture at this time.

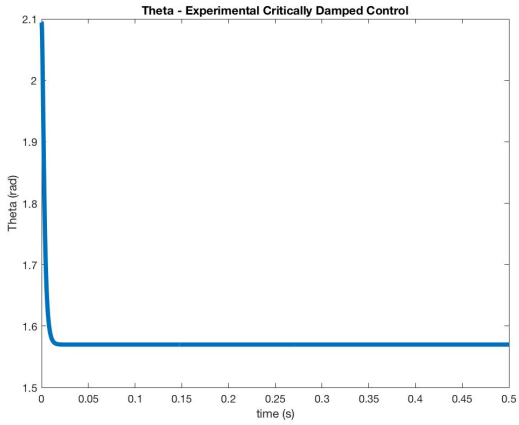


Fig. 14: A graph depicting a controlled system simulation through the angle of the platform over time given experimentally found PID values of K_p = -462, K_i = -.01, and Kd = -1.5. When compared to our derived critically damped system, the difference is a fraction of a second.

However, we do believe that we have found if not the fastest, at least a very fast stable control system that can correctly manipulate our system to a desired behavior. We have also performed extensive qualitative and quantitative analysis to ensure that we can both comprehend the system we were dealing with and control it with a self-created control mechanism.

For comparison, we conclude this paper with a simulation that uses a truly critically damped system with our control mechanism that was experimentally found through trial and error. Figure 14 shows this simulation, which used PID values of K_p = -462, K_i = -.01, and Kd = -1.5. Note that the curve of the graph is not that different from ours. In fact, when compared to the simulation graph of our derived critically damped system, the difference in time it takes for the angle of the platform to converge to the vertical is less than .01 seconds. The only major difference is that it doesn't overshoot slightly. With this in mind, we believe we found a very effective mechatronics control system for balancing self-balancing robots.

VI. Letter to the Editor

Thank you for the feedback on our technical paper you provided us with last week. We greatly valued your brilliant insights on our paper and the areas of possible improvement. In response to this feedback, we have edited several portions of the paper. We hope that the changes, as noted below, successfully address the concerns you had.

First, we wrote in significantly more interpretations of the graphs and visuals used in our report. We made sure that all visuals were adequately described, and that all graphics and all text could stand alone. We also included many more additional visuals as per the feedback, such as including system figures, adding in more descriptive and useful graphs for analysis, and remaking our graphs for the Control section.

Second, we made sure to more fully analyze any results that we arrived at within our report, such as when we derived the final versions of the governing equations, or whenever we arrived at a conclusion about an analytically predicted behavior in the uncontrolled and controlled sections. We also better explained our steps in derivations, including big picture explanations as well as clarifications on what we were trying to achieve on certain thought processes in the report.

Next, we clarified the context in the Information section and gave a better roadmap. We also reoriented the two free body diagrams in the Systems and Abstractions section such that they were right next to each other for better viewing. We also explained the Derivation of the Governing Equations section better such that readers could better understand the big picture and the approach we were using. We used a more step-by-step approach to ensure readers could follow our derivations.

We combined the sections for the Uncontrolled System into one, and did the same for the Controlled System. We reformatted the structure such that both sections explained and derived analytical behaviors for multiple cases to validate or better understand the system, which were followed by simulations to empirically prove the predicted behaviors. We were also more explicit in our explanations and provided more analytical analysis over mathematical derivations.

Next, we flushed out the Control Systems and the PID control section out such that it was robust and thorough. We also included additional validation and empirical evidence that our control mechanism does work as intended. We made sure to include a punchline.

Finally, there were a number of small changes to the paper, such as clarifying some graphics, adding additional explanations in some places for clearer understanding, and proofreading for grammatical and spelling errors. Thank you once again for your feedback. Despite its ups and downs, QEA was a great class, and we both believed we learned a lot from this course. Thank you for being our teachers for this entire year. We hope to have you again as mentors one day.

Acknowledgment

The authors would like to thank Mark Somerville, John Geddes, Siddhartan Govindasamy, Christopher Lee and Rebecca Christianson for their extensive guidance in the progression of this project and the learning of building blocks that happened prior to this experiment. Finally a big thanks to the entire Quantitative Engineering Analysis class for its assistance at various stages of the project.