Mechatronic Systems Engineering
School of Engineering Science
SIMON FRASER UNIVERSITY

MSE 483 – Modern Control Systems
- Project Report -

Two-Wheeled Balancing Robot Control

Prepared by:

Devpreet Bhullar 301217638

Kirill Salita 301209852

Abstract:

This report provides details our effort at designing a controller for a two-wheeled self-balancing robot. We began with developing a dynamics model that was converted into a fourth-order state-space representation. We studied and analyzed the controllability and observability of the state equations. A feedback controller was developed using state feedback control laws. The steady state performance of the system response was optimized by adding an integrator to the controller. The system was tested using MATLAB and a SimMechanics model was also developed in order to rigorously test the controller.

April 19th 2016

Table of Contents

Table of Figures

1. Introduction

Two-wheeled balancing robot consists of a body and two wheels actuated by electric motors (Figure 1). Its purpose is to perform translational motion and to remain in vertical position despite having only two wheels. It is an inherently unstable system that requires a well-tuned controller in order to operate. Traditionally the balancing robots are created using PID controllers [1] to control the body angle. In this project we attempt to develop a position control algorithm for a two-wheeled balancing robot using observer-based state-feedback controller instead of PID.

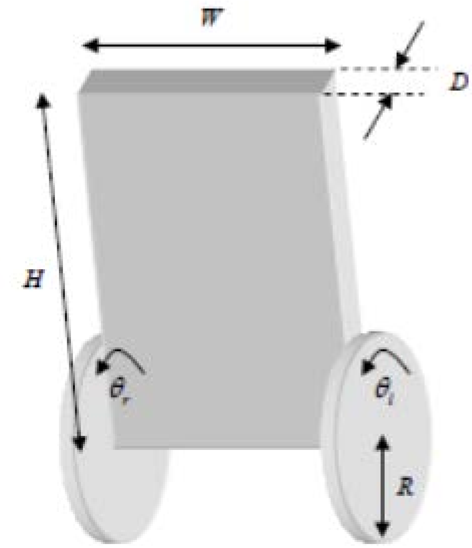

Figure 1: A two-wheeled self-balancing robot

In part 2 of this report we develop a dynamics model of the system using free body diagrams to determine the dynamic equation. We then convert the system of equations into a fourth-order state-space representation and determine the model parameters.

In part 3 we study the system's controllability and observability. Then we apply state feedback laws to develop a controller and we also develop an observer.

Part 4 describes our approach to modelling the linearized model of the system in Simulink and the non-linear model in Simmechanics.

In part 5 we discuss results of the simulations that we performed on our linearized system using MATLAB and Simulink. We then proceed to results of nonlinear system testing using SimMechanics.

In conclusion we discuss the effectiveness of the developed controller and observer systems and present our recommendations on how the controller can be improved to increase the system performance.

2. Analysis

2.1 Dynamic System Derivation

We start by drawing a free body diagram of the robot's wheels (Figure 2)

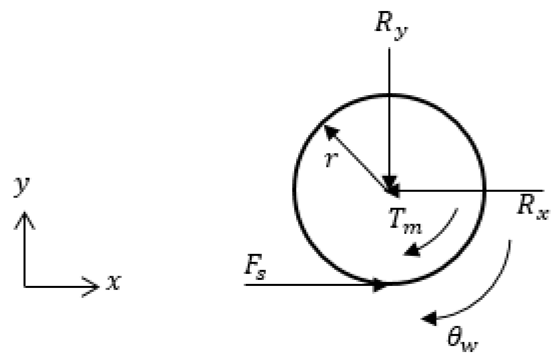

Figure 2: Free body diagram of the wheel of the robot

From the free body diagram we write:

$$\ddot{\theta}_w = \frac{1}{J_w}(T_m - r * F_s) \qquad (1)$$

$$\ddot{X}_w = \frac{1}{M_w}(F_s - R_x) \qquad (2)$$

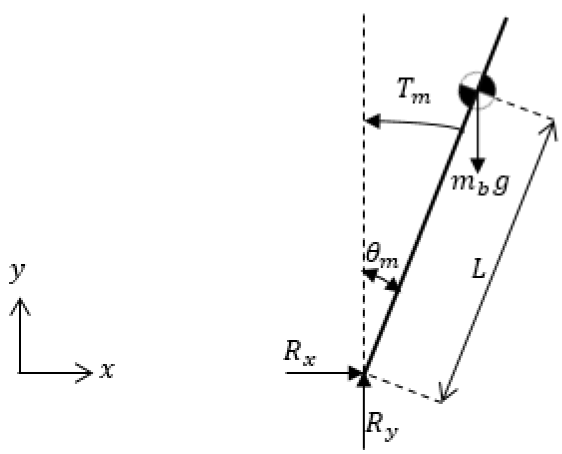

Figure 3: Free body diagram of the body of the robot

And from the free body diagram of the robot body:

$$\ddot{\theta}_b = \frac{1}{J_b}\left(R_x * L * \cos(\theta_b) + R_y * L * \sin(\theta_b) - T_m\right) \qquad (3)$$

$$\ddot{X}_b = \frac{1}{M_b}(R_x - M_b * g * \sin(\theta_b)) \qquad (4)$$

We also assume no-slip condition for the wheels and write:

$$\theta_w = r * X_w \qquad (5)$$

Solving (1) for F_s we get:

$$F_s = \frac{1}{r}\left(T_m - \ddot{\theta}_w * J_w\right) \qquad (6)$$

Substituting (5) into (6):

$$F_s = \frac{1}{r}\left(T_m - \frac{J_w}{r}\ddot{X}_w\right) \qquad (7)$$

Solving (2) for R_x and substituting (7) for F_s we get:

$$R_x = -\frac{M_w * r^2 + J_m}{r^2}\ddot{X}_w - \frac{1}{r}T_m \quad (8)$$

Noticing that $X_w = X_b = X$ we substitute (8) into (4) to get:

$$\ddot{X} = \ddot{X}_b = \frac{-r^2 * g * M_b}{(M_b + M_w)r^2 + J_w}\sin(\theta) + \frac{r}{(M_b + M_w)r^2 + J_w}T_m \quad (9)$$

Since we want to obtain a linearized model of the system about the point $\theta = 0$ we simplify the expression using small angle approximation $\sin(\theta) = \theta$ to obtain the final differential equation for the linear acceleration of the robot $\ddot{X}$:

$$\ddot{X} = \ddot{X}_b = \frac{-r^2 * g * M_b}{(M_b + M_w)r^2 + J_w}\theta + \frac{r}{(M_b + M_w)r^2 + J_w}T_m \qquad (I)$$

Moving on to the angular acceleration equation we note that:

$$R_y = M_b * g * \cos(\theta) \qquad (10)$$

Substituting (8) and (10) into (3) we get:

$$\ddot{\theta}_b = \frac{r^2\left(g * L * M_b * \sin(\theta_b) * \cos(\theta_b) + \ddot{X} * L * M_w * \cos(\theta_b) - T_m\right)}{J_b * r^2}$$

$$+ \frac{r * L * T_m * \cos(\theta_b) + L * J_m * \cos(\theta_b) *}{J_b * r^2}$$

Linearizing the above equation using small angle approximation $\sin(\theta) = \theta$, $\cos(\theta) = 1$ we obtain:

$$\ddot{\theta}_b = \frac{r^2\left(g * L * M_b * \theta_b + \ddot{X} * L * M_w - T_m\right) + L * r * T_m + L * J_w * \ddot{X}}{J_b * r^2}$$

Substituting (I) into the above and simplifying we obtain the expression for $\ddot{\theta}_b$:

$$\ddot{\theta}_b = \frac{g * L * M_b\left((M_b + 2 * M_w)r^2 + 2 * J_w\right)}{Jb((M_b + M_w)r^2 + J_w)}\theta + \frac{-r^3(M_b + M_w) - L * r^2(M_b + 2 * M_w) - r * J_w - 2 * L * J_w}{r * J_b(r^2(M_b + M_w) + J_w)} Tm \qquad (II)$$

2.2 State-Space Representation

Choosing the state variables as follows

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} X \\ \dot{X} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

We can then express equations (I) and (II) in a state space representation as

$$\begin{bmatrix} \dot{X} \\ \ddot{X} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \beta & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma \\ 0 \\ \delta \end{bmatrix} T_m \qquad \text{(III)a}$$

Choosing body position as the output, we can write the output state equation as

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{(III)b}$$

Where:

$$\alpha = \frac{-r^2 g M_b}{(M_b + M_w)r^2 + J_w}$$

$$\beta = \frac{gLM_b\left((M_b + 2M_w)r^2 + 2 * J_w\right)}{Jb\left((M_b + M_w)r^2 + J_w\right)}$$

$$\gamma = \frac{r}{(M_b + M_w)r^2 + J_w}$$

$$\delta = \frac{-r^3(M_b + M_w) - Lr^2(M_b + 2M_w) - rJ_w - 2LJ_w}{rJ_b(r^2(M_b + M_w) + J_w)}$$

For the scope of the project we define physical parameters of the balancing robot as follows:

Table 1: Parameters with their corresponding symbols and values

Parameter	Symbol	Value	Unit
Mass of The Body	M_b	0.5	kg
Mass of Two Wheels	M_w	0.04	kg
Body Moment of Inertia	J_b	0.0015	kgm^2
Wheel Radius	r	0.03	m
Wheel Moment of Inertia (Both Wheels)	J_w	5e-5	kgm^2
Distance from Axis To Center of Gravity of The Body	L	0.1	m

We also use value of gravitational acceleration g = 9.81 m/s^2

Using the values above the matrices of the state equation become:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -8.2360 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 379.47 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 55.970 \\ 0 \\ -3245.4 \end{bmatrix}$$

3. Controller Design

3.1 Controllability

In order to assess the controllability, we need to find the controllability matrix P.

$$P = [B \quad AB \quad A^2B \quad A^3B]$$

$$A^2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \beta & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \beta & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & \beta \end{bmatrix}$$

$$A^3 = A^2 \times A = \begin{bmatrix} 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & \beta \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \beta & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \alpha \\ 0 & 0 & \alpha\beta & 0 \\ 0 & 0 & 0 & \beta \\ 0 & 0 & \beta^2 & 0 \end{bmatrix}$$

$$\therefore P = \begin{bmatrix} 0 & \gamma & 0 & \alpha\delta \\ \gamma & 0 & \alpha\delta & 0 \\ 0 & \delta & 0 & \beta\delta \\ \delta & 0 & \beta\delta & 0 \end{bmatrix}$$

Since all column vectors of P are independent, rank of P is the same as number of columns, i.e. 4. Thus, using Theorem 3.2 [2], the linear state equation described in equation (III) is controllable.

3.2 Observability

Observability can be measured by finding the observability matrix Q as follows

$$Q = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix}$$

$$\therefore Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & \alpha \end{bmatrix}$$

Since all column/row vectors of P are independent, the rank of Q is 4.

Therefore, using Theorem 4.2 [2], we can deduce that the linear state equation described in equation (III) is observable.

3.3 State Feedback Controller

Given the system defined by matrices A, B, C as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -8.2360 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 379.47 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 55.970 \\ 0 \\ -3245.4 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

We first compute eigenvalues of the system:

$$\det(\lambda I - A) = 0$$

$$\lambda = [0, \quad 0, \quad 19.4799, -19.4799]$$

We note that the system is unstable due to double pole s = 0 and a positive real pole at s = 19.4799

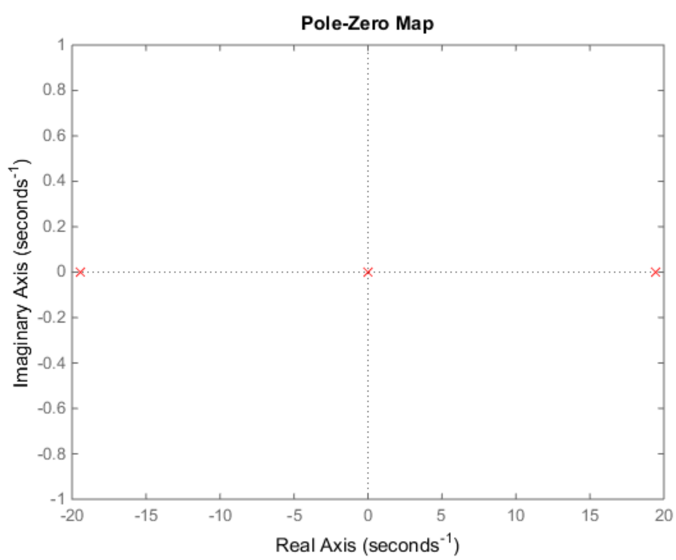

Figure 4: Pole-zero map showing the eigenvalues of the open loop system

We want to achieve settling time t_s = 1s and percent overshoot P.O. = 5%. Therefore, we can obtain values for desired ζ and ω_n.

$$P.O. = 100 * e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} = 5$$

$$\zeta = 0.69$$

$$t_s = \frac{4}{\zeta * \omega_n}$$

$$\omega_n = 5.797$$

Based on the obtained values of ζ and ω_n we determine that we want to place the two dominant poles of the system at locations:

$$-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -4 \pm j4.196$$

We also want to move the pole at 19.4799 + j0 to the LHP to location - 19.4799 + j0

The desired characteristic polynomial then becomes:

$$(s + 19.4799)^2(s + 4 - j4.196)(s + 4 + j4.196) = 0$$

$$s^4 + 46.9598\, s^3 + 724.751\, s^2 + 4345.03\, s + 12752.5 = 0$$

Adding state controller to the state equation of the system, the new state equation becomes:

$$\dot{X} = (A - Bk)X + Br$$

Or

$$\dot{X} = \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -8.2360 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 379.47 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 55.970 \\ 0 \\ -3245.4 \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \right) X + \begin{bmatrix} 0 \\ 55.970 \\ 0 \\ -3245.4 \end{bmatrix} r$$

Which is equivalent to

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -55.97\, k_1 & -55.97\, k_2 & -55.97\, k_3 - 8.236 & -55.97\, k_4 \\ 0 & 0 & 0 & 1 \\ 3245.4\, k_1 & 3245.4\, k_2 & 3245.4\, k_2 + 379.47 & 3245.4\, k_4 \end{bmatrix} X + Br$$

Finding characteristic equation of the above matrix:

$$\det\big(sI - (A - Bk)\big) =$$
$$= s^4 + 55.97(k_2 - 57.9846\, k_4)s^3 + 55.97\big(k_1 - 57.9846(k_3 + 0.116925)\big)s^2$$
$$+ (5490.18\, k_2)s + 5490.18\, k_1 = 0$$

Comparing this to the desired characteristic equation

$$s^4 + 46.9598\, s^3 + 724.751\, s^2 + 4345.03\, s + 12752.5 = 0$$

We obtain for following set of equations:

$$55.97(k_2 - 57.9846\, k_4) = 46.9598$$

$$55.97\big(k_1 - 57.9846(k_3 + 0.116925)\big) = 724.751$$

$$5490.18\, k_2 = 4345.03$$

$$5490.18\, k_1 = 12752.5$$

Solving these we obtain:

$$k_1 = 2.3228 \quad k_2 = 0.79142 \quad k_3 = -0.300183 \quad k_4 = -0.000821$$

3.4 Observer

Open-loop characteristic polynomial is as follows

$$a(s) = s^4 - 379.467s^2$$

$$a_3 = 0, a_2 = -379.467, a_1 = 0, a_0 = 0$$

Desired closed-loop eigenvalues for the controller are

$$(s + \ 19.4799)^2(s + 4 - \ \mathrm{j}4.196)(s + 4 + \ \mathrm{j}4.196) = 0$$

We can scale the eigenvalues by 10 to obtain the desired characteristic polynomial for the observer error dynamics

$$\alpha(s) = (s + \ 194.799)^2(s + 40 - \ \mathrm{j}41.96)(s + 40 + \ \mathrm{j}41.96)$$

$$\alpha_3 = -194.799, \alpha_2 = -194.799, \alpha_1 = -40 + j41.96, \alpha_0 = -40 - j41.96$$

The observer gain matrix is

$$L = \begin{bmatrix} 271.2273 & -19.1454 \\ 1.4888 \times 10^4 & -3.7377 \times 10^3 \\ 161.2748 & 198.3707 \\ 3.1416 \times 10^4 & 1.0752 \times 10^3 \end{bmatrix}$$

4. Modelling

4.1 Simulink Linear Model

We model the linearized system plant in Simulink using the model shown in Figure 5.

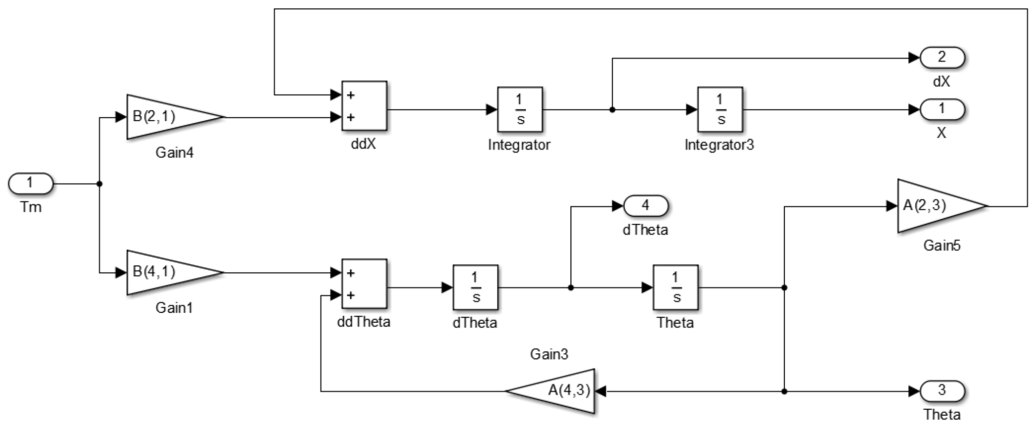

Figure 5: Simulink Model of the Linear Plant

We then model the system with the observer, the observer-based state feedback controller and the integrator using the model shown in Figure 6.

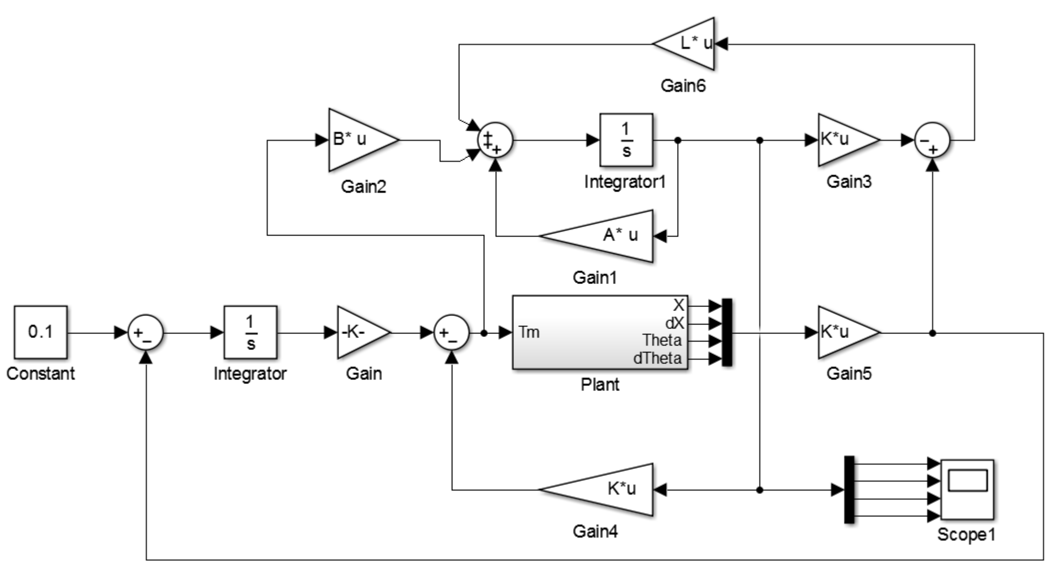

Figure 6: Simulink Model of the Observer-Based Controller

4.1 SimMechanics Non-Linear Model

We model the system in SimMechanics to be able to test the controller performance with non-linear system model. We create the non-linear system plant using the diagram shown in Figure 7.

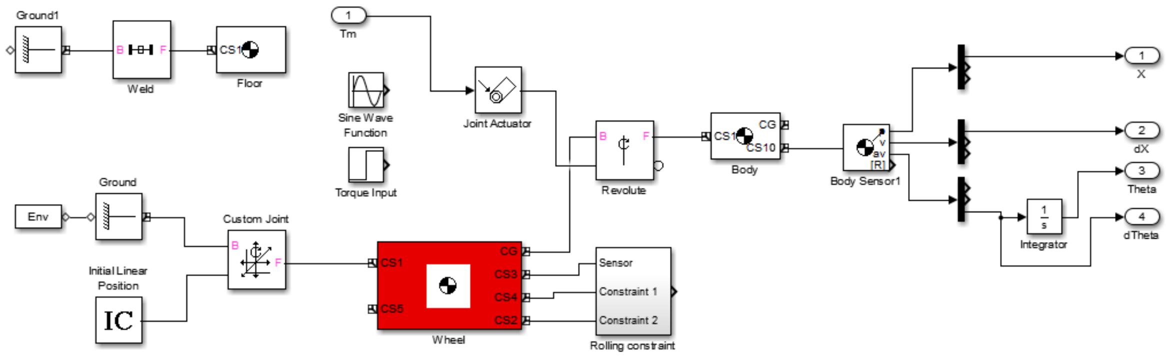

Figure 7: SimMechanics Non-Linear System Plant

Brief testing of this model shows that the system performs realistically similar to a real two-wheeled balancing robot except is doesn't model the collision with ground when the robot collapses. Instead the model allows the body of the robot to rotate through a full circle about the wheel axis as though the ground was not there. With that exception the laws of physics seem to hold for the model which can be visually confirmed by observing the animation (Figure 8). The robot acts like an inverted pendulum and tends to go to the stable upside down position (normal pendulum) if left uncontrolled.

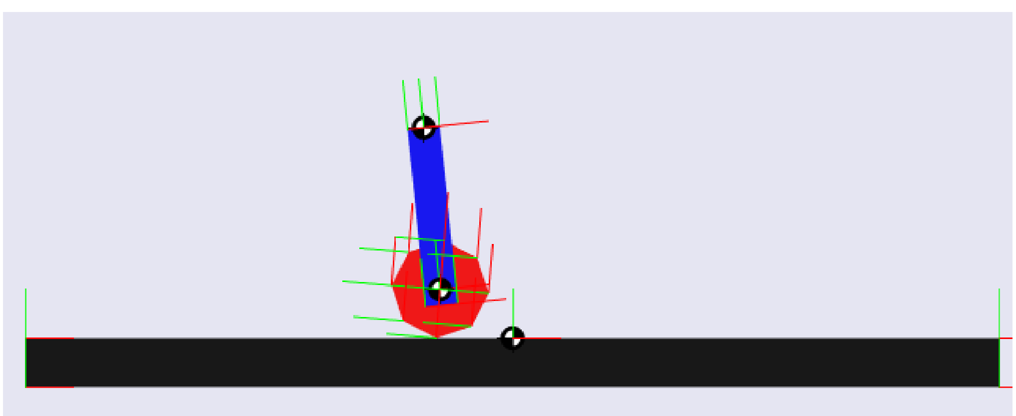

Figure 8: Capture of the SimMechanics Model Animation

5. Results

We start the testing by obtaining the system's open loop response to step input of magnitude 0.1 in MATLAB using the "lsim" function. As expected the system's response is unbounded and grows exponentially (Figure 9).

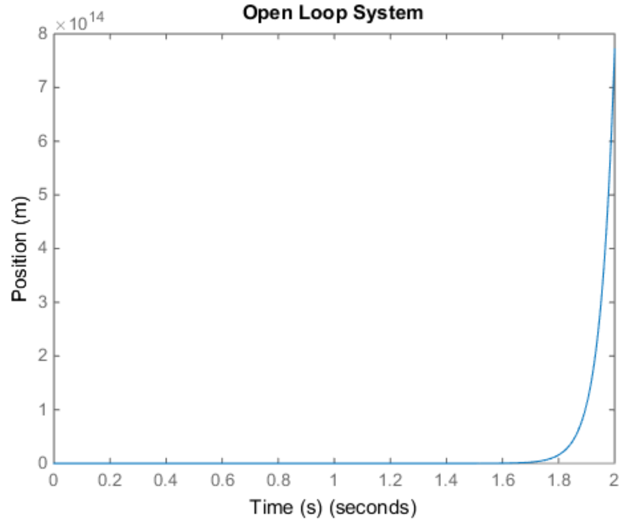

Figure 9: Open loop system step response

Next we apply the state feedback controller loop and again test the system's response to step input of magnitude 0.1. We observe that our goals of 1s rising time and 5% overshoot are achieved, but the response exhibits a very large steady state error of about 55% (Figure 10).

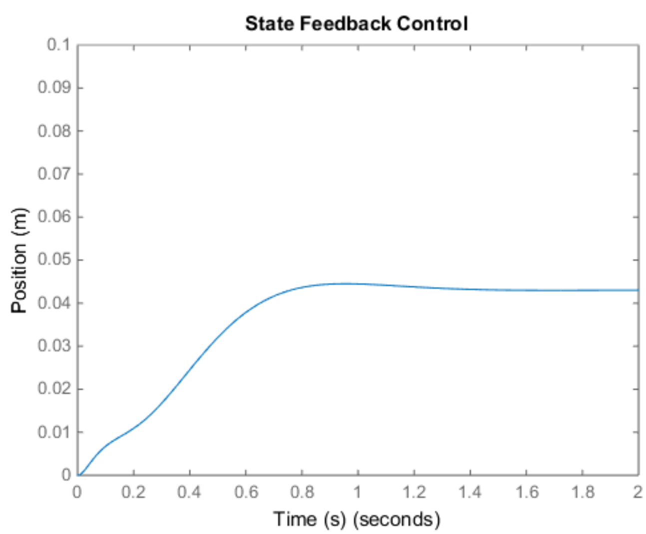

Figure 10: State Feedback Controller Step Response

Next we add an observer to the system and test the observer based controller performance. We see that the controller exhibits an identical performace when using the observed state and the true state. This indicates that the observer is performing well (Figure 11)

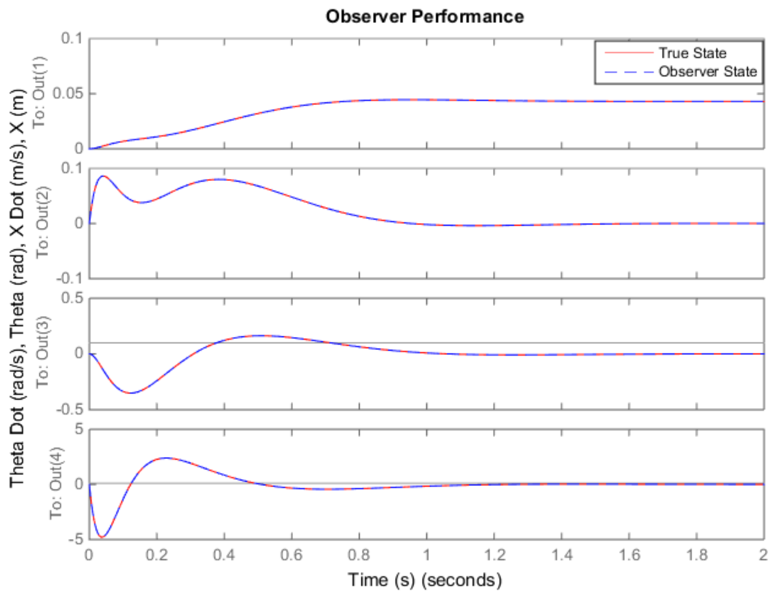

Figure 11: Observer Performance

Next we add an Integrator to the state feedback controller using ITAE method to calculate the controller gains. We observe that the steady state error of the response is now negligible. However the settling time of the system slightly increased after applying the integrator (Figure 12)

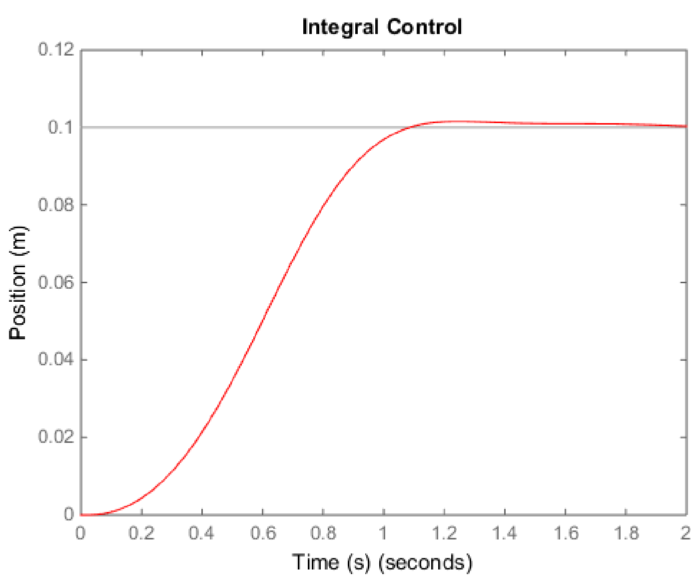

Figure 12: Step Response with Integrator

Next we move on to Simulink testing that would allow is to verify the results obtained in MATLAB and to compare the controller performance with linearized and the non-linear model. We test the step response of the system with observer based state feedback controller and integrator, using step input of magnitude 0.1 (Figure 13). We obtain the same response of the first state variable (position) that we observed in the previous test. This verifies that the Simulink model is valid and can be used for further testing.

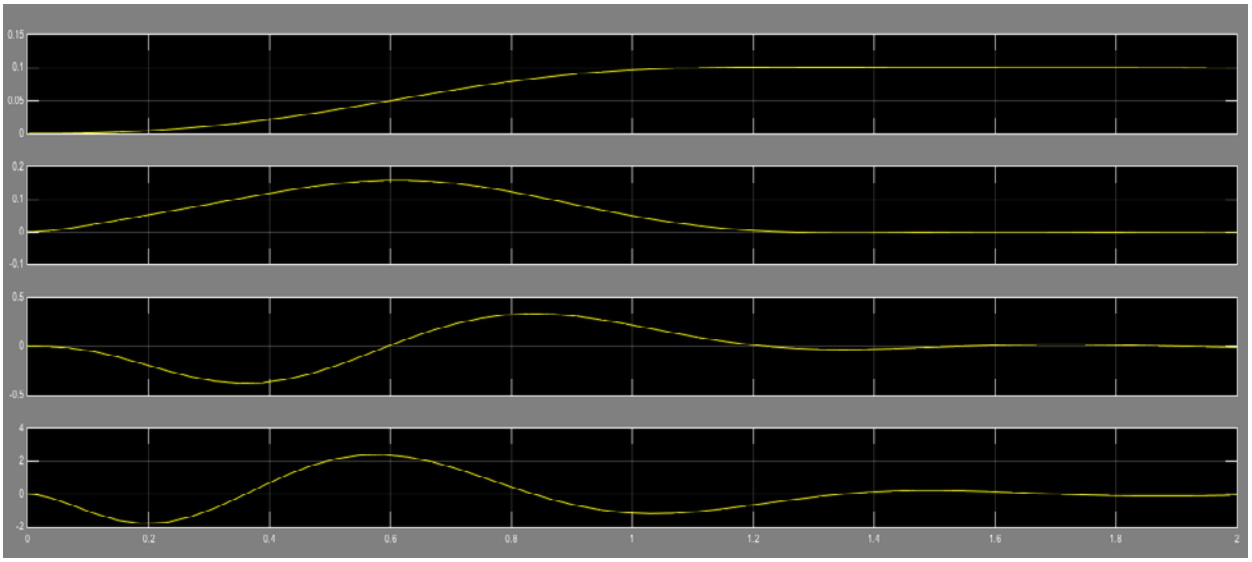

Figure 13: Step Response using Simulink (4 State Variables Respectively)

Next step is to verify that the non-linear system created in SimMechanics is valid and can be used for non-linear testing of the developed controller. To validate the model we run a test with a simple PID controller using the 3rd state variable (angle) as input to the controller and reference input value of 0.1 rad. To perform the test we use a Simulink model shown in Figure 14.

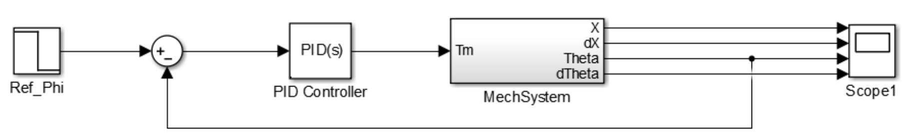

Figure 14: PID Controller used to verify the SimMechanics model

This test shows that the model created in SimMechanics is valid. The angle of the robot settles at desired value of 0.1 rad in about 0.1 seconds and remains at that value (Figure 15). The animation shows that the robot moves steadily at constant velocity maintaining a constant angle. The next step therefore is to test the state-feedback controller that we developed with this model to attempt position control.

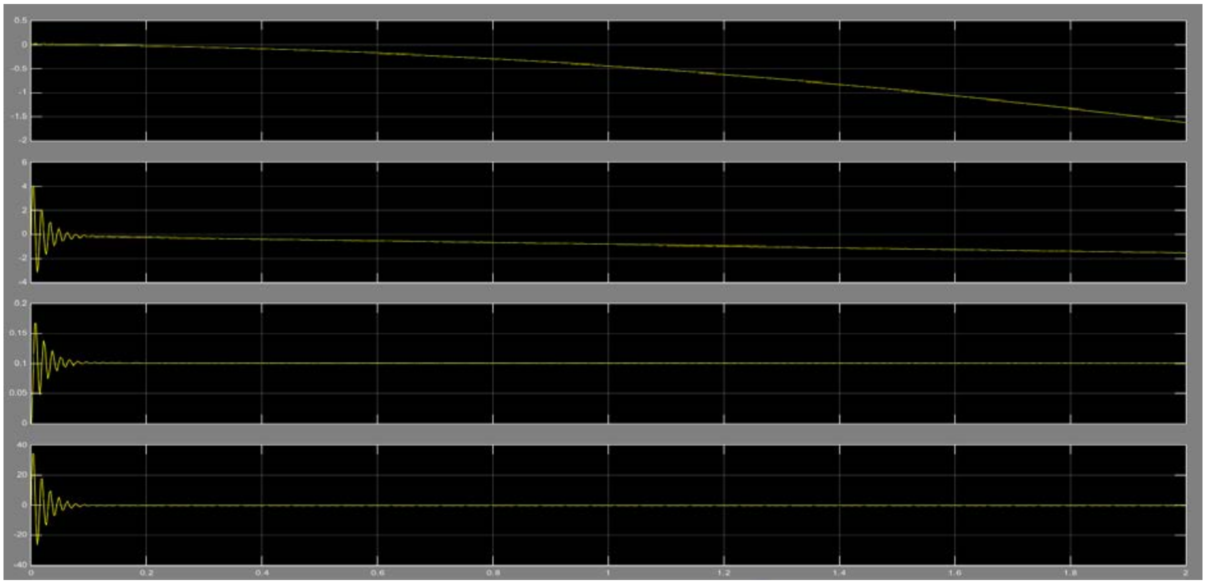

Figure 15: Response of the SimMechanics model to PID control (4 State Variables Respectively)

Having verified the nonlinear model we created in SimMechanics we now attempt to implement position control using the state feedback controller similarly to what we did previously with the linearized model. We see that the attempt to control the nonlinear model fails. The angle of the robot (3rd state variable) grows to large values (~4 rad) which indicates that the robot failed to balance in vertical position and collapsed (Figure 16).

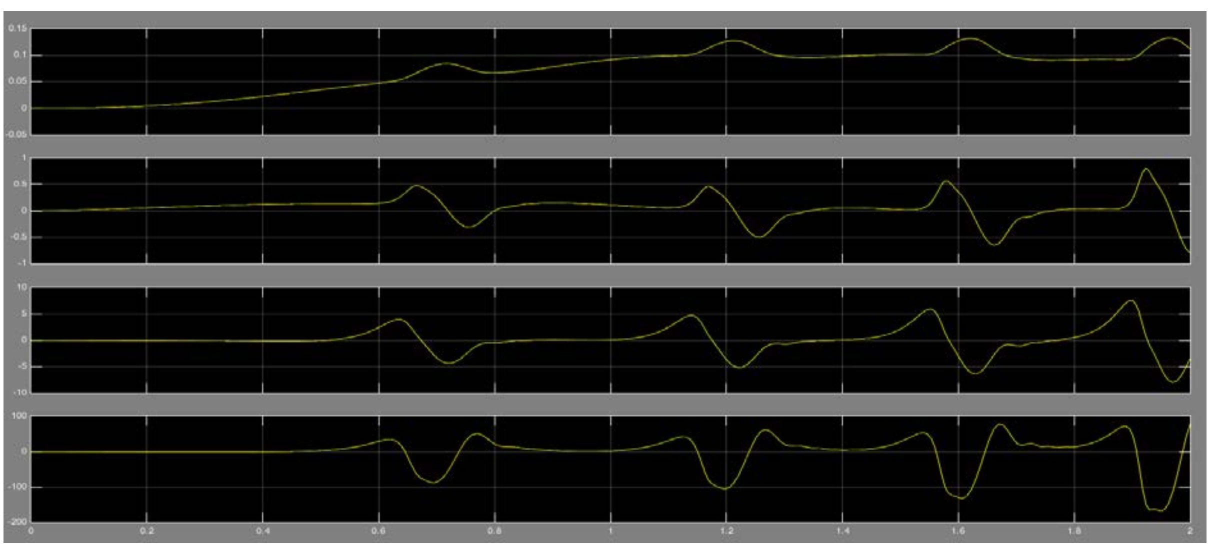

Figure 16: State Feedback Control of SimMechanics Model (4 State Variables Respectively)

6. Conclusion

We developed observer-based state-feedback controller algorithm for linearized model of two-wheeled balancing robot. We chose to control the robot's position. The goal of the controller design was to make the robot travel a distance of 0.1m with settling time of 1s and P.O. of 5%.

Testing of the controller with the linearized model in both MATLAB and Simulink produced good results as expected. The design goals defined for this project were achieved, the steady state error was negligible and the observer functioned properly.

However we quickly learned the limitations of the controller when SimMechanics non-linear model was used to test the controller. We found that SimMechanics displays more realistic results compared to using Simulink blocks or MATLAB plots. This is because Simulink and MATLAB generate linear responses, whereas SimMechanics simulation is nonlinear.

We concluded that our controller was unable to control the non-linear model because it was designed to control the position of the robot and not the angle directly. We were unable to control the angle instead of position with observer-based state-feedback controller because changing the system output from position to angle made the system unobservable.

During validation of the SimMechanics model we were able to control the angle of the robot with PID controller. We therefore can recommend PID control as one way of overcoming the issues we encountered with control of non-linear system.

This project has helped us learn various aspects of controller design, such as model derivation, analyzing controllability and observability, using state feedback control laws and ITAE, implementing observers etc. This project provided us an excellent opportunity to use our skills and knowledge learnt in the class as well as the labs. This knowledge and experience acquired through the project will be immensely beneficial to any future endeavours involving controller design.

References

[1] M. Xie, X. Youlon, C. Xiong and Z. Hu, Intelligent Robotics and Applications, Springer-Verlag Berlin Heidelberg, 2009.

[2] R. L. Williams II and D. A. Lawrence, Linear State-Space Control Systems, New Jersey: John Wiley & Sons, Inc., 2007.