ECE 5115
Control Systems Lab II
Project Report

State-Space Modeling and Analysis of Bicycle Dynamics

Written By:
Vedant Chopra
November 24, 2020

Under the Instruction of:
Dr. Rose Faghih

Abstract

Bicycles are amongst the most common modes of transportation in our world, and its use ranges from commuting to locations to use in competitive sport. While a bicycle might seem to be a simple vehicle, it is difficult to model as there are many complex variables to consider such as velocity, steering angle, tilt angle, torque, and inertia. In this report, we will model this machine as a state-space system and will examine the model's linearity, time-invariance, stability, controllability, and observability. Once deemed usable, the system's open-loop system will be evaluated using a linear quadratic regulator (LQR). Once these examinations are complete, we can design an optimized controller so that we can control a bicycle with utmost efficiency and performance. This optimized controller will be compared to other variations of the controller and open loop model.

Table of Contents

Problem Description..4
The Induction Motor..4
State-Space Analysis..4

Approach.. 5
The State-Space Model.. 5
Model Analysis.. 7
Controller Design.. 8

Simulations 9
Initial Simulink Model..9
Control Added..10
Reference Tracking Added 10

Results..11
Direct-Control System 11
Summary.. 14

References..15

Table of Figures

Figure 1. Bicycle's Dynamic Model[1]..4
Figure 2. Open Loop Model in Simulink...9
Figure 3. Closed Loop Model Modified for Controller in Simulink..10
Figure 4. Closed Loop Model With Reference Tracking in Simulink..11
Figure 5.Open Loop Model Output...12
Figure 6. Closed Loop Model Output………………………………………………..............12
Figure 7. Reference Tracking Model Output...13
Figure 8.Optimized Model Outputh observer-based control and error tracking……….........13

Problem Description

The Bicycle

Bicycles are amongst the most common modes of transportation in our world, and its uses ranges from everything to commuting to locations to use in competitive sport. While a bicycle might seem to be a simple vehicle, it is difficult to model as there are many complex variables to consider such as velocity, steering angle, tilt angle, torque, and inertia.

Figure 1 showcases a simple model of a bicycle. There are other values that are necessary to model a bicycle completely, such as forces experienced at the rear and front of the bicycle, but those are not included in the following figure as the fundamental state-space representation does not have these values included.

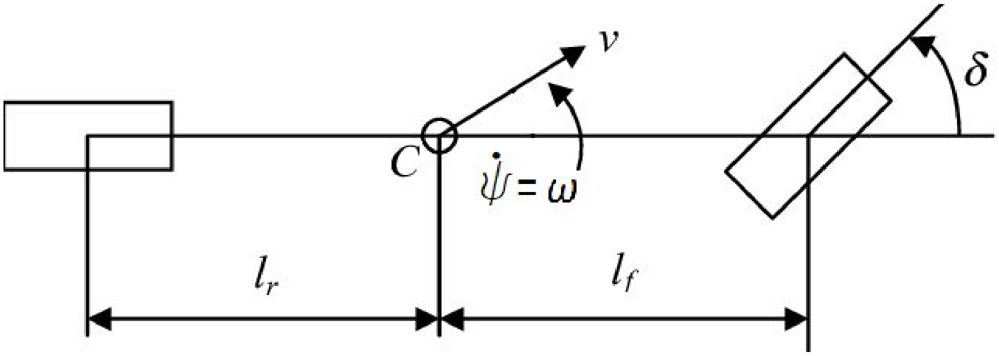

Figure 1. Bicycle's Dynamic Model [1]

There are other complex models to represent a bicycle, but I chose this model for simplicity as it will be easier for others to understand. Assumptions that were made were that the front and rear wheels are the same, reference point C is where the center of gravity is, and the bicycle's motion is restricted to a two-dimensional plane. Further assumptions include that the bike is considered to be a rigid body, only lateral tire forces are created by the wheels, the velocity of the bike is constant, and lastly the steering angle is small (which is reasonable as a rider would never have large steering angle as it would make the bike crash immediately).

State-Space Analysis

As we are trying to design a controller for the bicycle, one of the most appealing options we have is to model our system using state-space analysis. The advantages of state-space is that it allows us to model our bicycle system in the time domain and it allows us the option to model multiple-input-multiple-output (MIMO) systems. The general state-space model this report will use are Equation 1, Equation 2, and Equation 3:

$$\dot{x}(t) = \mathsf{A}x(t) + \mathsf{B}u(t), \tag{1}$$

$$y(t) = \mathsf{C}x(t) + \mathsf{D}u(t), \tag{2}$$

$$x(t_0) = x_0 \tag{3}$$

The purpose of this report is to use state-space methods to design a controller to provide us a stable output that can immediately track the input. We will then optimize this controller.

Approach

The State-Space Model

We need to calculate the lateral dynamics of the bicycle to find our state-space representation. We first begin by calculating the perpendicular forces experienced by the front and rear wheels. The front wheel's force takes into account the front wheel's steering angle. Since the front wheel steering angle will be small (a rider would not turn the front wheel with a large angle as it would make them instantly unstable), we can estimate the lateral force to be simply the sum of the lateral force experienced by both rear and front wheels. The mathematical representation can be seen in Equation 3:

$$\mathrm{ma}_y = \sum_i F_{\mathrm{yi}} = F_f + F_r\cos(\delta) \approx F_f + F_r \,. \tag{3}$$

We also need to represent the lateral acceleration as centripetal acceleration as the wheels are turning on a fixed point, and this representation is seen in Equation 4,

$$a_y = \dot{v}_{\mathrm{lat}} + \omega \mathrm{v}_{\mathrm{lon}}, \tag{4}$$

and then we substitute Equation 4 into Equation 3. This gives us the critical Equation 5, which is

$$m\left(\dot{v}_{\mathrm{lat}} + \omega \mathrm{v}_{\mathrm{lon}}\right) = F_f + F_r \,. \tag{5}$$

Since the bicycle turns by the angular motion about the vertical axis, we need to consider the yaw dynamics. We can represent this in the critical Equation 6 as

$$\dot{\omega} i_z = \sum_i M_i = F_f l_f - F_r l_r \cos(\delta) \approx F_f l_f + F_r l_r \,. \tag{6}$$

Again, since the steering angle of the front wheel will be small, we can estimate it to be negligible, hence the approximation in Equation 6. We now need to represent the front and rear wheel's with the linear tire model [2].This gives us Equation 7 and Equation 8, which are

$$F_f \approx \delta c_{\alpha \mathrm{f}} - \frac{c_{\alpha \mathrm{f}}\left(\omega \mathrm{l}_f + v_{\mathrm{lat}}\right)}{v_{\mathrm{lon}}}, \tag{7}$$

$$F_r = -\,c_{\alpha \mathrm{r}} \frac{v_{\mathrm{lat}} - \omega \mathrm{l}_r}{v_{\mathrm{lon}}} \,. \tag{8}$$

In Equation 7 and Equation 8, $c_{\alpha\text{f}}$ and $c_{\alpha\text{r}}$ represents the cornering stiffness of the front and rear wheels. The cornering stiffness is a constant, and it is the negative of the rate of change of lateral force with respect to change in the slip angle [3]. The slip angle is the ratio of forward and lateral velocities in the form of an angle [4]

We can now substitute Equation 7 and Equation 8 into Equation 5 and Equation 6 respectively, which brings us to the form needed to represent the bicycle in state-space form. Then we modify the equations so that we get the differential of the lateral velocity and the the differential of the angular velocity. We get Equation 9 and Equation 10, which are

$$\dot{v}_{\text{lat}} = \frac{\omega\left(c_{\alpha\text{r}} l_r - c_{\alpha\text{f}} l_f\right)}{\text{m}\text{v}_{\text{lon}}} - \frac{v_{\text{lat}}\left(c_{\alpha\text{f}} + c_{\alpha\text{r}}\right)}{\text{m}\text{v}_{\text{lon}}} + \frac{\delta c_{\alpha\text{f}}}{m} - \omega v_{\text{lon}} . \tag{9}$$

$$\dot{\omega} = \frac{c_{\alpha\text{r}} l_r - c_{\alpha\text{f}} l_f}{I_z v_{\text{lon}}} v_{\text{lat}} \text{ -} \frac{c_{\alpha\text{r}} l_r^2 + c_{\alpha\text{f}} l_f^2}{I_z v_{\text{lon}}} \omega + \frac{c_{\alpha\text{f}}}{I_z} l_f \delta . \tag{10}$$

Using Equation 9 and Equation 10, we can get our state space representation, as seen in Equation 11 and Equation 12. Note the substitution that occurs for the $\dot{x}$ vector:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} \dot{v}_{\text{lat}} \\ \dot{\psi} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} -\frac{c_{\alpha\text{f}} + c_{\alpha\text{r}}}{\text{m}v_{\text{lon}}} & 0 & \frac{c_{\alpha\text{r}} l_r - c_{\alpha\text{f}} l_f}{\text{m}v_{\text{lon}}} - v_{\text{lon}} \\ 0 & 0 & 1 \\ \frac{c_{\alpha\text{r}} l_r - c_{\alpha\text{f}} l_f}{i_z v_{\text{lon}}} & 0 & \frac{c_{\alpha\text{f}} l_f^2 + c_{\alpha\text{r}} l_r^2}{i_z v_{\text{lon}}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} \frac{c_{\alpha\text{f}}}{m} \\ 0 \\ \frac{c_{\alpha\text{f}} l_f}{i_z} \end{pmatrix} \mu \tag{11}$$

$$\text{y} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} . \tag{12}$$

Realistic values for the constant coefficients, as seen in Table 1, were researched and later assigned to each of the bicycle's values [5][6][7][8][9]. These estimated values represent a typical bicycle that an average consumer could buy. It is also assumed the same tires will be used for the front and the rear, and as a result the same cornering stiffness was used. The input will be the steering angle, and the out put is the angular acceleration.

Table 1. Values for Bicycle Components

Coefficient	Value	Units
$c_{\alpha f} = c_{\alpha r}$	150	N/deg
m	8.16	kg
v_{lon}	4.4	m/s^2
l_r	0.35	m
l_f	0.625	m
I_z	30*(10^5)	kg-m^2

Model Analysis

We need to check for certain requirements before designing a controller. These properties include linearity, time-invariance, stability, controllability, and observability of our model. In order for us to establish that our model is linear, we need to ensure our system consists of only real numbers. Since Equation 11 and Equation 12 were able to be evaluated, we know the system is linear.

If the entire bicycle's motion was modeled, such as the bike accelerating from no movement to movement, and eventually decelerating to a complete stop, we could not use state-space to design a controller. The reason why is due to the bicycle's speed being dependent on time when accelerating and decelerating. However, what makes us able to still use state-space is the clever usage of a constant velocity. Using the average speed of an average rider, we can treat velocity as a constant. This means the constant velocity used is not dependent on time, allowing us to make the model to be time-invariant. As a result, our model is a linear, time-invariant (LTI) system.

To check for stability, we need to evaluate the eigenvalues for the A matrix. Using MATLAB's eig( ) function, we were able to evaluate Equation 13,

$$\det(\lambda \mathrm{I} - A) = 0, \tag{13}$$

and we get the eigenvalues to be at locations λ= 0, -8.3556, 0. Since the eigenvalues are not positive in magnitude, this system is never unstable. However, the eigenvalues are not all less than zero, and as a result this model is marginally stable.

Since we determined the stability, we can determine the controllability and observability of the model. Using MATLAB's ctrb() and obsv() functions, we can evaluate Equation 14 and Equation 15 respectively:

$$M_c = \left(\begin{array}{ccc} B & A & \ldots A^{n-1}B \end{array} \right), \tag{14}$$

$$\left(\begin{array}{c} C \\ \mathrm{CA} \\ \mathrm{CA}^2 \\ \ldots \\ \mathrm{CA}^{n-1} \end{array} \right). \tag{15}$$

We can determine the controllability and observability of the states by subtracting the rank of each matrix from the length of the A matrix. Using MATLAB, this was achieved by using the rank( ) and length( ) functions. All of the states were controllable, but some of the states were not observable.

We need to evaluate if the system can be made into an observable state, so we need to evaluate the Diagonal Modal Form. We can use MATLAB function [csys, T] = canon( ), and evaluate it for the system in the 'modal' form. We can use csys and the product of the C matrix and the transformation matrix T. It was determined that two of the modes were unobservable, and as a result the system is not detectable.

We can use minimal realization to see if an altered version of the system can be controllable and observable. Using the MATLAB function minreal( ), we can get a new system where the x_3 state column and row for matrices A, B, C, and D are removed. Since this new system is minimally realized, we now know that the system now has the desired characteristics we need for designing the controllers.

With our new system, we need to reevaluate our eigenvalues to determine stability, realize the new system's controllability and observability, and lastly reevaluate if the new system has any state that is unobservable or uncontrollable. Now, using minimal realization, the system is stable, controllable, and observable.

Controller Design

Now that we established that our model has the desired characteristics needed to design a controller, a stable sinusoid and a step function were added to the model to create a baseline output for us to achieve. When riding, a biker makes slight adjustments in their steering angles when moving, and this baseline output helps simulate these adjustments. Further, this baseline output was created so that we can evaluate the effectiveness of the controller.

Now we need to design an optimized controller. Since there is no reference given to help us decide the placement of the poles, we can use a linear quadratic regulator (LQR) to design the controls. One could have considered using the Ackerman's formula, but this is not possible as we have no reference to where the pole should be. However, the LQR approach has some advantages that one should consider. LQR allows for immediate in its calculations for the optimal gain matrix and its approach's structure is the same as the Ackerman's approach.

Using the MATLAB lqr( ) function, we can create a model of our bicycle. The lqr( ) model needs the inputs including the open-loop system, Q matrix, and R matrix. The Q matrix adds a magnitude to the states, and the R matrix pertains to the control effort. To model use the Q matrix effectively, use the original Q matrix , as seen in Equation 16,

$$Q_{\text{orignal}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{16}$$

and change the values of 1 in the equation. This step will be further explored in the Simulations section. The 2x2 version of the Equation 16 will only be considered as the minimally realized A matrix is also a 2x2 matrix. The size of the A matrix and Q matrix need to be identical.

When considering the control effort R, which is dictated by the limitation of a machine, we need to realize that no restrictions were given to us. As a result, for simplicity, a value of one was assigned to the R.

After evaluating the lqr( ) funcion, we are given the gain matrix K, the solution of the algebraic Riccati equation (S), and the closed loop poles (P) for the input system. Now we can evaluate the closed loop representation of matrix A, which is defined as A_{cl}, and this transformation is represented by Equation 17,

$$A_{\text{cl}} = A - BK. \tag{17}$$

Using the new closed loop function, we can insert it into MATLAB’s Simulink function, which will show the outputs of the different models and controllers. We could also add the reference tracking tinto the control scheme. We can calculate the gain matrix N using Equation 18,

$$N= -(C (A - BK)^{-1} B)^{-1}, \tag{18}$$

and insert it into Simulink.

Simulations

Initial Simulink Model

We first start modeling our bicycle system with the initial open loop state-space model that was defined by Equation 11 and Equation 12, with the values in Table 1 included. The system’s structure in Simulink can be seen in Figure 2.

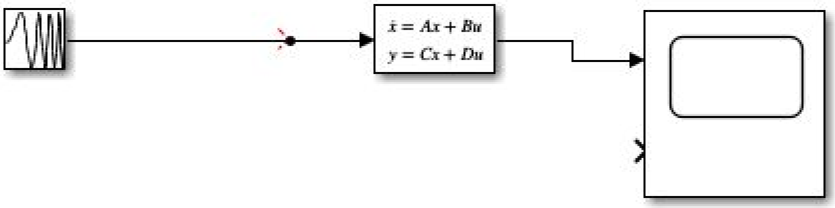

Figure 2. Open Loop Model in Simulink

Simulink is useful for analysis since it has a state-space model built in. An input ranging from 0 to 23 [Hz] was inserted to model the steering angle (so it models the steering angle and not frequency). The reason why the input range is limited to 0 to 23[Hz] is attributed to the fact we used the small angle assumption in our math. If we increase the inputs range, it would make the mathematical work wrong as it would violate the small angle assumptions used. It is worth mentioning again that a small steering is fine as a small steering angle still models most biking situations well. However, a variety of inputs, including a steady-state sinusoid, and the case when the frequency decreases from 23 to o [Hz].

Control Added

Now that we want to add a controller to our Simulink model, we have to break apart the convenient state-space block into its components. Keep in mind we do not need to model the D vector as it is simply zero in magnitude. The new simulink model now includes the feedback gain K matrix, which turns the structure of the model to a closed loop model. Figure 3 showcases the new simulink model.

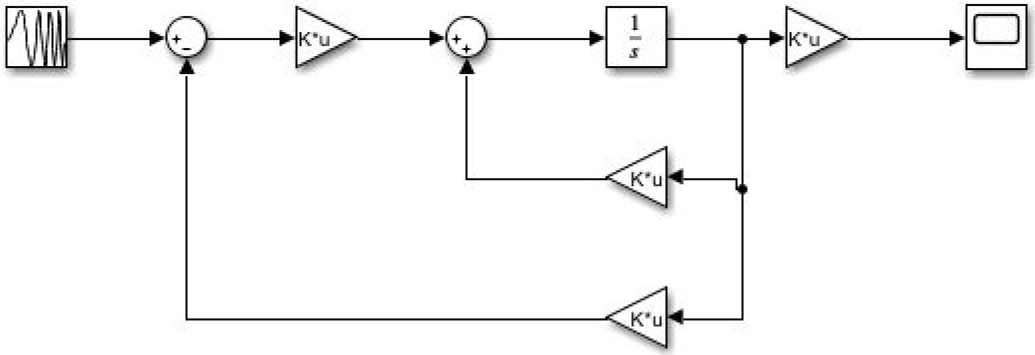

Figure 3. Closed Loop Model Modified for Controller in Simulink

Recall the Approach section of this report. Using the optimal control method, we calculated the K matrix. The input is the same as that from the Initial Simulink Model, including the variety of inputs used.

Reference Tracking Added

In order to add reference tracking, we use MATLAB to calculate the gain Matrix N. We add the gain Matrix N into our Simulink model. Other than this simple addition, the Simulink model is fundamentally identical to Figure 3. The new model with the added Matrix N can be seen in Figure 4.

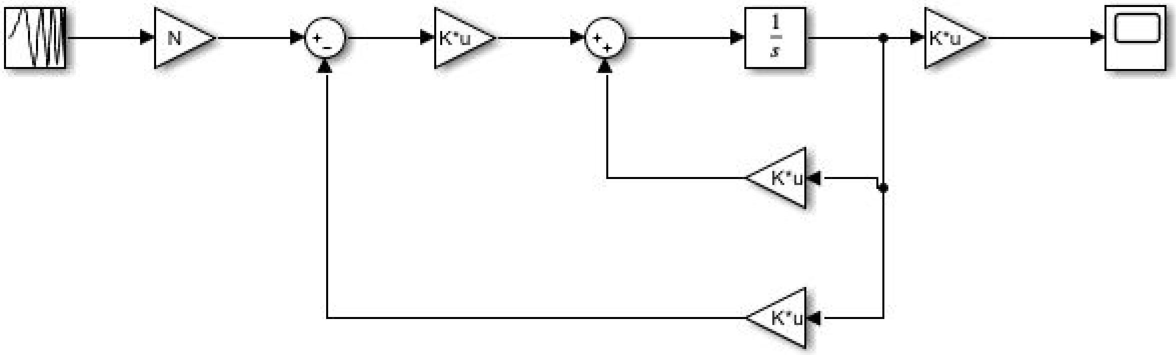

Figure 4. Closed Loop Model With Reference Tracking in Simulink

The new simulink model has the same conditions as the previous models, which includes the variety of inputs and a scope to measure the output

If we were to simply use our initial Matrix N, which was derived from the original Q matrix, which can be seen in Equation 16, our controller would still not be optimized. While it is much better than the initial models, which will be explored in the forthcoming Results section, it can still be improved by altering the Q matrix. In our case, we only needed to alter the top-left one from Equation 16.

Results

Direct Control System

While other inputs were tested and verified, for the case of simplicity and impactfulness of this report, the only responses that will be discussed is when we have the input ranging from 0 to 23 [Hz] was inserted to model the steering angle. The reason for this decision is that if we were to use only one value, which is represented as a step function, it would only give a small idea of how the system is affected by changing the steering angle. By having an input that changes from 0 to 23 [Hz] (again Hz is what Simulink uses, it actually represents the steering angle), we can have a better idea of how the steering angle actually affects the angular acceleration.

Figure 5 showcases the output of the open loop model in Simulink. What is supring is the relatively small value of the output angular acceleration. This can possibly be attributed to the amounts of assumptions we made to make the math simpler.This model still showcases the relatively large amount of variation experienced by the output experiences without a controller.

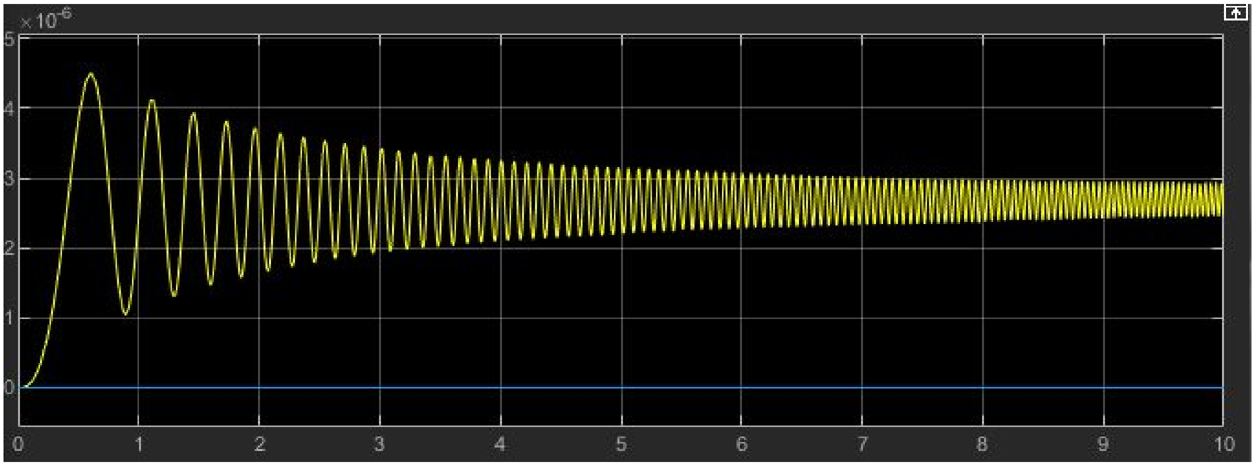

Figure 5. Open Loop Model Output

We can now look at the case when the closed loop model was implemented in Figure 6. We can already see a proportionally large difference between the open loop and closed loop. While the open loop model's output was actually closer to approaching stability in terms of magnitude, the closed loop model approaches stability much faster. The closed loop model has less variability in the magnitude of the output when oscillating. To reiterate, the closed loop model has a much larger output, which we do not want, but it has less magnitude change when it does reach stability. Adding a controller will help bring this desired attribute.

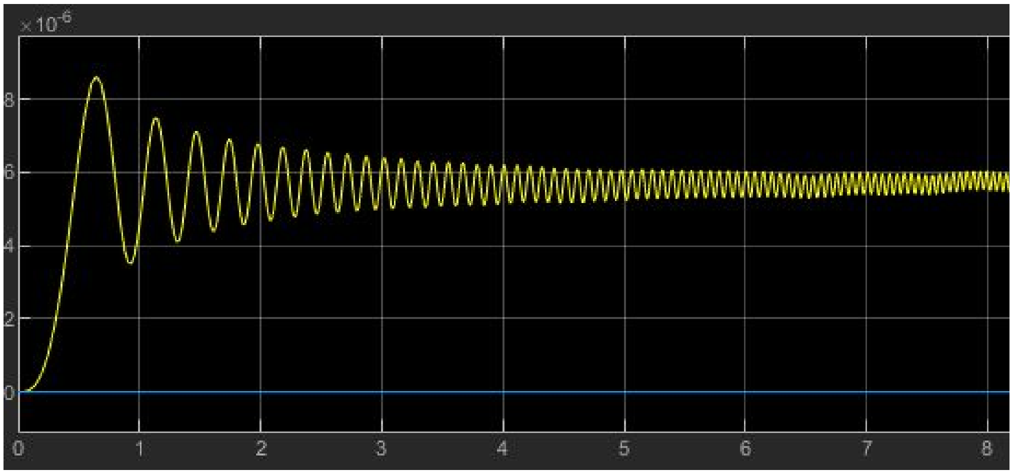

Figure 6. Closed Loop Model Output

Now we can add a controller to our model. Figure 7 showcases the output angular acceleration when the reference gain was added. While this looks like the output seen in figure 5, after closer inspection, we notice that this new output has less magnitude of error (by 0.5 respectively) and the oscillation of the error reduced by half. This is the first reasonable response as it could be used in some applications as a usable controller.

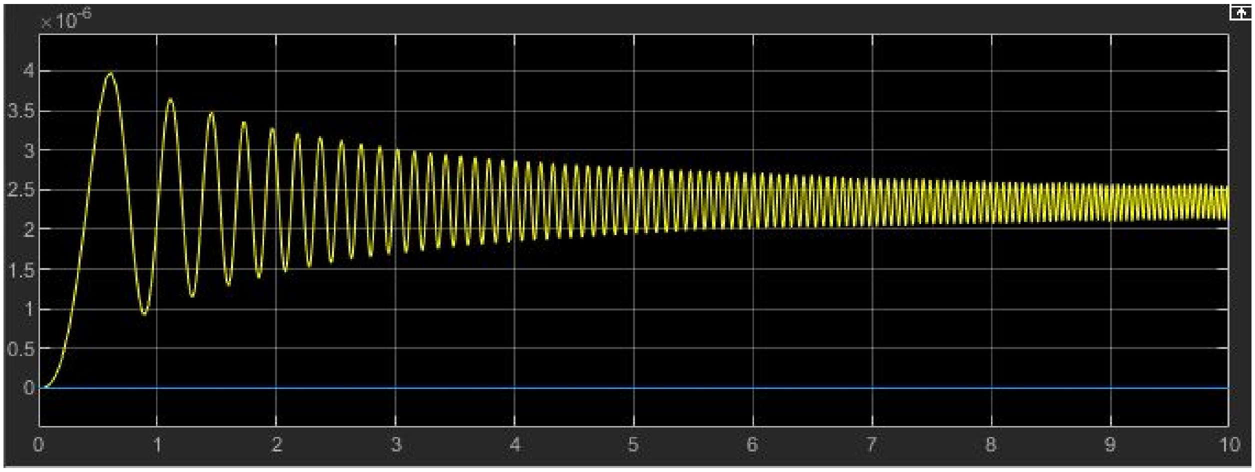

Figure 7. Reference Tracking Model Output

However, as much as an improvement the previous case was, we can still optimize it into a much more effective controller. The previous case uses the ordinary Q matrix, which can be seen in Equation 16. To optimize our controller, we can use the same LQR analysis, allowing us to do pole placement, by altering the Q matrix. We simply need to only alter the top-left one from Equation 16. We can keep R to remain the same value of one, and change Q to [21 0; 0 1]. This new output can be seen in Figure 8. If we were to increase the 21 value by anymore, we would see an overshoot occur, which we do not want.

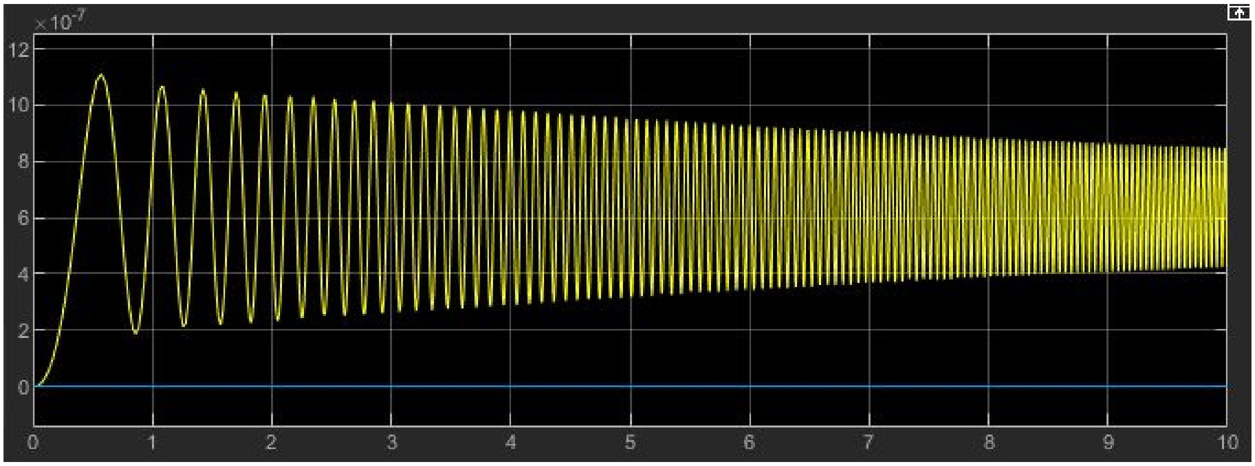

Figure 8. Optimized Model Output

At first glance, it seems that altering the Q matrix actually made our output worse. However, after closer inspection, we notice that the magnitude of the output reduces almost by a factor of five. This shows extremely well that our bike is becoming more stable as time increases since the angular acceleration, or in other words the change in angular velocity, approached stability almost immediately. This model could become even more accurate if we were given a R value to use as the one we chose was arbitrary.

Summary

While it may be amongst the most common modes of transportation and simple at first glance, bicycles are very complex machines. Our bicycle model is one of the simplest models, which includes a lot of idealized assumptions constrained in a two-dimensional plane. However, this very model is still analized by the automotive gives valuable information that, when expanded upon, can even model complex car dynamics.

Using state-space analysis for control analysis, we were able to design an optimized controller. First we had to solve preliminary physics problems to model the bicycle in state-space form. After tham, before we could attempt design, we had to check for linearity, time-invariance, stability, observability, and controllability.

Using minimal realization, we were able to get a model that we could use to design a controller. Using Simulink and MATLAB, we were able to see how the steering angle affects the change in angular acceleration. The faster the angular acceleration stabilizes and remains low, the more stable the system is. We then realized an optimized controller.

While this is an extremely good model for our case, it is only valid in an extremely idealized version of the bicycle model. The model can quickly become complex once we add a third dimension to the model, use different sizes of wheels, and even use a nonlinear state-space model.

References

[1] B. Zheng, “Active steering control with front wheel steering,” Jan-2004. [Online]. Available: https://www.researchgate.net/figure/Bicycle-model-for-steering-dynamics-The-corresponding-linearized-dynamic-equation-is_fig1_4119228. [Accessed: 22-Nov-2020].

[2] W. Bombardier, *A Brief Introduction to Vehicle Dynamics*, 16-May-2006. [Online]. Available: http://www.minibaja.uwindsor.ca/Downloads/Vehicle%20Dynamics_white2.pdf. [Accessed: 23-Nov-2020].

[3] A. Yates, “Cornering Stiffness,” *Academic Dictionaries and Encyclopedias*, 2013. [Online]. Available: https://mech.enacademic.com/220/cornering_stiffness. [Accessed: 23-Nov-2020].

[4] C. Smith, *Slip Angle*, 2015. [Online]. Available: http://formula1-dictionary.net/slip_angle.html. [Accessed: 23-Nov-2020].

[5] P. Windes, M. Archibald, and B. Joseph, “Experimental Determination of Bike Tire Stiffnesses,” *http://people.cst.cmich.edu/*, 2013. [Online]. Available: http://people.cst.cmich.edu/yelam1k/asee/proceedings/2013/papers/97.pdf. [Accessed: 23-Nov-2020].

[6] M. White, “How Much Does A Bike Weigh?,” *Bicycle Universe*, 15-Mar-2020. [Online]. Available: https://bicycleuniverse.com/how-much-does-bike-weigh/. [Accessed: 23-Nov-2020].

[7] C. Woods, “How Fast Does the Average Person Ride a Bike?,” *Reference*, 04-Apr-2020. [Online]. Available: https://www.reference.com/world-view/fast-average-person-ride-bike-cf053029272716e0 . [Accessed: 23-Nov-2020].

[8] D. D. Coles, “Classic Road Bicycle Dimensions,” *webflow*, 10-Aug-2014. [Online]. Available: https://global-uploads.webflow.com/5b44edefca321a1e2d0c2aa6/5b74db0363ff825040103c0b_Dimensions-Guide-Transport-Bicycle-dimensions.svg. [Accessed: 23-Nov-2020].

[9] L. Zinn, “Calculating a wheel's moment of inertia,” *VeloNews.com*, 28-Jan-2011. [Online]. Available: https://www.velonews.com/gear/calculating-a-wheel%E2%80%99s-moment-of-inertia/. [Accessed: 23-Nov-2020].