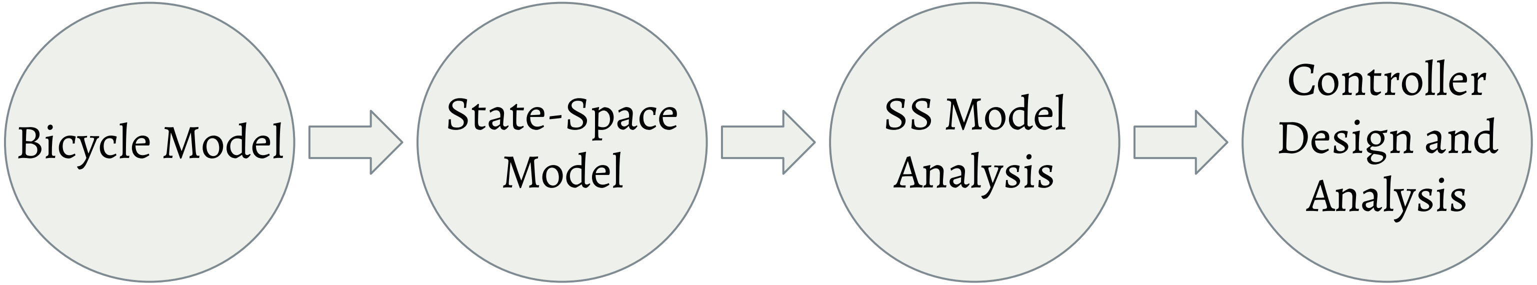
State-Space Modeling and Analysis of Bicycle Dynamics

Vedant Chopra
November 24, 2020

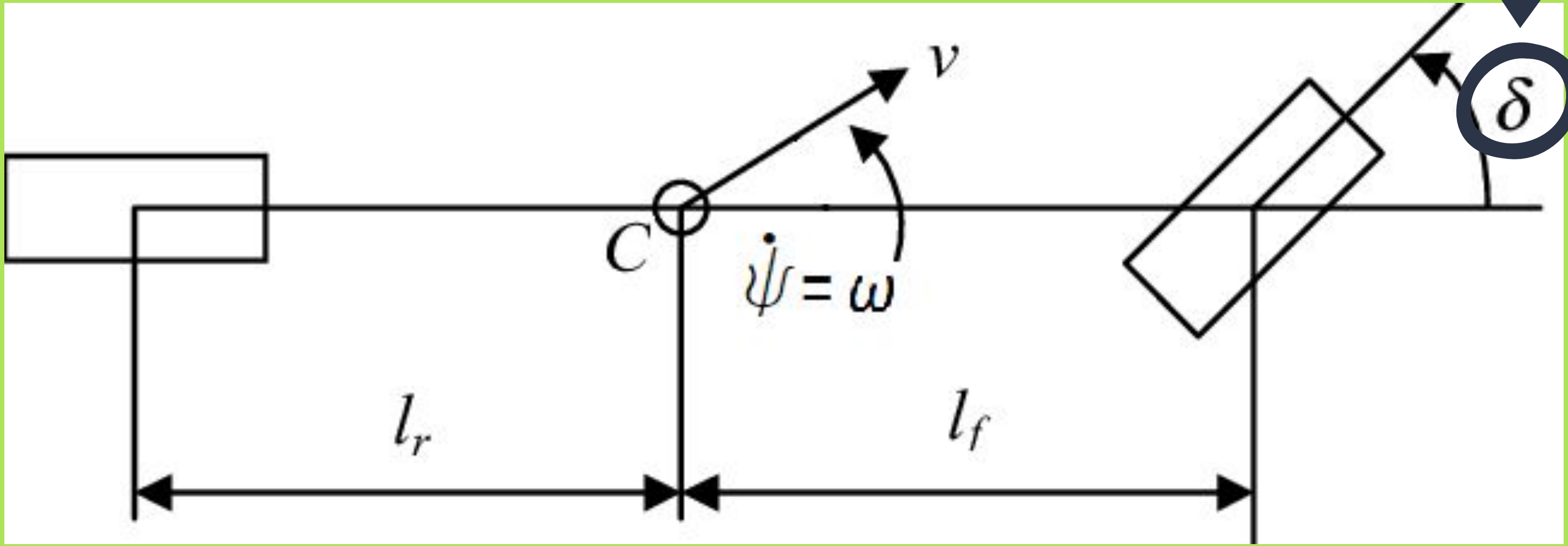
ECE 5115
Controls System Lab II

Overview


Bicycle Model
State-Space Model
SS Model Analysis
Controller Design and Analysis

Bicycle Model

Steering Angle



Input: Steering Angle

Output: Angular Acceleration

Bicycle Model Analysis to SS Model Conversion

$$\mathrm{m}a_y = \sum_i F_{yi} = F_f + F_r\cos(\delta) \approx F_f + F_r$$

$$a_y = \dot{v}_{\text{lat}} + \omega \mathrm{v}_{\text{lon}}$$

$$\Rightarrow m\left(\dot{v}_{\text{lat}} + \omega \mathrm{v}_{\text{lon}}\right) = F_f + F_r$$

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

$$F_f \approx \delta c_{\alpha f} - \frac{c_{\alpha f}\left(\omega l_f + v_{\text{lat}}\right)}{v_{\text{lon}}}$$

$$F_r = -c_{\alpha r}\frac{v_{\text{lat}} - \omega l_r}{v_{\text{lon}}}$$

We can now substitute force into equations!

State-Space Form

$$\dot{v}_{lat} = \frac{\omega\,(c_{\alpha r} l_r - c_{\alpha f} l_f)}{m v_{lon}} - \frac{v_{lat}\,(c_{\alpha f} + c_{\alpha r})}{m v_{lon}} + \frac{\delta c_{\alpha f}}{m} - \omega v_{lon}$$

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

=

A **B**

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

$$y = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

C

D= 0

Add Values to Coefficients

- Researched values to substitute into state-space model
- Will represent an average bike with a average biker

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

SS Model Analysis

By hand analysis would be arduous and time-consuming

Preliminary MATLAB Code

```
%Setup
syms u x1 x2 x3 x1p x2p x3p;

car = 150;
caf = 150;
m = 8.16;
vlon = 4.4;
lr =0.35;
lf = 0.625;
iz = 30*(10^5);

eq1 = -(x1p)-((car +caf)/(m*vlon))*x1 +((car*lr
+caf*lf)/(m*vlon))*x3 -vlon*x3+(caf/m)*u==0;
eq2 = x3==x2p;
eq3 = -x3p+((lr*car
+lf*caf)/(iz*vlon))*x1-(((lf^2)*caf
+(lr^2)*car)/(iz*vlon))*x3+(caf/iz)*lf*u==0;

A = [-((car +caf)/(m*vlon)) 0 ((((car*lr
+caf*lf)/(m*vlon))-vlon)); 0 0 1;((lr*car
-lf*caf)/(iz*vlon)) 0 -(((lf^2)*caf
+(lr^2)*car)/(iz*vlon))];

B = [(caf/m); 0 ;(caf/iz)*lf];
C = [0 0 1];
D = 0;
sys=ss(A,B,C,D);
```

Checking Requirements

Our model is also linear and time-invariant

```
%Check for stability: eigenvalues
e = eig(A);% 0, -8.3556, 0

%Check for observability and controllability
Mo = obsv(A,C);
Mc = ctrb(A,B);
%Check for number of unobservable and
uncontrollable states
uobs = length(A) - rank(Mo); %1, so unobservable
uctr = length(A) - rank(Mc); %0, so controllable
```

```
%Convert to Diagonal Modal Form
[csys,T] = canon(sys,'modal');

%Evaluate Detectability
detect=csys.C*T; %Evaluates to [0 0 1], two modes are
%unobservable
```

Minimal Realization

Removes the x3 variable, which results in a controllable and observable system

Minimal Realization and Repetition

```
%Use Minimal Realization and Revaluation
nsys = minreal(sys); %x3 state was removed from A,B,C,D


ne = eig(nsys);
nMo = obsv(nsys.A,nsys.C);
nMc = ctrb(nsys.A,nsys.B);
nuobs = length(nsys.A) - rank(nMo); %0, so observable
nuctr = length(nsys.A) - rank(nMc); %0, so controllable
```

Controller Design

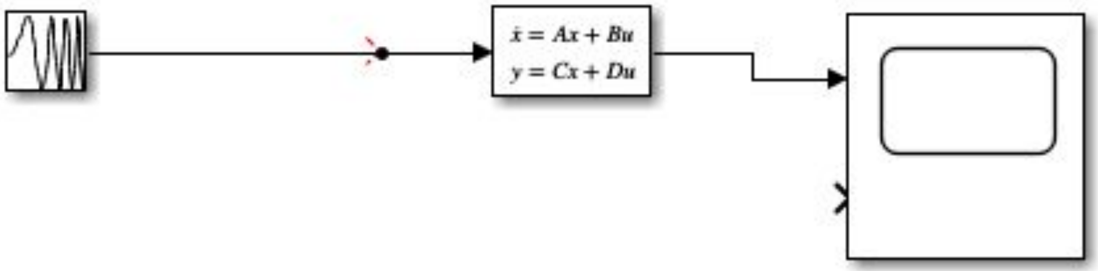
For Feedback Gain K:

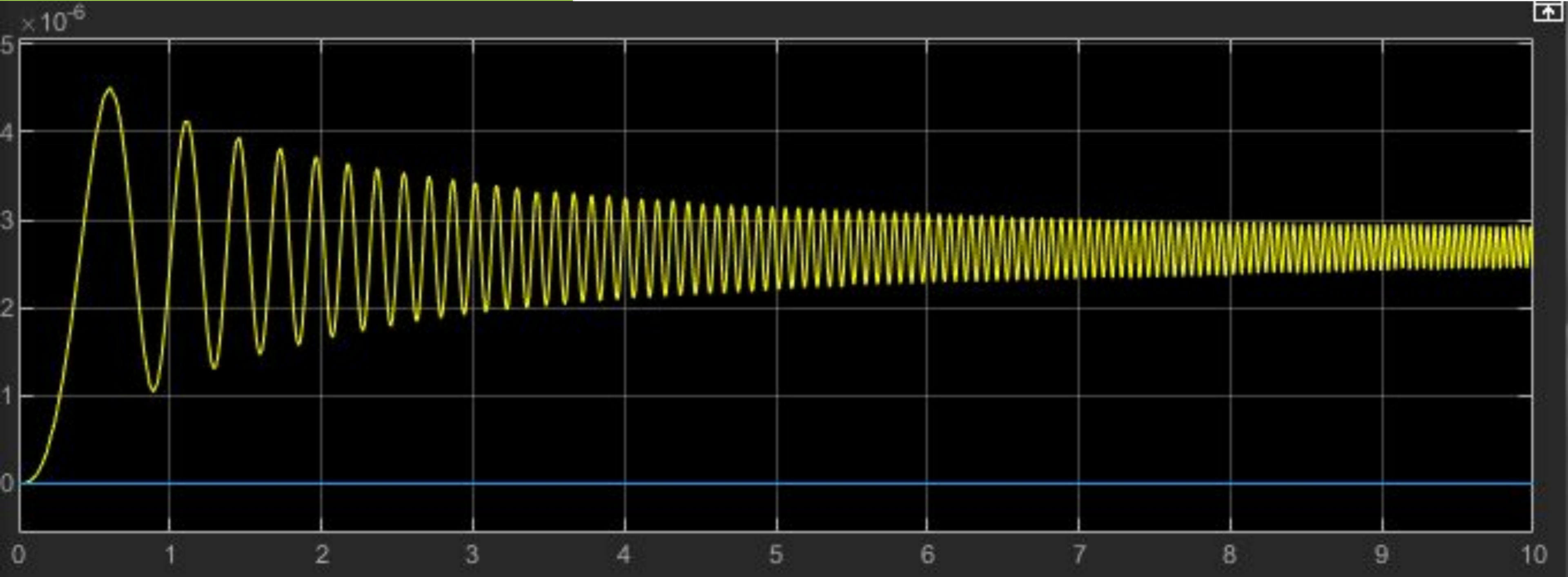
```
%Define Q and R
Q = [21 0; 0 1]; %Started with Original Q=[1 0; 0 1],
adjusted to meet most
R = 1; %Made 1 since we were given no machine
%limits, good for simple math

%Calculate Gain K, ARE Solution S, and Closed-loop
%Poles (P)
[K,S,P] = lqr(nsys,Q,R);
```

For Reference Gain N:

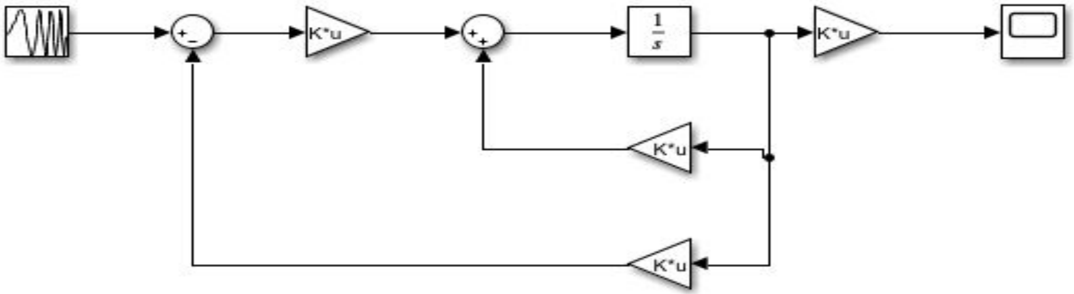
```
%Calculate Gain N for error tracking
N = -(nsys.C*(nsys.A-nsys.B*K)^-1*nsys.B)^-1;
```

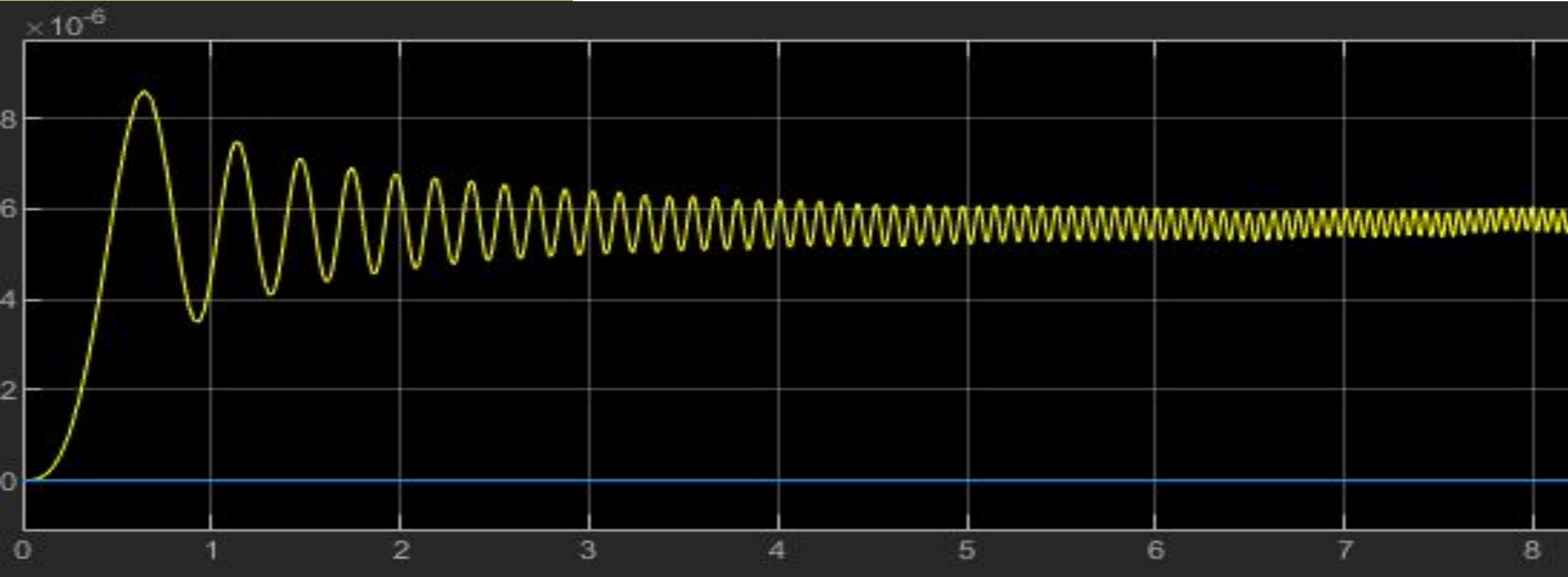
Open Loop Model



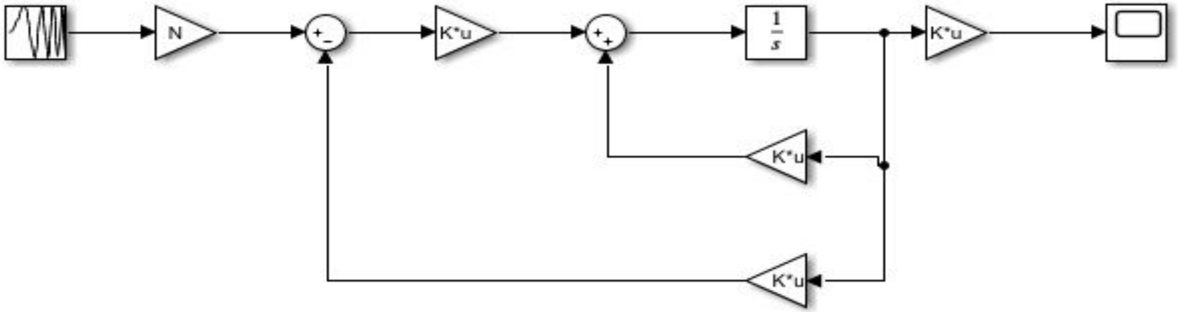


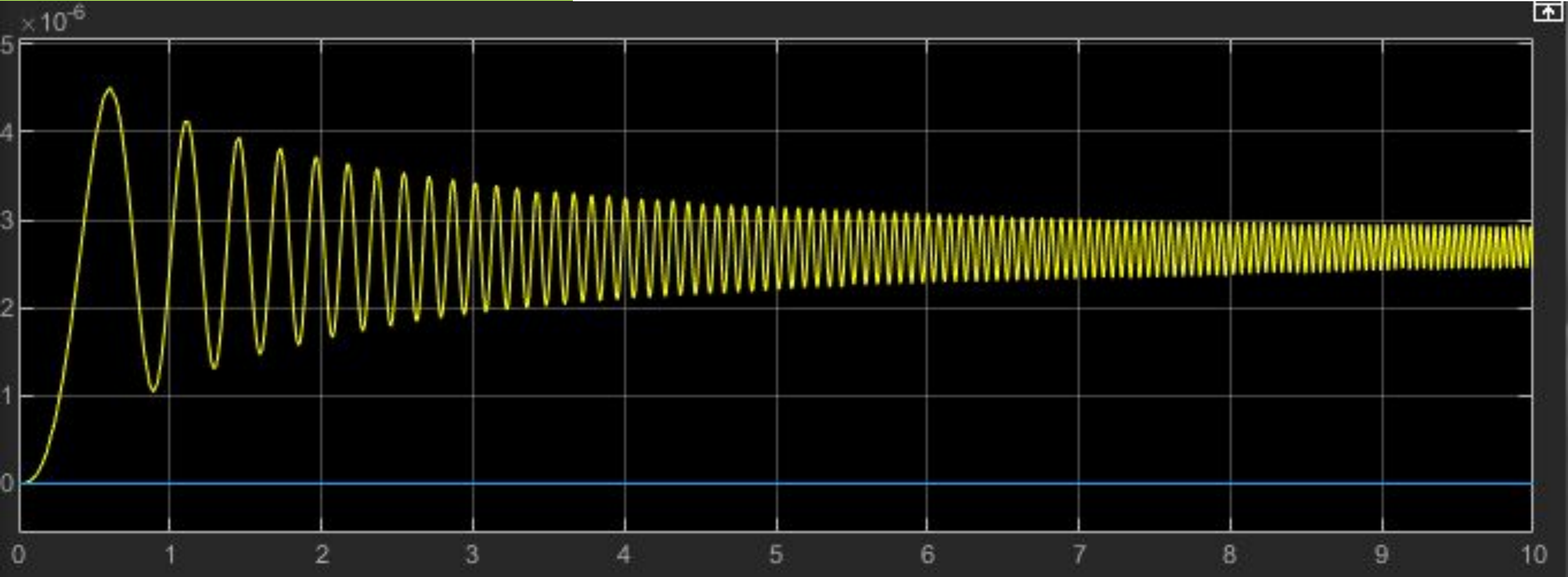
Close Loop Model



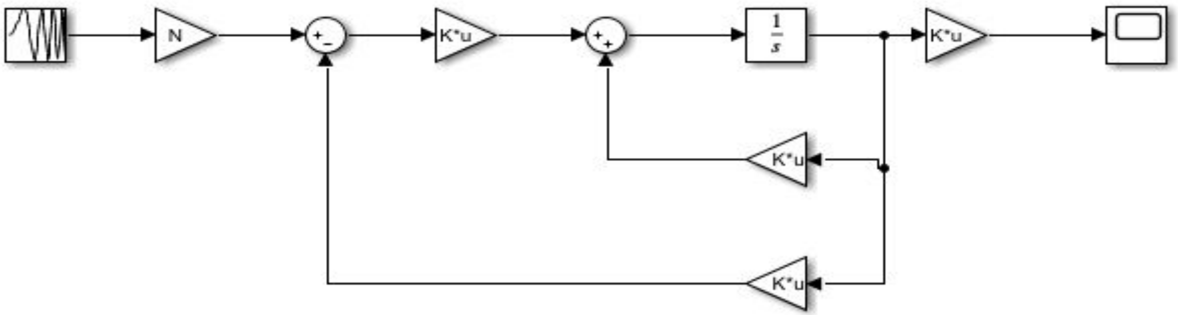


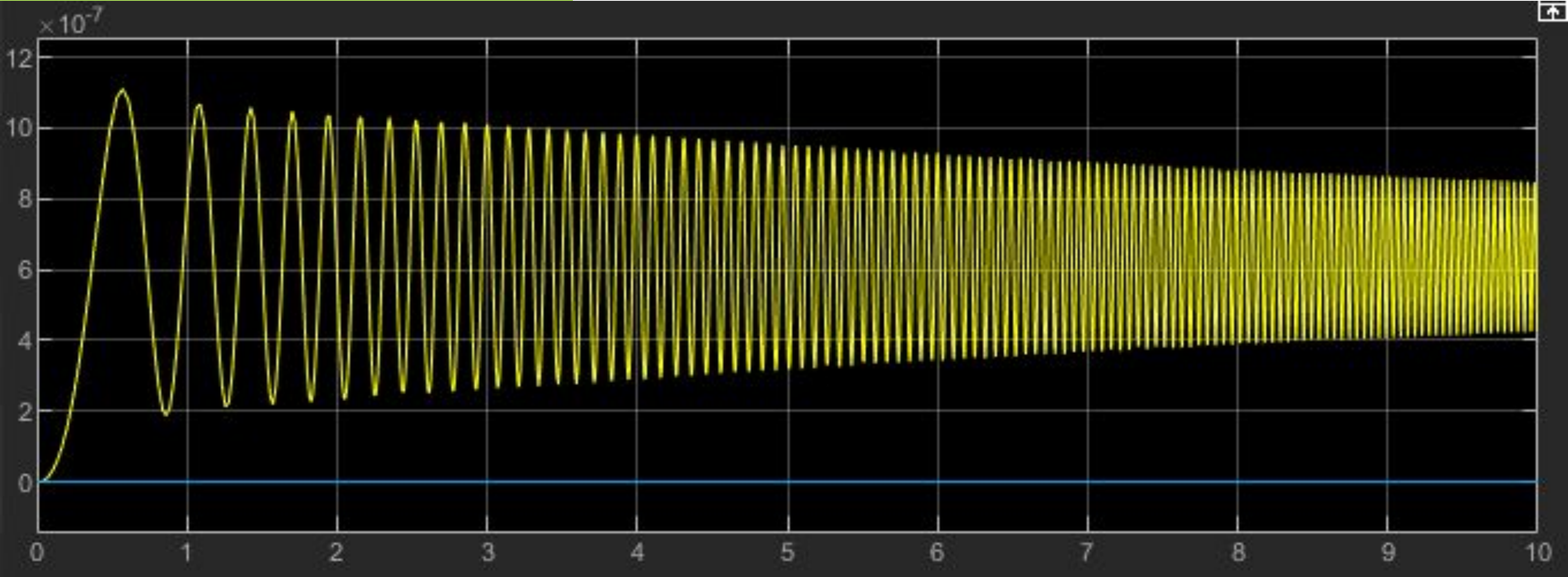
Reference Tracking Model





Optimized Model





Summary

- Bicycle Model
- State-Space Model
- Requirement Check
- Create Optimized Controller

Acknowledgements

Work Cited

[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].

THIS VIDEO WAS PRODUCED AS PART OF THE REQUIREMENTS FOR
ECE 5115 CONTROL LAB II AT THE
CULLEN COLLEGE OF ENGINEERING
UNIVERSITY OF HOUSTON
HOUSTON, TEXAS