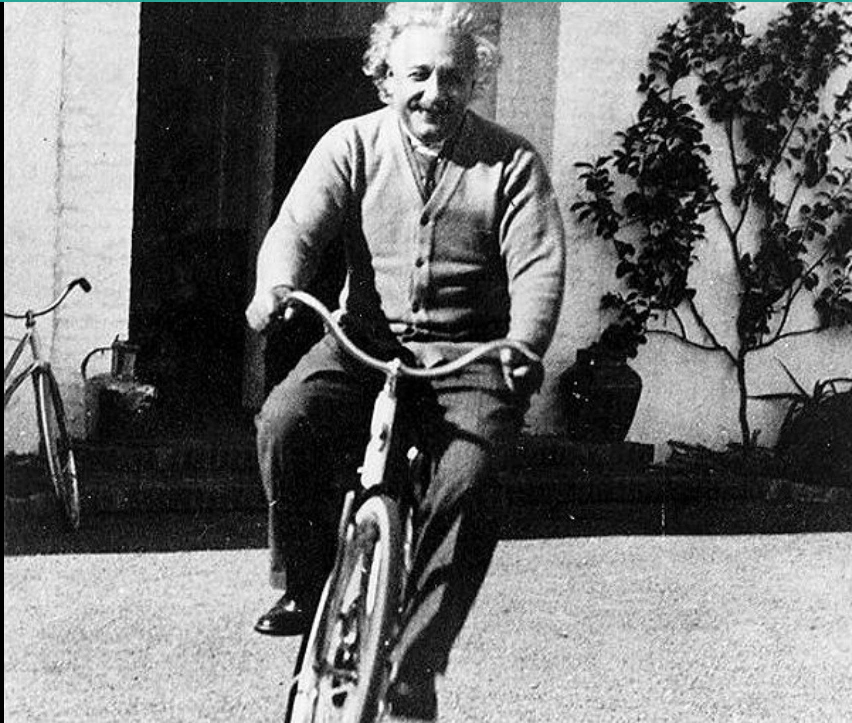
Bicycle Dynamics

Modern Control Course Final Project

Hossein Soltani



Roadmap

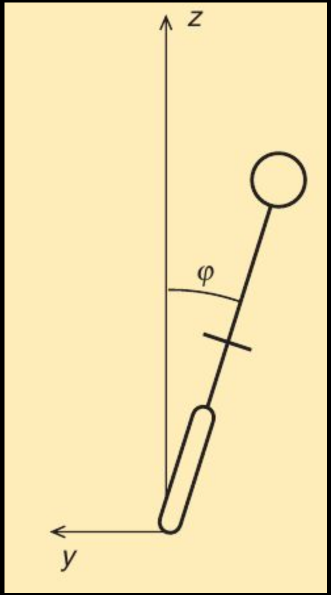
- Description
- Parameters and Models
- State Space Analysis
- Xtra

Description

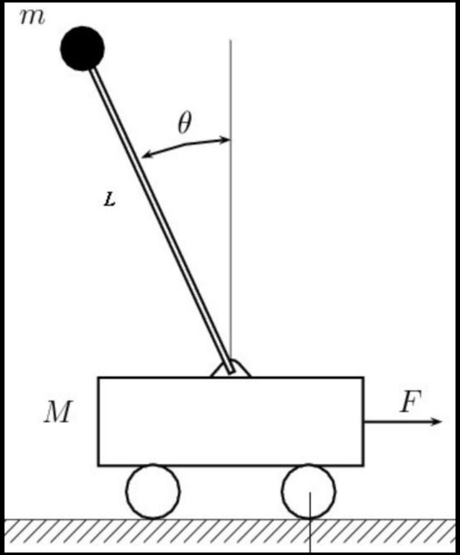


- Bicycle is an interesting dynamical system.
- Bicycle dynamics focuses on understanding the mechanical behavior of a bicycle during motion.
- There are two control tasks for it: Maneuvering, **Stabilization.**
- **In this project, our control task is "Bicycle Stabilization".**

Description - Bicycle Stabilization



Bicycle Rear View



Cart Pendulum System

- Bicycle Stabilization is the task of **keeping the bicycle from falling.**
- It is similar to **inverted pendulum stabilization.**
- Both Bicycle and Inverted Pendulum are **Inherently Unstable. (I'll explain more about this)**

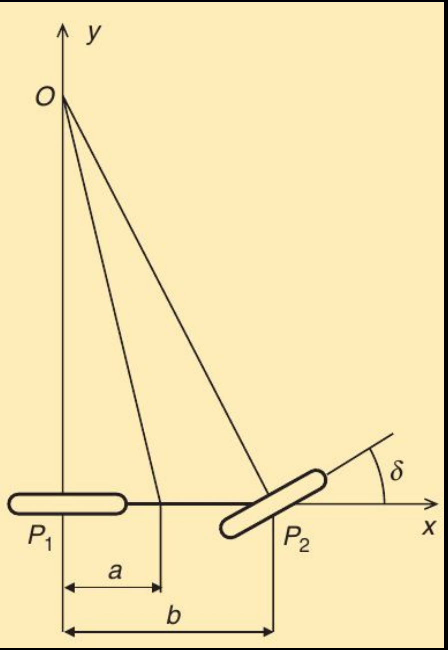
Description - Coordinate System

Description

Parameters and Models

State Space Analysis

Xtra



- We assume that the coordinate system, xyz, is **attached to the bicycle** and has its origin at the contact point of the rear wheel.

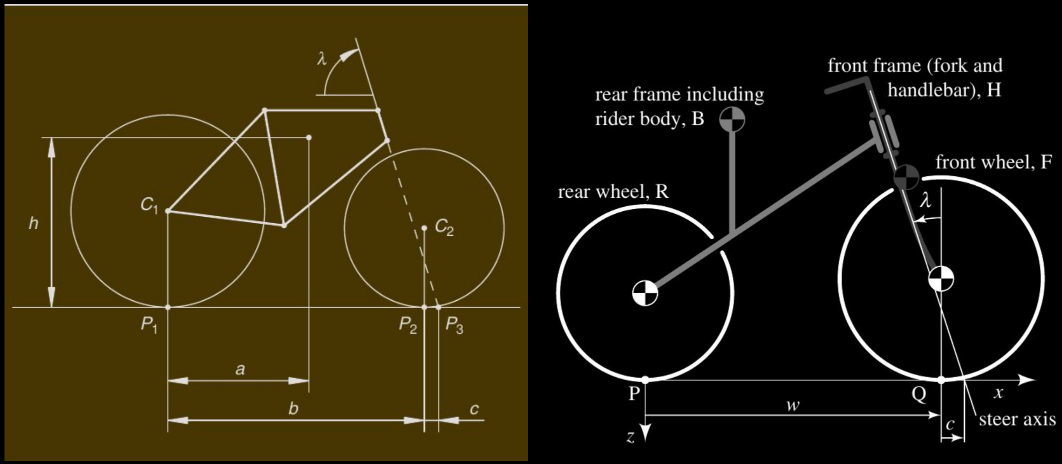
Parameters and Models - Parameters

Description

Parameters and Models

State Space Analysis

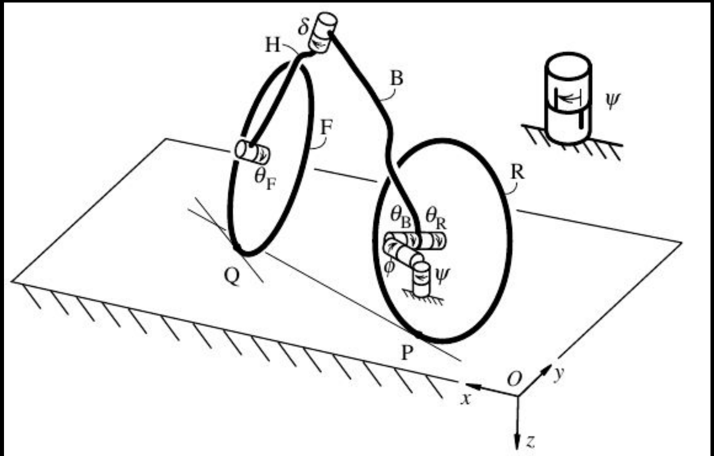
Xtra



Bicycle view from side and the parameters

c	trail distance between P2 and P3
a	x_{COM} distance from a vertical line through the center of mass to P1
h	z_{COM} height of the center of mass
m	mass of the bicycle (including the rider)
λ	head angle
b	wheelbase distance from P1 to P2
C1 and C2	wheels rotation axes
P1 and P2	the contact points of the wheels with the ground
P3	intersection of the steer axis with the horizontal plane
φ	roll angle positive when frame leans to right negative when frame leans to right
δ	steer angle
V	forward velocity of the bicycle

Models



Description

Parameters and Models

State Space Analysis

Xtra

- The first step in **designing a controller** for a system is to develop an **appropriate and thorough model** for it.
- In this project I'm going to use the **Analytical Modeling** or **White-box modeling** method.
- Since the control task is "Bicycle Stabilization", only **rotational movement of different parts of the bicycle** matters, so our model will be built based on **rotational motion relations** and **principles.**
- 3 types of models will be discussed. **#0 - Base Model, #1 - The Model with FeedBack** and **#2 - Whipple Model**.

Models - Model#0 - Base Model

- Description
- Parameters and Models
- State Space Analysis
- Xtra

- When there are **no disturbances** and **forces from the outside** in **an ideal environment**, Bicycle can be considered **an isolated system.**

- So based on the **Angular Momentum Balance principle** and **Newton's second law for rotational motion**, the net torque applied to the frame is 0:

$$\tau_{net} = I * \frac{\partial^2 \varphi}{\partial t^2} \Rightarrow I * \frac{\partial^2 \varphi}{\partial t^2} = mgh * sin(\varphi) + \frac{DV*sin(\lambda)}{b}\frac{\partial \delta}{\partial t} + \frac{m(V^2 h - acg)*sin(\lambda)}{b}\delta$$

$$\Rightarrow I * \frac{\partial^2 \varphi}{\partial t^2} - mgh * sin(\varphi) = \frac{DV*sin(\lambda)}{b}\frac{\partial \delta}{\partial t} + \frac{m(V^2 h - acg)*sin(\lambda)}{b}\delta$$

- $I * \frac{\partial^2 \varphi}{\partial t^2}$ is the angular momentum of the frame.
- $mgh * sin(\varphi)$ is the torque generated by gravity.
- $\frac{DV*sin(\lambda)}{b}\frac{\partial \delta}{\partial t}$ is the inertial torque generated by steering.
- $\frac{m(V^2 h - acg)*sin(\lambda)}{b}\delta$ is the torque due to centrifugal forces

This model is non-linear, because of the sin term. Applying linearization, by considering small range of movement:

$$I * \frac{\partial^2 \varphi}{\partial t^2} - mgh\varphi = \frac{DV*sin(\lambda)}{b}\frac{\partial \delta}{\partial t} + \frac{m(V^2 h - acg)*sin(\lambda)}{b}\delta$$

Models - Model#0 - Base Model

$I \approx m z_{com}^2 = m h^2$ and $D = -I_{xz} \approx m x_{COM} z_{COM} = mah$:

$$\frac{\partial^2 \varphi}{\partial t^2} - \frac{g}{h}\varphi = \frac{aV^* \sin(\lambda)}{bh}\frac{\partial \delta}{\partial t} + \frac{(V^2 h - acg)^* \sin(\lambda)}{bh^2}\delta$$

the final model

Description

Parameters and Models

State Space Analysis

Xtra

Assumptions of this model:

- Small range of movement of the frame.
- The rider sits **rigidly on the bicycle**, so the effect of his/her leaning will be neglected.

Problems of this model:

1) This model is completely unstable.

(Then how come a bicycle maintains its balance? especially when the rider rides with no hands!)

$$\frac{\varphi(s)}{\delta(s)} = \frac{\frac{aV^* \sin(\lambda)}{bh} {}^* s + \frac{(V^2 h - acg)^* \sin(\lambda)}{bh^2}}{s^2 - \frac{g}{h}}$$

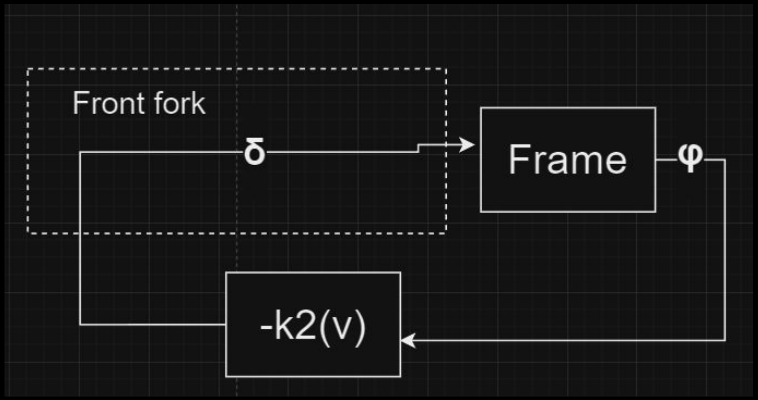
- One pole in the right half plane!
- Inverted Pendulum transfer function:

$$\frac{\theta(s)}{U(s)} = \frac{1}{Ml} * \frac{-1}{s^2 - \frac{g}{h}}$$

2. This model won't take into account the importance of the **Bicycle's Velocity** into the stabilization of it at all. No V in the denominator.

Models - Model#1: The model with FeedBack

- It turns out that there's **feedback between the frame angle (lean angle)** and **steering angle** that helps the bicycle to become and remain stable.



- This feedback relation can explain why it's **possible to ride with no hands** at sufficiently large speed.

The bicycle will **stabilize itself** because of this **internal feedback mechanism.**

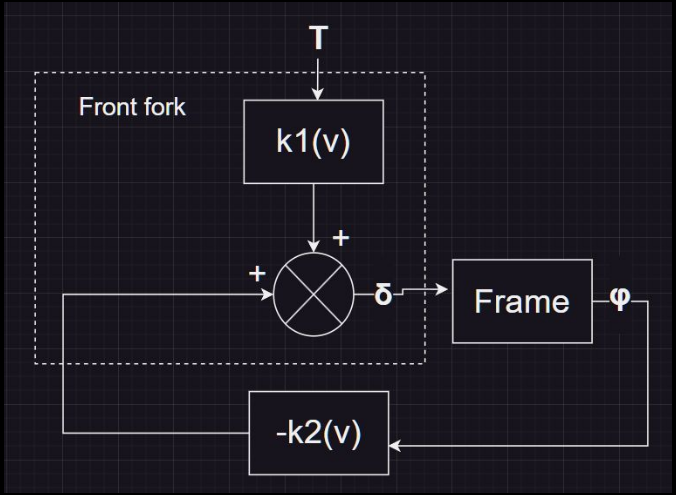
Description

Parameters and Models

State Space Analysis

Xtra

Models - Model#1: The model with FeedBack



- Based on the block diagram, we have a new relation as such: $\delta = k_1(v)T - k_2(v)\varphi$

- So in this model, input is T and the output is φ

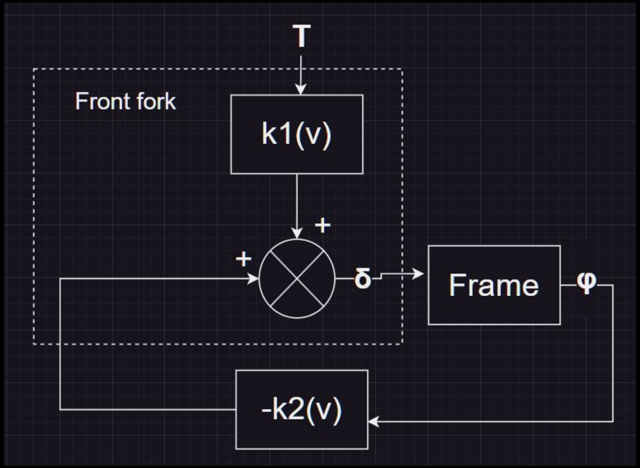
$$k_1(v) = \frac{b^2}{(V^2 * sin(\lambda) - bg * cos(\lambda)) * mac * sin(\lambda)}$$

$$k_2(V) = \frac{bg}{V^2 \sin\lambda - bg\cos\lambda}.$$

By substituting this into the **model#0** and doing necessary simplifications, we will have:

$$\frac{\partial^2\varphi}{\partial t^2} + \frac{aV * sin(\lambda)}{bh} k_2(v) \frac{\partial\varphi}{\partial t} + \varphi[-\frac{g}{h} + k_2(v)\frac{(V^2h - acg) * sin(\lambda)}{bh^2}] = \frac{aV * sin(\lambda)}{bh} k_1(v)\frac{\partial T}{\partial t} + \frac{(V^2h - acg) * sin(\lambda)}{bh^2} k_1(v)T$$

Models - Model#1: The model with FeedBack



- Two new variables, k1(v) and k2(v). Which are the **values of the gains** based on the block diagram. They **both depend on V.**

- So it takes into account Bicycle velocity at a very important part of the model, **feedback gain!**

The model's transfer function becomes:

$$\frac{\varphi(s)}{T(s)} = \frac{\frac{aV*sin(\lambda)}{bh}k_1(v)*s + \frac{(V^2h-acg)*sin(\lambda)}{bh^2}k_1(v)}{s^2 + \frac{aV*sin(\lambda)}{bh}k_2(v)*s + [-\frac{g}{h} + k_2(v)\frac{(V^2h-acg)*sin(\lambda)}{bh^2}]}$$

$$s_{1,2} = \frac{-\frac{aV*sin(\lambda)}{bh}k_2(v) \pm \sqrt{(\frac{aV*sin(\lambda)}{bh})^2 - 4*[-\frac{g}{h} + k_2(v)\frac{(V^2h-acg)*sin(\lambda)}{bh^2}]}}{2}$$

This shows that zeros of the system depend on k1(v) and poles of the system depend on k2(v).

Models - Model#1 Stability Analysis

The main goal of this part, is to show the importance of Velocity in Stability.

Description

Parameters and Models

State Space Analysis

Xtra

We choose these values for the parameters:

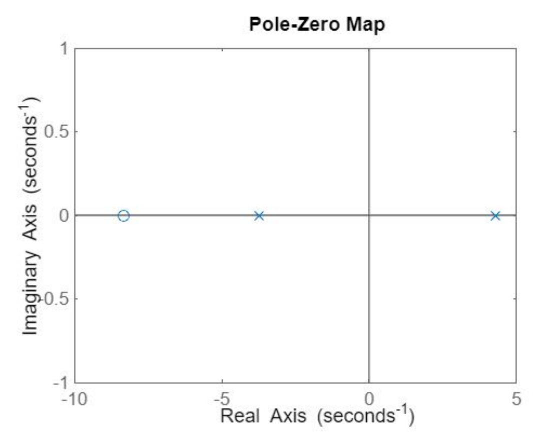
```
b = 1.02;      c = 0.08;      h = 0.8603;    lambda = pi/10;
V = 3;         a = 0.347;     g = 9.81;      m = 84;
k1 = b^2/((V^2*sin(lambda) - b*g*cos(lambda))*m*a*c*sin(lambda));
k2 = b*g/(V^2*sin(lambda) - b*g*cos(lambda));
```

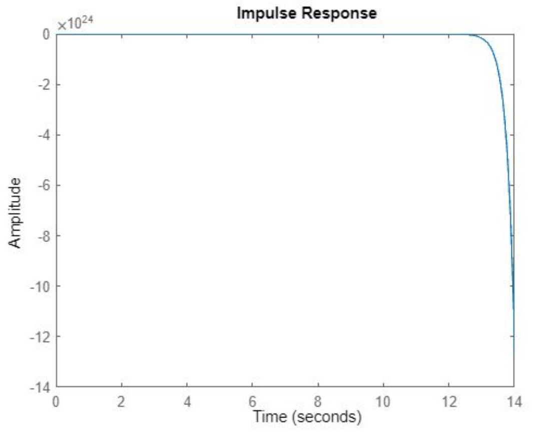
The transfer function will be:

$$\frac{\varphi(s)}{\delta(s)} = \frac{-0.07859*s - 0.6555}{s^2 - 0.5446*s - 15.95}$$

- Based on the PZ map, the system does have one pole at the right half plane ⇒ the system is **unstable!**
- Impulse response will go to **negative infinity**. So this proves that **the system described by this model** isn't **BIBO stable.**

⇒ This proves that at **lower speeds**, the bicycle **isn't stable.**





Models - Model#1 Stability Analysis

Let’s change the value of V to something higher.

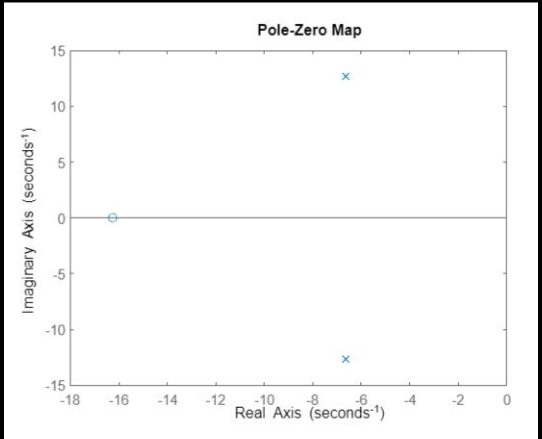
```
V = 5.7;            (other parameters as before)
```

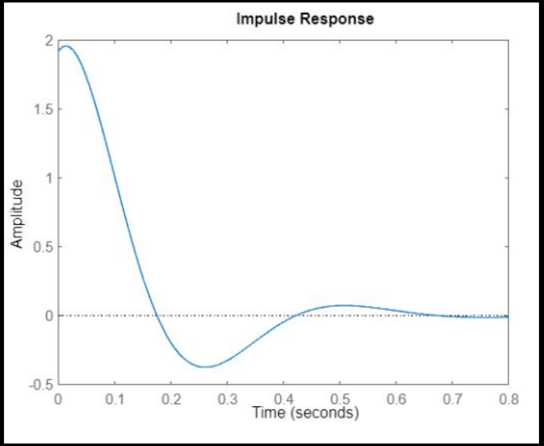
The transfer function will be: $$\frac{\varphi(s)}{\delta(s)} = \frac{1.921{*}s + 31.25}{s^2+13.31{*}s+205.2}$$

- Description
- Parameters and Models
- State Space Analysis
- Xtra

- Based on the PZ map, this model **doesn’t have any pole at the right half plane**. So this proves that this model is **asymptotically stable**.
- This model’s impulse response is **limited** and goes to **zero** at infinity. So this proves that the system described by this model is **BIBO stable**.

⇒ This proves that at **higher speeds**, the bicycle is **stable**.





⇒ Based on these two analysis with different values of V, we can say, Bicycle’s stability depends on its **velocity**.

Models - Model#1 Stability Analysis

As we saw, this model takes into account the importance of V in the stability of the Bicycle very well. But this model also have **some limitations**:

Description

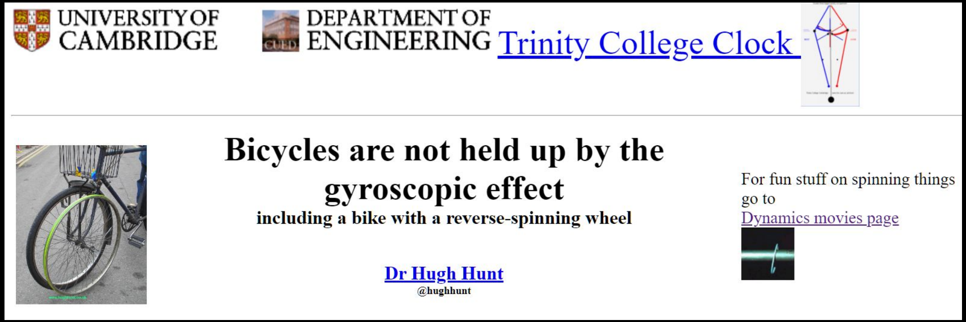
Parameters and Models

State Space Analysis

Xtra

1) This model completely neglects the Gyroscopic Effect in stability.

*Although, based on this research, Gyro Effect isn't very important, especially at **lower speeds**. Because the Angular Momentum of the Bicycle Wheels isn't very high.*
Low speed +
Wheels doesn't have much of mass



*But at **higher speeds,** it can't be negligible.*

2) It also doesn't take into account the interaction of different parts of the bicycle with each other **very well.** $\Rightarrow$ Lower Precision

Models - Model#2 Whipple Model

Description

Parameters and Models

State Space Analysis

Xtra

About this model:

- The dynamics equations for this model follow from **linear** and **angular momentum balance** applied to each part.
- It **neglects** the **motion of the rider relative to the frame**.
- It also takes “Gyro Effect” into account.

$$M\frac{\partial^2 q}{\partial t^2} + CV\frac{\partial q}{\partial t} + (K_1 + K_2 * V^2)q = f$$

$$M = \begin{bmatrix} m1 & m2 \\ m3 & m4 \end{bmatrix}$$

$$C = \begin{bmatrix} c1 & c2 \\ c3 & c4 \end{bmatrix}$$

$$K_1 = \begin{bmatrix} K_{11} & K_{12} \\ K_{13} & K_{14} \end{bmatrix}$$

$$K_2 = \begin{bmatrix} K_{21} & K_{22} \\ K_{23} & K_{24} \end{bmatrix}$$

in which:

$$q = [\varphi\ \delta]^T$$
$$f = [0\ T]^T$$

- M - Symmetric mass matrix of the bicycle model
- CV - a damping-like matrix
- K1 and K2 - Stiffness matrices

- The elements of the matrices depend on the **geometry** and **mass distribution** of the bicycle.

⇒ I’ll choose this model as the main model to do State Space Analysis (rest of the project).

State Space Analysis

At first, let's derive the Dynamical Equations of this model in State Space.

We define the state variables as follows:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \phi \\ \delta \\ \dot{\phi} \\ \dot{\delta} \end{pmatrix} \Longrightarrow \dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ \dot{\delta} \\ \ddot{\phi} \\ \ddot{\delta} \end{pmatrix}$$

Trivially, we have:

$$\dot{x}_1 = x_3,\ \dot{x}_2 = x_4$$

Now we have to find $\dot{x}_3,\ \dot{x}_4$

If we multiply and expand the main equations from matrix form, we'll have:

$$m_1 \cdot \ddot{\varphi} + m_2 \cdot \ddot{\delta} + c_1 V \cdot \dot{\varphi} + c_2 V \cdot \dot{\varphi} + \left(k_{11} + k_{21} V^2\right)\varphi + \left(k_{12} + k_{22} V^2\right)\delta = 0$$
$$m_3 \cdot \ddot{\varphi} + m_4 \cdot \ddot{\delta} + c_3 V \cdot \dot{\varphi} + c_4 V \cdot \dot{\varphi} + \left(k_{13} + k_{23} V^2\right)\varphi + \left(k_{14} + k_{24} V^2\right)\delta = T$$

We have to find $\ddot{\varphi}$ and $\ddot{\delta}$, so we consider these two as **two unknowns** and **others** as **known** and then try to solve the 2 variables 2 equations system.

State Space Analysis

After solving it and simplifications, we'll have:

$$\ddot{\varphi} = \dot{\varphi}[A_1A_2c_1 \cdot V - A_1c_3 \cdot V] + \varphi[A_1A_2a_1 - A_1a_3] + \dot{\delta}[A_1A_2c_2 \cdot V - A_1c_4 \cdot V] + \delta[A_1A_2a_2 - A_1a_4] + A_1 \cdot T$$

Description

Parameters and Models

State Space Analysis

Xtra

we define:

$$A_1 = \frac{m_2}{m_2 * m_3 - m_1 * m_4}$$

$$A_2 = \frac{m_4}{m_2}$$

$$A_3 = \frac{m_1}{m_2}$$

$$a_1 = (k_{11} + k_{21}V^2)$$

$$a_2 = (k_{12} + k_{22}V^2)$$

$$a_3 = (k_{13} + k_{23}V^2)$$

$$a_4 = (k_{14} + k_{24}V^2)$$

substituting:

$$x_1 = \varphi,\ x_2 = \delta,\ x_3 = \dot{\varphi},\ x_4 = \dot{\delta}$$

$$\dot{x}_3 = x_1[A_1A_2a_1 - Aa_3] + x_2[A_1A_2a_2 - A_1a_4] + x_3[A_1A_2c_1 \cdot V - A_1c_3 \cdot V] + x_4[A_1A_2c_2 \cdot V - A_1c_4 \cdot V] + A_1 \cdot T$$

$$\dot{x}_4 = x_1\left[-A_1A_2A_3a_1 + A_1A_3a_3 - \frac{a_1}{m_2}\right] + x_2\left[-A_1A_2A_3a_2 + A_1A_3a_4 - \frac{a_2}{m_2}\right] + x_3\left[-A_1A_2A_3c_1 \cdot V + A_1A_3c_3 \cdot V - \frac{c_1 \cdot V}{m_2}\right] + x_4\left[-A_1A_2A_3c_2 \cdot V + A_1A_3c_4 \cdot V - \frac{c_2 \cdot V}{m_2}\right] - A_1A_3 \cdot T$$

State Space Analysis

So the State Space model of this system becomes:

- Description
- Parameters and Models
- State Space Analysis
- Xtra

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A_1A_2a_1 - Aa_3 & A_1A_2a_2 - A_1a_4 & A_1A_2c_1 \cdot V - A_1c_3 \cdot V & A_1A_2c_2 \cdot V - A_1c_4 \cdot V \\ -A_1A_2A_3a_1 + A_1A_3a_3 - \frac{a_1}{m_2} & -A_1A_2A_3a_2 + A_1A_3a_4 - \frac{a_2}{m_2} & -A_1A_2A_3c_1 \cdot V + A_1A_3c_3 \cdot V - \frac{c_1 \cdot V}{m_2} & -A_1A_2A_3c_2 \cdot V + A_1A_3c_4 \cdot V - \frac{c_2 \cdot V}{m_2} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ A_1 \\ -A_1A_3 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad D = 0$$

- Now let's analyse this model further...

State Space Analysis
Model#2 Stability Analysis in State Space

Values for the parameters:

```
% M matrix
m1 = 96.8;      m2 = -3.57;     m3 = -3.57;     m4 = 0.258;

% C matrix
c1 = 0;         c2 = -50.8;     c3 = 0.436;     c4 = 2.20;

% K0 matrix
k11 = -901.0;  k12 = 35.17;    k13 = 35.17;    k14 = -12.03;

% K2 matrix
k21 = 0;        k22 = -87.06;  k23 = 0;        k24 = 3.50;
```


1 - Stable Model

- It turns out that at the bicycle at V larger than 6 m/s is **stable**, so I choose V=6.1 m/s for velocity.

Matrices in MATLAB:

A =

	x1	x2	x3	x4
x1	0	0	1	0
x2	0	0	0	1
x3	8.741	33.09	-0.7764	2.62
x4	-15.36	-0.227	-21.05	-15.76

B =

	u1
x1	0
x2	0
x3	0.2919
x4	7.915

C =

	x1	x2	x3	x4
y1	1	0	0	0

D =

	u1
y1	0

State Space Analysis
Model#2 Stability Analysis in State Space

1 - Stable Model

A =

	x1	x2	x3	x4
x1	0	0	1	0
x2	0	0	0	1
x3	8.741	33.09	-0.7764	2.62
x4	-15.36	-0.227	-21.05	-15.76

B =

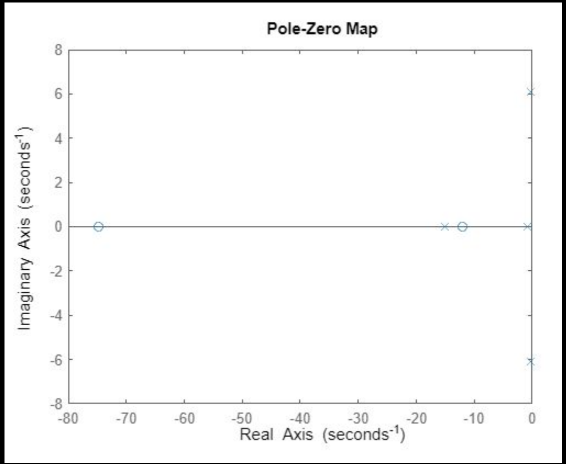
	u1
x1	0
x2	0
x3	0.2919
x4	7.915

C =

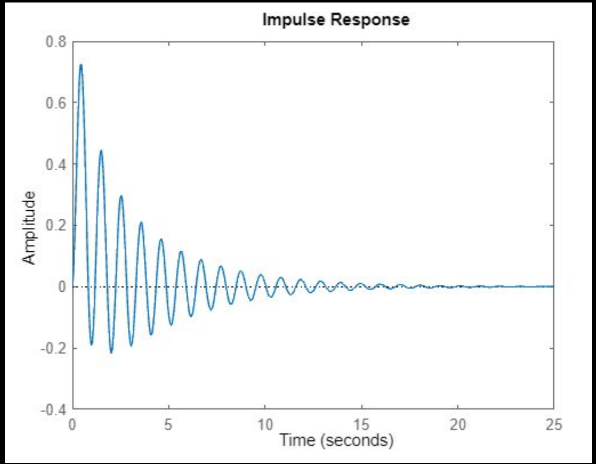
	x1	x2	x3	x4
y1	1	0	0	0

D =

	u1
y1	0




```
>> eig(A)
ans =
  -0.9063 + 0.0000i
  -0.2564 + 6.0736i
  -0.2564 - 6.0736i
 -15.1212 + 0.0000i
```



⇒ This result shows that system (bicycle) is **Asymptotically Stable + BIBO Stable.**

State Space Analysis
Model#2 Stability Analysis in State Space

1 - Stable Model

transfer function

Description

Parameters and Models

State Space Analysis

Xtra

A =

	x1	x2	x3	x4
x1	0	0	1	0
x2	0	0	0	1
x3	8.741	33.09	-0.7764	2.62
x4	-15.36	-0.227	-21.05	-15.76

B =

	u1
x1	0
x2	0
x3	0.2919
x4	7.915

C =

	x1	x2	x3	x4
y1	1	0	0	0

D =

	u1
y1	0

```
sys_tf =

            0.2919 s^2 + 25.34 s + 262
  ---------------------------------------------
  s^4 + 16.54 s^3 + 58.88 s^2 + 599.3 s + 506.5

Continuous-time transfer function.
```

System transfer function

```
>> pole(sys_tf)
ans =
 -15.1212 + 0.0000i
  -0.2564 + 6.0736i
  -0.2564 - 6.0736i
  -0.9063 + 0.0000i
```

⇒ No right half plane poles.

⇒ The system is **asymptotically stable.**

State Space Analysis
Model#2 Stability Analysis in State Space

2 - Unstable Model

As mentioned before, the bicycle at V **lower than 6 m/s** is **unstable**, so I choose **V=3.5 m/s** as velocity. Matrices in MATLAB:

A =

	x1	x2	x3	x4
x1	0	0	1	0
x2	0	0	0	1
x3	8.741	23.97	-0.6491	2.19
x4	-15.36	25.41	-17.6	-13.18

B =

	u1
x1	0
x2	0
x3	0.2919
x4	7.915

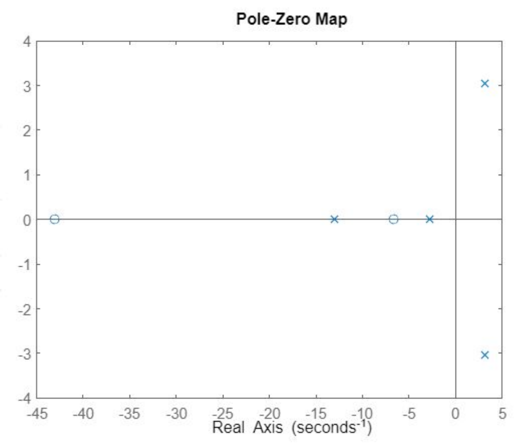
C =

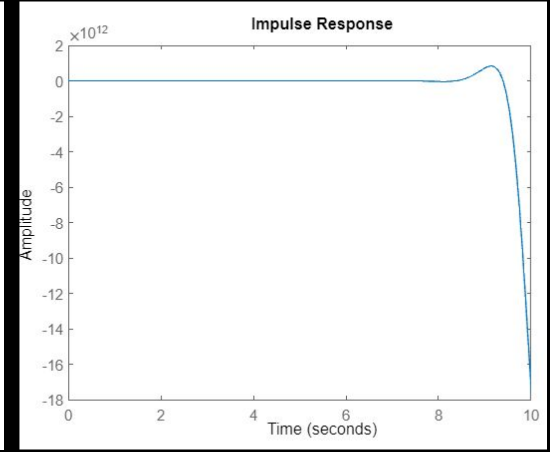
	x1	x2	x3	x4
y1	1	0	0	0

D =

	u1
y1	0

```
>> eig(A)
ans =
   3.1367 + 3.0396i
   3.1367 - 3.0396i
  -2.8048 + 0.0000i
 -12.9589 + 0.0000i
```





⇒ This result shows that system (bicycle) isn't stable!

⇒ Based on the last two analyses also, we can conclude that the stability of the Bicycle depends on V.

And Model#2 captures that very well.

State Space Analysis
Controllability and Observability Analysis

Both of these properties are **very important** for **designing State Feedback** for control and the **Observer** for **state estimation.**

Description

Parameters and Models

State Space Analysis

Xtra

Since only stability was depending on V and Controllability and Observability properties are separate properties of the system, the value for V doesn't matter.

Controllability check in MATLAB:

```
conty = ctrb(sys_ss.A,
sys_ss.B);
>> rank(conty)
ans = 4
```

Controllability matrix is **Full rank.**
⇒ State Space model is **fully Controllable.**

Observability check in MATLAB:

```
obsy = obsv(sys_ss.A, sys_ss.C);
>> rank(obsy)
ans = 4
```

Observability matrix is **Full rank.**
⇒ State Space model is **fully Observable.**

⇒ Since this realization is **both Controllable and Observable**, it's also **Minimal!**

⇒ Every other 4th order realization is also minimal.

State Space Analysis

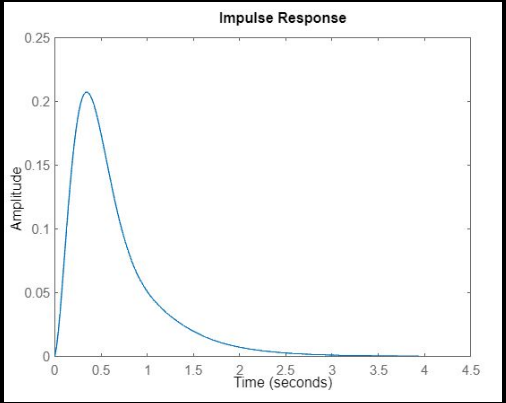
State Feedback Control Using Pole Placement Method

Since the State Space model is **fully controllable**, a State Feedback Controller using Pole Placement method can be designed for that.


```
new_poles = [
  -2 -5+6i -5-6i -10
];
K = place(A,B,new_poles)
```

```
A_pp = A-B*K;
B_pp = B;
C_pp = C;
D_pp = D;

sys_ss_pp = ss(A_pp, B_pp, C_pp, D_pp);
```



Observer for State Estimation:

The State Space model is **fully Observable.** So it's possible to design a **Continuous Luenberger Observer** for that.

```
obsv_poles = [-20, -20, -20, -20];
L_t = acker(A', C', obsv_poles);
L = L_t'
```

$$\begin{cases} \dot{\hat{x}} = (A - LC)\hat{x} + \begin{bmatrix} B & L \end{bmatrix} \begin{bmatrix} u \\ Y \end{bmatrix} \\ Y = C\hat{x} + D \begin{bmatrix} u \\ Y \end{bmatrix} \end{cases}$$

The State Space Dynamical Equations for the Observer

```
A_obs = A-L*C
B_obs = [B L]
C_obs = eye(length(A_obs))
D_obs = zeros(size(B_obs))
```

State Space Analysis

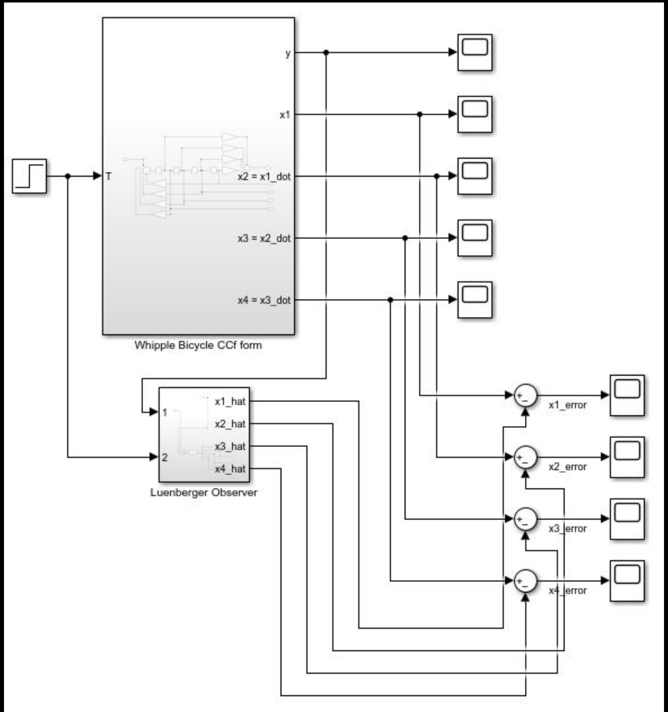
Observer Implementation in Simulink

Description

Parameters and Models

State Space Analysis

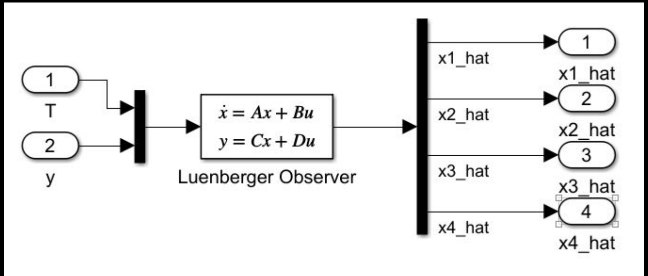
Xtra



Observer + System in Simulink

```
obsv_poles = [-20, -20, -20, -20];
L_t = acker(A_ccf', C_ccf', obsv_poles );
L = L_t';

A_obs = A_ccf-L*C_ccf;
B_obs = [B_ccf L];
C_obs = eye(length(A_obs));
D_obs = zeros(size(B_obs));
```

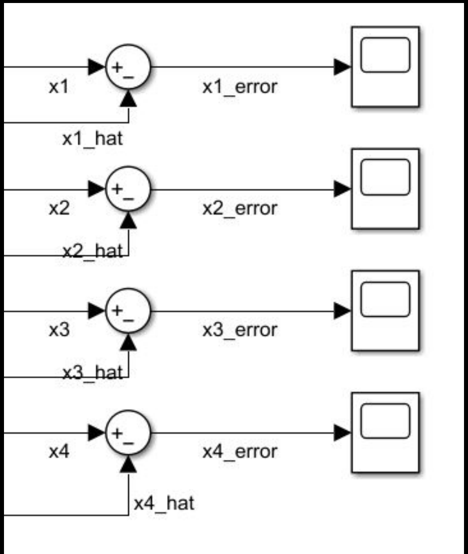


Observer Dynamical Sub System

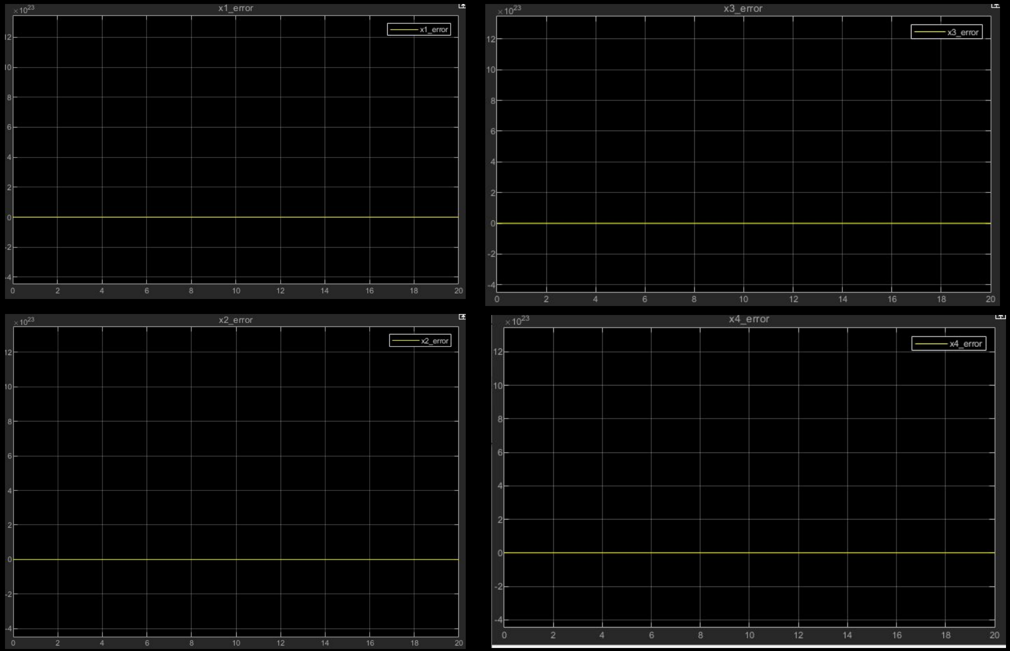
State Space Analysis

Observer Implementation in Simulink - Error Analysis

Let's check whether the Observer can estimate the states properly.



The circuit for Error checking



⇒ Since the error between the estimated states and real ones is **0 at all times**, we can say **Observer estimated states properly**.

State Space Analysis - State FB + Observer

- A state feedback controller requires all of the states of the system.
- But in reality, **not all states are directly measurable**. This is where the observer steps in.

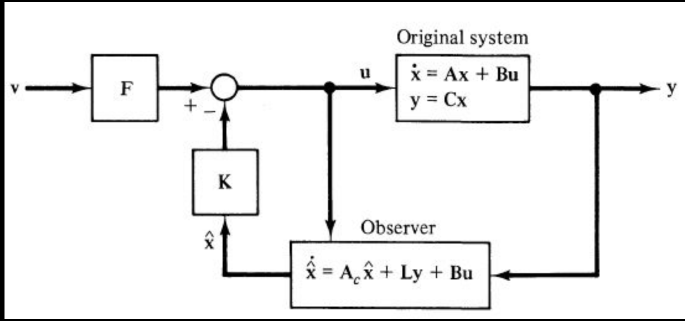
Description

Parameters and Models

State Space Analysis

Xtra

- For designing the Full State Feedback Controller (+ Observer), we consider the "Separation Principle for Feedback Controller".
- Based on this principle, we have to at first design **both** of the **State Feedback Controller** and the **Observer separately**, then **couple them together** by feeding the output of the Observer, which is the system estimated states into the State Feedback Controller.



State Space Analysis - State FB + Observer in Simulink

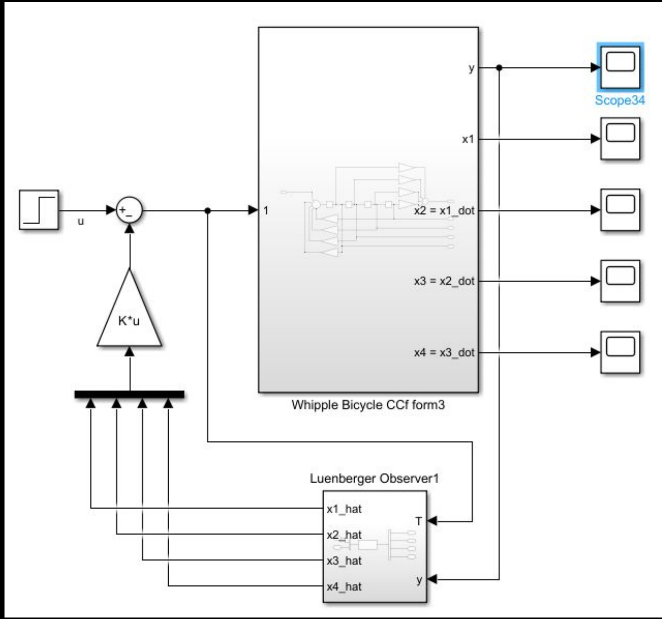
- Using the previously designed State FB and Luenberger Observer, we build our Controller.

Description

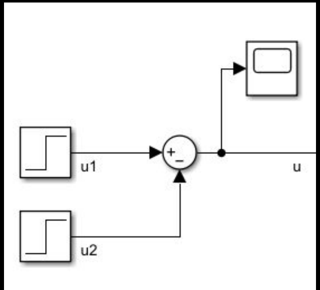
Parameters and Models

State Space Analysis

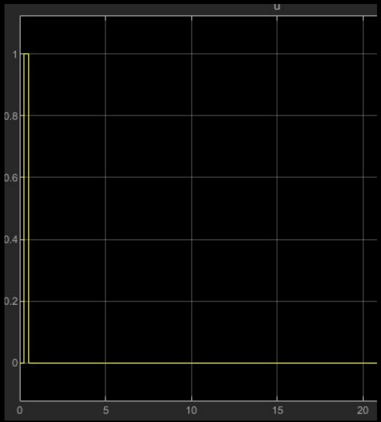
Xtra

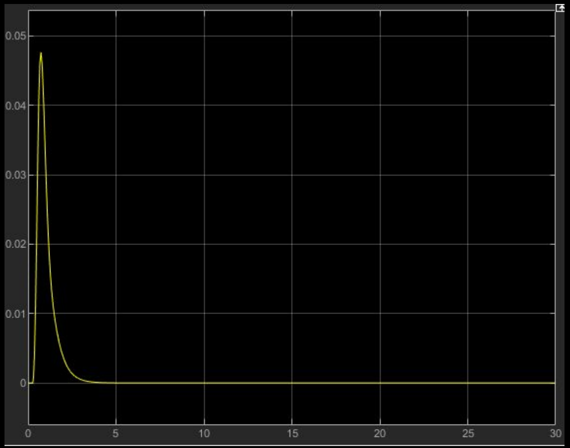


System + Controller (State FB Controller + Observer)



Impulse (kind of) signal as input

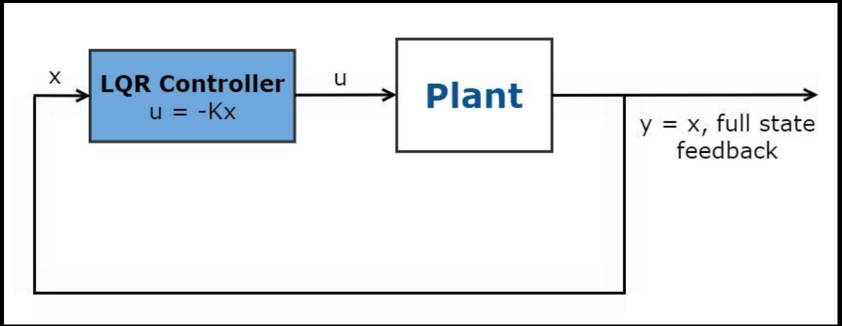




⇒ This result shows that system (bicycle) is **Asymptotically Stable**

⇒ Controller (State FB Controller + Observer) worked perfectly!

Xtra
LQR Controller



Description

Parameters and Models

State Space Analysis

Xtra

- The Linear Quadratic Regulator (LQR) is a control strategy used to **design Feedback Controllers** for **linear systems.**
- It aims to **minimize a cost function** that represents **the system's performance**.
- The case where the system dynamics are described by **a set of linear differential equations** and the cost is **described by a quadratic function** is called the **LQ problem.**
- LQR controller cost function is defined as follows:

$$J = \int_0^{\infty} (x^T Q x + u^T R u)\, dt$$

- The term x^T Q x penalizes **deviations of the state variables from desired values**, promoting **stability** and **tracking performance**.
- The term u^T R u penalizes **excessive control effort**, ensuring **efficient control** and **reducing actuator wear**.

- This cost function does have **an analytical solution**. By solving the **Algebraic Riccati equation**, the K matrix of the State Feedback will be found.
- We don’t do that by ourselves, we simply let MATLAB solve that for us.

$\Rightarrow$ `[K,S,P] = lqr(sys,Q,R)`

Xtra - Designing LQR Controller

The procedure is as follows:

1) Choosing the appropriate Q and R matrices.

I used the ***Bryson's Rule*** *for determining Q and R matrices.*

$$Q_{ii} = \frac{1}{\text{maximum acceptable value of } z_i^2},$$
$$R_{jj} = \frac{1}{\text{maximum acceptable value of } u_j^2},$$

```
Q = [
      1/0.01   0     0    0;
      0     1/0.04   0    0;
      0        0     0    0;
      0        0     0    0;
];
R = 1/4;
```

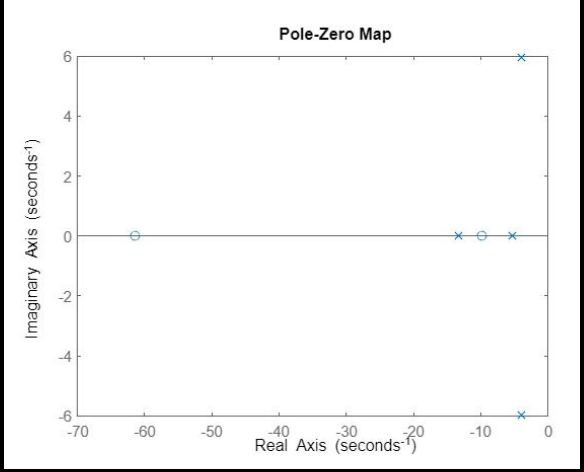
2) Using the MATLAB `lqr` command to find the K matrix.

```
[K, S, P] = lqr(sys_ss, Q, R)
>> K =
   25.8018   20.1905    4.4580    1.4741
```

Stability Analysis after adding the LQR controller:

```
sys_lqr = ss(A-B*K, B, C, D);
```

```
>> pole(sys_lqr)
ans =
 -13.3297 + 0.0000i
  -3.9217 + 5.9548i
  -3.9217 - 5.9548i
  -5.3538 + 0.0000i
```



⇒ Based on these, we can conclude that the system + LQR controller is **asymptotically stable.**

The END