Seesaw rotational dynamics

1. Newton's Second Law for Rotation

For any rigid body rotating about a pivot, Newton's second law takes the form

$$I\,\ddot{\theta} = \sum \tau,$$

where

- I is the total moment of inertia,
- θ is the rotation angle, and
- $\sum \tau$ is the sum of the torques acting on the system.
  Express each term of the torque sum and then divide by I to isolate $\ddot{\theta}$.

2. Gravitational Torque via the 2D Cross Product

a. **The Setup**

Assume two point masses m_1 and m_2 are attached at the ends of a beam (or seesaw) of length L. If the pivot is at the beam's center, each mass is located at a distance $r = L/2$ from the pivot.

b. **Using the Cross Product**

The gravitational force on each mass is $\mathbf{F}_g$ directed downward. The position vectors from the pivot to each mass are, for example, $\mathbf{r}_1$ and $\mathbf{r}_2$

$$\mathbf{F}_g = \begin{pmatrix} 0 \\ -m_i\, g \end{pmatrix} \quad , \quad \mathbf{r}_1 = \frac{L}{2}\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \quad , \quad \mathbf{r}_2 = -\frac{L}{2}\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$

The choice of sign is made for convenience so that the gravitational term "supports" or "resists" depending on the difference in masses.
In two dimensions, the torque for each mass is given by the scalar (the "z-component" of the cross product):

$$\tau = \mathbf{r} \times \mathbf{F} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \times \begin{bmatrix} F_x \\ F_y \\ 0 \end{bmatrix} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x & y & 0 \\ F_x & F_y & 0 \end{vmatrix} = \mathbf{i} \cdot \begin{vmatrix} y & 0 \\ F_y & 0 \end{vmatrix} - \mathbf{j} \cdot \begin{vmatrix} x & 0 \\ F_x & 0 \end{vmatrix} + \mathbf{k} \cdot \begin{vmatrix} x & y \\ F_x & F_y \end{vmatrix} = xF_y - yF_x$$

For mass m_1 (located at $-\frac{L}{2}$ on one side)

$$\tau_1 = \mathbf{r}_1 \times \begin{pmatrix} 0 \\ -m_1 g \end{pmatrix} = \left(\frac{L}{2}\cos\theta\right)(-m_1\,g) - \left(\frac{L}{2}\sin\theta\right)(0) = -\frac{L}{2}\,m_1\,g\,\cos\theta.$$

For mass m_2 (located on the opposite side)

$$\tau_2 = \mathbf{r}_2 \times \begin{pmatrix} 0 \\ -m_2 g \end{pmatrix} = \left(-\frac{L}{2}\cos\theta\right)(-m_2\,g) - \left(-\frac{L}{2}\sin\theta\right)(0) = \frac{L}{2}\,m_2\,g\,\cos\theta.$$

c. **Net Gravitational Torque**

The net gravitational torque is then

$$\tau_g = \tau_1 + \tau_2 = \frac{(m_2 - m_1)g\,L}{2}\cos\theta,$$

Thus, when we divide by I (to write the angular acceleration), the gravitational term becomes

$$\tau_g = \gamma\,\cos\theta \quad , \quad \gamma = \frac{(m_2 - m_1)g\,L}{2\,I}.$$

3. Friction Torque

a. **Viscous Friction**

Viscous friction is modeled as being proportional to the angular velocity, where b is the viscous damping coefficient:

$$\tau_{\text{viscous}} = -b\,\dot{\theta},$$

b. **Coulomb Friction**

The Coulomb (dry) friction torque is given by

$$\tau_{\text{Coulomb}} = -\mu_c\,N\,\frac{L}{2}\,\operatorname{sgn}(\dot{\theta}),$$

where

- μ_c is the Coulomb friction coefficient,
- N is the normal force (which we approximate using g in our gravitational context), and
- $L/2$ is the effective lever arm.
  Because the sign function $\operatorname{sgn}(\dot{\theta})$ is discontinuous at zero, we use a smooth approximation:

  $$\operatorname{sgn}(\dot{\theta}) \approx \tanh\left(\frac{\dot{\theta}}{\epsilon}\right),$$

  This approximation facilitates numerical simulation (continuity and derivability), especially useful if it is then linearized. With ϵ as a small positive constant.

$$\tau_{\text{Coulomb}} \approx -\mu_c \frac{g\,L}{2} \tanh\left(\frac{\dot\theta}{\epsilon}\right).$$

c. **Total Friction Torque**

$$\tau_{\text{friction}} = \tau_{\text{viscous}} + \tau_{\text{Coulomb}} = -b\,\dot\theta - \mu_c \frac{g\,L}{2} \tanh\left(\frac{\dot\theta}{\epsilon}\right)$$

When divided by the total moment of inertia I, the friction terms in the angular acceleration become

$$\tau_{\text{friction}} = -\alpha\,\dot\theta - \beta\,\tanh\left(\frac{\dot\theta}{\epsilon}\right) \quad , \quad \alpha = \frac{b}{I}, \quad \beta = \frac{\mu_c\,g\,L}{2\,I}$$

4. Total Moment of Inertia

a. **Beam's Moment of Inertia**

For a uniform bar (beam) of mass m_b and length L rotating about its center, the moment of inertia is obtained by integrating:

1. **Density**: $\lambda = m_b/L$
2. **Elemental mass**: $dm = \lambda\,dx$
3. **Moment contribution**: $dI = x^2\,dm = x^2\,\lambda\,dx$
4. **Integrate**:

$$I_b = \int_{-L/2}^{L/2} x^2\,\lambda\,dx = \lambda \int_{-L/2}^{L/2} x^2\,dx = \frac{m_b}{L}\,\frac{L^3}{12} = \frac{1}{12} m_b L^2$$

b. **Point Masses' Moments of Inertia**

Each point mass m_i located at a distance $r = L/2$ from the pivot contributes

$$I_i = m_i \left(\frac{L}{2}\right)^2 = \frac{1}{4} m_i L^2.$$

Thus, for both masses:

$$I_{\text{points}} = \frac{1}{4}\,(m_1 + m_2)\,L^2.$$

c. **Total Moment of Inertia**

Adding the beam's and the point masses' moments of inertia gives

$$I = I_b + I_{\text{points}} = \frac{1}{12} m_b L^2 + \frac{1}{4}(m_1 + m_2) L^2.$$

5. Adding the Control Force Torque

To include the contribution of a **control force** F generated by a **drone motor fixed vertically** on one end of the beam (e.g., at $x = \frac{L}{2}$), we must analyze the **torque generated** by that force around the pivot.
We start from the general expression in scalar 2D form $\tau = xF_y - yF_x$. Alternatively, torque magnitude in terms of the angle between the vectors is $\tau = r \cdot F \cdot \sin(\theta_{\text{rel}})$

where:

- $r = \|\mathbf{r}\| = \frac{L}{2}$ is the distance from the pivot to the point where the force is applied,
- $F = \|\mathbf{F}\|$,
- θ_{rel} is the angle between $\mathbf{r}$ and $\mathbf{F}$.

In our system:

- The motor is mounted **perpendicular to the beam** in the **vertical direction**, and thus, $\mathbf{F} = (0, F)$ in the beam's reference frame.
- The point of application is at $\mathbf{r} = (-\frac{L}{2}, 0)$, i.e., the left end of the beam.
- The angle between $\mathbf{r}$ and $\mathbf{F}$ is exactly 90°, so $\sin(\theta_{\text{rel}}) = 1$.
  Then:

$$\tau_{\text{control}} = \frac{L}{2} \cdot F \cdot \sin(90^\circ) = \frac{L}{2} \cdot F$$

  Dividing both sides by I, we define $\delta = L/(2I)$:

$$\tau_{\text{control}} = \delta F$$

6. Assembling the Equation

Writing Newton’s second law with all the torques:

$$I\,\ddot{\theta} = -b\,\dot{\theta} - \mu_c \frac{g\,L}{2} \tanh\left(\frac{\dot{\theta}}{\epsilon}\right) + \frac{(m_1 - m_2) g\,L}{2} \cos\theta + \frac{L}{2} F.$$

Dividing through by I gives

$$\ddot{\theta} = -\frac{b}{I}\,\dot{\theta} - \frac{\mu_c\,g\,L}{2I} \tanh\left(\frac{\dot{\theta}}{\epsilon}\right) + \frac{(m_1 - m_2) g\,L}{2I} \cos\theta + \delta\,F.$$

With the constant definitions

$$\alpha = \frac{b}{I}, \quad \beta = \frac{\mu_c\, g\, L}{2I}, \quad \gamma = \frac{(m_1 - m_2) g\, L}{2I}, \quad \delta = \frac{L}{2I},$$

this becomes

$$\ddot{\theta} + \alpha\,\dot{\theta} + \beta\,\tanh\left(\frac{\dot{\theta}}{\epsilon}\right) = \gamma\,\cos\theta + \delta\,F.$$

7. State-Space Representation

To express the model in state-space form, define the state vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}.$$

Then, the state equations are

$$\boxed{\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\alpha\, x_2 - \beta\,\tanh\left(\frac{x_2}{\epsilon}\right) + \gamma\,\cos x_1 + \delta\,F. \end{aligned}}$$

As $\epsilon \to 0$, $\tanh(x_2/\epsilon) \to \text{sign}(x_2)$ but this function is undefined at zero

$$\boxed{\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\alpha\, x_2 - \beta\,\text{sign}\,(x_2) + \gamma\,\cos x_1 + \delta\,F. \end{aligned}}$$

with the constants defined as:

- $\alpha = \frac{b}{I}$,
- $\beta = \frac{\mu_c\, g\, L}{2I}$,
- $\gamma = \frac{(m_1 - m_2) g\, L}{2I}$,
- $\delta = \frac{L}{2I}$.

Linear Approximation of Dynamical Systems with Inputs and Outputs

System Description

Consider a dynamical system with **state** $x \in \mathbb{R}^n$, **input** $u \in \mathbb{R}^m$, and **output** $y \in \mathbb{R}^p$:

$$\dot{x} = f(x, u) \quad , \quad y = g(x, u, t),$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^p$ are smooth functions.

Equilibrium Points

An equilibrium point $(x_{\text{eq}}, u_{\text{eq}})$ satisfies:

$$f(x_{\text{eq}}, u_{\text{eq}}) = 0.$$

At equilibrium, the state remains stationary if undisturbed, and u_{eq} is the steady-state input.

Multivariable Taylor Series Linearization

State Dynamics Linearization: Expand $f(x, u)$ around $(x_{\text{eq}}, u_{\text{eq}})$ using a first-order Taylor series:

$$f(x, u) \approx f(x_{\text{eq}}, u_{\text{eq}}) + \frac{\partial f}{\partial x}\bigg|_{\text{eq}} (x - x_{\text{eq}}) + \frac{\partial f}{\partial u}\bigg|_{\text{eq}} (u - u_{\text{eq}}).$$

Since $f(x_{\text{eq}}, u_{\text{eq}}) = 0$, the linearized dynamics become:

$$\dot{x} \approx \underbrace{\frac{\partial f}{\partial x}\bigg|_{\text{eq}}}_{A} (x - x_{\text{eq}}) + \underbrace{\frac{\partial f}{\partial u}\bigg|_{\text{eq}}}_{B} (u - u_{\text{eq}}).$$

Output Equation Linearization: Similarly, linearize $y = g(x, u, t)$ around $(x_{\text{eq}}, u_{\text{eq}})$:

$$y \approx g(x_{\text{eq}}, u_{\text{eq}}, t) + \underbrace{\frac{\partial g}{\partial x}\bigg|_{\text{eq}}}_{C} (x - x_{\text{eq}}) + \underbrace{\frac{\partial g}{\partial u}\bigg|_{\text{eq}}}_{D} (u - u_{\text{eq}}).$$

Linearized System Representation

Perturbation Variables: Define deviations from equilibrium, where $y_{\text{eq}} = g(x_{\text{eq}}, u_{\text{eq}}, t)$ is the equilibrium output:

$$\begin{aligned} z &= x - x_{\text{eq}}, \\ v &= u - u_{\text{eq}}, \\ \Delta y &= y - y_{\text{eq}}, \end{aligned}$$

Linear State-Space Model: The linearized system becomes:

$$\begin{aligned} \dot{z} &= Az + Bv, \\ \Delta y &= Cz + Dv. \end{aligned}$$

Stability Analysis

- **Stability** depends on the eigenvalues of $A = \frac{\partial f}{\partial x}\big|_{\text{eq}}$:
  - **Stable**: All eigenvalues of A have $\text{Re}(\lambda_i) < 0$.
  - **Unstable**: Any eigenvalue has $\text{Re}(\lambda_i) > 0$.
  - **Marginal**: Eigenvalues with $\text{Re}(\lambda_i) = 0$ (nonlinear terms dominate).

Solution of the Linearized System

1. **The initial condition** for the linearized system z_0 is defined as the deviation of the original system's initial state x_0 at t_0 from equilibrium x_{eq}:

$$z_0 = x_0 - x_{\text{eq}},$$

2. **State Trajectory**:

$$z(t) = e^{At} z(0) + \int_0^t e^{A(t-\tau)} B v(\tau) d\tau \quad , \quad x(t) = x_{\text{eq}} + z(t)$$

3. **Output Response**:

$$\Delta y(t) = C z(t) + D v(t) \quad , \quad y(t) = y_{\text{eq}} + \Delta y(t)$$

Equilibrium and Linearization Analysis

1. System Dynamics

The state-space model is:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\alpha\, x_2 - \beta\, \tanh\left(\frac{x_2}{\epsilon}\right) + \gamma\, \cos x_1 + \delta\, F \end{aligned}$$

where $x_1 = \theta$, $x_2 = \dot{\theta}$, and the constants are:

- $\alpha = b/I$ (viscous damping)
- $\beta = \mu_c g L/(2I)$ (Coulomb friction magnitude)
- $\gamma = (m_2 - m_1) g L/(2I)$ (gravitational torque factor)
- $\delta = L/(2I)$ (control input gain)

2. Equilibrium Points

Equilibrium points ($\mathbf{x}_{eq}$) occur when $\dot{\mathbf{x}} = \mathbf{0}$.

1. $\dot{x}_1 = x_2 = 0 \implies \dot{\theta}_{eq} = 0$.
2. Substituting $x_2 = 0$ into $\dot{x}_2$:
   $0 = -0 - \beta \tanh(0) + \gamma \cos x_1 + \delta F$

$$\gamma \cos \theta_{eq} + \delta F = 0 \quad \implies \quad \cos \theta_{eq} = -\frac{\delta F}{\gamma}$$

- **Case 1: No Control ($F = 0$)**
  $\cos\theta_{eq} = 0 \implies \theta_{eq} = \frac{\pi}{2} + n\pi$, for any integer n.
  The distinct equilibrium points are $(\theta_{eq}, \dot{\theta}_{eq}) = (\pm\frac{\pi}{2}, 0)$. These correspond to the vertical positions.
- **Case 2: With Control ($F \neq 0$)**
  Equilibrium points exist if $|\frac{\delta F}{\gamma}| \leq 1$.
  If so, $\theta_{eq} = \pm\arccos(-\frac{\delta F}{\gamma}) + 2k\pi$.

3. Linearization Analysis

1. Non-linear System Equations

Let the state vector be $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$.
The system can be written as $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, F)$, where

$$\mathbf{f}(\mathbf{x}, F) = \begin{pmatrix} x_2 \\ -\alpha\, x_2 - \beta\, \tanh\left(\frac{x_2}{\epsilon}\right) + \gamma\, \cos x_1 + \delta\, F \end{pmatrix} = \begin{pmatrix} f_1(\mathbf{x}, F) \\ f_2(\mathbf{x}, F) \end{pmatrix}$$

Note on $\tanh$: Using $\tanh(x_2/\epsilon)$ instead of $\operatorname{sgn}(x_2)$ makes the function differentiable at $x_2 = 0$, allowing for straightforward Jacobian calculation at equilibrium.

3. Compute Jacobians

We linearize $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, F)$ around an operating point $(\mathbf{x}_{op}, F_{op})$

Matrix A (Jacobian):

$$A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix}$$

The partial derivatives are:

- $\frac{\partial f_1}{\partial x_1} = \frac{\partial (x_2)}{\partial x_1} = 0$
- $\frac{\partial f_1}{\partial x_2} = \frac{\partial (x_2)}{\partial x_2} = 1$
- $\frac{\partial f_2}{\partial x_1} = \frac{\partial}{\partial x_1}(-\alpha x_2 - \beta \tanh(\frac{x_2}{\epsilon}) + \gamma \cos x_1 + \delta F) = -\gamma \sin x_1$
- $\frac{\partial f_2}{\partial x_2} = \frac{\partial}{\partial x_2}(-\alpha x_2 - \beta \tanh(\frac{x_2}{\epsilon}) + \gamma \cos x_1 + \delta F) = -\alpha - \beta \frac{1}{\epsilon} \operatorname{sech}^2(\frac{x_2}{\epsilon})$

Evaluate at the equilibrium point $(\mathbf{x}_{eq}, F_{eq}) = (\theta_{eq}, 0, F_{eq})$:

- $\left.\frac{\partial f_1}{\partial x_1}\right|_{eq} = 0$
- $\left.\frac{\partial f_1}{\partial x_2}\right|_{eq} = 1$

- $\left.\frac{\partial f_2}{\partial x_1}\right|_{eq} = -\gamma \sin\theta_{eq}$
- $\left.\frac{\partial f_2}{\partial x_2}\right|_{eq} = -\alpha - \frac{\beta}{\epsilon}\text{sech}^2(0) = -\alpha - \frac{\beta}{\epsilon}$

So, the Jacobian matrix at equilibrium is:

$$A = \begin{pmatrix} 0 & 1 \\ -\gamma \sin\theta_{eq} & -\alpha - \beta/\epsilon \end{pmatrix}$$

Matrix B (Input Matrix):

$$B = \frac{\partial \mathbf{f}}{\partial F} = \begin{pmatrix} \frac{\partial f_1}{\partial F} \\ \frac{\partial f_2}{\partial F} \end{pmatrix}$$

- $\frac{\partial f_1}{\partial F} = \frac{\partial (x_2)}{\partial F} = 0$
- $\frac{\partial f_2}{\partial F} = \frac{\partial}{\partial F}(-\alpha x_2 - \beta \tanh(\frac{x_2}{\epsilon}) + \gamma \cos x_1 + \delta F) = \delta$

Evaluating at equilibrium (or anywhere, as it's constant):

$$B = \begin{pmatrix} 0 \\ \delta \end{pmatrix}$$

The Linearized System

The initial conditions for the linearized system are $\mathbf{z}(0)$ and $\mathbf{x}(0)$ are the initial conditions of the non linear system

$$\mathbf{z}(0) = \mathbf{x}(0) - \mathbf{x}_{eq}$$

The real input is $\Delta F(t) = F(t) - F_{\text{eq}}$

$$\boxed{\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\gamma \sin\theta_{eq} & -\alpha - \beta/\epsilon \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \delta \end{pmatrix} \Delta F}$$

The restore the original coordinates use

$$\mathbf{x}(t) = \mathbf{x}_{eq} + \mathbf{z}(t)$$

This represents the linearized dynamics of the deviations ($z_1 = \Delta\theta$, $z_2 = \Delta\dot{\theta}$) from the equilibrium point $(\theta_{eq}, 0)$ due to input deviations ΔF.

Analysis at Specific Points (assuming $F = 0$)

- **Around $\theta = 0$ (Horizontal):**
  - This is **not an equilibrium point** unless $\gamma = 0$ ($m_1 = m_2$).

- Linearizing here yields $A = \begin{pmatrix} 0 & 1 \\ 0 & -\alpha - \beta/\epsilon \end{pmatrix}$.
- Linearized equation: $\Delta\ddot{\theta} + (\alpha + \beta/\epsilon)\Delta\dot{\theta} = \delta\Delta F$.
- **Issue:** The restoring force term dependent on $\Delta\theta$ (from gravity) vanishes ($-\gamma\sin(0) = 0$). The linearization **loses the essential angular dynamics** and doesn't capture the tendency to fall away from horizontal if $m_1 \neq m_2$.

- **Around $\theta_{eq} = \pi/2$ (Vertical):**
  - $A = \begin{pmatrix} 0 & 1 \\ -\gamma & -\alpha - \beta/\epsilon \end{pmatrix}$.
  - Linearized equation: $\Delta\ddot{\theta} + (\alpha + \beta/\epsilon)\Delta\dot{\theta} + \gamma\Delta\theta = \delta\Delta F$.
  - **Usefulness:** This is a standard 2nd order system. It captures the local dynamics. If $\gamma > 0$ (i.e., $m_2 > m_1$), the $+\gamma\Delta\theta$ term represents a **restoring torque**, making this equilibrium **stable** (assuming damping $\alpha, \beta > 0$).
- **Around $\theta_{eq} = -\pi/2$ (Vertical):**
  - $A = \begin{pmatrix} 0 & 1 \\ \gamma & -\alpha - \beta/\epsilon \end{pmatrix}$.
  - Linearized equation: $\Delta\ddot{\theta} + (\alpha + \beta/\epsilon)\Delta\dot{\theta} - \gamma\Delta\theta = \delta\Delta F$.
  - **Instability:** If $\gamma > 0$ ($m_2 > m_1$), the $-\gamma\Delta\theta$ term represents an **anti-restoring torque**, pushing the system away from equilibrium. The linearized system has a positive eigenvalue, indicating **instability**, matching simulation observations.

4. Stability vs. n and γ

The equilibrium angles are $\theta_{eq} = \frac{\pi}{2} + n\pi$. Stability depends on n (even/odd) and the sign of $\gamma = (m_2 - m_1)gL/(2I)$.

- **If n is even** (e.g., -2, 0, 2,...): $\theta_{eq} \equiv \pi/2$.
  - Linearized term: $+\gamma\Delta\theta$.
  - **Stable** if $\gamma > 0$ ($m_2 > m_1$, heavier mass m_2 is down).
  - **Unstable** if $\gamma < 0$ ($m_1 > m_2$, heavier mass m_1 is up).
- **If n is odd** (e.g., -1, 1, 3,...): $\theta_{eq} \equiv -\pi/2$.
  - Linearized term: $-\gamma\Delta\theta$.
  - **Stable** if $\gamma < 0$ ($m_1 > m_2$, heavier mass m_1 is down).
  - **Unstable** if $\gamma > 0$ ($m_2 > m_1$, heavier mass m_2 is up).

Conclusion on negative n:
The stability for $\theta_{eq} = \pi/2 + n\pi$ with *negative* n depends on whether n is even or odd:

- If n is negative **odd** ($\equiv -\pi/2$): Unstable if $\gamma > 0$, Stable if $\gamma < 0$.
- If n is negative **even** ($\equiv \pi/2$): Stable if $\gamma > 0$, Unstable if $\gamma < 0$.

Swapping the masses ($m_1 \leftrightarrow m_2$) flips the sign of γ, thus **reversing the stability** of both vertical equilibrium configurations ($\pi/2$ and $-\pi/2$).

Lyapunov Stability Analysis for Linearized System

1. Linear System Model

The system under consideration is the linearized dynamics of a seesaw around the vertical equilibrium point $\theta_{\text{eq}} = \frac{\pi}{2}$. This equilibrium is assumed to be stable in the absence of damping and friction, requiring $m_2 > m_1$. Equivalent to $\theta_{\text{eq}} = -\frac{\pi}{2}$ for $m_1 > m_2$. The linearized dynamics are described by the linear time-invariant (LTI) system:

$$\dot{\mathbf{z}} = A\mathbf{z},$$

where $\mathbf{z}$ is the state vector representing the deviation from the equilibrium and the angular velocity:

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \theta - \theta_{\text{eq}} \\ \dot{\theta} \end{bmatrix}.$$

The system matrix A is given by:

$$A = \begin{pmatrix} 0 & 1 \\ -\gamma & -\alpha - \frac{\beta}{\epsilon} \end{pmatrix},$$

where the parameters are defined in terms of the physical properties of the seesaw (m_1, m_2 are masses, g is gravity, L is half the seesaw length, I is the moment of inertia, b is viscous damping coefficient, μ_c is Coulomb friction magnitude, and ϵ is a small positive constant used to handle the sign function in the original nonlinear model):

$$\gamma = \frac{(m_2 - m_1)gL}{2I}, \quad \alpha = \frac{b}{I}, \quad \beta = \frac{\mu_c g L}{2I}, \quad \epsilon > 0.$$

For the equilibrium $\theta_{\text{eq}} = \pi/2$ to be a point we can analyze for stability in this linearized model, we typically need $\gamma > 0$, which corresponds to $m_2 > m_1$. We also assume positive viscous damping ($b > 0$, so $\alpha > 0$) and non-zero Coulomb friction effects ($\mu_c > 0$, so $\beta > 0$). The moment of inertia I is always positive.

2. Lyapunov Function Candidate

To analyze the stability of the equilibrium point $\mathbf{z} = \mathbf{0}$ using Lyapunov's Direct Method, we choose a quadratic Lyapunov function candidate:

$$V(\mathbf{z}) = \mathbf{z}^T P \mathbf{z},$$

where P is a symmetric, positive definite matrix ($P = P^T \succ 0$). For a 2×2 system, the matrix P has the form:

$$P = \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix}.$$

For P to be positive definite, its leading principal minors must be positive: $p_{11} > 0$ and $\det(P) = p_{11}p_{22} - p_{12}^2 > 0$. The positive definiteness of P ensures that $V(\mathbf{z}) > 0$ for all $\mathbf{z} \neq \mathbf{0}$ and $V(\mathbf{0}) = 0$.

3. Solving the Lyapunov Equation

For a linear system $\dot{\mathbf{z}} = A\mathbf{z}$, asymptotic stability at the origin is proven if there exists a positive definite matrix P such that the time derivative of the Lyapunov function, $\dot{V}(\mathbf{z})$, is negative definite. The derivative of $V(\mathbf{z})$ along the system trajectories is given by:

$$\dot{V}(\mathbf{z}) = \frac{d}{dt}(\mathbf{z}^T P \mathbf{z}) = \dot{\mathbf{z}}^T P \mathbf{z} + \mathbf{z}^T P \dot{\mathbf{z}}.$$

Substituting $\dot{\mathbf{z}} = A\mathbf{z}$:

$$\dot{V}(\mathbf{z}) = (A\mathbf{z})^T P \mathbf{z} + \mathbf{z}^T P (A\mathbf{z}) = \mathbf{z}^T A^T P \mathbf{z} + \mathbf{z}^T P A \mathbf{z} = \mathbf{z}^T (A^T P + PA) \mathbf{z}.$$

To ensure $\dot{V}(\mathbf{z})$ is negative definite, we can choose a positive definite matrix Q and solve the algebraic **Lyapunov equation**:

$$A^T P + PA = -Q.$$

A standard approach is to choose Q as the identity matrix, $Q = I$. Substituting the matrix A, the general form of P, and $Q = I$:

$$\begin{pmatrix} 0 & -\gamma \\ 1 & -\left(\alpha + \frac{\beta}{\epsilon}\right) \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} + \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -\gamma & -\left(\alpha + \frac{\beta}{\epsilon}\right) \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Performing the matrix multiplications:

$$\begin{pmatrix} -\gamma p_{12} & -\gamma p_{22} \\ p_{11} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{12} & p_{12} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{22} \end{pmatrix} + \begin{pmatrix} -\gamma p_{12} & p_{11} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{12} \\ -\gamma p_{22} & p_{12} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{22} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Adding the matrices on the left-hand side and equating corresponding elements to the matrix on the right-hand side results in the following system of linear equations for the elements of P:

$$\begin{pmatrix} -2\gamma p_{12} & p_{11} - \gamma p_{22} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{12} \\ p_{11} - \gamma p_{22} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{12} & 2\left(p_{12} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{22}\right) \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

This matrix equation gives us three distinct equations (due to the symmetry of P and $-Q$):

1. $-2\gamma p_{12} = -1$
2. $p_{11} - \gamma p_{22} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{12} = 0$
3. $2\left(p_{12} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{22}\right) = -1$

From equation (1), assuming $\gamma \neq 0$, we can solve for p_{12}:

$$p_{12} = \frac{1}{2\gamma}.$$

Since we assume $\gamma > 0$ for stability at $\theta_{eq} = \pi/2$, p_{12} is a well-defined positive value.

From equation (3):

$$p_{12} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{22} = -\frac{1}{2}.$$

Substitute the value of p_{12}:

$$\frac{1}{2\gamma} - \left(\alpha + \frac{\beta}{\epsilon}\right) p_{22} = -\frac{1}{2}.$$

Rearranging and solving for p_{22}:

$$\left(\alpha + \frac{\beta}{\epsilon}\right) p_{22} = \frac{1}{2\gamma} + \frac{1}{2} = \frac{1+\gamma}{2\gamma}.$$

Since $\alpha > 0$, $\beta > 0$, and $\epsilon > 0$, the term $\left(\alpha + \frac{\beta}{\epsilon}\right) > 0$. As $\gamma > 0$, the right-hand side is also positive. Thus, p_{22} is positive:

$$p_{22} = \frac{1+\gamma}{2\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)}.$$

From equation (2), we can solve for p_{11}:

$$p_{11} = \gamma p_{22} + \left(\alpha + \frac{\beta}{\epsilon}\right) p_{12}.$$

Substitute the derived values of p_{12} and p_{22}:

$$p_{11} = \gamma\left(\frac{1+\gamma}{2\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)}\right) + \left(\alpha + \frac{\beta}{\epsilon}\right)\left(\frac{1}{2\gamma}\right).$$

Simplifying the expression for p_{11}:

$$p_{11} = \frac{1+\gamma}{2\left(\alpha + \frac{\beta}{\epsilon}\right)} + \frac{\alpha + \frac{\beta}{\epsilon}}{2\gamma}.$$

Given that $\gamma > 0$, $\alpha > 0$, $\beta > 0$, and $\epsilon > 0$, all terms in the expression for p_{11} are positive, which means $p_{11} > 0$.

Thus, we have found the unique symmetric matrix P that solves the Lyapunov equation $A^T P + PA = -I$:

$$P = \begin{pmatrix} \frac{1+\gamma}{2\left(\alpha+\frac{\beta}{\epsilon}\right)} + \frac{\alpha+\frac{\beta}{\epsilon}}{2\gamma} & \frac{1}{2\gamma} \\ \frac{1}{2\gamma} & \frac{1+\gamma}{2\gamma\left(\alpha+\frac{\beta}{\epsilon}\right)} \end{pmatrix}.$$

4. Positive Definiteness of P

For P to be a valid matrix for constructing a Lyapunov function, it must be positive definite. For a symmetric 2×2 matrix, the conditions for positive definiteness are that the leading principal minors are positive: $p_{11} > 0$ and $\det(P) > 0$.

We have already shown that $p_{11} = \frac{1+\gamma}{2\left(\alpha+\frac{\beta}{\epsilon}\right)} + \frac{\alpha+\frac{\beta}{\epsilon}}{2\gamma}$. Since $\gamma > 0$, $\alpha > 0$, $\beta > 0$, and $\epsilon > 0$, all terms are positive, so $p_{11} > 0$.

Now, we compute the determinant of P:

$$\det(P) = p_{11}p_{22} - p_{12}^2.$$

Substituting the derived values of p_{11}, p_{22}, and p_{12}:

$$\det(P) = \left(\frac{1+\gamma}{2\left(\alpha + \frac{\beta}{\epsilon}\right)} + \frac{\alpha + \frac{\beta}{\epsilon}}{2\gamma}\right)\left(\frac{1+\gamma}{2\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)}\right) - \left(\frac{1}{2\gamma}\right)^2.$$

Expanding the first term:

$$\det(P) = \frac{(1+\gamma)^2}{4\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)^2} + \frac{\left(\alpha + \frac{\beta}{\epsilon}\right)(1+\gamma)}{4\gamma^2\left(\alpha + \frac{\beta}{\epsilon}\right)} - \frac{1}{4\gamma^2}.$$

Simplifying the second term:

$$\det(P) = \frac{(1+\gamma)^2}{4\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)^2} + \frac{1+\gamma}{4\gamma^2} - \frac{1}{4\gamma^2}.$$

$$\det(P) = \frac{(1+\gamma)^2}{4\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)^2} + \frac{1+\gamma-1}{4\gamma^2}.$$

$$\det(P) = \frac{(1+\gamma)^2}{4\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)^2} + \frac{\gamma}{4\gamma^2}.$$

Since $\gamma \neq 0$, we can simplify the second term:

$$\det(P) = \frac{(1+\gamma)^2}{4\gamma\left(\alpha + \frac{\beta}{\epsilon}\right)^2} + \frac{1}{4\gamma}.$$

Given that $\gamma > 0$, $\alpha > 0$, $\beta > 0$, and $\epsilon > 0$, both terms in the expression for $\det(P)$ are positive. Therefore, $\det(P) > 0$.

Since $p_{11} > 0$ and $\det(P) > 0$, the matrix P obtained by solving the Lyapunov equation with $Q = I$ is indeed positive definite.

5. Time Derivative of the Lyapunov Function

By solving the Lyapunov equation $A^T P + PA = -Q$ with $Q = I$, the time derivative of the Lyapunov function $V(\mathbf{z})$ along the trajectories of the system $\dot{\mathbf{z}} = A\mathbf{z}$ is directly given by:

$$\dot{V}(\mathbf{z}) = \mathbf{z}^T(A^T P + PA)\mathbf{z} = \mathbf{z}^T(-Q)\mathbf{z}.$$

With $Q = I$, we have:

$$\dot{V}(\mathbf{z}) = -\mathbf{z}^T I\mathbf{z} = -\|\mathbf{z}\|^2.$$

The term $\|\mathbf{z}\|^2$ represents the squared Euclidean norm of the state vector $\mathbf{z}$. It is positive for any $\mathbf{z} \neq \mathbf{0}$ and zero only when $\mathbf{z} = \mathbf{0}$. Therefore, $\dot{V}(\mathbf{z})$ is negative definite: $\dot{V}(\mathbf{z}) < 0$ for all $\mathbf{z} \neq \mathbf{0}$ and $\dot{V}(\mathbf{0}) = 0$.

6. Conclusion

We have successfully applied Lyapunov's Direct Method to analyze the stability of the equilibrium point $\mathbf{z} = \mathbf{0}$ for the linearized seesaw system $\dot{\mathbf{z}} = A\mathbf{z}$. We found a quadratic Lyapunov function candidate $V(\mathbf{z}) = \mathbf{z}^T P\mathbf{z}$ by solving the Lyapunov equation $A^T P + PA = -I$ for the matrix P.

The matrix P was found to be positive definite under the physical assumptions that lead to $\gamma > 0$, $\alpha > 0$, and $\beta > 0$ (i.e., $m_2 > m_1$, positive viscous damping, and non-zero Coulomb friction effects). This ensures that $V(\mathbf{z})$ is positive definite ($V(\mathbf{z}) > 0$ for $\mathbf{z} \neq \mathbf{0}$ and $V(\mathbf{0}) = 0$). Furthermore, quadratic forms with a positive definite matrix P are radially unbounded, meaning $V(\mathbf{z}) \to \infty$ as $\|\mathbf{z}\| \to \infty$.

The time derivative of $V(\mathbf{z})$ along the system trajectories was found to be $\dot{V}(\mathbf{z}) = -\|\mathbf{z}\|^2$, which is negative definite ($\dot{V}(\mathbf{z}) < 0$ for $\mathbf{z} \neq \mathbf{0}$ and $\dot{V}(\mathbf{0}) = 0$).

According to Lyapunov's Direct Method, if there exists a positive definite, radially unbounded Lyapunov function whose derivative is negative definite, then the equilibrium point is globally asymptotically stable. In this case, the equilibrium $\mathbf{z} = \mathbf{0}$ of the linearized system is globally asymptotically stable.

The global asymptotic stability of the linearized system at $\mathbf{z} = \mathbf{0}$ implies that the corresponding equilibrium point of the original nonlinear system, $\theta_{\text{eq}} = \frac{\pi}{2}$, is locally asymptotically stable.

Okay, here is the explanation of LQR control applied specifically to the linearized seesaw model, following your formatting rules and excluding the concepts of Jacobi-Bellman and detectability.

LQR Control Applied to the Linearized Seesaw System

1. Linearized Seesaw System Model

As derived previously, the linearized dynamics of the seesaw around a stable equilibrium point (e.g., $\theta_{\text{eq}} = \pi/2$ with $m_2 > m_1$, where $\gamma > 0$) are given by the state-space equation:

$$\dot{\mathbf{z}}(t) = A\mathbf{z}(t) + B\Delta F(t)$$

where $\mathbf{z}(t) \in \mathbb{R}^2$ is the state vector representing the deviation from the equilibrium, $\Delta F(t) \in \mathbb{R}$ is the deviation in the control input from its equilibrium value F_{eq}, A is the system matrix (2×2), and B is the input matrix (2×1).

The state vector is $\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \theta - \theta_{\text{eq}} \\ \dot{\theta} \end{pmatrix}$.

The matrices A and B (evaluated at the equilibrium with $F_{\text{eq}} = 0$ and $\theta_{\text{eq}} = \pi/2$) are:

$$A = \begin{pmatrix} 0 & 1 \\ -\gamma & -\alpha - \frac{\beta}{\epsilon} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \delta \end{pmatrix}$$

The parameters are $\gamma = \frac{(m_2 - m_1)gL}{2I}$, $\alpha = \frac{b}{I}$, $\beta = \frac{\mu_c g L}{2I}$, $\delta = \frac{L}{2I}$, and $\epsilon > 0$. For the equilibrium at $\theta_{\text{eq}} = \pi/2$ to be relevant for LQR design assuming linear stability (which LQR will optimize), we require $\gamma > 0$, $\alpha > 0$, $\beta > 0$, and $\epsilon > 0$.

Part 1: Continuous-Time Infinite-Horizon LQR

The goal is to find a control input deviation $\Delta F(t)$ that stabilizes the linearized system $\dot{\mathbf{z}} = A\mathbf{z} + B\Delta F$ and minimizes a quadratic cost function over an infinite time horizon.

1. Performance Index

The cost function penalizes deviations in the state vector $\mathbf{z}$ and the control input deviation ΔF:

$$J = \int_0^\infty \big(\mathbf{z}(t)^T Q \mathbf{z}(t) + \Delta F(t)^T R \Delta F(t)\big)\, dt$$

where:

- Q is a 2×2 symmetric, positive semi-definite matrix ($Q \geq 0$). This matrix weights the importance of keeping the state deviations $z_1 = \Delta\theta$ and $z_2 = \Delta\dot{\theta}$ close to zero. For example, $Q = \begin{pmatrix} q_{11} & 0 \\ 0 & q_{22} \end{pmatrix}$ with $q_{11}, q_{22} \geq 0$ would penalize angular position and velocity deviations.
- R is a 1×1 symmetric, positive definite matrix ($R > 0$). Since ΔF is a scalar input, R is a positive scalar, often denoted as R. It penalizes the magnitude of the control input deviation ΔF. A larger R means control effort is more penalized, resulting in smaller control actions.

The LQR problem seeks the control law $\Delta F(t)$ that minimizes J subject to $\dot{\mathbf{z}} = A\mathbf{z} + B\Delta F$. The solution is a linear state feedback control law:

$$\Delta F(t) = -K\mathbf{z}(t)$$

where K is the constant feedback gain matrix (1×2).

2. The Algebraic Riccati Equation (ARE)

The matrix P (2×2) that determines the optimal control gain and relates to the minimum cost satisfies the **Algebraic Riccati Equation (ARE)**:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

For the infinite-horizon LQR problem with a stabilizable system pair (A, B), there exists a unique symmetric, positive semi-definite solution P to the ARE that stabilizes the closed-loop system $A - BK$.

3. Solving the ARE for the Seesaw

Substitute the matrices A and B for the seesaw into the ARE. Let R be a scalar $r > 0$, so $R^{-1} = 1/r$.

$$\begin{pmatrix} 0 & -\gamma \\ 1 & -\alpha - \frac{\beta}{\epsilon} \end{pmatrix} P + P \begin{pmatrix} 0 & 1 \\ -\gamma & -\alpha - \frac{\beta}{\epsilon} \end{pmatrix} - P \begin{pmatrix} 0 \\ \delta \end{pmatrix} \frac{1}{r} \begin{pmatrix} 0 & \delta \end{pmatrix} P + Q = 0$$

$$(A^TP + PA) - \frac{\delta^2}{r} P \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} P + Q = 0$$

Let $P = \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix}$. The term $P \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} P$ becomes:

$$P \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} P = \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} = \begin{pmatrix} 0 & p_{12} \\ 0 & p_{22} \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} = \begin{pmatrix} p_{12}^2 & p_{12}p_{22} \\ p_{12}p_{22} & p_{22}^2 \end{pmatrix}$$

The $(A^TP + PA)$ term was calculated in the Lyapunov analysis:

$$A^TP + PA = \begin{pmatrix} -2\gamma p_{12} & p_{11} - \gamma p_{22} - (\alpha + \beta/\epsilon)p_{12} \\ p_{11} - \gamma p_{22} - (\alpha + \beta/\epsilon)p_{12} & 2(p_{12} - (\alpha + \beta/\epsilon)p_{22}) \end{pmatrix}$$

Let $\tilde{\alpha} = \alpha + \beta/\epsilon$. The ARE becomes:

$$\begin{pmatrix} -2\gamma p_{12} & p_{11} - \gamma p_{22} - \tilde{\alpha} p_{12} \\ p_{11} - \gamma p_{22} - \tilde{\alpha} p_{12} & 2(p_{12} - \tilde{\alpha} p_{22}) \end{pmatrix} - \frac{\delta^2}{r} \begin{pmatrix} p_{12}^2 & p_{12}p_{22} \\ p_{12}p_{22} & p_{22}^2 \end{pmatrix} + Q = 0$$

This is a system of coupled nonlinear algebraic equations in terms of p_{11}, p_{12}, p_{22} and the elements of Q. This system must be solved to find the P matrix. Using the command LQR in Octave/MATLAB.

4. Optimal Gain Calculation

Once the correct solution P is found, the optimal feedback gain matrix $K = (k_1 \quad k_2)$ is computed:

$$K = R^{-1}B^TP = \frac{1}{r}(0 \quad \delta) \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} = \frac{1}{r}(\delta p_{12} \quad \delta p_{22})$$

So, the feedback gains are $k_1 = \frac{\delta p_{12}}{r}$ and $k_2 = \frac{\delta p_{22}}{r}$. The optimal control input deviation is $\Delta F(t) = -k_1 z_1(t) - k_2 z_2(t)$.

5. Stability

The positive semi-definite solution P to the ARE, when it exists for a stabilizable system pair (A, B), guarantees that the closed-loop system $\dot{\mathbf{z}} = (A - BK)\mathbf{z}$ is asymptotically stable. The quadratic function $V(\mathbf{z}) = \mathbf{z}^T P \mathbf{z}$ serves as a Lyapunov function for the closed-loop system.

The derivative of V along the closed-loop trajectories is:

$$\dot{V}(\mathbf{z}) = \mathbf{z}^T(A^TP + PA)\mathbf{z} + \mathbf{z}^T PB\Delta F + \Delta F^T B^T P\mathbf{z}$$

Substitute $\Delta F = -K\mathbf{z} = -R^{-1}B^TP\mathbf{z}$:

$$\dot{V}(\mathbf{z}) = \mathbf{z}^T(A^TP + PA)\mathbf{z} - \mathbf{z}^T PBR^{-1}B^TP\mathbf{z} - (R^{-1}B^TP\mathbf{z})^T B^T P\mathbf{z}$$

$$\dot{V}(\mathbf{z}) = \mathbf{z}^T(A^TP + PA)\mathbf{z} - \mathbf{z}^TPBR^{-1}B^TP\mathbf{z} - \mathbf{z}^TPBR^{-1}B^TP\mathbf{z}$$

$$\dot{V}(\mathbf{z}) = \mathbf{z}^T(A^TP + PA - 2PBR^{-1}B^TP)\mathbf{z}$$

From the ARE ($A^TP + PA = PBR^{-1}B^TP - Q$), substitute $A^TP + PA$:

$$\dot{V}(\mathbf{z}) = \mathbf{z}^T(PBR^{-1}B^TP - Q - 2PBR^{-1}B^TP)\mathbf{z}$$

$$\dot{V}(\mathbf{z}) = \mathbf{z}^T(-Q - PBR^{-1}B^TP)\mathbf{z} = -\mathbf{z}^TQ\mathbf{z} - \mathbf{z}^TPBR^{-1}B^TP\mathbf{z}$$

Since $Q \geq 0$ and $R^{-1} > 0$, $PBR^{-1}B^TP = (B^TP)^TR^{-1}(B^TP) \geq 0$. Thus, $\dot{V}(\mathbf{z}) \leq 0$ for all $\mathbf{z}$. The conditions for existence of the stabilizing P (system stabilizability) ensure that $\dot{V}(\mathbf{z}) = 0$ only when $\mathbf{z} = \mathbf{0}$, proving asymptotic stability.

Step-by-step for Continuous-Time LQR for the Seesaw:

1. Identify the linearized system matrices A and B.
2. Choose weighting matrices $Q \geq 0$ (2×2) and $R > 0$ (scalar r) to define the trade-off between state deviation penalties (z_1, z_2) and control effort penalty (ΔF).
3. Solve the Algebraic Riccati Equation $A^TP + PA - PBR^{-1}B^TP + Q = 0$ for the unique symmetric, positive semi-definite matrix P.
4. Compute the optimal constant feedback gain $K = R^{-1}B^TP = \frac{1}{r}(\delta p_{12} \quad \delta p_{22})$.
5. Implement the control law $\Delta F(t) = -K\mathbf{z}(t) = -k_1(\theta - \theta_{\text{eq}}) - k_2\dot{\theta}$. The total control input is then $F(t) = F_{\text{eq}} + \Delta F(t)$.

Part 2: Discrete-Time Finite-Horizon LQR

For implementing the LQR controller on a digital system, we need a discrete-time model and a discrete-time LQR design. We consider a finite time horizon N.

1. Discrete-Time Seesaw Model (using ZOH)

Assuming the control input deviation $\Delta F(t)$ is held constant at ΔF_k over each sampling interval $[t_k, t_{k+1})$ of duration T_s, the linearized seesaw system discretizes to:

$$\mathbf{z}_{k+1} = G\mathbf{z}_k + H\Delta F_k$$

where $\mathbf{z}_k = \mathbf{z}(t_k)$, $\Delta F_k = \Delta F(t_k)$, and the matrices G (2×2) and H (2×1) are computed from the continuous-time matrices A, B, and the sampling period T_s:

$$G = e^{AT_s}, \quad H = \int_0^{T_s} e^{A\tau}Bd\tau$$

These matrices G and H are specific to the seesaw model defined by its parameters $\gamma, \alpha, \beta, \epsilon, \delta$.

2. Discrete-Time Performance Index

The objective is to minimize a sum of quadratic costs over a finite horizon of N steps, from $k = 0$ to $k = N - 1$:

$$J_N = \sum_{k=0}^{N-1} (\mathbf{z}_k^T Q_d \mathbf{z}_k + \Delta F_k^T R_d \Delta F_k) + \mathbf{z}_N^T Q_f \mathbf{z}_N$$

where:

- $Q_d \geq 0$ (2×2) penalizes state deviations at each step.
- $R_d > 0$ (scalar) penalizes control input deviations at each step.
- $Q_f \geq 0$ (2×2) penalizes the final state deviation at the end of the horizon.

The problem is to find the sequence of control input deviations $\Delta F_0, \Delta F_1, \ldots, \Delta F_{N-1}$ that minimizes J_N subject to $\mathbf{z}_{k+1} = G\mathbf{z}_k + H\Delta F_k$, given $\mathbf{z}_0$.

3. Iterative Solution via Riccati Difference Equation

The optimal solution to this finite-horizon problem is a time-varying linear feedback control law:

$$\Delta F_k = -K_k \mathbf{z}_k$$

where K_k is the feedback gain matrix (1×2) at step k. The matrices P_k (2×2), representing the future optimal cost from step k, satisfy the **Riccati Difference Equation (RDE)**:

$$P_k = G^T P_{k+1} G - G^T P_{k+1} H (R_d + H^T P_{k+1} H)^{-1} H^T P_{k+1} G + Q_d$$

This equation is solved *backward* in time, starting from the terminal cost defined by Q_f.

3.1 Bellman's Principle and Backward Computation (Intuition)

The backward computation inherent in solving the RDE stems directly from **Bellman's Principle of Optimality**.

Bellman's Principle of Optimality: Simply put, it states that if you have found the optimal sequence of decisions (control inputs ΔF_k) to get from a starting point to an endpoint while minimizing some cost, then any *sub-sequence* of that optimal sequence must also be optimal for the corresponding sub-problem.

For example, if the optimal path from state $\mathbf{z}_0$ to $\mathbf{z}_N$ (minimizing J_N) involves passing through state $\mathbf{z}_k$ at time k, then the part of the path from $\mathbf{z}_k$ to $\mathbf{z}_N$ must be the optimal path for minimizing the cost *starting* from state $\mathbf{z}_k$ at time k.

Why Backward Computation?
Consider the decision we need to make at time step k: choosing the control input ΔF_k. This choice affects the next state $\mathbf{z}_{k+1} = G\mathbf{z}_k + H\Delta F_k$. The total cost J_N depends not only on the

immediate cost incurred at step k ($\mathbf{z}_k^T Q_d \mathbf{z}_k + \Delta F_k^T R_d \Delta F_k$) but also on *all future costs* starting from $\mathbf{z}_{k+1}$.

According to Bellman's principle, to make the optimal choice for ΔF_k, we must assume that all *future* decisions ($\Delta F_{k+1}, \ldots, \Delta F_{N-1}$) will also be made optimally, given the state $\mathbf{z}_{k+1}$ that results from our choice of ΔF_k. Therefore, the decision at step k depends on knowing the *minimum possible future cost* starting from step $k+1$.

This creates a dependency chain working backward from the end:

1. To find the optimal ΔF_{N-1}, we only need to consider the immediate cost at $N-1$ and the final state cost at N, because there are no future decisions. The "future cost" from state $\mathbf{z}_N$ is simply $\mathbf{z}_N^T Q_f \mathbf{z}_N$.
2. To find the optimal ΔF_{N-2}, we need to consider the immediate cost at $N-2$ and the *minimum possible cost* from step $N-1$ onwards (which we found in step 1).
3. To find the optimal ΔF_k, we need to consider the immediate cost at k and the *minimum possible cost* from step $k+1$ onwards (which depends on knowing the optimal policy from $k+1$ to N).

This process naturally starts at the final step N, where the future cost is explicitly defined by Q_f, and iterates backward to determine the optimal policy (K_k) and minimum cost (P_k) at each preceding step.

3.2 Rigorous Derivation of the RDE using Bellman's Principle

Let $J_k^*(\mathbf{z}_k)$ denote the minimum cost-to-go from time step k to the final time N, starting in state $\mathbf{z}_k$.

$$J_k^*(\mathbf{z}_k) = \min_{\Delta F_k, \ldots, \Delta F_{N-1}} \left[\sum_{i=k}^{N-1} (\mathbf{z}_i^T Q_d \mathbf{z}_i + \Delta F_i^T R_d \Delta F_i) + \mathbf{z}_N^T Q_f \mathbf{z}_N \right]$$

subject to $\mathbf{z}_{i+1} = G\mathbf{z}_i + H\Delta F_i$.

The cost at the final step N is simply the terminal penalty:

$$J_N^*(\mathbf{z}_N) = \mathbf{z}_N^T Q_f \mathbf{z}_N$$

We assume (and can prove by induction) that the optimal cost-to-go is a quadratic form in the state:

$$J_k^*(\mathbf{z}_k) = \mathbf{z}_k^T P_k \mathbf{z}_k$$

where P_k is a symmetric positive semi-definite matrix. For $k = N$, we have $P_N = Q_f$.

Now, apply Bellman's Principle for step $k < N$. The minimum cost from step k is achieved by choosing the current control ΔF_k optimally, considering the immediate cost and the minimum future cost from step $k+1$:

$$J_k^*(\mathbf{z}_k) = \min_{\Delta F_k} \left[\underbrace{\mathbf{z}_k^T Q_d \mathbf{z}_k + \Delta F_k^T R_d \Delta F_k}_{\text{Cost at step k}} + \underbrace{J_{k+1}^*(\mathbf{z}_{k+1})}_{\text{Min future cost from k+1}} \right]$$

Substitute the quadratic form assumption and the system dynamics $\mathbf{z}_{k+1} = G\mathbf{z}_k + H\Delta F_k$:

$$\mathbf{z}_k^T P_k \mathbf{z}_k = \min_{\Delta F_k} \left[\mathbf{z}_k^T Q_d \mathbf{z}_k + \Delta F_k^T R_d \Delta F_k + (G\mathbf{z}_k + H\Delta F_k)^T P_{k+1} (G\mathbf{z}_k + H\Delta F_k) \right]$$

Let $\mathcal{L}(\Delta F_k)$ be the expression inside the minimization:

$$\mathcal{L}(\Delta F_k) = \mathbf{z}_k^T Q_d \mathbf{z}_k + \Delta F_k^T R_d \Delta F_k + \mathbf{z}_k^T G^T P_{k+1} G \mathbf{z}_k + \mathbf{z}_k^T G^T P_{k+1} H \Delta F_k + \Delta F_k^T H^T P_{k+1} G \mathbf{z}_k + \Delta F_k^T H^T$$

This is a quadratic function of ΔF_k. To find the minimum, we take the gradient with respect to ΔF_k and set it to zero (assuming $R_d + H^T P_{k+1} H$ is invertible, which holds if $R_d > 0$):

$$\frac{\partial \mathcal{L}}{\partial \Delta F_k} = 2R_d \Delta F_k + 2H^T P_{k+1} G \mathbf{z}_k + 2H^T P_{k+1} H \Delta F_k = 0$$

$$(R_d + H^T P_{k+1} H) \Delta F_k = -H^T P_{k+1} G \mathbf{z}_k$$

The optimal control input at step k, denoted ΔF_k^*, is therefore:

$$\Delta F_k^* = -(R_d + H^T P_{k+1} H)^{-1} H^T P_{k+1} G \mathbf{z}_k$$

This is a linear state feedback law $\Delta F_k^* = -K_k \mathbf{z}_k$, where the optimal gain matrix at step k is:

$$K_k = (R_d + H^T P_{k+1} H)^{-1} H^T P_{k+1} G$$

Now, substitute ΔF_k^* back into the expression for $\mathbf{z}_k^T P_k \mathbf{z}_k = \mathcal{L}(\Delta F_k^*)$:

$$\mathbf{z}_k^T P_k \mathbf{z}_k = \mathbf{z}_k^T Q_d \mathbf{z}_k + (\Delta F_k^*)^T R_d \Delta F_k^* + (G\mathbf{z}_k + H\Delta F_k^*)^T P_{k+1} (G\mathbf{z}_k + H\Delta F_k^*)$$

Substitute $\Delta F_k^* = -K_k \mathbf{z}_k$:

$$\mathbf{z}_k^T P_k \mathbf{z}_k = \mathbf{z}_k^T Q_d \mathbf{z}_k + (-K_k \mathbf{z}_k)^T R_d (-K_k \mathbf{z}_k) + (G\mathbf{z}_k - HK_k \mathbf{z}_k)^T P_{k+1} (G\mathbf{z}_k - HK_k \mathbf{z}_k)$$

$$\mathbf{z}_k^T P_k \mathbf{z}_k = \mathbf{z}_k^T Q_d \mathbf{z}_k + \mathbf{z}_k^T K_k^T R_d K_k \mathbf{z}_k + \mathbf{z}_k^T (G - HK_k)^T P_{k+1} (G - HK_k) \mathbf{z}_k$$

Since this equality must hold for any state $\mathbf{z}_k$, we can equate the matrices:

$$P_k = Q_d + K_k^T R_d K_k + (G - HK_k)^T P_{k+1} (G - HK_k)$$

Expanding the last term:

$$P_k = Q_d + K_k^T R_d K_k + (G^T - K_k^T H^T) P_{k+1} (G - HK_k)$$

$$P_k = Q_d + K_k^T R_d K_k + G^T P_{k+1} G - G^T P_{k+1} H K_k - K_k^T H^T P_{k+1} G + K_k^T H^T P_{k+1} H K_k$$

$$P_k = Q_d + G^T P_{k+1} G - G^T P_{k+1} H K_k - K_k^T H^T P_{k+1} G + K_k^T (R_d + H^T P_{k+1} H) K_k$$

Now substitute the expression defining K_k: $(R_d + H^T P_{k+1} H) K_k = H^T P_{k+1} G$.

$$P_k = Q_d + G^T P_{k+1} G - G^T P_{k+1} H K_k - K_k^T H^T P_{k+1} G + K_k^T (H^T P_{k+1} G)$$

The last two terms cancel: $K_k^T H^T P_{k+1} G$.

$$P_k = Q_d + G^T P_{k+1} G - G^T P_{k+1} H K_k$$

Finally, substitute the full expression for K_k:

$$P_k = Q_d + G^T P_{k+1} G - G^T P_{k+1} H (R_d + H^T P_{k+1} H)^{-1} H^T P_{k+1} G$$

This is precisely the Riccati Difference Equation (RDE) presented earlier. It defines how to compute P_k from P_{k+1}, starting with the known terminal condition $P_N = Q_f$ and iterating backward in time.

4. Backward Iteration

The RDE is solved by starting at the final time step $k = N$ and iterating backward to $k = 0$.

- **Terminal Condition:** At $k = N$, the remaining cost is only the final state penalty, so $P_N = Q_f$.
- **Iteration:** Calculate P_k and K_k for $k = N-1, N-2, \ldots, 0$ using the following steps:
  1. Compute P_k using the RDE with the previously computed P_{k+1}:

  $$P_k = G^T P_{k+1} G - G^T P_{k+1} H (R_d + H^T P_{k+1} H)^{-1} H^T P_{k+1} G + Q_d$$

  2. Compute the time-varying gain K_k using P_{k+1}:

  $$K_k = (R_d + H^T P_{k+1} H)^{-1} H^T P_{k+1} G$$

This process yields a sequence of gain matrices $K_0, K_1, \ldots, K_{N-1}$.

Step-by-step for Discrete-Time Finite-Horizon LQR for the Seesaw:

1. Identify the continuous-time matrices A and B for the linearized seesaw.
2. Choose the sampling period T_s and the finite horizon length N.
3. Compute the discrete-time system matrices $G = e^{AT_s}$ and $H = \int_0^{T_s} e^{A\tau} B d\tau$.
4. Choose weighting matrices $Q_d \geq 0$, $R_d > 0$ (scalar), and $Q_f \geq 0$.
5. Set the terminal condition for the RDE: $P_N = Q_f$.
6. Iterate backward from $k = N-1$ down to 0. In each iteration, calculate P_k using the RDE and $K_k = (R_d + H^T P_{k+1} H)^{-1} H^T P_{k+1} G$.
7. Store the sequence of gains $K_0, K_1, \ldots, K_{N-1}$.

8. Implement the control law online: at each step k, measure or estimate the state $\mathbf{z}_k$ and apply the control input deviation $\Delta F_k = -K_k \mathbf{z}_k$. The total control input is $F_k = F_{\text{eq}} + \Delta F_k$.

For infinite-horizon discrete-time LQR, the RDE iteration converges to a constant matrix P_∞ and a constant gain K_∞ if the discrete-time system (G, H) is stabilizable. This P_∞ satisfies the discrete-time Algebraic Riccati Equation (DARE).

Conclusion

LQR provides a systematic way to design optimal controllers for the linearized seesaw system by minimizing a quadratic performance index. The continuous-time infinite-horizon approach yields a constant feedback gain by solving the ARE, guaranteeing asymptotic stability. The discrete-time finite-horizon approach, suitable for digital implementation, yields a time-varying sequence of gains by solving the RDE backward in time, a process derived directly from Bellman's Principle of Optimality. Both methods require selecting appropriate weighting matrices (Q, R or Q_d, R_d, Q_f) to tune the controller's performance and control effort.