TU/e
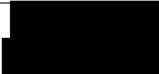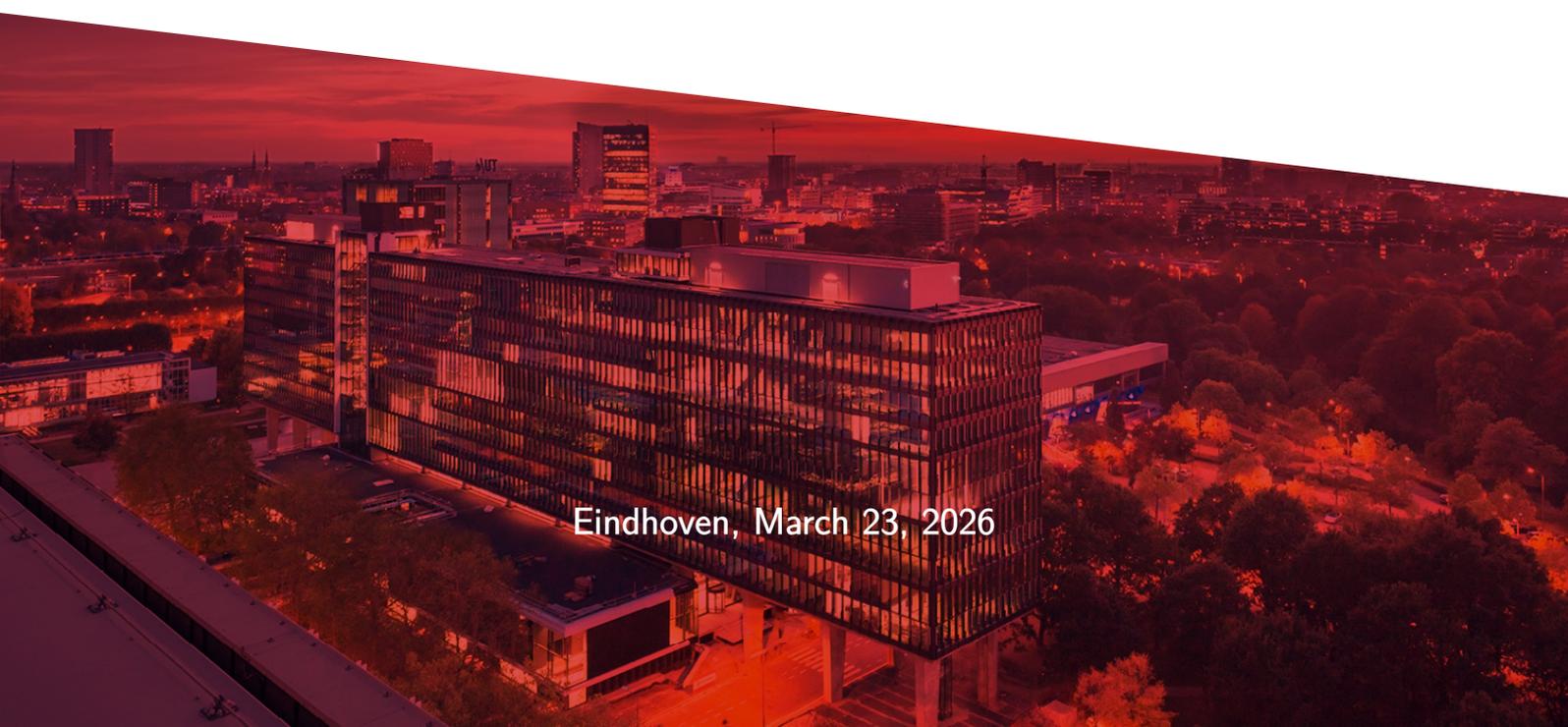
Assignment 2

4DM10, Multibody and Nonlinear Dynamics - Q2 (2025)

Group 17

Full Name	Student ID
Tobi Wejbora	

Eindhoven, March 23, 2026

1 | Problem 1:

Question 1.1

Position Vectors

The position of CM_1 and P in the fixed frame $\vec{e}^0$:

$$\vec{r}_{CM_1} = (R_o\theta)\vec{e}_1^0 + R_o\vec{e}_2^0 \tag{1.1}$$

$$\vec{r}_P = \vec{r}_{CM_1} + \vec{r}_{P/CM_1} = \vec{r}_{CM_1} + \ell\vec{e}_1^1 \tag{1.2}$$

$$\vec{e}_1^1 = \cos(\theta)\vec{e}_1^0 + \sin(\theta)\vec{e}_2^0 \tag{1.3}$$

$$\vec{r}_P = (R_o\theta + \ell\cos\theta)\vec{e}_1^0 + (R_o + \ell\sin\theta)\vec{e}_2^0 \tag{1.4}$$

Velocity Vectors

Simplified velocity of CM_1 and P:

$$\dot{\vec{r}}_{CM_1} = R_o\dot{\theta}\vec{e}_1^0 \tag{1.5}$$

$$\dot{\vec{r}}_P = \frac{d}{dt}\vec{r}_P = (R_o\dot{\theta} + \dot{\ell}\cos\theta - \ell\dot{\theta}\sin\theta)\vec{e}_1^0 + (\dot{\ell}\sin\theta + \ell\dot{\theta}\cos\theta)\vec{e}_2^0 \tag{1.6}$$

Kinetic Energy of the Ring

The ring has both translational and rotational kinetic energy. The moment of inertia of a thin ring about its center is:

$$J_{ring} = M\frac{R_i^2 + R_o^2}{2} \tag{1.7}$$

The kinetic energy of the ring:

$$T_{ring} = \frac{1}{2}M\dot{\vec{r}}_{CM_1} \cdot \dot{\vec{r}}_{CM_1} + \frac{1}{2}J_{ring}\dot{\theta}^2 \tag{1.8}$$

$$T_{ring} = \frac{1}{2}MR_o^2\dot{\theta}^2 + \frac{1}{2} \cdot M\frac{R_i^2 + R_o^2}{2}\dot{\theta}^2 \tag{1.9}$$

$$T_{ring} = \frac{1}{2}M\left(R_o^2 + \frac{R_i^2 + R_o^2}{2}\right)\dot{\theta}^2 = \frac{1}{4}M(3R_o^2 + R_i^2)\dot{\theta}^2 \tag{1.10}$$

Kinetic Energy of the Point Mass

$$T_m = \frac{1}{2}m\dot{\vec{r}}_P \cdot \dot{\vec{r}}_P \tag{1.11}$$

$$\dot{\vec{r}}_P \cdot \dot{\vec{r}}_P = (\dot{\vec{r}}_{CM_1} + \dot{\ell}\vec{e}_1^1 + \ell\dot{\theta}\vec{e}_2^1) \cdot (\dot{\vec{r}}_{CM_1} + \dot{\ell}\vec{e}_1^1 + \ell\dot{\theta}\vec{e}_2^1) \tag{1.12}$$

$$= R_o^2\dot{\theta}^2 + \dot{\ell}^2 + \ell^2\dot{\theta}^2 + 2R_o\dot{\theta}\dot{\ell}(\vec{e}_1^0 \cdot \vec{e}_1^1) + 2R_o\ell\dot{\theta}^2(\vec{e}_1^0 \cdot \vec{e}_2^1) \tag{1.13}$$

Using $\vec{e}_1^0 \cdot \vec{e}_1^1 = \cos\theta$ and $\vec{e}_1^0 \cdot \vec{e}_2^1 = -\sin\theta$:

$$\dot{\vec{r}}_P \cdot \dot{\vec{r}}_P = R_o^2\dot{\theta}^2 + \dot{\ell}^2 + \ell^2\dot{\theta}^2 + 2R_o\dot{\theta}\dot{\ell}\cos\theta - 2R_o\ell\dot{\theta}^2\sin\theta \tag{1.14}$$

$$T_m = \frac{1}{2}m\left(R_o^2\dot{\theta}^2 + \dot{\ell}^2 + \ell^2\dot{\theta}^2 + 2R_o\dot{\theta}\dot{\ell}\cos\theta - 2R_o\ell\dot{\theta}^2\sin\theta\right) \tag{1.15}$$

Total Kinetic Energy

$$T = T_{ring} + T_m = \frac{1}{4}M(3R_o^2 + R_i^2)\dot{\theta}^2 + \frac{1}{2}m\left((R_o^2 + \ell^2 - 2R_o\ell\sin\theta)\dot{\theta}^2 + \dot{\ell}^2 + 2R_o\dot{\ell}\dot{\theta}\cos\theta\right) \tag{1.16}$$

Gravitational Potential Energy

The gravitational potential energy is defined with respect to the fixed frame $\vec{\underline{e}}^0$ with $\vec{g} = -g\vec{e}_2^0$. For the ring (center of mass at height R_o) and point mass P:

$$V_{g,ring} = MgR_o \tag{1.17}$$

$$V_{g,m} = mg(R_o + \ell\sin\theta) \tag{1.18}$$

Spring Potential Energy

Each spring has stiffness k and rest length R_i. The two springs connect the point mass to the ring at attachment points A_1 and A_2.
Looking at the geometry in Figure 1:

- When $\ell = 0$, the point mass is at the center, and each spring has length R_i (rest length) $\Rightarrow$ no spring force

- When $\ell \neq 0$, one spirng streches, the other compresses

The total spring potential energy:

$$V_{spring} = \frac{1}{2}k(R_i + \ell - R_i)^2 + \frac{1}{2}k(R_i - \ell - R_i)^2 = \frac{1}{2}k\ell^2 + \frac{1}{2}k\ell^2 = k\ell^2 \tag{1.19}$$

Total Potential Energy

$$V = MgR_o + mg(R_o + \ell\sin\theta) + k\ell^2 = (M+m)gR_o + mg\ell\sin\theta + k\ell^2 \tag{1.20}$$

Generalized Non-Conservative Forces

The actuator applies force $\vec{F} = F\vec{e}_1^1$ on the point mass P. The reaction forces on the ring are each $-\frac{1}{2}F\vec{e}_1^1$ at points A_1 and A_2.
Using the virtual work principle and computing the partial derivatives:

$$Q^{nc} = \left(\frac{\partial\vec{r}_P}{\partial q}\right)^\top \cdot \vec{F} \tag{1.21}$$

$$\frac{\partial\vec{r}_P}{\partial\theta} = R_o\vec{e}_1^0 - \ell\sin\theta\vec{e}_1^0 + \ell\cos\theta\vec{e}_2^0 = R_o\vec{e}_1^0 + \ell\vec{e}_2^1 \tag{1.22}$$

$$\frac{\partial\vec{r}_P}{\partial\ell} = \cos\theta\vec{e}_1^0 + \sin\theta\vec{e}_2^0 = \vec{e}_1^1 \tag{1.23}$$

$$Q_\theta^{nc} = \frac{\partial\vec{r}_P}{\partial\theta} \cdot F\vec{e}_1^1 = (R_o\vec{e}_1^0 + \ell\vec{e}_2^1) \cdot F\vec{e}_1^1 = FR_o\cos\theta \tag{1.24}$$

$$Q_\ell^{nc} = \frac{\partial\vec{r}_P}{\partial\ell} \cdot F\vec{e}_1^1 = \vec{e}_1^1 \cdot F\vec{e}_1^1 = F \tag{1.25}$$

$$Q^{nc} = \begin{bmatrix} Q_\theta^{nc} \\ Q_\ell^{nc} \end{bmatrix} = \begin{bmatrix} FR_o\cos\theta \\ F \end{bmatrix} \tag{1.26}$$

Partial Derivatives of Kinetic Energy

Let us define for convenience:

$$J_{eff} = \frac{M(3R_o^2 + R_i^2)}{2} + m(R_o^2 + \ell^2 - 2R_o\ell\sin\theta) \tag{1.27}$$

Then:

$$T = \frac{1}{2}J_{eff}\dot{\theta}^2 + \frac{1}{2}m\dot{\ell}^2 + mR_o\dot{\ell}\dot{\theta}\cos\theta \tag{1.28}$$

Computing the required partial derivatives:

$$\frac{\partial T}{\partial \dot{\theta}} = J_{eff}\dot{\theta} + mR_o\dot{\ell}\cos\theta \tag{1.29}$$

$$\frac{\partial T}{\partial \theta} = -mR_o\ell\dot{\theta}^2\cos\theta - mR_o\dot{\ell}\dot{\theta}\sin\theta \tag{1.30}$$

$$\frac{\partial T}{\partial \dot{\ell}} = m\dot{\ell} + mR_o\dot{\theta}\cos\theta \tag{1.31}$$

$$\frac{\partial T}{\partial \ell} = m\ell\dot{\theta}^2 - mR_o\dot{\theta}^2\sin\theta \tag{1.32}$$

Time derivatives:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) = J_{eff}\ddot{\theta} + 2m(\ell - R_o\sin\theta)\dot{\ell}\dot{\theta} - 2mR_o\ell\cos\theta\dot{\theta}^2 \tag{1.33}$$

$$+ mR_o\ddot{\ell}\cos\theta - mR_o\dot{\ell}\dot{\theta}\sin\theta \tag{1.34}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\ell}}\right) = m\ddot{\ell} + mR_o\ddot{\theta}\cos\theta - mR_o\dot{\theta}^2\sin\theta \tag{1.35}$$

Partial Derivatives of Potential Energy

$$\frac{\partial V}{\partial \theta} = mg\ell\cos\theta \tag{1.36}$$

$$\frac{\partial V}{\partial \ell} = mg\sin\theta + 2k\ell \tag{1.37}$$

Equations of Motion

Equation for θ:

$$J_{eff}\ddot{\theta} + 2m(\ell - R_o\sin\theta)\dot{\ell}\dot{\theta} - 2mR_o\ell\cos\theta\dot{\theta}^2 + mR_o\ddot{\ell}\cos\theta - mR_o\dot{\ell}\dot{\theta}\sin\theta \tag{1.38}$$

$$+ mR_o\ell\dot{\theta}^2\cos\theta + mR_o\dot{\ell}\dot{\theta}\sin\theta + mg\ell\cos\theta = FR_o\cos\theta \tag{1.39}$$

$$J_{eff}\ddot{\theta} + 2m\ell\dot{\ell}\dot{\theta} - 2mR_o\sin\theta\dot{\ell}\dot{\theta} - mR_o\ell\cos\theta\dot{\theta}^2 + mR_o\ddot{\ell}\cos\theta + mg\ell\cos\theta = FR_o\cos\theta \tag{1.40}$$

Equation for ℓ:

$$m\ddot{\ell} + mR_o\ddot{\theta}\cos\theta - mR_o\dot{\theta}^2\sin\theta - m\ell\dot{\theta}^2 + mR_o\dot{\theta}^2\sin\theta + mg\sin\theta + 2k\ell = F \tag{1.41}$$

$$m\ddot{\ell} + mR_o\ddot{\theta}\cos\theta - m\ell\dot{\theta}^2 + mg\sin\theta + 2k\ell = F \tag{1.42}$$

$$\left[\frac{M(3R_o^2 + R_i^2)}{2} + m(R_o^2 + \ell^2 - 2R_o\ell\sin\theta)\right]\ddot{\theta} + 2m(\ell - R_o\sin\theta)\dot{\ell}\dot{\theta}$$

$$- mR_o\ell\cos\theta\dot{\theta}^2 + mR_o\cos\theta\ddot{\ell} + mg\ell\cos\theta = FR_o\cos\theta \tag{1.43}$$

$$m\ddot{\ell} + mR_o\cos\theta\ddot{\theta} - m\ell\dot{\theta}^2 + mg\sin\theta + 2k\ell = F \tag{1.44}$$

Question 1.2

Total Energy

The total energy (Hamiltonian) is the sum of kinetic and potential energies:

$$H(q,\dot{q}) = T + V \tag{1.45}$$

$$= \frac{1}{4}M(3R_o^2 + R_i^2)\dot{\theta}^2 + \frac{1}{2}m\left((R_o^2 + \ell^2 - 2R_o\ell\sin\theta)\dot{\theta}^2 + \dot{\ell}^2 + 2R_o\dot{\ell}\dot{\theta}\cos\theta\right)$$

$$+ (M + m)gR_o + mg\ell\sin\theta + k\ell^2 \tag{1.46}$$

Minimum Energy

The minimum energy is attained when:

1. Kinetic energy is zero: $\dot{\theta} = 0$ and $\dot{\ell} = 0$

2. Potential energy is minimized

The potential energy (excluding constants):

$$V_{var} = mg\ell \sin\theta + k\ell^2 \tag{1.47}$$

To find the minimum, we set the partial derivatives to zero:

$$\frac{\partial V}{\partial \theta} = mg\ell \cos\theta = 0 \Rightarrow \theta = \pm\frac{\pi}{2} \text{ or } \ell = 0 \tag{1.48}$$

$$\frac{\partial V}{\partial \ell} = mg\sin\theta + 2k\ell = 0 \Rightarrow \ell = -\frac{mg\sin\theta}{2k} \tag{1.49}$$

For $\theta = -\frac{\pi}{2}$ (point mass below center):

$$\ell^* = -\frac{mg(-1)}{2k} = \frac{mg}{2k} \tag{1.50}$$

The minimum potential energy:

$$V_{min} = (M+m)gR_o + mg \cdot \frac{mg}{2k} \cdot (-1) + k\left(\frac{mg}{2k}\right)^2 \tag{1.51}$$

$$V_{min} = (M+m)gR_o - \frac{m^2g^2}{2k} + \frac{m^2g^2}{4k} = (M+m)gR_o - \frac{m^2g^2}{4k} \tag{1.52}$$

$$H_{min} = (M+m)gR_o - \frac{m^2g^2}{4k} \tag{1.53}$$

This minimum is achieved at $\theta = -\frac{\pi}{2} + 2n\pi$, $\ell = \frac{mg}{2k}$, $\dot{\theta} = 0$, $\dot{\ell} = 0$.

Question 1.3:

Time Derivative of Total Energy

The time derivative of the total energy equals the power supplied by non-conservative forces, which for our system becomes:

$$\dot{H} = \dot{q}^\top Q^{nc} \tag{1.54}$$

$$\dot{H} = \dot{\theta} \cdot FR_o \cos\theta + \dot{\ell} \cdot F = F(R_o\dot{\theta}\cos\theta + \dot{\ell}) \tag{1.55}$$

Verification by Direct Computation

Let us verify this by computing $\dot{H}$ directly:

$$\dot{H} = \frac{\partial H}{\partial \theta}\dot{\theta} + \frac{\partial H}{\partial \ell}\dot{\ell} + \frac{\partial H}{\partial \dot{\theta}}\ddot{\theta} + \frac{\partial H}{\partial \dot{\ell}}\ddot{\ell} \tag{1.56}$$

Using the Euler-Lagrange equations and the structure of mechanical systems, after substitution and simplification, we obtain:

$$\dot{H} = F(R_o\dot{\theta}\cos\theta + \dot{\ell}) \tag{1.57}$$

Passivity Output

A system is passive from input u to output y if there exists a storage function $V(x) \geq 0$ such that:

$$\dot{V} \leq y^\top u \tag{1.58}$$

Taking the total energy H as the storage function and the actuator force F as the input:

$$\dot{H} = F \cdot (R_o \dot{\theta} \cos\theta + \dot{\ell}) \tag{1.59}$$

For the system to be passive from input F to output y, we choose:

$$\boxed{y = R_o \dot{\theta} \cos\theta + \dot{\ell}} \tag{1.60}$$

This output represents the velocity of the point mass along the direction of the applied force in the body-fixed frame, which is the natural passive output for Euler-Lagrange systems with collocated actuation.

Note: The output can also be interpreted as $y = \dot{\ell} + R_o \dot{\theta} \cos\theta$, which represents the rate of change of the position of P projected onto $\vec{e}_1^1$.

Question 1.4: Lyapunov Analysis of Control Law

The feedback control law is:

$$F = -\gamma \dot{\ell}(H(q, \dot{q}) - H^*) \tag{1.61}$$

where $\gamma > 0$ is the control gain and $H^* > 0$ is the desired total energy.

Consider the Lyapunov function candidate:

$$V(q, \dot{q}) = \frac{1}{2\gamma}(H(q, \dot{q}) - H^*)^2 \tag{1.62}$$

This function satisfies:

- $V \geq 0$ for all $(q, \dot{q})$

- $V = 0$ if and only if $H = H^*$

Time Derivative of Lyapunov Function:

$$\dot{V} = \frac{1}{\gamma}(H - H^*)\dot{H} \tag{1.63}$$

$$= \frac{1}{\gamma}(H - H^*) \cdot F \cdot (R_o \dot{\theta} \cos\theta + \dot{\ell}) \tag{1.64}$$

Substituting the control law and simplifying:

$$\dot{V} = \frac{1}{\gamma}(H - H^*) \cdot (-\gamma \dot{\ell})(H - H^*) \cdot (R_o \dot{\theta} \cos\theta + \dot{\ell}) \tag{1.65}$$

$$= -(H - H^*)^2 \dot{\ell}(R_o \dot{\theta} \cos\theta + \dot{\ell}) \tag{1.66}$$

$$\dot{V} = -(H - H^*)^2 \left[\dot{\ell}^2 + R_o \dot{\ell}\dot{\theta} \cos\theta\right] \tag{1.67}$$

Analysis of $\dot{V}$

The sign of $\dot{V}$ depends on the term $[\dot{\ell}^2 + R_o \dot{\ell}\dot{\theta} \cos\theta]$:

- When $\dot{\ell} = 0$: $\dot{V} = 0$

- When $\dot{\ell} \neq 0$: The sign depends on $\dot{\ell}^2 + R_o \dot{\ell}\dot{\theta} \cos\theta$

The term $\dot{\ell}^2 + R_o \dot{\ell}\dot{\theta} \cos\theta$ is not necessarily positive definite. It can be written as:

$$\dot{\ell}^2 + R_o \dot{\ell}\dot{\theta} \cos\theta = \dot{\ell}\left(\dot{\ell} + R_o \dot{\theta} \cos\theta\right) = \dot{\ell} \cdot y \tag{1.68}$$

where y is the passive output from Question 1.3.

Conclusions

1. $\dot{V} \leq 0$ is NOT guaranteed:
The time derivative $\dot{V} = -(H - H^*)^2 \dot{\ell} \cdot y$ can be positive, negative, or zero depending on the signs of $\dot{\ell}$ and y.

2. Boundedness:
When $\dot{\ell} \cdot y > 0$, we have $\dot{V} < 0$ (for $H \neq H^*$), and energy approaches H^*.
When $\dot{\ell} \cdot y < 0$, we have $\dot{V} > 0$, and energy moves away from H^*.

3. Does the control law guarantee $\lim_{t \to \infty} H(q(t), \dot{q}(t)) = H^*$?
No, the control law does NOT guarantee convergence to H^* in general. The Lyapunov analysis only shows that when $\dot{\ell} = 0$, we have $\dot{V} = 0$, which means the energy stops changing. This occurs when:

- The point mass is not moving relative to the ring ($\dot{\ell} = 0$)

- At this condition, $F = 0$ regardless of the energy level

However, if $H^* > H_{min}$, the system can evolve such that the energy is "pumped" or "damped" depending on the motion, potentially achieving energy tracking asymptotically through the natural coupling dynamics, even though convergence is not guaranteed by this Lyapunov analysis alone.

Physical Interpretation: The control law acts as an energy-shaping controller that tries to inject or remove energy based on the error $(H - H^*)$. However, since the control force acts only along the $\vec{e}_1^1$ direction, it can only do work when $\dot{\ell} \neq 0$. When the point mass oscillates in phase with the wheel rotation, energy can be effectively transferred.

Question 1.5

Key Energy Values

From the theoretical analysis in Question 1.2, the minimum achievable energy is:

$$H_{min} = (M + m)gR_o - \frac{m^2 g^2}{4k} = (4+1)(10)(1) - \frac{(1)^2(10)^2}{4(20)} = 50 - 1.25 = \boxed{48.75 \text{ J}} \tag{1.69}$$

The initial energy of the system is calculated as:

$$H_0 = H(q(0), \dot{q}(0)) = T(0) + V(0) = \boxed{50.38 \text{ J}} \tag{1.70}$$

Control Law

The feedback control law from Question 1.4 is:

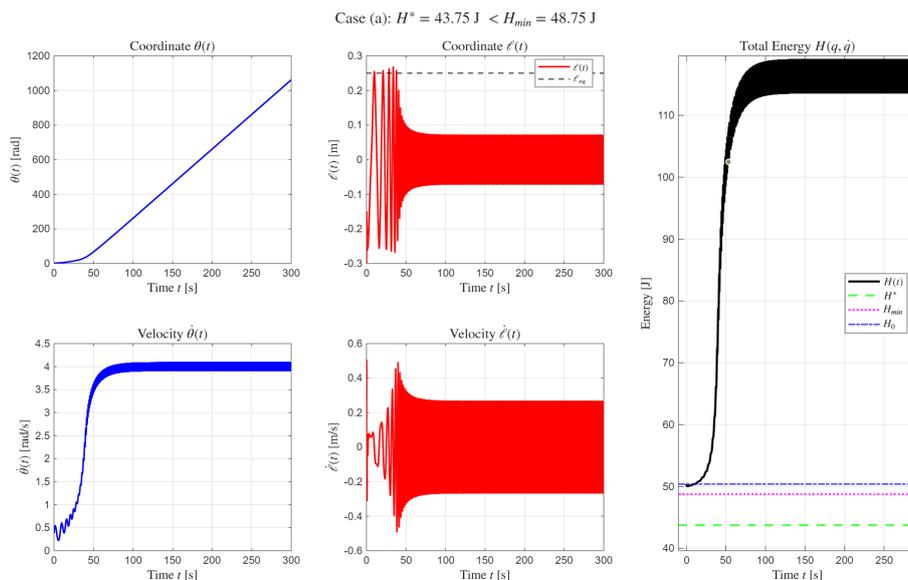$$F = -\gamma \dot{\ell} \left(H(q, \dot{q}) - H^* \right) \tag{1.71}$$

From the passivity analysis (Question 1.3), the power flow is:

$$\dot{H} = F \cdot y, \quad \text{where} \quad y = \dot{\ell} + R_o \dot{\theta} \cos \theta \tag{1.72}$$

Case (a): $H^* < H_{min}$

We set $H^* = 43.75$ J, which is 5 J below the theoretical minimum $H_{min} = 48.75$ J.

Since $H^* < H_{min}$, the desired energy level is **physically impossible** to achieve. The minimum energy H_{min} can only be attained when the system is completely at rest ($\dot{\theta} = 0$, $\dot{\ell} = 0$)

Figure 1.1: Case (a): System response for $H^* = 43.75$ J $< H_{\min} = 48.75$ J. The simulation runs for 300 seconds.

From the simulation results, we observe the following asymptotic behavior:

1. **Energy behavior**: The total energy $H(t)$ does **not** converge to H^*. Instead, the energy increases significantly from $H_0 = 50.38$ J to approximately 115 J.

2. **Ring motion**: The angle $\theta(t)$ increases continuously, indicating that the ring **rotates indefinitely** in one direction. The angular velocity stabilizes around $\dot\theta \approx 4$ rad/s.

3. **Point mass motion**: The coordinate $\ell(t)$ oscillates around a mean value close to zero, with the oscillations coupled to the rotation of the ring.

Why is H^* not achieved?
The desired energy $H^* = 43.75$ J is below the physical minimum $H_{\min} = 48.75$ J. No configuration of the system can have energy less than $H_{\min}$, so the control law cannot drive the system to H^*.

Why does energy increase instead of decrease?
The control law $F = -\gamma\dot\ell(H - H^*)$ is designed to extract energy when $H > H^*$. The power flow is:

$$\dot H = F \cdot y = -\gamma\dot\ell(H - H^*) \cdot (\dot\ell + R_o\dot\theta\cos\theta)$$

Analysis shows that:

- The term $|\dot\ell| \approx 0.18$ (small)

- The term $|R_o\dot\theta\cos\theta| \approx 2.17$ (12× larger!)

When the wheel rotates, the rotational term $R_o\dot\theta\cos\theta$ dominates the passive output y. This term oscillates with θ, causing y to change sign periodically. As a result:

- For ~55% of the time: $F \cdot y > 0$ (energy injection)

- For ~45% of the time: $F \cdot y < 0$ (energy extraction)

The net effect is energy **injection**, causing the system to speed up rather than slow down.

Case (b): Threshold H_p

We seek the threshold value H_p such that:

- For $H^* > H_p$: the ring rotates indefinitely

- For $H^* < H_p$: the ring oscillates indefinitely

To find H_p, we use a bisection method:

1. Set initial search range: $[H_{\min} + \epsilon, 55]$ J

2. For each test value H^*, simulate the system for sufficient time

3. Classify the motion as "oscillation" (bounded θ) or "rotation" (unbounded θ)

4. Narrow the search range based on the classification
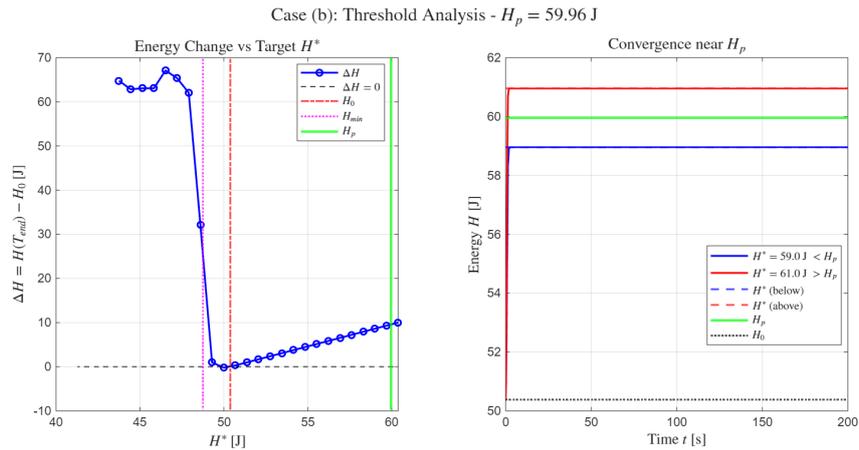
5. Converge to H_p within tolerance

Figure 1.2: Case (b): Analysis of control law effectiveness and threshold behavior.
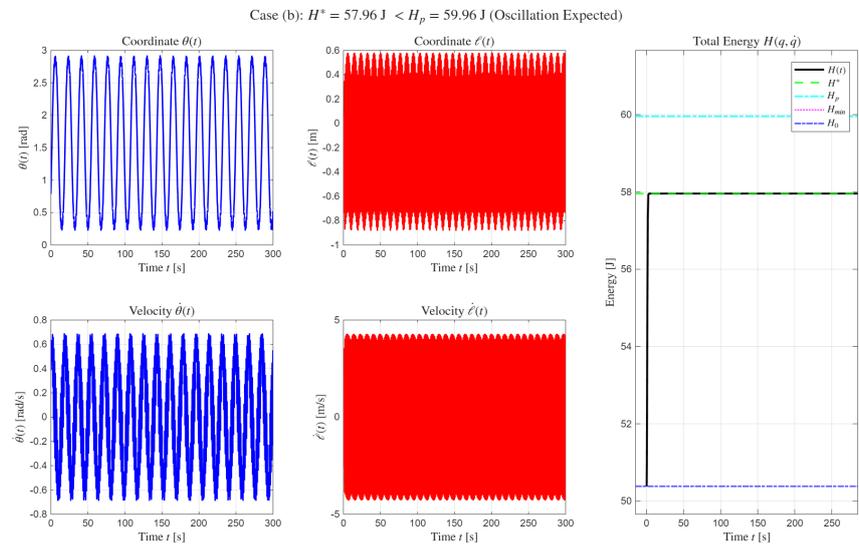

Figure 1.3: Case (b): System response for $H^* < H_p$ showing oscillatory motion of the ring.
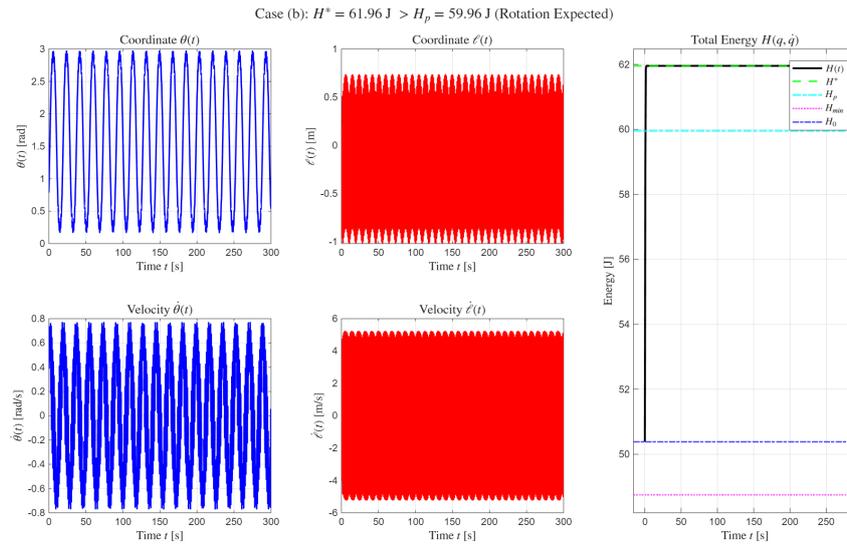
Figure 1.4: Case (b): System response for $H^* > H_p$ showing oscillatory motion of the ring.

For our wheel system, the results in both cases $H^* > H_p$ and $H^* < H_p$ that the gravitational potential acts as a barrier, causing the ring to oscillate.

Case (c): Convergence Comparison

We compare two scenarios where the control law is expected to work effectively:

Table 1.1: Case (c) scenarios

Case	H^* **Range**	H^* **Value**	**Expected Behavior**
c1	$H_p < H^* < H_0$	49.5 J	Energy decreasing
c2	$H_0 < H^* < 52.7$ J	51.5 J	Energy increasing

Note: We choose H^* values close to $H_0 = 50.38$ J where the control law is most effective.
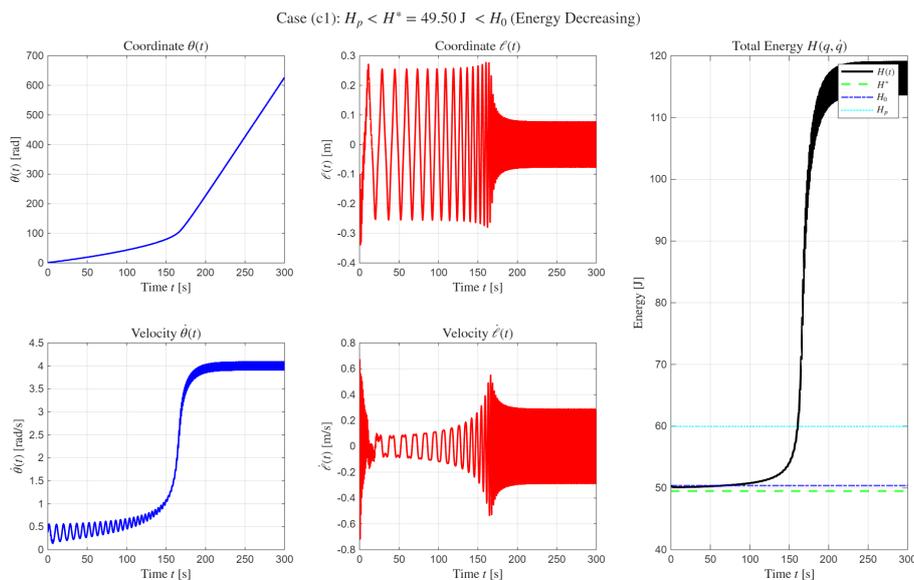
Figure 1.5: Case (c1): System response for $H^* = 49.5$ J $< H_0$.
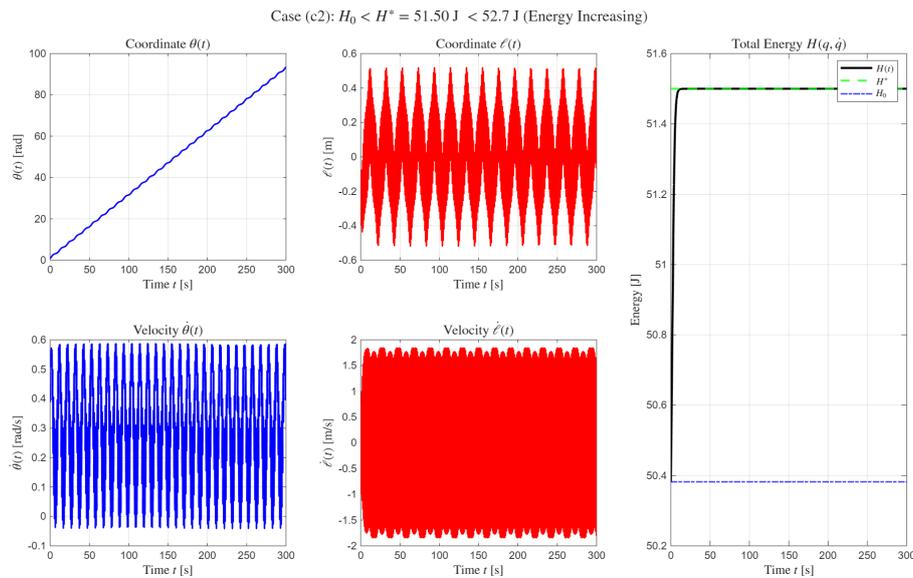
Figure 1.6: Case (c2): System response for $H^* = 51.5$ J $> H_0$.
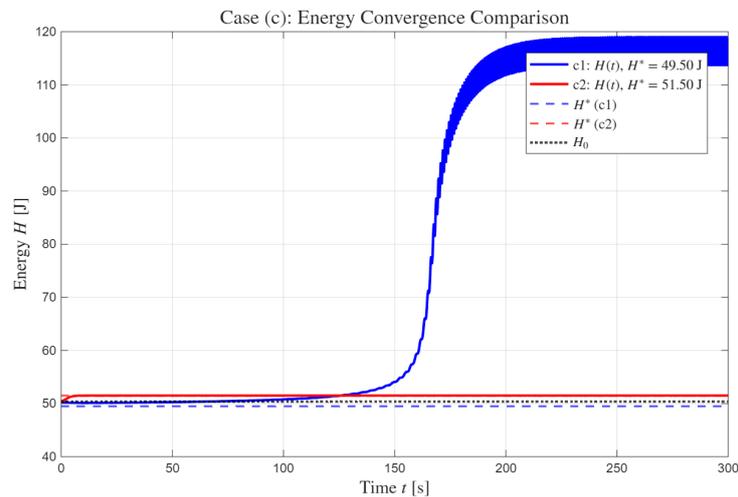
Figure 1.7: Case (c): Comparison of energy convergence for cases c1 and c2.

Case c1:

- Initial energy: $H_0 = 50.38$ J

- Target energy: $H^* = 49.5$ J

- The energy however increases and does not converge to H^*. Instead it goes to 118.9696 J . This results in an error of 69.5 J

- System does not converge to the expected value of H^*

Case c2:

- Initial energy: $H_0 = 50.38$ J

- Target energy: $H^* = 51.5$ J

- The energy increases and converges to H^*

- The ring motion increases in amplitude as energy is injected

- Time to converge (within 0.5 J): 2.7 s

2 | Problem 2:

Question 2.1:

How the state-space representation (4) was derived:

The state-space representation can be derived by first looking at the input (births) vs output which includes deaths, infections, recovery of the population. For x_1 (vulnerable population), we need to take into consideration newborns which enter the population at a rate of βN. Also vulnerable individuals die at rate β, which contributes to this part of the equation: $-\beta x_1$. Finally, the rate of new people infected depends on how much the vulnerable and infected individuals contact each other. Assuming homogeneous mixing, the probability that a vulnerable person encounters an infected person is proportional to the infected density x_2/N. With transmission rate γ, the infection rate is $\frac{\gamma}{N} x_2 x_1$.
Therefore: $\dot{x}_1 = \beta N - \beta x_1 - \frac{\gamma}{N} x_2 x_1$

For x_2 (contaminated population), New infections from the vulnerable population: $\frac{\gamma}{N} x_2 x_1$. Infected people die at rate β: $-\beta x_2$. The infected people also recover at rate of μ: $-\mu x_2$.
Therefore: $\dot{x}_2 = \frac{\gamma}{N} x_2 x_1 - \beta x_2 - \mu x_2$

For x_3 (immune population), Recovered individuals from the infected compartment: μx_2. Immune individuals die at rate β: $-\beta x_3$.
Therefore: $\dot{x}_3 = \mu x_2 - \beta x_3$

This now gives us the full state space representation.

Reduced normalized state-space model:

Given that $v = \frac{x_1}{N}$, $c = \frac{x_2}{N}$, $i = \frac{x_3}{N}$, we know that $x_1 + x_2 + x_3 = N$ for a constant population. We also then have $v + c + i = 1$, which implies $i = 1 - v - c$.
Substituting into the original equations and dividing by N:

$$\dot{v} = \frac{\dot{x}_1}{N} = \beta - \beta v - \gamma cv = \beta(1 - v) - \gamma cv \tag{2.1}$$

$$\dot{c} = \frac{\dot{x}_2}{N} = \gamma cv - \beta c - \mu c = \gamma cv - (\beta + \mu)c \tag{2.2}$$
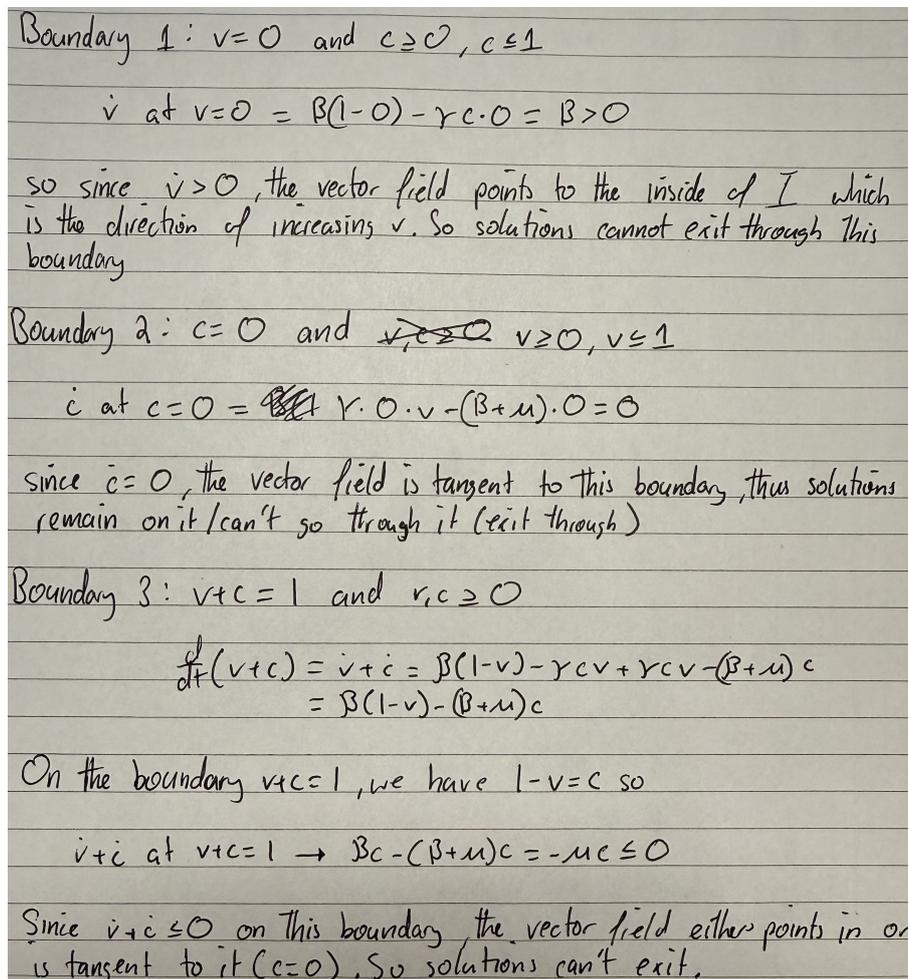
The equation for i becomes:

$$\dot{i} = \mu c - \beta i \tag{2.3}$$

This reduced model is sufficient for further analysis since i can be completely determined by v and c from $i = 1 - v - c$. Thus (2.1),(2.2) is sufficient for the complete dynamical analysis. Simply put, once $v(t)$ and $c(t)$ are known, $i(t) = 1 - v(t) - c(t)$ is known.

Question 2.2:

We must show that the set $\mathcal{I}$ is positively invariant, which says that trajectories starting in $\mathcal{I}$ must remain in $\mathcal{I}$ for all future time. We can verify this by checking that the vector field does not point outward on each boundary.

Boundary 1: $v=0$ and $c \geq 0, c \leq 1$

$\dot{v}$ at $v=0 = \beta(1-0) - \gamma c \cdot 0 = \beta > 0$

So since $\dot{v} > 0$, the vector field points to the inside of $\mathcal{I}$ which is the direction of increasing v. So solutions cannot exit through this boundary

Boundary 2: $c=0$ and ~~$v,c \geq 0$~~ $v \geq 0, v \leq 1$

$\dot{c}$ at $c=0 = ~~\beta(\cdot)~~ \gamma \cdot 0 \cdot v - (\beta+\mu) \cdot 0 = 0$

since $\dot{c} = 0$, the vector field is tangent to this boundary, thus solutions remain on it / can't go through it (exit through)

Boundary 3: $v+c=1$ and $v, c \geq 0$

$\frac{d}{dt}(v+c) = \dot{v}+\dot{c} = \beta(1-v) - \gamma c v + \gamma c v - (\beta+\mu) c$
$\qquad\qquad = \beta(1-v) - (\beta+\mu) c$

On the boundary $v+c=1$, we have $1-v = c$ so

$\dot{v}+\dot{c}$ at $v+c=1 \rightarrow \beta c - (\beta+\mu) c = -\mu c \leq 0$

Since $\dot{v}+\dot{c} \leq 0$ on this boundary, the vector field either points in or is tangent to it ($c=0$). So solutions can't exit.

Figure 2.1: Derivation for the positive invariance of the set

This shows that the vector field does not point outside $\mathcal{I}$ on any boundary. So the set $\mathcal{I}$ is positively invariant.
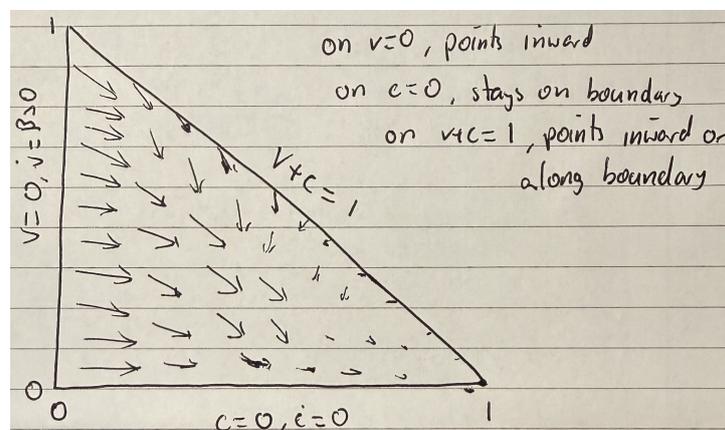
Figure 2.2: The invariant set $\mathcal{I}$ with vector field directions on boundaries.

Question 2.3:

To find equilibrium points, we set $\dot{v} = 0$ and $\dot{c} = 0$:

$$\beta(1 - v) - \gamma cv = 0 \tag{2.4}$$

$$c[\gamma v - (\beta + \mu)] = 0 \tag{2.5}$$

From equation (2.5), either $c^* = 0$ or $\gamma v^* = \beta + \mu$.
For $c^* = 0$, we substitute it into (2.4):

$$\beta(1 - v^*) = 0 \implies v^* = 1 \tag{2.6}$$

The interpretation of this is that it would be the equilibrium where the population is disease free: $(v^*, c^*) = (1, 0)$. The entire population is vulnerable; the disease has been eradicated. So this equilibrium would always exist.
For the $c^* \neq 0$ case, we know that $\gamma v^* = \beta + \mu$ and from this we can get:

$$v^* = \frac{\beta + \mu}{\gamma} = \frac{1}{\theta} \tag{2.7}$$

Substituting into (2.4):

$$\beta\left(1 - \frac{1}{\theta}\right) - \gamma c^* \cdot \frac{1}{\theta} = 0$$

$$\beta\left(\frac{\theta - 1}{\theta}\right) = \frac{\gamma c^*}{\theta}$$

$$c^* = \frac{\beta(\theta - 1)}{\gamma} = \frac{\beta}{\gamma}\left(1 - \frac{1}{\theta}\right)$$

This gives the following equilibrium:

$$(v^*, c^*) = \left(\frac{1}{\theta}, \frac{\beta}{\gamma}\left(1 - \frac{1}{\theta}\right)\right) \tag{2.8}$$

This would be the equilibrium where the disease persists indefinitely in the population where a constant amount of people are always infected.
It's important to notice that when $\theta \leq 1$: Only the disease-free equilibrium exists (since $c^* \leq 0$ doesn't mean anything). Furthermore for $\theta > 1$: Both equilibria exist. The equilibrium where people are always infected lies in $\mathcal{I}$ since $v^* = 1/\theta < 1$ and $c^* > 0$. Lastly, when $\theta = 1$, we are at a critical point since if $\theta > 1$, each infected person infects more than one other on average, and the disease becomes regularly occurring.

Question 2.4:

We will analyze the stability by linearization. Firstly, the Jacobian matrix of the system (2.1),(2.2) is:

$$J = \begin{pmatrix} \frac{\partial \dot{v}}{\partial v} & \frac{\partial \dot{v}}{\partial c} \\ \frac{\partial \dot{c}}{\partial v} & \frac{\partial \dot{c}}{\partial c} \end{pmatrix} = \begin{pmatrix} -\beta - \gamma c & -\gamma v \\ \gamma c & \gamma v - (\beta + \mu) \end{pmatrix} \tag{2.9}$$

At the disease free equilibrium $(v^*, c^*) = (1, 0)$:

$$J_{DFE} = \begin{pmatrix} -\beta & -\gamma \\ 0 & \gamma - (\beta + \mu) \end{pmatrix} \tag{2.10}$$

This is upper triangular, so the eigenvalues are simply the diagonal entries:

$$\lambda_1 = -\beta < 0 \tag{2.11}$$
$$\lambda_2 = \gamma - (\beta + \mu) = (\beta + \mu)(\theta - 1) \tag{2.12}$$

- If $\theta < 1$: $\lambda_2 < 0$, so both eigenvalues are negative. The disease free equilibrium is a stable node.

- If $\theta = 1$: $\lambda_2 = 0$. This is a bifurcation point.

- If $\theta > 1$: $\lambda_2 > 0$, so eigenvalues have opposite signs. The disease free equilibrium is an unstable saddle point.

At the persistent disease equilibrium $(v^*, c^*) = \left(\frac{1}{\theta}, \frac{\beta(\theta - 1)}{\gamma} \right)$ for $\theta > 1$:
Substituting the equilibrium values:

$$\gamma v^* = \gamma \cdot \frac{1}{\theta} = \frac{\gamma}{\theta} = \beta + \mu \tag{2.13}$$

$$\gamma c^* = \gamma \cdot \frac{\beta(\theta - 1)}{\gamma} = \beta(\theta - 1) \tag{2.14}$$

The Jacobian becomes:

$$J_{PDE} = \begin{pmatrix} -\beta - \beta(\theta - 1) & -\frac{\gamma}{\theta} \\ \beta(\theta - 1) & (\beta + \mu) - (\beta + \mu) \end{pmatrix} = \begin{pmatrix} -\beta\theta & -\frac{\gamma}{\theta} \\ \beta(\theta - 1) & 0 \end{pmatrix} \tag{2.15}$$

The trace and determinant are:

$$\text{tr}(J_{PDE}) = -\beta\theta < 0 \tag{2.16}$$

$$\det(J_{PDE}) = 0 - \left(-\frac{\gamma}{\theta} \right) \cdot \beta(\theta - 1) = \frac{\gamma\beta(\theta - 1)}{\theta} > 0 \quad \text{(for } \theta > 1) \tag{2.17}$$

Since $\text{tr}(J_{PDE}) < 0$ and $\det(J_{PDE}) > 0$, both eigenvalues have negative real parts. To find the equilibrium type, we find the discriminant:

$$\Delta = \text{tr}^2 - 4 \det = \beta^2 \theta^2 - \frac{4\gamma\beta(\theta - 1)}{\theta} \tag{2.18}$$

- If the discriminant is > 0: The equilibrium is a stable node (two distinct negative real eigenvalues).

- If the discriminant $= 0$: The equilibrium is a stable degenerate node.

- If the discriminant < 0: The equilibrium is a stable focus/spiral (complex eigenvalues with negative real part).

Question 2.5:

For $\theta > 1$, we show global asymptotic stability of the endemic equilibrium using a Lyapunov function. Using the hint:

$$V(v,c) = \left(v - v^* \ln \frac{v}{v^*}\right) + \left(c - c^* \ln \frac{c}{c^*}\right) \tag{2.19}$$

where $(v^*, c^*) = \left(\frac{1}{\theta}, \frac{\beta(\theta-1)}{\gamma}\right)$ is the endemic equilibrium.

We can compute $\dot{V}$ next:

$$\dot{V} = \frac{\partial V}{\partial v}\dot{v} + \frac{\partial V}{\partial c}\dot{c} \quad \rightarrow \quad \left(1 - \frac{v^*}{v}\right)\dot{v} + \left(1 - \frac{c^*}{c}\right)\dot{c}$$

now putting in $\dot{v} = \beta(1-v) - \gamma cv$ and $\dot{c} = \gamma cv - (\beta + \mu)c$
and at equilibrium v^*, c^*, both $\dot{v} = 0$ and $\dot{c} = 0$

So from $\dot{v} = 0 \rightarrow \beta(1-v^*) = \gamma c^* v^* \rightarrow \beta = \beta v^* + \gamma c^* v^*$
and from $\dot{c} = 0 \rightarrow \gamma v^* = \beta + \mu \rightarrow \gamma v^* = \beta + \mu$

$$\dot{V} = \left(1 - \frac{v^*}{v}\right)\left[\beta(1-v) - \gamma cv\right] + \left(1 - \frac{c^*}{c}\right)\left[\gamma cv - (\beta+\mu)c\right]$$

$$\dot{V} = \left(1 - \frac{v^*}{v}\right)\left[\gamma c^* v^* + \beta v^* - \beta v - \gamma cv\right] + \left(1 - \frac{c^*}{c}\right)\gamma c(v - v^*)$$

This becomes after some simplifying

$$\dot{V} = (\gamma c^* v^* + \beta v^*)\left(2 - \frac{v}{v^*} - \frac{v^*}{v}\right)$$

Using $2 - x - \frac{1}{x} \leq 0$ for $x > 0$, where $x = \frac{v}{v^*} = v\theta$

$$2 - \frac{v}{v^*} - \frac{v^*}{v} = 2 - v\theta - \frac{1}{v\theta} \leq 0$$

So $\dot{V} \leq 0$ with $\dot{V} = 0$ only when $v = v^*$

Solutions converge to the largest invariant set where $\dot{V} = 0$

The set where $\dot{V} = 0$ is $(v,c): v = v^*$

Checking if the solutions stay there: $\dot{v} = \beta(1 - v^*) - \gamma cv^* = 0 \rightarrow$

$$c = \frac{\beta(1 - v^*)}{\gamma v^*} = c^* \quad \text{so the only point (invariant) where } \dot{v} = 0$$

is (v^*, c^*), itself

The epidemic equilibrium (v^*, c^*) is globally asymptotically stable for $\theta > 1$

Figure 2.3: Computing $\dot{V}$

The endemic equilibrium (v^*, c^*) is globally asymptotically stable in the interior of $\mathcal{I}$ for $\theta > 1$.

Question 2.6:

For the first row parameter(Set 1): $\beta = 1/80$, $\gamma = 20$, $\mu = 8$, we get $\theta = \frac{\gamma}{\beta+\mu} = \frac{20}{1/80+8} = \frac{20}{8.0125} \approx 2.496 > 1$
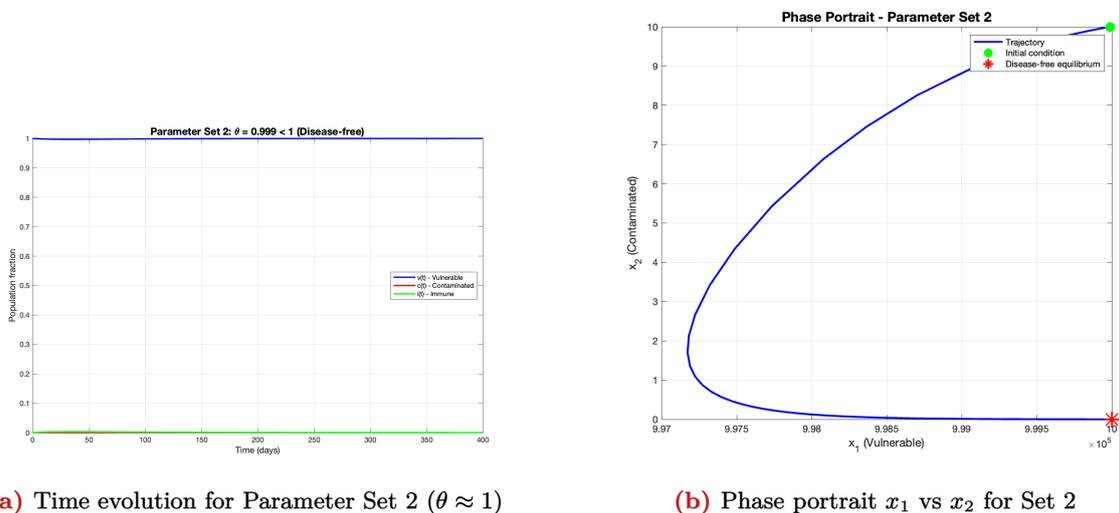
Since $\theta > 1$, the disease becomes endemic. The endemic equilibrium values are:

$$v^* = \frac{1}{\theta} \approx 0.401, \quad c^* = \frac{\beta(\theta - 1)}{\gamma} \approx 9.35 \times 10^{-4} \quad (2.20)$$

For the second row parameter(Set 2): $\beta = 1/80$, $\gamma = 20$, $\mu = 20$, we get $\theta = \frac{20}{1/80+20} = \frac{20}{20.0125} \approx 0.999 < 1$

Since $\theta < 1$, the disease goes to zero and converges to the disease free equilibrium $(v^*, c^*) = (1, 0)$.

(a) Time evolution for Parameter Set 1 ($\theta \approx 2.5$)

(b) Phase portrait x_1 vs x_2 for Set 1
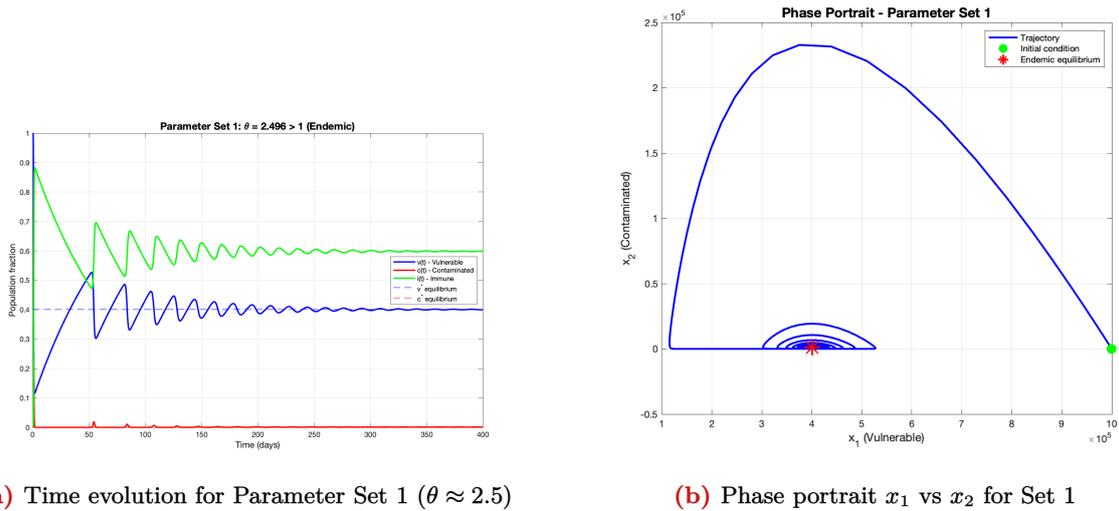
Figure 2.4: Simulation results for Parameter Set 1 (endemic case).

(a) Time evolution for Parameter Set 2 ($\theta \approx 1$)

(b) Phase portrait x_1 vs x_2 for Set 2

Figure 2.5: Simulation results for Parameter Set 2 (disease free case).

The results can be interpreted as follows:

For parameter set 1 ($\theta > 1$, endemic case):

- The time evolution shows that after an initial outbreak, the system undergoes damped oscillations before settling to the endemic equilibrium.
- The vulnerable fraction decreases from nearly 1 to approximately $v^* = 0.4$, while the immune fraction grows to about $i^* \approx 0.6$.

■ The phase clearly shows a spiral trajectory, confirming that the endemic equilibrium is a stable focus.

■ Physically, this represents a scenario where the disease stays alive indefinitely in the population at a low but constant level.

For parameter set 2 ($\theta < 1$, disease free case):

■ The time evolution shows that the contaminated fraction $c(t)$ rises briefly but then decays monotonically to zero.

■ The vulnerable fraction remains close to 1, with only a small fraction becoming immune.

■ The phase shows a trajectory that moves directly toward the $x_2 = 0$ axis (disease free state) without spiraling, consistent with a stable node.

■ Physically, each infected person infects fewer than one other person on average ($\theta < 1$), so the disease cannot sustain itself and dies out by itself.

The simulations confirm our theoretical predictions: the threshold $\theta = 1$ determines whether the disease becomes endemic ($\theta > 1$) or dies out ($\theta < 1$).

Question 2.7:

Derivation of the vaccination model:

With the specific rate of vaccination of newborns at rate u, the modified state-space representation is reasoned as follows:

For x_1, a fraction u of newborns (who would otherwise be vulnerable) are vaccinated and become immune. This removes $\beta u N$ from the vulnerable equation:

$$\dot{x}_1 = \beta N - \beta x_1 - \frac{\gamma}{N} x_2 x_1 - \beta u N \tag{2.21}$$

Then for x_2, there are no changes, as vaccination only affects newborns, not the infection equation:

$$\dot{x}_2 = -\beta x_2 + \frac{\gamma}{N} x_2 x_1 - \mu x_2 \tag{2.22}$$

Finally for x_3, the vaccinated newborns $\beta u N$ become immune:

$$\dot{x}_3 = -\beta x_3 + \mu x_2 + \beta u N \tag{2.23}$$

So the reduced normalized model is as follows:

Using $v = x_1/N$, $c = x_2/N$:

$$\dot{v} = \beta(1-u) - \beta v - \gamma cv = \beta(1-u-v) - \gamma cv \tag{2.24}$$
$$\dot{c} = \gamma cv - (\beta + \mu)c \tag{2.25}$$

Vaccination rate for 90% effective vaccine

With 90% vaccine efficacy, only $0.9u$ of newborns actually get immunity. The effective vaccination term becomes $0.9 \cdot \beta u N$ for x_3. The normalized model is then:

$$\dot{v} = \beta(1 - 0.9u - v) - \gamma cv \tag{2.26}$$
$$\dot{c} = \gamma cv - (\beta + \mu)c \tag{2.27}$$

We can now find the equilibria with vaccination:

From $\dot{c} = c[\gamma v - (\beta + \mu)] = 0$:

- Disease free: $c^* = 0$, and from $\dot{v} = 0$: $v^* = 1 - 0.9u$

- Endemic: $v^* = \frac{1}{\theta}$

For the endemic equilibrium to not exist, we need $v^* = 1/\theta$ to not be possible within the invariant set, i.e., $1 - 0.9u < 1/\theta$. So we can make the condition to eliminate endemic equilibrium:

$$1 - 0.9u < \frac{1}{\theta} \implies 0.9u > 1 - \frac{1}{\theta} \implies u > \frac{1}{0.9}\left(1 - \frac{1}{\theta}\right) \tag{2.28}$$

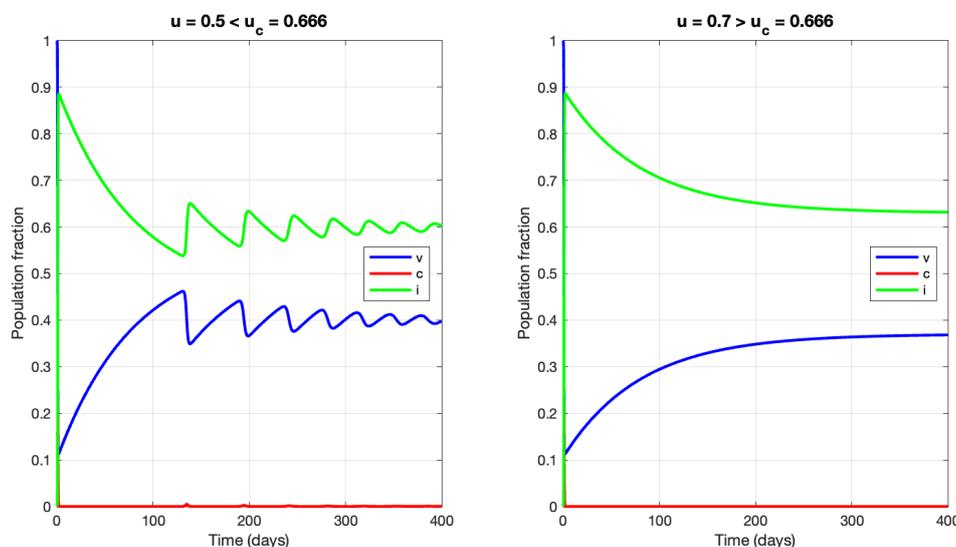$$u > \frac{10}{9}\left(1 - \frac{1}{\theta}\right) = \frac{10(\theta - 1)}{9\theta} \tag{2.29}$$

For Parameter Set 1 with $\theta \approx 2.496$:

$$u > \frac{10(2.496 - 1)}{9 \times 2.496} = \frac{10 \times 1.496}{22.464} \approx 0.666 \tag{2.30}$$

This means that at least 66.6% of newborns must be vaccinated (with the 90% effective vaccine) to prevent an endemic equilibrium.

Simulations with vaccination

The model has been simulated with the 90% effective vaccine using Parameter Set 1 ($\theta \approx 2.496$) for different vaccination rates.

Figure 2.6: Effect of vaccination rate on disease spreading with 90% vaccine efficacy. Left: $u = 0.5 < u_c$ (endemic keeps going). Right: $u = 0.7 > u_c$ (disease eliminated). The critical vaccination rate is $u_c \approx 0.666$. Note: the contaminated fraction $c(t)$ in red is very small and not clearly visible at this scale.

These results can be interpreted as follows:
Below the threshold ($u < u_c = 0.666$):

- For $u = 0.5$, the disease still becomes endemic, but at a lower level than without vaccination.
- It still shows damped oscillations converging to a positive endemic equilibrium $c^* \approx 2.3 \times 10^{-4}$ (approximately 230 infected people out of 1 million).
- Vaccination reduces the magnitude of the epidemic but does not eliminate it.

Above the threshold ($u > u_c = 0.666$):

- For $u = 0.7$, the contaminated fraction $c(t)$ decays to zero.
- The disease is eliminated from the population.
- When the vaccination rates are higher then the disease is eliminated faster.

The simulations show what we also calculated theoretically; which is that vaccinating more than 66.6% of newborns (with a 90% effective vaccine) is enough to eliminate the endemic equilibrium and make the disease vanish.

Bifurcation analysis with 100% effective vaccine

With 100% vaccine efficacy, the normalized model is:

$$\dot{v} = \beta(1 - u - v) - \gamma c v \tag{2.31}$$

$$\dot{c} = \gamma c v - (\beta + \mu)c = c[\gamma v - (\beta + \mu)] \tag{2.32}$$

Finding the equilibria:
Disease-free equilibrium: $c^* = 0$, $v^* = 1 - u$ Endemic equilibrium: $v^* = 1/\theta$, and from $\dot{v} = 0$:

$$\beta(1 - u - 1/\theta) = \gamma c^*/\theta \implies c^* = \frac{\beta\theta}{\gamma}\left(1 - u - \frac{1}{\theta}\right) = \frac{\beta}{\gamma}(\theta - \theta u - 1) \tag{2.33}$$

The endemic equilibrium exists when $c^* > 0$:

$$\theta - \theta u - 1 > 0 \implies u < 1 - \frac{1}{\theta} \tag{2.34}$$

Bifurcation at $u = 1 - 1/\theta$:

At the critical vaccination rate $u_c = 1 - 1/\theta$ we notice the following:

- For $u < u_c$: Both disease free and endemic equilibria exist. The endemic equilibrium is stable, the disease free is unstable.

- At $u = u_c$: The endemic equilibrium collides with the disease-free equilibrium at $(v^*, c^*) = (1/\theta, 0) = (1 - u_c, 0)$.

- For $u > u_c$: Only the disease free equilibrium exists and is stable.

Hence we have two equilibria which exchange stability as they pass through each other. We can now make a bifurcation diagram where the endemic equilibrium value c^* is plotted as a function of u:

$$c^*(u) = \begin{cases} \frac{\beta}{\gamma}(\theta - 1 - \theta u) & \text{for } 0 \leq u < 1 - \frac{1}{\theta} \\ 0 & \text{for } u \geq 1 - \frac{1}{\theta} \end{cases} \tag{2.35}$$
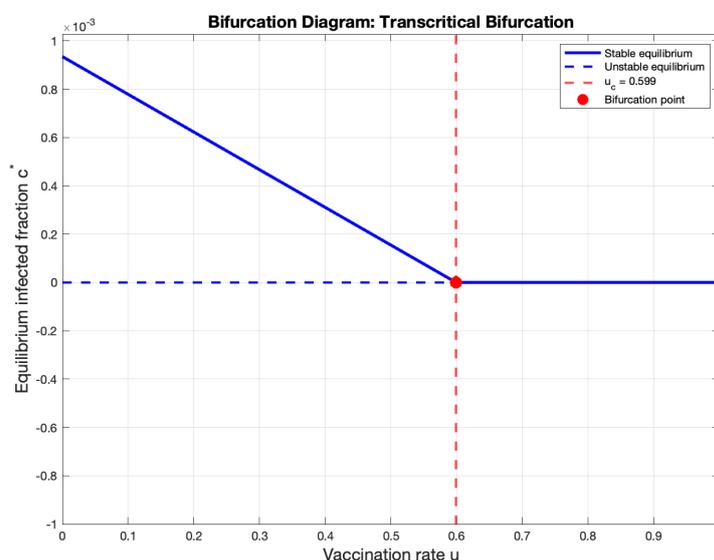
Figure 2.7: Bifurcation diagram showing c^* vs. vaccination rate u. Solid lines show stable equilibria, dashed lines show unstable equilibria. The critical bifurcation occurs at $u_c = 1 - 1/\theta \approx 0.599$.

The physical interpretation of this is that the critical vaccination rate $u_c = 1 - 1/\theta$ represents the herd immunity threshold. When this fraction of newborns is vaccinated, $\theta \cdot (1 - u) = 1$ and Each infected person infects exactly one other person on average. Also any higher vaccination rate $(u > u_c)$, the disease has nowhere to go because too many people are immune, so it stops spreading and disappears. So with $\theta \approx 2.496$, this means vaccinating approximately 60% of newborns (with a 100% effective vaccine) achieves herd immunity and prevents endemic disease.

3 │ Appendix A - Question 1 Matlab Code

```matlab
clear; close all; clc;

params.M  = 4;          % Ring mass [kg]
params.Ri = 0.8;        % Inner radius [m]
params.Ro = 1;          % Outer radius [m]
params.m  = 1;          % Point mass [kg]
params.k  = 20;         % Spring stiffness [N/m]
params.g  = 10;         % Gravitational acceleration [m/s^2]
params.gamma = 0.5;     % Control gain [-]

theta0  = pi/4;         % Initial angle [rad]
ell0    = -0.3;         % Initial position of point mass [m]
dtheta0 = 0.4;          % Initial angular velocity [rad/s]
dell0   = 0;            % Initial velocity of point mass [m/s]

x0 = [theta0; ell0; dtheta0; dell0];

Hmin = calculate_Hmin(params);
H0 = calculate_H(x0, params);


%% Case A
Hstar_a = Hmin - 5;  % H* = 43.75 J < Hmin = 48.75 J
Tend_a = 300;

[t_a, X_a, H_a] = run_simulation(x0, params, Hstar_a, Tend_a);


fig_a = figure('Name', 'Figure 1: Case (a)', 'Position', [50, 100, 1400, 800]);

subplot(2,3,1);
plot(t_a, X_a(:,1), 'b', 'LineWidth', 1.5);
xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
ylabel('$\theta(t)$ [rad]', 'Interpreter', 'latex', 'FontSize', 12);
title('Coordinate $\theta(t)$', 'Interpreter', 'latex', 'FontSize', 13);
grid on;

subplot(2,3,2);
ell_eq = params.m*params.g/(2*params.k);
plot(t_a, X_a(:,2), 'r', 'LineWidth', 1.5);
hold on;
yline(ell_eq, 'k--', 'LineWidth', 1.2);
xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
ylabel('$\ell(t)$ [m]', 'Interpreter', 'latex', 'FontSize', 12);
title('Coordinate $\ell(t)$', 'Interpreter', 'latex', 'FontSize', 13);
legend('$\ell(t)$', '$\ell_{eq}$', 'Interpreter', 'latex', 'Location', 'best');
grid on;

subplot(2,3,4);
plot(t_a, X_a(:,3), 'b', 'LineWidth', 1.5);
xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
ylabel('$\dot{\theta}(t)$ [rad/s]', 'Interpreter', 'latex', 'FontSize', 12);
title('Velocity $\dot{\theta}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
grid on;

subplot(2,3,5);
plot(t_a, X_a(:,4), 'r', 'LineWidth', 1.5);
xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
ylabel('$\dot{\ell}(t)$ [m/s]', 'Interpreter', 'latex', 'FontSize', 12);
title('Velocity $\dot{\ell}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
grid on;

subplot(2,3,[3,6]);
plot(t_a, H_a, 'k', 'LineWidth', 2);
hold on;
yline(Hstar_a, 'g--', 'LineWidth', 2);
yline(Hmin, 'm:', 'LineWidth', 2);
yline(H0, 'b-.', 'LineWidth', 1.5);
xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
ylabel('Energy [J]', 'Interpreter', 'latex', 'FontSize', 12);
```

```matlab
71  title('Total Energy $H(q,\dot{q})$', 'Interpreter', 'latex', 'FontSize', 13);
72  legend('$H(t)$', '$H^*$', '$H_{min}$', '$H_0$', ...
73      'Interpreter', 'latex', 'Location', 'east', 'FontSize', 10);
74  grid on;
75
76  sgtitle(sprintf('Case (a): $H^* = %.2f$ J $< H_{min} = %.2f$ J', Hstar_a, Hmin), ...
77      'Interpreter', 'latex', 'FontSize', 15);
78  %% Case B
79  % Find Hp using bisection
80  Hp = find_threshold_Hp(x0, params, Hmin);
81
82  % Test range of H* values for effectiveness analysis
83  Hstar_test = linspace(Hmin - 5, H0 + 10, 25);
84  Tend_test = 100;
85  delta_H = zeros(size(Hstar_test));
86
87  for i = 1:length(Hstar_test)
88      [~, ~, H_test] = run_simulation(x0, params, Hstar_test(i), Tend_test);
89      delta_H(i) = H_test(end) - H_test(1);
90  end
91
92  fig_b1 = figure('Name', 'Figure 2: Case (b) Threshold Analysis', 'Position', [100, 100,
        1100, 500]);
93
94  subplot(1,2,1);
95  plot(Hstar_test, delta_H, 'b-o', 'LineWidth', 1.5, 'MarkerSize', 5);
96  hold on;
97  yline(0, 'k--', 'LineWidth', 1);
98  xline(H0, 'r-.', 'LineWidth', 1.5);
99  xline(Hmin, 'm:', 'LineWidth', 1.5);
100 xline(Hp, 'g-', 'LineWidth', 2);
101 xlabel('$H^*$ [J]', 'Interpreter', 'latex', 'FontSize', 12);
102 ylabel('$\Delta H = H(T_{end}) - H_0$ [J]', 'Interpreter', 'latex', 'FontSize', 12);
103 title('Energy Change vs Target $H^*$', 'Interpreter', 'latex', 'FontSize', 13);
104 legend('$\Delta H$', '$\Delta H = 0$', '$H_0$', '$H_{min}$', '$H_p$', ...
105     'Interpreter', 'latex', 'Location', 'best');
106 grid on;
107
108 subplot(1,2,2);
109 % Show specific cases near Hp
110 Hstar_near_below = Hp - 1;
111 Hstar_near_above = Hp + 1;
112 Tend_near = 200;
113
114 [t_near_below, ~, H_near_below] = run_simulation(x0, params, Hstar_near_below, Tend_near
        );
115 [t_near_above, ~, H_near_above] = run_simulation(x0, params, Hstar_near_above, Tend_near
        );
116
117 plot(t_near_below, H_near_below, 'b', 'LineWidth', 1.5);
118 hold on;
119 plot(t_near_above, H_near_above, 'r', 'LineWidth', 1.5);
120 yline(Hstar_near_below, 'b--', 'LineWidth', 1.5);
121 yline(Hstar_near_above, 'r--', 'LineWidth', 1.5);
122 yline(Hp, 'g-', 'LineWidth', 2);
123 yline(H0, 'k:', 'LineWidth', 1.5);
124 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
125 ylabel('Energy $H$ [J]', 'Interpreter', 'latex', 'FontSize', 12);
126 title('Convergence near $H_p$', 'Interpreter', 'latex', 'FontSize', 13);
127 legend(sprintf('$H^*=%.1f$ J $< H_p$', Hstar_near_below), ...
128         sprintf('$H^*=%.1f$ J $> H_p$', Hstar_near_above), ...
129         '$H^*$ (below)', '$H^*$ (above)', '$H_p$', '$H_0$', ...
130     'Interpreter', 'latex', 'Location', 'best');
131 grid on;
132
133 sgtitle(sprintf('Case (b): Threshold Analysis - $H_p = %.2f$ J', Hp), ...
134     'Interpreter', 'latex', 'FontSize', 15);
135
136 % Simulations for below and above Hp
137 Hstar_below_Hp = Hp - 2;  % Below threshold (should oscillate)
138 Hstar_above_Hp = Hp + 2;  % Above threshold (should rotate)
139 Tend_b = 300;
140
```

```matlab
141 [t_below_Hp, X_below_Hp, H_below_Hp] = run_simulation(x0, params, Hstar_below_Hp, Tend_b
        );
142 [t_above_Hp, X_above_Hp, H_above_Hp] = run_simulation(x0, params, Hstar_above_Hp, Tend_b
        );
143
144 fig_b2 = figure('Name', 'Figure 3: Case (b) Below Hp', 'Position', [50, 50, 1400, 800]);
145
146 subplot(2,3,1);
147 plot(t_below_Hp, X_below_Hp(:,1), 'b', 'LineWidth', 1.5);
148 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
149 ylabel('$\theta(t)$ [rad]', 'Interpreter', 'latex', 'FontSize', 12);
150 title('Coordinate $\theta(t)$', 'Interpreter', 'latex', 'FontSize', 13);
151 grid on;
152
153 subplot(2,3,2);
154 plot(t_below_Hp, X_below_Hp(:,2), 'r', 'LineWidth', 1.5);
155 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
156 ylabel('$\ell(t)$ [m]', 'Interpreter', 'latex', 'FontSize', 12);
157 title('Coordinate $\ell(t)$', 'Interpreter', 'latex', 'FontSize', 13);
158 grid on;
159
160 subplot(2,3,4);
161 plot(t_below_Hp, X_below_Hp(:,3), 'b', 'LineWidth', 1.5);
162 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
163 ylabel('$\dot{\theta}(t)$ [rad/s]', 'Interpreter', 'latex', 'FontSize', 12);
164 title('Velocity $\dot{\theta}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
165 grid on;
166
167 subplot(2,3,5);
168 plot(t_below_Hp, X_below_Hp(:,4), 'r', 'LineWidth', 1.5);
169 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
170 ylabel('$\dot{\ell}(t)$ [m/s]', 'Interpreter', 'latex', 'FontSize', 12);
171 title('Velocity $\dot{\ell}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
172 grid on;
173
174 subplot(2,3,[3,6]);
175 plot(t_below_Hp, H_below_Hp, 'k', 'LineWidth', 2);
176 hold on;
177 yline(Hstar_below_Hp, 'g--', 'LineWidth', 2);
178 yline(Hp, 'c-.', 'LineWidth', 2);
179 yline(Hmin, 'm:', 'LineWidth', 1.5);
180 yline(H0, 'b-.', 'LineWidth', 1.5);
181 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
182 ylabel('Energy [J]', 'Interpreter', 'latex', 'FontSize', 12);
183 title('Total Energy $H(q,\dot{q})$', 'Interpreter', 'latex', 'FontSize', 13);
184 legend('$H(t)$', '$H^*$', '$H_p$', '$H_{min}$', '$H_0$', ...
185     'Interpreter', 'latex', 'Location', 'best', 'FontSize', 10);
186 grid on;
187
188 sgtitle(sprintf('Case (b): $H^* = %.2f$ J $< H_p = %.2f$ J (Oscillation Expected)', ...
189     Hstar_below_Hp, Hp), 'Interpreter', 'latex', 'FontSize', 15);
190
191 fig_b3 = figure('Name', 'Figure 4: Case (b) Above Hp', 'Position', [100, 50, 1400, 800])
        ;
192
193 subplot(2,3,1);
194 plot(t_above_Hp, X_above_Hp(:,1), 'b', 'LineWidth', 1.5);
195 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
196 ylabel('$\theta(t)$ [rad]', 'Interpreter', 'latex', 'FontSize', 12);
197 title('Coordinate $\theta(t)$', 'Interpreter', 'latex', 'FontSize', 13);
198 grid on;
199
200 subplot(2,3,2);
201 plot(t_above_Hp, X_above_Hp(:,2), 'r', 'LineWidth', 1.5);
202 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
203 ylabel('$\ell(t)$ [m]', 'Interpreter', 'latex', 'FontSize', 12);
204 title('Coordinate $\ell(t)$', 'Interpreter', 'latex', 'FontSize', 13);
205 grid on;
206
207 subplot(2,3,4);
208 plot(t_above_Hp, X_above_Hp(:,3), 'b', 'LineWidth', 1.5);
209 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
210 ylabel('$\dot{\theta}(t)$ [rad/s]', 'Interpreter', 'latex', 'FontSize', 12);
```

```matlab
211 title('Velocity $\dot{\theta}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
212 grid on;
213
214 subplot(2,3,5);
215 plot(t_above_Hp, X_above_Hp(:,4), 'r', 'LineWidth', 1.5);
216 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
217 ylabel('$\dot{\ell}(t)$ [m/s]', 'Interpreter', 'latex', 'FontSize', 12);
218 title('Velocity $\dot{\ell}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
219 grid on;
220
221 subplot(2,3,[3,6]);
222 plot(t_above_Hp, H_above_Hp, 'k', 'LineWidth', 2);
223 hold on;
224 yline(Hstar_above_Hp, 'g--', 'LineWidth', 2);
225 yline(Hp, 'c-.', 'LineWidth', 2);
226 yline(Hmin, 'm:', 'LineWidth', 1.5);
227 yline(H0, 'b-.', 'LineWidth', 1.5);
228 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
229 ylabel('Energy [J]', 'Interpreter', 'latex', 'FontSize', 12);
230 title('Total Energy $H(q,\dot{q})$', 'Interpreter', 'latex', 'FontSize', 13);
231 legend('$H(t)$', '$H^*$', '$H_p$', '$H_{min}$', '$H_0$', ...
232     'Interpreter', 'latex', 'Location', 'best', 'FontSize', 10);
233 grid on;
234
235 sgtitle(sprintf('Case (b): $H^* = %.2f$ J $> H_p = %.2f$ J (Rotation Expected)', ...
236     Hstar_above_Hp, Hp), 'Interpreter', 'latex', 'FontSize', 15);
237
238 %% Case C
239 Hstar_c1 = 49.5; % (energy should decrease)
240 Hstar_c2 = 51.5; % (energy should increase)
241 Tend_c = 300;
242
243 [t_c1, X_c1, H_c1] = run_simulation(x0, params, Hstar_c1, Tend_c);
244 [t_c2, X_c2, H_c2] = run_simulation(x0, params, Hstar_c2, Tend_c);
245
246 fig_c1 = figure('Name', 'Figure 5: Case (c1)', 'Position', [50, 50, 1400, 800]);
247
248 subplot(2,3,1);
249 plot(t_c1, X_c1(:,1), 'b', 'LineWidth', 1.5);
250 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
251 ylabel('$\theta(t)$ [rad]', 'Interpreter', 'latex', 'FontSize', 12);
252 title('Coordinate $\theta(t)$', 'Interpreter', 'latex', 'FontSize', 13);
253 grid on;
254
255 subplot(2,3,2);
256 plot(t_c1, X_c1(:,2), 'r', 'LineWidth', 1.5);
257 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
258 ylabel('$\ell(t)$ [m]', 'Interpreter', 'latex', 'FontSize', 12);
259 title('Coordinate $\ell(t)$', 'Interpreter', 'latex', 'FontSize', 13);
260 grid on;
261
262 subplot(2,3,4);
263 plot(t_c1, X_c1(:,3), 'b', 'LineWidth', 1.5);
264 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
265 ylabel('$\dot{\theta}(t)$ [rad/s]', 'Interpreter', 'latex', 'FontSize', 12);
266 title('Velocity $\dot{\theta}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
267 grid on;
268
269 subplot(2,3,5);
270 plot(t_c1, X_c1(:,4), 'r', 'LineWidth', 1.5);
271 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
272 ylabel('$\dot{\ell}(t)$ [m/s]', 'Interpreter', 'latex', 'FontSize', 12);
273 title('Velocity $\dot{\ell}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
274 grid on;
275
276 subplot(2,3,[3,6]);
277 plot(t_c1, H_c1, 'k', 'LineWidth', 2);
278 hold on;
279 yline(Hstar_c1, 'g--', 'LineWidth', 2);
280 yline(H0, 'b-.', 'LineWidth', 1.5);
281 yline(Hp, 'c:', 'LineWidth', 1.5);
282 xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
283 ylabel('Energy [J]', 'Interpreter', 'latex', 'FontSize', 12);
```

```matlab
284  title('Total Energy $H(q,\dot{q})$', 'Interpreter', 'latex', 'FontSize', 13);
285  legend('$H(t)$', '$H^*$', '$H_0$', '$H_p$', 'Interpreter', 'latex', 'Location', 'best');
286  grid on;
287
288  sgtitle(sprintf('Case (c1): $H_p < H^* = %.2f$ J $< H_0$ (Energy Decreasing)', Hstar_c1)
         , ...
289      'Interpreter', 'latex', 'FontSize', 15);
290
291  fig_c2 = figure('Name', 'Figure 6: Case (c2)', 'Position', [100, 50, 1400, 800]);
292
293  subplot(2,3,1);
294  plot(t_c2, X_c2(:,1), 'b', 'LineWidth', 1.5);
295  xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
296  ylabel('$\theta(t)$ [rad]', 'Interpreter', 'latex', 'FontSize', 12);
297  title('Coordinate $\theta(t)$', 'Interpreter', 'latex', 'FontSize', 13);
298  grid on;
299
300  subplot(2,3,2);
301  plot(t_c2, X_c2(:,2), 'r', 'LineWidth', 1.5);
302  xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
303  ylabel('$\ell(t)$ [m]', 'Interpreter', 'latex', 'FontSize', 12);
304  title('Coordinate $\ell(t)$', 'Interpreter', 'latex', 'FontSize', 13);
305  grid on;
306
307  subplot(2,3,4);
308  plot(t_c2, X_c2(:,3), 'b', 'LineWidth', 1.5);
309  xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
310  ylabel('$\dot{\theta}(t)$ [rad/s]', 'Interpreter', 'latex', 'FontSize', 12);
311  title('Velocity $\dot{\theta}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
312  grid on;
313
314  subplot(2,3,5);
315  plot(t_c2, X_c2(:,4), 'r', 'LineWidth', 1.5);
316  xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
317  ylabel('$\dot{\ell}(t)$ [m/s]', 'Interpreter', 'latex', 'FontSize', 12);
318  title('Velocity $\dot{\ell}(t)$', 'Interpreter', 'latex', 'FontSize', 13);
319  grid on;
320
321  subplot(2,3,[3,6]);
322  plot(t_c2, H_c2, 'k', 'LineWidth', 2);
323  hold on;
324  yline(Hstar_c2, 'g--', 'LineWidth', 2);
325  yline(H0, 'b-.', 'LineWidth', 1.5);
326  xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 12);
327  ylabel('Energy [J]', 'Interpreter', 'latex', 'FontSize', 12);
328  title('Total Energy $H(q,\dot{q})$', 'Interpreter', 'latex', 'FontSize', 13);
329  legend('$H(t)$', '$H^*$', '$H_0$', 'Interpreter', 'latex', 'Location', 'best');
330  grid on;
331
332  sgtitle(sprintf('Case (c2): $H_0 < H^* = %.2f$ J $< 52.7$ J (Energy Increasing)', ...
         Hstar_c2), ...
333      'Interpreter', 'latex', 'FontSize', 15);
334
335  fig_compare = figure('Name', 'Figure 7: Case (c) Comparison', 'Position', [150, 100,
         900, 550]);
336
337  plot(t_c1, H_c1, 'b', 'LineWidth', 2);
338  hold on;
339  plot(t_c2, H_c2, 'r', 'LineWidth', 2);
340  yline(Hstar_c1, 'b--', 'LineWidth', 1.5);
341  yline(Hstar_c2, 'r--', 'LineWidth', 1.5);
342  yline(H0, 'k:', 'LineWidth', 2);
343  xlabel('Time $t$ [s]', 'Interpreter', 'latex', 'FontSize', 14);
344  ylabel('Energy $H$ [J]', 'Interpreter', 'latex', 'FontSize', 14);
345  title('Case (c): Energy Convergence Comparison', 'Interpreter', 'latex', 'FontSize', 16)
         ;
346  legend(sprintf('c1: $H(t)$, $H^*=%.2f$ J', Hstar_c1), ...
347        sprintf('c2: $H(t)$, $H^*=%.2f$ J', Hstar_c2), ...
348        '$H^*$ (c1)', '$H^*$ (c2)', '$H_0$', ...
349        'Interpreter', 'latex', 'Location', 'best', 'FontSize', 11);
350  grid on;
351
352  % Calculate convergence times
```

```
353  tol_conv = 0.5;  % Within 0.5 J of target
354  error_c1 = abs(H_c1 - Hstar_c1);
355  error_c2 = abs(H_c2 - Hstar_c2);
356
357  idx_c1_conv = find(error_c1 < tol_conv, 1, 'first');
358  idx_c2_conv = find(error_c2 < tol_conv, 1, 'first');
359  %% Functions
360
361  function Hmin = calculate_Hmin(p)
362      Hmin = (p.M + p.m) * p.g * p.Ro - (p.m^2 * p.g^2) / (4 * p.k);
363  end
364
365  function H = calculate_H(x, p)
366      theta = x(1); ell = x(2); dtheta = x(3); dell = x(4);
367      Jeff = (p.M * (3*p.Ro^2 + p.Ri^2)) / 2 + ...
368             p.m * (p.Ro^2 + ell^2 - 2*p.Ro*ell*sin(theta));
369      T = 0.5 * Jeff * dtheta^2 + 0.5 * p.m * dell^2 + ...
370          p.m * p.Ro * dell * dtheta * cos(theta);
371      V = (p.M + p.m) * p.g * p.Ro + p.m * p.g * ell * sin(theta) + p.k * ell^2;
372      H = T + V;
373  end
374
375  function dx = system_ode(t, x, p, Hstar)
376      theta = x(1); ell = x(2); dtheta = x(3); dell = x(4);
377      H = calculate_H(x, p);
378      F = -p.gamma * dell * (H - Hstar);
379
380      Jeff = (p.M * (3*p.Ro^2 + p.Ri^2)) / 2 + ...
381             p.m * (p.Ro^2 + ell^2 - 2*p.Ro*ell*sin(theta));
382      M11 = Jeff;
383      M12 = p.m * p.Ro * cos(theta);
384      M22 = p.m;
385
386      C1 = 2*p.m*(ell - p.Ro*sin(theta))*dell*dtheta - p.m*p.Ro*ell*cos(theta)*dtheta^2;
387      C2 = -p.m*ell*dtheta^2;
388
389      G1 = p.m*p.g*ell*cos(theta);
390      G2 = p.m*p.g*sin(theta) + 2*p.k*ell;
391
392      Qnc1 = F * p.Ro * cos(theta);
393      Qnc2 = F;
394
395      RHS1 = Qnc1 - C1 - G1;
396      RHS2 = Qnc2 - C2 - G2;
397
398      detM = M11*M22 - M12^2;
399      ddtheta = (M22*RHS1 - M12*RHS2) / detM;
400      ddell = (-M12*RHS1 + M11*RHS2) / detM;
401
402      dx = [dtheta; dell; ddtheta; ddell];
403  end
404
405  function [t, X, H] = run_simulation(x0, params, Hstar, Tend)
406      opts = odeset('RelTol', 1e-9, 'AbsTol', 1e-11, 'MaxStep', 0.1);
407      [t, X] = ode45(@(t,x) system_ode(t, x, params, Hstar), [0, Tend], x0, opts);
408      H = zeros(length(t), 1);
409      for i = 1:length(t)
410          H(i) = calculate_H(X(i,:)', params);
411      end
412  end
413
414  function Hp = find_threshold_Hp(x0, params, Hmin)
415      Hstar_low = Hmin + 0.05;
416      Hstar_high = 60;
417      tol = 0.1;
418      Tend = 400;
419
420      while (Hstar_high - Hstar_low) > tol
421          Hstar_test = (Hstar_low + Hstar_high) / 2;
422          [~, X_test, ~] = run_simulation(x0, params, Hstar_test, Tend);
423          theta_test = X_test(:,1);
424
425          if is_oscillating(theta_test)
```

```matlab
426            Hstar_low = Hstar_test;
427        else
428            Hstar_high = Hstar_test;
429        end
430    end
431    Hp = (Hstar_low + Hstar_high) / 2;
432 end
433
434 function is_osc = is_oscillating(theta)
435    dtheta_sign = sign(diff(theta));
436    dtheta_sign(dtheta_sign == 0) = 1;
437    sign_changes = sum(abs(diff(dtheta_sign)) > 0);
438    theta_range = max(theta) - min(theta);
439    is_osc = (sign_changes > 20) || (theta_range < 4*pi);
440 end
441
442 function behavior = get_behavior_string(theta)
443    if is_oscillating(theta)
444        behavior = 'OSCILLATION (bounded theta)';
445    else
446        behavior = 'ROTATION (unbounded theta)';
447    end
448 end
```

Listing 1: MATLAB simulation script for oscillating mass on a wheel problem

4 │ Appendix B – Question 2 Matlab Code

```matlab
clear all; close all; clc;

N = 1e6;
beta = 1/80;
gamma_param = 20;
Tend = 400;
x0 = [N - 10; 10; 0];

mu1 = 8;
theta1 = gamma_param / (beta + mu1);
v_star1 = 1/theta1;
c_star1 = (beta/gamma_param) * (theta1 - 1);

odefun1 = @(t, x) [beta*N - beta*x(1) - (gamma_param/N)*x(2)*x(1);
                   -beta*x(2) + (gamma_param/N)*x(2)*x(1) - mu1*x(2);
                   -beta*x(3) + mu1*x(2)];

[t1, x1] = ode45(odefun1, [0 Tend], x0);

figure('Position', [100 100 800 500]);
plot(t1, x1(:,1)/N, 'b-', 'LineWidth', 2); hold on;
plot(t1, x1(:,2)/N, 'r-', 'LineWidth', 2);
plot(t1, x1(:,3)/N, 'g-', 'LineWidth', 2);
yline(v_star1, 'b--', 'LineWidth', 1);
yline(c_star1, 'r--', 'LineWidth', 1);
xlabel('Time (days)', 'FontSize', 12);
ylabel('Population fraction', 'FontSize', 12);
legend('v(t) - Vulnerable', 'c(t) - Contaminated', 'i(t) - Immune', ...
       'v^* equilibrium', 'c^* equilibrium', 'Location', 'east');
title(['Parameter Set 1: \theta = ', num2str(theta1, '%.3f'), ' > 1 (Endemic)'], '
    FontSize', 14);
grid on;
saveas(gcf, 'sim_set1_time.png');

figure('Position', [100 100 600 500]);
plot(x1(:,1), x1(:,2), 'b-', 'LineWidth', 2); hold on;
plot(x1(1,1), x1(1,2), 'go', 'MarkerSize', 10, 'MarkerFaceColor', 'g');
plot(v_star1*N, c_star1*N, 'r*', 'MarkerSize', 15, 'LineWidth', 2);
xlabel('x_1 (Vulnerable)', 'FontSize', 12);
ylabel('x_2 (Contaminated)', 'FontSize', 12);
legend('Trajectory', 'Initial condition', 'Endemic equilibrium', 'Location', 'northeast'
    );
title('Phase Portrait - Parameter Set 1', 'FontSize', 14);
grid on;
saveas(gcf, 'sim_set1_phase.png');

mu2 = 20;
theta2 = gamma_param / (beta + mu2);

odefun2 = @(t, x) [beta*N - beta*x(1) - (gamma_param/N)*x(2)*x(1);
                   -beta*x(2) + (gamma_param/N)*x(2)*x(1) - mu2*x(2);
                   -beta*x(3) + mu2*x(2)];

[t2, x2] = ode45(odefun2, [0 Tend], x0);

figure('Position', [100 100 800 500]);
plot(t2, x2(:,1)/N, 'b-', 'LineWidth', 2); hold on;
plot(t2, x2(:,2)/N, 'r-', 'LineWidth', 2);
plot(t2, x2(:,3)/N, 'g-', 'LineWidth', 2);
xlabel('Time (days)', 'FontSize', 12);
ylabel('Population fraction', 'FontSize', 12);
legend('v(t) - Vulnerable', 'c(t) - Contaminated', 'i(t) - Immune', 'Location', 'east');
title(['Parameter Set 2: \theta = ', num2str(theta2, '%.3f'), ' < 1 (Disease-free)'], '
    FontSize', 14);
grid on;
saveas(gcf, 'sim_set2_time.png');

figure('Position', [100 100 600 500]);
plot(x2(:,1), x2(:,2), 'b-', 'LineWidth', 2); hold on;
plot(x2(1,1), x2(1,2), 'go', 'MarkerSize', 10, 'MarkerFaceColor', 'g');
```

```matlab
68  plot(N, 0, 'r*', 'MarkerSize', 15, 'LineWidth', 2);
69  xlabel('x_1 (Vulnerable)', 'FontSize', 12);
70  ylabel('x_2 (Contaminated)', 'FontSize', 12);
71  legend('Trajectory', 'Initial condition', 'Disease-free equilibrium', 'Location', '
        northeast');
72  title('Phase Portrait - Parameter Set 2', 'FontSize', 14);
73  grid on;
74  saveas(gcf, 'sim_set2_phase.png');
75
76  mu = 8;
77  theta = gamma_param / (beta + mu);
78  efficacy = 0.9;
79  u_critical = (1/efficacy) * (1 - 1/theta);
80
81  figure('Position', [100 100 1000 400]);
82
83  subplot(1,2,1);
84  u = 0.5;
85  odefun_vacc = @(t, x) [beta*N - beta*x(1) - (gamma_param/N)*x(2)*x(1) - efficacy*beta*u*
        N;
86                          -beta*x(2) + (gamma_param/N)*x(2)*x(1) - mu*x(2);
87                          -beta*x(3) + mu*x(2) + efficacy*beta*u*N];
88  [t_v1, x_v1] = ode45(odefun_vacc, [0 Tend], x0);
89  plot(t_v1, x_v1(:,1)/N, 'b-', 'LineWidth', 2); hold on;
90  plot(t_v1, x_v1(:,2)/N, 'r-', 'LineWidth', 2);
91  plot(t_v1, x_v1(:,3)/N, 'g-', 'LineWidth', 2);
92  xlabel('Time (days)', 'FontSize', 11);
93  ylabel('Population fraction', 'FontSize', 11);
94  title(sprintf('u = %.1f < u_c = %.3f', u, u_critical), 'FontSize', 12);
95  legend('v', 'c', 'i', 'Location', 'east');
96  grid on;
97
98  subplot(1,2,2);
99  u = 0.7;
100 odefun_vacc = @(t, x) [beta*N - beta*x(1) - (gamma_param/N)*x(2)*x(1) - efficacy*beta*u*
        N;
101                         -beta*x(2) + (gamma_param/N)*x(2)*x(1) - mu*x(2);
102                         -beta*x(3) + mu*x(2) + efficacy*beta*u*N];
103 [t_v2, x_v2] = ode45(odefun_vacc, [0 Tend], x0);
104 plot(t_v2, x_v2(:,1)/N, 'b-', 'LineWidth', 2); hold on;
105 plot(t_v2, x_v2(:,2)/N, 'r-', 'LineWidth', 2);
106 plot(t_v2, x_v2(:,3)/N, 'g-', 'LineWidth', 2);
107 xlabel('Time (days)', 'FontSize', 11);
108 ylabel('Population fraction', 'FontSize', 11);
109 title(sprintf('u = %.1f > u_c = %.3f', u, u_critical), 'FontSize', 12);
110 legend('v', 'c', 'i', 'Location', 'east');
111 grid on;
112
113 sgtitle('Effect of Vaccination Rate (90% Efficacy)', 'FontSize', 14);
114 saveas(gcf, 'vacc_comparison.png');
115
116 figure('Position', [100 100 1000 400]);
117
118 subplot(1,2,1);
119 u = 0.5;
120 c_star_vacc = (beta/gamma_param) * (theta*(1 - efficacy*u) - 1);
121 if c_star_vacc < 0; c_star_vacc = 0; end
122 plot(t_v1, x_v1(:,2)/N, 'r-', 'LineWidth', 2); hold on;
123 if c_star_vacc > 0
124     yline(c_star_vacc, 'r--', 'LineWidth', 1.5, 'Label', sprintf('c^* = %.4f',
        c_star_vacc));
125 end
126 xlabel('Time (days)', 'FontSize', 11);
127 ylabel('Contaminated fraction c(t)', 'FontSize', 11);
128 title(sprintf('u = 0.5 < u_c: Zoomed view of c(t)'), 'FontSize', 12);
129 grid on;
130 ylim([0, max(x_v1(:,2)/N)*1.1 + 0.001]);
131
132 subplot(1,2,2);
133 u = 0.7;
134 c_star_vacc2 = (beta/gamma_param) * (theta*(1 - efficacy*u) - 1);
135 if c_star_vacc2 < 0; c_star_vacc2 = 0; end
136 plot(t_v2, x_v2(:,2)/N, 'r-', 'LineWidth', 2); hold on;
```

```
137  yline(0, 'r--', 'LineWidth', 1.5, 'Label', 'c^* = 0');
138  xlabel('Time (days)', 'FontSize', 11);
139  ylabel('Contaminated fraction c(t)', 'FontSize', 11);
140  title(sprintf('u = 0.7 > u_c: Zoomed view of c(t)'), 'FontSize', 12);
141  grid on;
142  ylim([0, max(x_v2(:,2)/N)*1.1 + 0.001]);
143  sgtitle('Zoomed View of Contaminated Fraction c(t)', 'FontSize', 14);
144  saveas(gcf, 'vacc_comparison_zoomed.png');
145
146  u_c_100 = 1 - 1/theta;
147  u_range = linspace(0, 1, 200);
148  c_star_stable = zeros(size(u_range));
149
150  for i = 1:length(u_range)
151      u = u_range(i);
152      if u < u_c_100
153          c_star_stable(i) = (beta/gamma_param) * (theta - 1 - theta*u);
154      else
155          c_star_stable(i) = 0;
156      end
157  end
158
159  figure('Position', [100 100 700 500]);
160  plot(u_range(u_range < u_c_100), c_star_stable(u_range < u_c_100), 'b-', 'LineWidth',
         2.5);
161  hold on;
162  plot(u_range(u_range >= u_c_100), c_star_stable(u_range >= u_c_100), 'b-', 'LineWidth',
         2.5);
163  plot(u_range(u_range < u_c_100), zeros(sum(u_range < u_c_100), 1), 'b--', 'LineWidth',
         2);
164  xline(u_c_100, 'r--', 'LineWidth', 2);
165  plot(u_c_100, 0, 'ro', 'MarkerSize', 10, 'MarkerFaceColor', 'r');
166  xlabel('Vaccination rate u', 'FontSize', 14);
167  ylabel('Equilibrium infected fraction c^*', 'FontSize', 14);
168  title('Bifurcation Diagram: Transcritical Bifurcation', 'FontSize', 14);
169  legend('Stable equilibrium', '', 'Unstable equilibrium', ...
170          ['u_c = ', num2str(u_c_100, '%.3f')], 'Bifurcation point', ...
171          'Location', 'northeast');
172  grid on;
173  ylim([-0.0001, max(c_star_stable)*1.1]);
174  saveas(gcf, 'bifurcation.png');
175
176  figure('Position', [100 100 600 500]);
177  fill([0 1 0], [0 0 1], [0.9 0.95 1], 'EdgeColor', 'b', 'LineWidth', 2);
178  hold on;
179  [V, C] = meshgrid(0.05:0.1:0.95, 0.05:0.1:0.95);
180  beta_n = 0.1; gamma_n = 0.3; mu_n = 0.1;
181  Vdot = beta_n*(1-V) - gamma_n*C.*V;
182  Cdot = gamma_n*C.*V - (beta_n + mu_n)*C;
183  mask = (V + C <= 1) & (V >= 0) & (C >= 0);
184  Vdot(~mask) = NaN;
185  Cdot(~mask) = NaN;
186  quiver(V, C, Vdot, Cdot, 'k', 'LineWidth', 1);
187  plot([0 1], [0 0], 'b-', 'LineWidth', 2);
188  plot([0 0], [0 1], 'b-', 'LineWidth', 2);
189  plot([0 1], [1 0], 'b-', 'LineWidth', 2);
190  text(0.5, -0.08, 'c = 0: c'' = 0', 'FontSize', 11, 'HorizontalAlignment', 'center');
191  text(-0.15, 0.5, 'v = 0: v'' > 0', 'FontSize', 11, 'Rotation', 90, 'HorizontalAlignment'
         , 'center');
192  text(0.6, 0.5, 'v + c = 1', 'FontSize', 11, 'Rotation', -45);
193  text(0.4, 0.25, 'I', 'FontSize', 18, 'FontWeight', 'bold');
194  xlabel('v (vulnerable fraction)', 'FontSize', 12);
195  ylabel('c (contaminated fraction)', 'FontSize', 12);
196  title('Invariant Set I with Vector Field', 'FontSize', 14);
197  axis equal;
198  xlim([-0.1 1.1]);
199  ylim([-0.15 1.1]);
200  grid on;
201  saveas(gcf, 'invariant_set.png');
```

Listing 2: MATLAB simulation script for VCI model dynamics