Final Project - Acrobot Swing Up Control

Raj Harshit Srirangam | Rohith Poola | Kodanda Rama Naidu Dumpala | Varun Reddy Patlolla

1 Project Overview

1.1 Background

The Acrobot is a two-link underactuated robotic system resembling a gymnast swinging on a high bar, where only the second joint is actuated, and the first joint is passive. Due to its non-linear, unstable, and underactuated nature, the Acrobot presents a challenging control problem.

1.2 Objectives & Approach

The primary goal of this project is to design and implement a control strategy that enables the Acrobot to perform a swing-up maneuver, moving from the stable downward position to the unstable upright position, and then stabilize it at the top. This requires careful handling of the system's non-linear dynamics and energy-based behavior.

Initially, the dynamics of the Acrobot system is linearized about the upright equilibrium point, and then a Linear Quadratic Regulator (LQR) is designed to stabilize it. For the swing-up phase, an energy-based controller is developed to gradually increase the total energy of Acrobot, enabling it to reach the upright configuration. Once the acrobot is near the top, control is switched to the LQR for stabilization. This project allows us to extend what we learnt in class and apply it on a highly non linear system using energy shaping, optimal control, and switching strategies.

1.3 Simulator - Drake

The simulator used for this project is Drake (Dynamic Robot Autonomy and Kernel Environment), an open-source toolbox developed by the Robot Locomotion Group at MIT. Drake provides a rich set of tools for modeling, simulating, and controlling robotic systems, particularly those with complex dynamics and constraints. It offers symbolic computation for deriving equations of motion, built-in solvers for optimal control, and seamless support for system linearization and LQR design. Its support for multibody dynamics and feedback systems made it especially suitable for implementing and analyzing the nonlinear, underactuated Acrobot system in this project.

2 System Modeling

2.1 Acrobot Description

System Parameters:

- l_1, l_2: link lengths
- m_1, m_2: link masses
- I_1, I_2: link inertia about pivots
- g: gravitational acceleration

Simulation parameter values:

- $l_1 = 1\text{m}$, $l_2 = 2\text{m}$
- $m_1 = 1\text{kg}$, $m_2 = 1\text{kg}$
- $I_1 = 0.0833\ \text{kg}\cdot\text{m}^2, \quad I_2 = 0.3333\ \text{kg}\cdot\text{m}^2$
- $g = 9.8\ \text{m/s}^2$

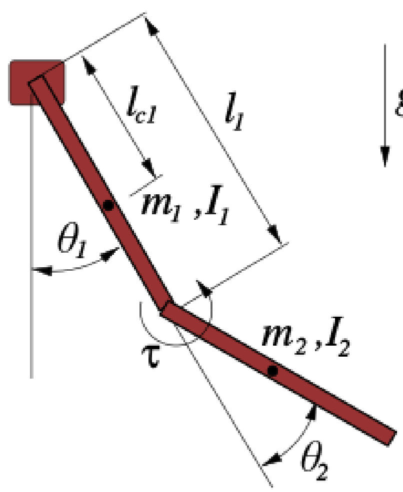


Figure 1: Acrobot schematic

2.2 Lagrange Equations & Manipulator Dynamics

2.2.1 Kinematic Equations

$$x_1 = l_1 \sin\theta_1, \qquad y_1 = -l_1 \cos\theta_1$$

$$x_2 = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2), \qquad y_2 = -l_1 \cos\theta_1 - l_2 \cos(\theta_1 + \theta_2)$$

2.2.2 Energy Equations

$$\text{Total Kinetic Energy:}\quad T = T_1 + T_2$$

$$T_1 = \frac{1}{2} I_1 \dot{\theta}_1^2$$

$$T_2 = \frac{1}{2}\left(m_2 l_1^2 + I_2 + 2m_2 l_1 l_{c2}\right)\dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + \left(I_2 + m_2 l_1 l_{c2}\cos\theta_2\right)\dot{\theta}_1\dot{\theta}_2$$

$$\text{Total Potential Energy:}\quad V = -m_1 g l_{c1}\cos\theta_1 - m_2 g\left(l_1\cos\theta_1 + l_2\cos\left(\theta_1+\theta_2\right)\right)$$

2.2.3 Lagrange Equations

$$\mathcal{L} = T - V$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) - \frac{\partial \mathcal{L}}{\partial \theta_i} = \tau_i, \quad i = 1, 2$$

$$\tau = \begin{bmatrix} 0 \\ u \end{bmatrix}$$

Equations of motion:

$$\begin{aligned}&\left(I_1 + I_2 + m_2 l_1^2 + 2m_2 l_1 l_{c2}\cos\theta_2\right)\ddot{q}_1 + \left(I_2 + m_2 l_1 l_{c2}\cos\theta_2\right)\ddot{q}_2 \\ &\quad - 2m_2 l_1 l_{c2}\sin\theta_2 \dot{q}_1\dot{q}_2 - m_2 l_1 l_{c2}\sin\theta_2\dot{\theta}_2^2 + \left(m_1 l_{c1} + m_2 l_1\right) g\sin\theta_1 + m_2 g l_2 \sin\left(\theta_1+\theta_2\right) = 0\end{aligned}$$

$$\left(I_2 + m_2 l_1 l_{c2}\cos\theta_2\right)\ddot{q}_1 + I_2\ddot{q}_2 + m_2 l_1 l_{c2}\sin\theta_2\dot{q}_1^2 + m_2 g l_2\sin\left(\theta_1+\theta_2\right) = \tau$$

2.2.4 Manipulator Equation Form

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \tau$$

$$M(\theta) = \begin{bmatrix} I_1 + m_2 l_1^2 + I_2 + 2m_2 l_1 l_{c2}\cos\theta_2 & I_2 + m_2 l_1 l_{c2}\cos\theta_2 \\ I_2 + m_2 l_1 l_{c2}\cos\theta_2 & I_2 \end{bmatrix}$$

$$C(\theta,\dot{\theta})\dot{\theta} = \begin{bmatrix} -m_2 l_1 l_{c2}\sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ m_2 l_1 l_{c2}\sin\theta_2\dot{\theta}_1^2 \end{bmatrix}$$

$$C(\theta,\dot{\theta}) = \begin{bmatrix} -2m_2 l_1 l_{c2}\sin\theta_2\dot{\theta}_2 & -m_2 l_1 l_{c2}\sin\theta_2\dot{\theta}_2 \\ m_2 l_1 l_{c2}\sin\theta_2\dot{\theta}_1 & 0 \end{bmatrix}$$

$$G(\theta) = \begin{bmatrix} m_1 g l_{c1}\sin\theta_1 - m_2 g\left(l_1\sin\theta_1 + l_2\sin\theta_{1+2}\right) \\ m_2 g l_2 \sin\theta_{1+2} \end{bmatrix}$$

2.2.5 State Space Representation

$$\ddot{\theta} = M^{-1}(\theta)\left[-C(\theta,\dot{\theta})\dot{\theta} - G(\theta) + Bu\right]$$

3 Controller Development

The primary control objective is to bring the system from its natural downward hanging position (stable equilibrium) to the upright position (unstable equilibrium) and then stabilize it there. To achieve this, a hybrid control strategy is employed.

Linear Quadratic Regulator (LQR) is a widely used optimal control technique for linear systems. It minimizes a quadratic cost function that penalizes deviations from a desired state and the control effort. In this project, LQR is designed around the linearization of the Acrobot dynamics at the upright equilibrium point. Although effective in stabilizing the system near this point, LQR is inherently a local controller. It relies on the assumption that the system operates within a small neighborhood of the linearization point. When the initial state is far from the upright position, such as when the robot is hanging down, LQR alone is incapable of generating the required energy or motion to reach the desired configuration.

To overcome this limitation, an energy-based swing-up controller is introduced. The swing-up controller takes advantage of the natural dynamics of the system to increase the total energy of the Acrobot. It is achieved by injecting control input that shape the energy of the system toward the desired level corresponding to the upright position. Once the total energy is sufficiently close to the target energy and the state lies within a specified region near the upright, the control law switches from the swing-up controller to the LQR controller.

This hybrid approach leverages the strengths of both controllers: the nonlinear energy shaping controller to move the Acrobot into the basin of attraction of the upright equilibrium, and the LQR to ensure efficient and stable convergence to the final goal state. A well-defined switching logic is implemented to ensure a smooth transition between the two control modes.

4 LQR Stabilization

4.1 Linearization Around Upright Equilibrium

Equilibrium point at upright position:

$$X = \begin{bmatrix} \pi & 0 & 0 & 0 \end{bmatrix}^T$$

$$\begin{aligned} \dot{X} &= \begin{bmatrix} \dot{\theta} \\ M^{-1}(\theta)\left[-C(\theta,\dot{\theta})\dot{\theta} - G(\theta) + Bu\right] \end{bmatrix} \\ &\approx A(x - x^*) + B(u - u^*) \end{aligned}$$

$$\begin{aligned} f_1 &= \dot{\theta} \\ f_2 &= M^{-1}(\theta)\left[-C(\theta,\dot{\theta})\dot{\theta} - G(\theta) + Bu\right] \end{aligned}$$

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \dot{\theta}} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \dot{\theta}} \end{bmatrix}$$

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix}$$

4.1.1 Linearized terms

$$A_{11} = \frac{\partial f_1}{\partial \theta} = \frac{\partial \dot{\theta}}{\partial \theta} = 0$$

$$A_{12} = \frac{\partial f_1}{\partial \dot{\theta}} = \frac{\partial \dot{\theta}}{\partial \dot{\theta}} = I$$

$$\begin{aligned} A_{21} = \frac{\partial f_2}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(M^{-1}(\theta)\left[-C(\theta,\dot{\theta})\dot{\theta} - G(\theta) + Bu\right]\right) \\ &= \frac{\partial M^{-1}(\theta)}{\partial \theta}\left(Bu - C(\theta,\dot{\theta})\dot{\theta} - G(\theta)\right) \\ &\quad + M^{-1}(\theta)\left(\frac{\partial B}{\partial \theta}u - \frac{\partial C(\theta,\dot{\theta})}{\partial \theta}\dot{\theta} - \frac{\partial G(\theta)}{\partial \theta}\right) \\ &= 0 - M^{-1}(\theta)\frac{\partial G(\theta)}{\partial \theta} \\ &= \begin{bmatrix} -g(m_1 l_{c1} + m_2 l_1 + m_2 l_2) & -m_2 g l_2 \\ -m_2 g l_2 & -m_2 g l_2 \end{bmatrix}_{x=x^*,\, u=u^*} \end{aligned}$$

$$
\begin{aligned}
A_{22} = \frac{\partial f_2}{\partial \dot{\theta}} &= \frac{\partial}{\partial \dot{\theta}} \left(M^{-1}(\theta) \left[-C(\theta, \dot{\theta})\dot{\theta} - G(\theta) + Bu \right] \right) \\
&= \frac{\partial M^{-1}(\theta)}{\partial \dot{\theta}} \left(Bu - C(\theta, \dot{\theta})\dot{\theta} - G(\theta) \right) \\
&\quad + M^{-1}(\theta) \left(\frac{\partial B}{\partial \dot{\theta}} u - \frac{\partial C(\theta, \dot{\theta})}{\partial \dot{\theta}} \dot{\theta} - \frac{\partial G(\theta)}{\partial \dot{\theta}} \right) \\
&= 0 - M^{-1}(\theta) \frac{\partial C(\theta, \dot{\theta})}{\partial \dot{\theta}} \dot{\theta} \\
&= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}_{x=x^*,\, u=u^*}
\end{aligned}
$$

$$
B_1 = \frac{\partial f_1}{\partial u} = \frac{\partial \dot{\theta}}{\partial u} = 0
$$

$$
B_2 = \frac{\partial f_2}{\partial u} = M^{-1}(\theta) \cdot B
$$

4.1.2 State and Control matrices:

$$
A = \begin{bmatrix} 0 & I \\ -H^{-1} \dfrac{\partial G}{\partial \theta} & -H^{-1} C \end{bmatrix}_{x=x^*,\, u=u^*}
$$

$$
B = \begin{bmatrix} 0 \\ H^{-1} B \end{bmatrix}_{x=x^*,\, u=u^*}
$$

Substituting the system parameters:

$$
A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 12.6292 & -12.6926 & 0 & 0 \\ -14.7488 & 29.6119 & 0 & 0 \end{bmatrix}
$$

$$
B = \begin{bmatrix} 0 \\ 0 \\ -3.0147 \\ 6.0332 \end{bmatrix}
$$

Controllability matrix:

$$
P = \begin{bmatrix} 0 & -3.0147 & 0 & -114.6497 \\ 0 & 6.0332 & 0 & 223.1171 \\ -3.0147 & 0 & -114.6497 & 0 \\ 6.0332 & 0 & 223.1171 & 0 \end{bmatrix}
$$

We can see that the controllability matrix has a rank of 4. Thus, it is controllable. But, by carefully observing the rows, we can see that thet are almost linearly dependent i.e a slight change in the entries could make it uncontrollable. This suggests weak controllability which adds to the difficulty of the problem. The "weakness" of controllability can be quantified by taking the SVD of the P matrix

$$
\mathrm{SVD}(P): \quad S = \begin{bmatrix} 250.940807 & 250.940807 & 0.0760446155 & 0.0760446155 \end{bmatrix}
$$

That last two entries are a lot smaller compared to the others which means that the controllability is "weak".

4.2 LQR Design and Implementation

Once the Acrobot reaches the proximity of the upright equilibrium, the system dynamics can be accurately approximated using a linearized model as defined above

where A and B are the Jacobian matrices computed around the upright equilibrium point $x^* = [\pi, 0, 0, 0]^T$, and $u^* = 0$ is the corresponding control input. This linearized model captures the local behavior of the nonlinear Acrobot system near the goal configuration and serves as the foundation for designing a stabilizing controller.

To regulate the system and ensure convergence to the upright position, a Linear Quadratic Regulator (LQR) is employed. LQR provides an optimal state feedback control law of the form $u = -Kx$, where the gain matrix K minimizes a quadratic cost function:

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt$$

In this implementation, the weighting matrices Q and R are both chosen as identity matrices:

$$Q = I, \quad R = I$$

This choice reflects equal emphasis on minimizing both the state deviations and the control effort, leading to a balanced and robust controller. While more sophisticated tuning of Q and R can emphasize certain states or penalize aggressive control, the identity matrices serve as a reasonable starting point that ensures all states and inputs are treated with equal importance.

The resulting gain matrix K is computed by solving the continuous-time algebraic Riccati equation. Once the gain is obtained, the LQR controller is activated when the Acrobot state enters a predefined region around the upright, taking over from the energy shaping swing-up controller to stabilize the system smoothly and efficiently.

obtained gain matrix:

$$K = \begin{bmatrix} -246.7157 & -98.7840 & -106.5109 & -50.1602 \end{bmatrix}$$

$$S = \begin{bmatrix} 12963.5849 & 5822.7661 & 5671.7846 & 2793.1748 \\ 5822.7661 & 2629.2514 & 2549.4332 & 1257.5231 \\ 5671.7846 & 2549.4332 & 2481.9793 & 1222.5371 \\ 2793.1748 & 1257.5231 & 1222.5371 & 602.5613 \end{bmatrix}$$

5 Energy Shaping Swing-Up Controller

5.1 Energy Shaping

The objective of the swing-up controller is to generate appropriate commands for the second link (denoted q_2^d) such that it excites the unactuated first link and drives the system from the stable downward equilibrium ($q_1 = 0$) toward the unstable upright equilibrium ($q_1 = \pi$). The key idea is to design the input q_2^d, which indirectly affects the dynamics of the first link through their coupling.

A simple and intuitive swing-up strategy involves oscillating the actuated link (link 2) back and forth between angles $+\alpha$ and $-\alpha$. This periodic motion injects energy into the system. The switching of the second link's input torque is carefully scheduled based on the direction of motion of the first link, such that the amplitude of its swing increases over time.

Mathematically, the command to the second link can be expressed as a function of the velocity of the first joint $\dot{q}_1$. One of the simplest strategies is to swing q_2 between fixed values whenever $\dot{q}_1$ is nonzero, thereby ensuring continuous energy injection. In this implementation, we adopt an energy shaping approach. The control input to the second joint is designed based on the difference between the current total energy of the system and the desired energy required to reach the upright position.

The control law is defined as:

$$E_{\text{desired}} = m_1 l_{c1} + m_2(l_1 + l_{c2}) + g$$

$$\tilde{E} = E - E_{\text{desired}}$$

$$u_e = -k_e \cdot \tilde{E} \cdot \dot{\theta}_2$$

where:

- $\tilde{E}$ is the energy error,
- E_{desired} is the total mechanical energy at the upright configuration,
- $\dot{\theta}_2$ is the velocity of the actuated joint,
- k_e is a tunable gain that scales the energy error.

This energy shaping controller ensures that the system accumulates sufficient energy to reach the desired configuration, after which a switching logic hands over control to the LQR controller for local stabilization.

5.2 Partial Feedback Linearization (PFL)

The Acrobot is an underactuated and highly nonlinear system, making it infeasible to fully linearize the dynamics using traditional feedback linearization. Instead, we adopt a Partial Feedback Linearization (PFL) approach, where only a subset of the system—specifically, the actuated second joint—is linearized. This allows us to control the system in a structured way by treating the dynamics as two interconnected subsystems.

By applying PFL, we effectively linearize and decouple the dynamics of the second link through a nonlinear inner-loop controller. This creates a collocated linearized subsystem for the actuated joint, while the dynamics of the unactuated first link form the internal (zero) dynamics. The control input is designed such that the second link follows a desired trajectory, which in turn injects energy into the first link and influences its behavior indirectly. This approach simplifies control design by reducing the complexity of the overall system while still exploiting the coupling between the links to achieve the desired swing-up behavior.

$$\ddot{q} = M^{-1}(q)\,(B\,u - h(q,\dot{q}))$$

$$where, \quad h(q,\dot{q}) = C(q,\dot{q})\,\dot{q} + G(q)$$

$$M^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \qquad h(q,\dot{q}) = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

$$\ddot{\theta}_2 = a_{21}(0 \cdot u - h_1) + a_{22}(u - h_2) = -a_{21}\,h_1 + a_{22}(u - h_2)$$

$$let, \quad \ddot{\theta}_{2desired} = -k_p\theta_2 + -k_d\dot{\theta}_2$$

$$then, \; u_p = \left(\frac{y + a_{21}h_1}{a_{22}}\right) + h_2$$

$$u = u_e + u_p$$

The above input can be used and the controller is switched to an LQR when the quadratic cost, is less than 1000.

6 Simulation & Plots

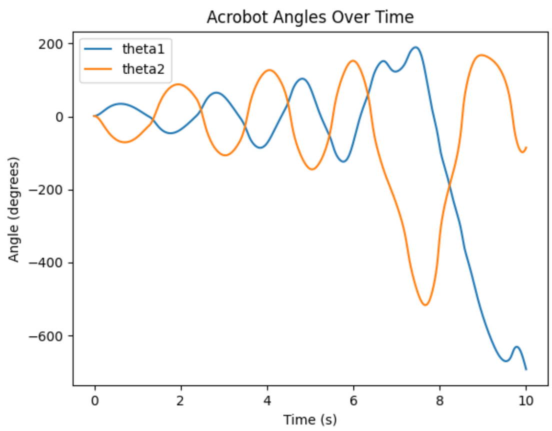
6.1 Comparison Plots

Below are the resulting energy plots and state plots for LQR, Energy shaping, Energy Shaping + PFL(collocated), Energy Shaping + PFL(non-collocated) and Energy Shaping + PFL + LQR respectively. The parameters used are as follows,

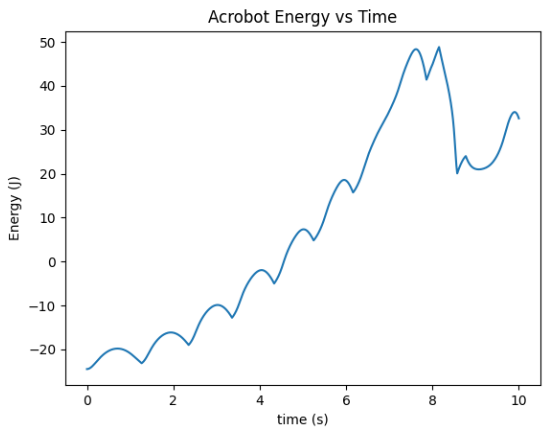
$$k_e = 2.5 \quad k_p = 16 \quad k_d = 8$$

6.2 Performance analysis

Looking at all the plots, we can evaluate the performance of each controller. The LQR is highly unstable at the stable equilibrium and is oscillating with increasing amplitudes. Energy shaping does a good job of pumping energy into the system and keeping the energy around the desired level but the states are erratic. The non-collocated PFL controller makes link 1 stable but makes link 2 spin around. The collocated PFL performs similar to the energy shaping only controller at pumping energy while keeping the states a lot smoother. The final cmbined controller achieves all of the requirements and manages to quickly swing up and balance the acrobot.

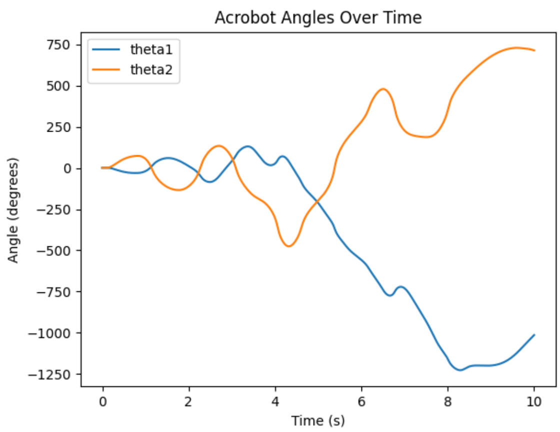


(a) Acrobot angles over time.

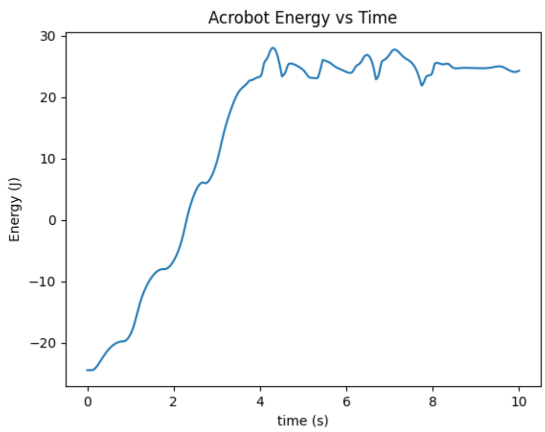


(b) Acrobot energy over time.

Figure 2: LQR controller

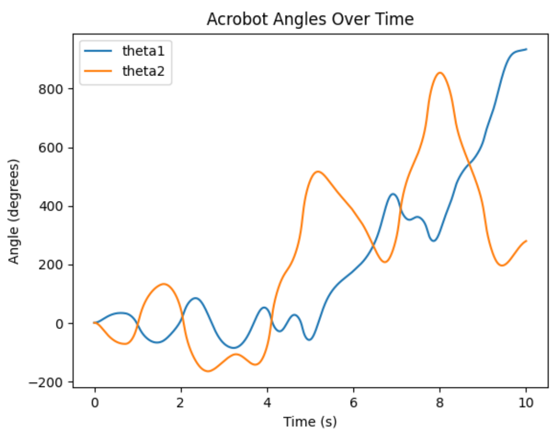


(a) Acrobot angles over time.

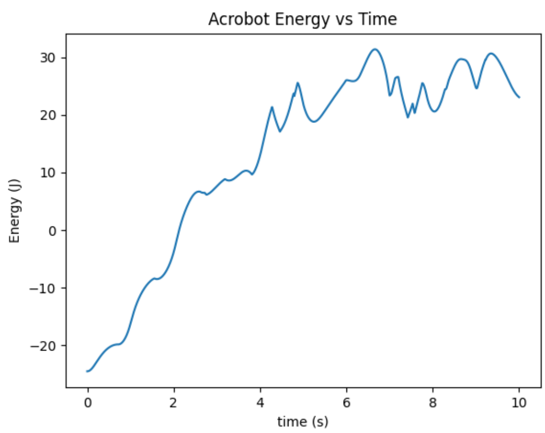


(b) Acrobot energy over time.

Figure 3: Energy shaping controller

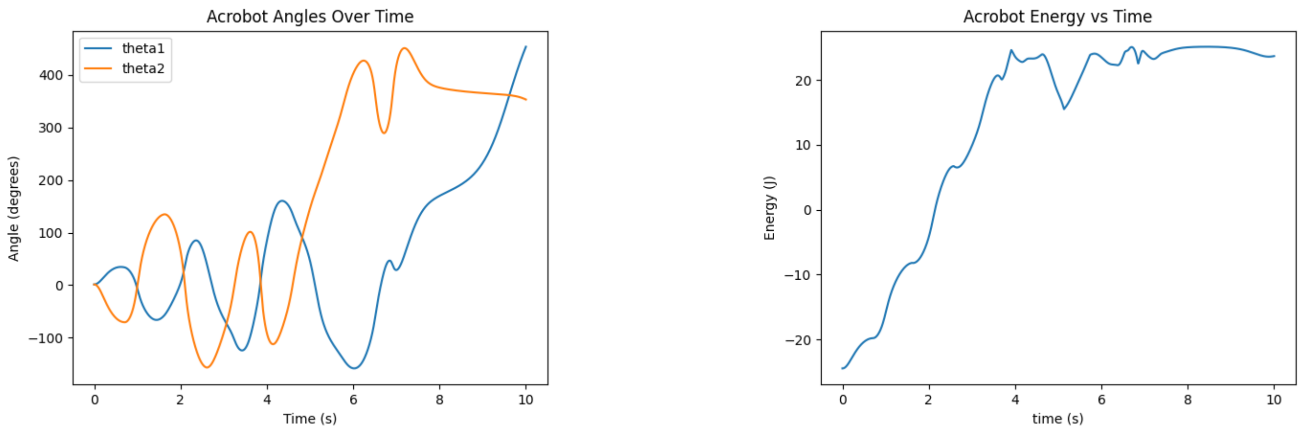


(a) Acrobot angles over time.



(b) Acrobot energy over time.

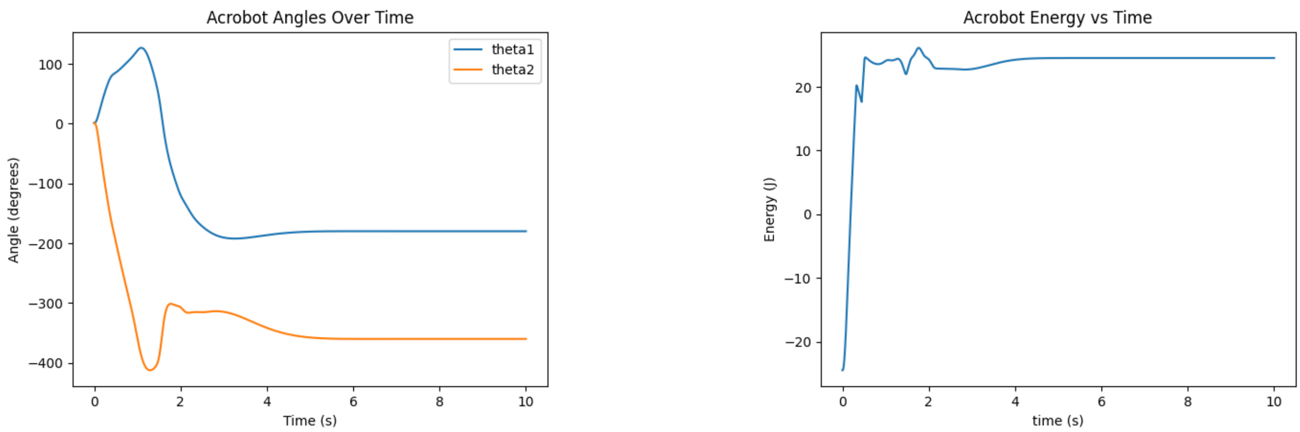
Figure 4: Energy shaping + Non collocated PFL controller



(a) Acrobot angles over time.

(b) Acrobot energy over time.

Figure 5: Energy shaping + collocated PFL controller



(a) Acrobot angles over time.

(b) Acrobot energy over time.

Figure 6: Energy shaping + collocated PFL + LQR controller

7 Discussion & Future Work

7.0.1 Describe the major design decisions in your implementation

One of the major design decisions in our implementation was to use a hybrid control strategy combining an energy-based swing-up controller and a Linear Quadratic Regulator (LQR) for stabilization. Since the Acrobot is an underactuated and nonlinear system, a single controller could not achieve both swing-up and balance. The swing-up controller increases the system's energy, while the LQR ensures stability near the upright position. Another critical design

decision was learning how to integrate multiple controllers in a coordinated manner to achieve a unified control objective. In our project, this involved designing the switching logic between the swing-up phase and the stabilization phase. The transition is triggered when the system's total energy and joint configuration fall within a predefined threshold near the upright position. This smooth handoff ensures stability without introducing oscillations or instability at the switching point.

7.0.2 What challenges did you encounter

One of the main challenges we faced was dealing with the nonlinear and underactuated nature of the Acrobot system. It was particularly difficult to understand and implement partial feedback linearization for the second link while ensuring that its motion effectively injected energy into the first link to achieve swing-up.

Another significant challenge was the linearization of the Acrobot dynamics around the upright equilibrium. The symbolic derivation of the Jacobian matrices from the nonlinear equations was mathematically intensive and error-prone. Even slight inaccuracies in the Jacobian entries led to degraded LQR performance. Achieving a functional and stable LQR controller required high levels of precision in both symbolic derivation and numerical implementation. Lastly, Careful tuning of controller gains and switching thresholds was necessary, and we relied heavily on iterative testing and fine-tuning to ensure smooth transitions without instability or oscillation.

7.0.3 What did you learn through implementing the project? Any major takeaways?

This project provided valuable insight into the practical challenges of controlling nonlinear, underactuated systems. One of the key takeaways was learning how to effectively combine multiple control strategies—specifically, using an energy-based swing-up controller followed by an LQR stabilizer to address different phases of the control problem. While these methods were conceptually introduced in class, implementing them on a real system deepened our understanding significantly.

Also, the importance of thoroughly understanding the underlying physics and system dynamics before designing controllers. By deriving the equations of motion using Lagrangian mechanics, we developed a clearer intuition for how the second joint's motion influences the behavior of the entire system. This foundation was essential for designing both the swing-up and stabilization controllers.

7.0.4 What are the "next steps" in the project (if you were to continue working on it)?

There are a few improvements that can be done to the switching of controllers. The basin of attraction i.e. the region for which the LQR can stabilize the system can be quantified exactly using control lyapunov functions. The switch can also be made smoother since currently we just abruptly switch between the controllers. The natural next step for this project would be to implement the controller on real hardware, such as a cart-pendulum system or a simplified Acrobot setup. This would allow us to gain hands-on experience with physical systems and address real-world challenges like sensor noise, actuator delays, and hardware limitations.

Beyond hardware implementation, this project laid a strong foundation in understanding and controlling underactuated systems, which has inspired us to explore more complex systems such as bipedal or humanoid robots. It gave us good intuition into the field of underactuated robotics, and our goal moving forward is to apply these principles to higher-dimensional systems, developing more advanced control strategies for real-world robotic platforms.

Github Repository

All the code used in the project can be accessed in this `github link`

References

[1] R. M. Murray and M. W. Spong, "A case study in approximate linearization: The acrobot example," *IFAC Proceedings Volumes*, vol. 24, no. 2, pp. 1149–1154, 1991.

[2] M. W. Spong, "The swing up control problem for the acrobot," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 49–55, 1994.

[3] R. Tedrake, "Underactuated robotics – Model systems Chapter 3: Acrobots, cart-poles, and quadrotors," https://underactuated.csail.mit.edu/acrobot.html, 2024. Accessed: Feb. 21, 2025.