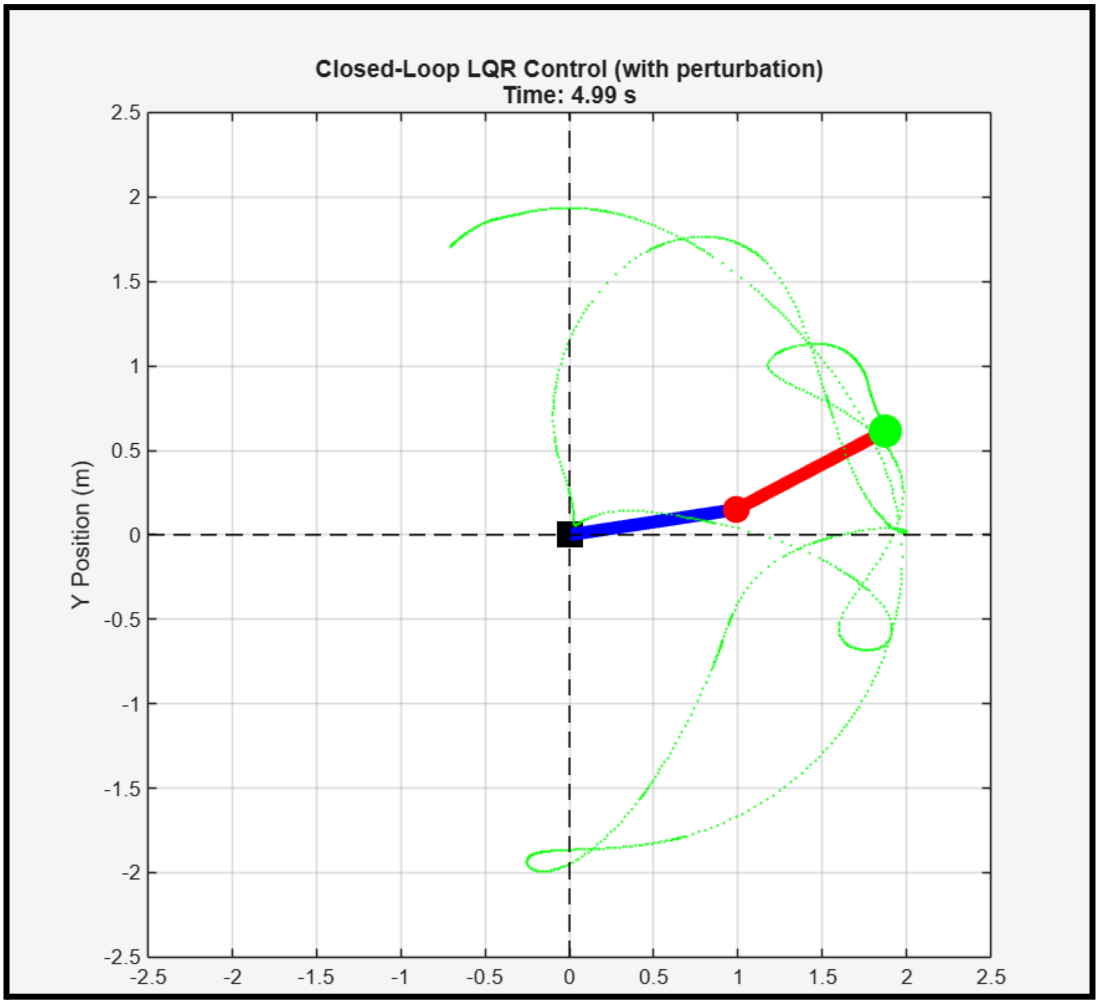
Feedback Controller- TO-LQR+MPC

Acrobot



Jash Shah

1 Dec. 2025

ECE 238 - ADV CONTROL DES LAB

Link to Code:

https://github.com/Jash-2000/Adv_Control_Systems/tree/main/Acrobot_Controller

INTRODUCTION

The Acrobot is a canonical example of an underactuated nonlinear robotic system consisting of two links connected by rotary joints. Only the elbow joint is actuated, making the system highly nonlinear, energy-dependent, and dynamically challenging to control. It is frequently used in research on swing-up control, optimal control, and nonlinear stabilization.

Project Emphasize

- **Trajectory Optimization (TO)** to generate a feasible swing-up motion
- **Open-loop validation** to evaluate sensitivity
- **Time-Varying LQR (TVLQR)** to stabilize the optimized trajectory
- **Model Predictive Control (MPC)** to transition between upright equilibrium points using linearized discrete-time dynamics

Variables and System Definitions

The **Acrobot** is a two-link, underactuated robotic system composed of two rigid segments connected by rotational joints. The first joint, located at the shoulder, is unactuated, while the second joint at the elbow is actuated by an external torque. The configuration of the system is described by two generalized coordinates: θ1\theta_1θ1, the angle of the upper link measured from the vertical downward direction, and θ2\theta_2θ2, the relative angle of the lower link with respect to the upper link. The corresponding angular velocities, θ˙1\dot{\theta}_1θ˙1 and θ˙2\dot{\theta}_2θ˙2, together with the angles, form the full state vector x=[θ1,θ2,θ˙1,θ˙2]Tx = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^Tx=[θ1,θ2,θ˙1,θ˙2]T. The system has a single control input, u=τ2u = \tau_2u=τ2, representing the torque applied at the elbow joint. The dynamics of the Acrobot follow from the standard manipulator equations of motion, where the inertia matrix M(q)M(q)M(q), Coriolis and centrifugal matrix C(q,q˙)C(q, \dot{q})C(q,q˙), and gravitational vector G(q)G(q)G(q) depend on the system's configuration and motion. Physical parameters—including the masses of the two links (m1,m2m_1, m_2m1,m2), their lengths (l1,l2l_1, l_2l1,l2), center-of-mass distances (lc1,lc2l_{c1}, l_{c2}lc1,lc2), moments of inertia (I1,I2I_1, I_2I1,I2), and gravitational acceleration ggg—define the system's mechanical behavior and appear explicitly in the dynamic equations. All parameters are defined in a single section of the code to support easy modification for testing different physical configurations or future hardware applications. As an underactuated system with only one actuator controlling two degrees of freedom, the Acrobot exhibits nonlinear and energy-dependent behavior that must be carefully managed using advanced control methods such as trajectory optimization, LQR, and MPC.

The system is defined precisely using the physics in MATLAB as follows:

```matlab
function dx = acrobotDynamics(t, x, u)
% ACROBOTDYNAMICS Dynamics for a 2-link underactuated robot (Acrobot)
% Second joint (elbow) is actuated; first joint is unactuated.
%
% State x = [theta1; theta2; theta1_dot; theta2_dot]
% Control u = torque applied at joint 2 (scalar)
% Physical parameters
m1 = 1.0;       % mass of link 1 (kg)
m2 = 1.0;       % mass of link 2 (kg)
L1 = 1.0;       % length of link 1 (m)
L2 = 1.0;       % length of link 2 (m)
Lc1 = 0.5;      % distance to center of mass of link 1 (m)
Lc2 = 0.5;      % distance to center of mass of link 2 (m)
I1 = m1*L1^2/12;  % moment of inertia of link 1
I2 = m2*L2^2/12;  % moment of inertia of link 2
g = 9.81;       % gravity (m/s^2)
b1 = 0.1;       % damping coefficient at joint 1
b2 = 0.1;       % damping coefficient at joint 2
% Extract states (handle both column vectors and matrices)
theta1 = x(1,:);      % first link angle
theta2 = x(2,:);      % second link angle (relative)
theta1_dot = x(3,:);  % first link angular velocity
theta2_dot = x(4,:);  % second link angular velocity
% Inertia matrix elements (same structure as two-link robot)
M11 = I1 + I2 + m1*Lc1^2 + m2*(L1^2 + Lc2^2 + 2*L1*Lc2.*cos(theta2));
M12 = I2 + m2*(Lc2^2 + L1*Lc2.*cos(theta2));
M21 = M12;
M22 = I2 + m2*Lc2^2;
% Coriolis/centrifugal terms
h = -m2*L1*Lc2.*sin(theta2);
C11 = h.*theta2_dot;
C12 = h.*(theta1_dot + theta2_dot);
C21 = -h.*theta1_dot;
C22 = 0;
% Gravity vector
G1 = (m1*Lc1 + m2*L1)*g.*sin(theta1) + m2*g*Lc2.*sin(theta1 + theta2);
G2 = m2*g*Lc2.*sin(theta1 + theta2);
```

```
% Damping torques
D1 = b1*theta1_dot;
D2 = b2*theta2_dot;
% Control input: torque applied at joint 2 (acrobot)
tau1 = 0;      % joint 1 passive
tau2 = u;      % actuation at joint 2
% Right-hand side
rhs1 = tau1 - C11.*theta1_dot - C12.*theta2_dot - G1 - D1;
rhs2 = tau2 - C21.*theta1_dot - C22.*theta2_dot - G2 - D2;
% Determinant
det_M = M11.*M22 - M12.*M21;
% Solve for accelerations
theta1_ddot = (M22.*rhs1 - M12.*rhs2) ./ det_M;
theta2_ddot = (-M21.*rhs1 + M11.*rhs2) ./ det_M;
% State derivatives
dx = [theta1_dot;
     theta2_dot;
     theta1_ddot;
     theta2_ddot];
end
```

Implementation Details

The control pipeline for the Acrobot consisted of four stages: trajectory optimization, open-loop validation, time-varying LQR stabilization, and linear MPC regulation. First, in **Part A**, a nonlinear trajectory optimization problem was formulated using OptimTraj to generate a swing-up motion from the downward-hanging configuration to the fully upright equilibrium. The Acrobot dynamics were supplied through the `acrobotDynamics` function, with a cost function minimizing squared torque and boundary conditions enforcing start and end states with zero velocities. State and torque constraints were applied, and an initial guess for states and controls was provided. OptimTraj solved for the optimal trajectory, producing a feasible swing-up motion which was plotted and animated.

In **Part B**, this optimal trajectory was validated via open-loop simulation with `ode45`, using an interpolated version of the optimized torque. The resulting trajectory was compared to the optimization solution by computing tracking errors of the joint angles and velocities. The open-loop simulation exhibited deviations from the optimized path, underscoring the sensitivity of the Acrobot and the need for feedback stabilization.

To address this, **Part C** implemented a time-varying LQR controller. The system was linearized along the optimal trajectory by numerically computing Jacobians for each time sample, yielding time-indexed A(t)A(t)A(t) and B(t)B(t)B(t) matrices. Using these, a time-varying gain K(t)K(t)K(t) was computed with MATLAB's `lqr` function. Closed-loop control combined the feedforward optimal torque with a feedback correction term that penalized deviations from the reference trajectory. Simulations showed that the TVLQR significantly reduced tracking error and ensured convergence to the upright configuration even under perturbations.

Finally, **Part D** introduced a linear MPC controller for local regulation around the upright equilibrium. The nonlinear system was linearized at the equilibrium and discretized using a zero-order hold approach. A finite-horizon LQR-based MPC scheme was implemented using a 10-step prediction horizon and quadratic cost weighting. A small reference shift was commanded, and a backward Riccati recursion produced the required sequence of MPC gains. The nonlinear system was simulated using a fixed-step RK4 integrator, applying the first MPC gain at each step with torque saturation. The controller successfully guided the Acrobot from the upright configuration to a nearby equilibrium with stable, accurate tracking. Together, these components formed a complete control pipeline for both large-swing maneuvers and small equilibrium transitions of the Acrobot.

Results and Discussion

1. PART A – Trajectory Optimization Results

```
Final time: 5.000 seconds
Control effort: 72.875
Max torque used: 5.000 N-m
```

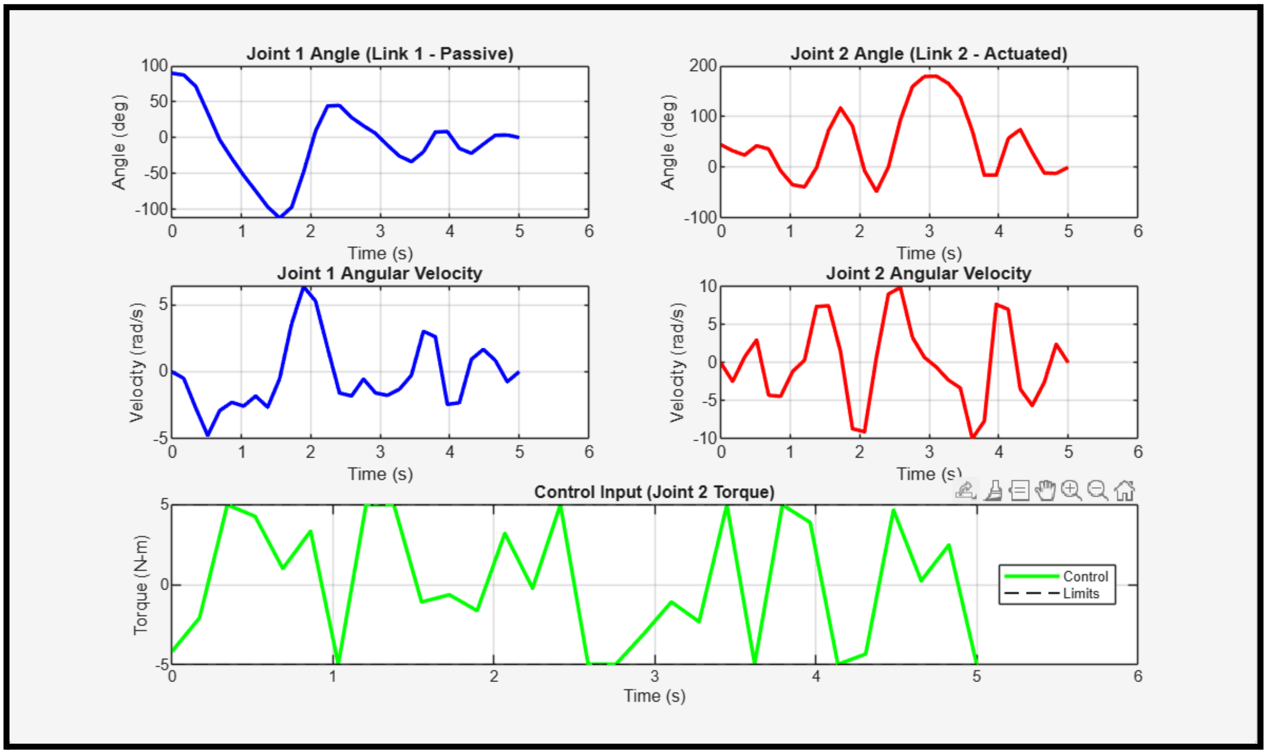


Fig1: Trajectory Optimization Results

2. PART B – ODE45 solver results and errors for open loop control

```
Max angle 1 error: 0.667285 rad (38.233 deg)
Max angle 2 error: 3.965455 rad (227.204 deg)
RMS angle 1 error: 0.277572 rad
RMS angle 2 error: 1.615168 rad
```

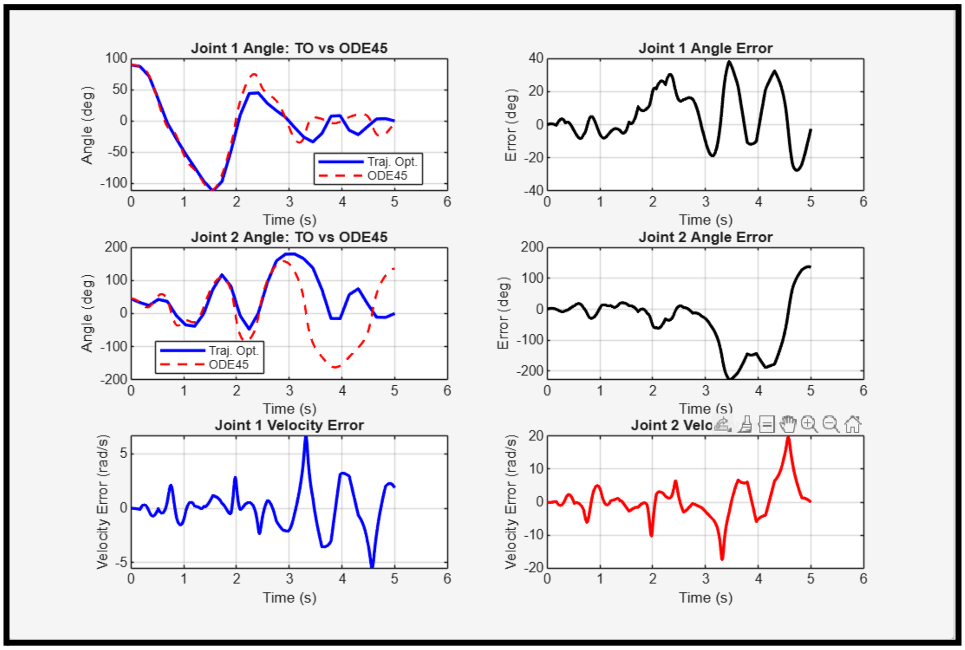


Fig2: Open Loop error analysis for TO path vs ODE45 simulated path

3. PART C – Closed loop solutions with LQR feedback

```
Final angle error (open-loop, perturbed start): 2.3654 rad (135.52 deg)
Final angle error (closed-loop): 0.3584 rad (20.54 deg)
```

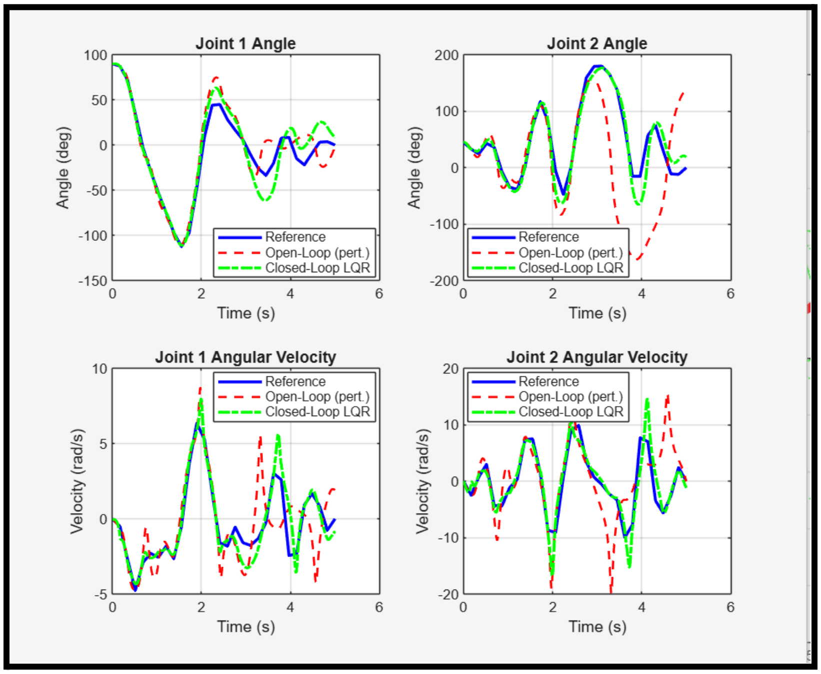


Fig3: State Variables comparison for all three methods

4. PART D - MPC based shift to new equilibrium

```
MPC simulation finished. Final state error to desired eq: 0.1245 (norm)
```

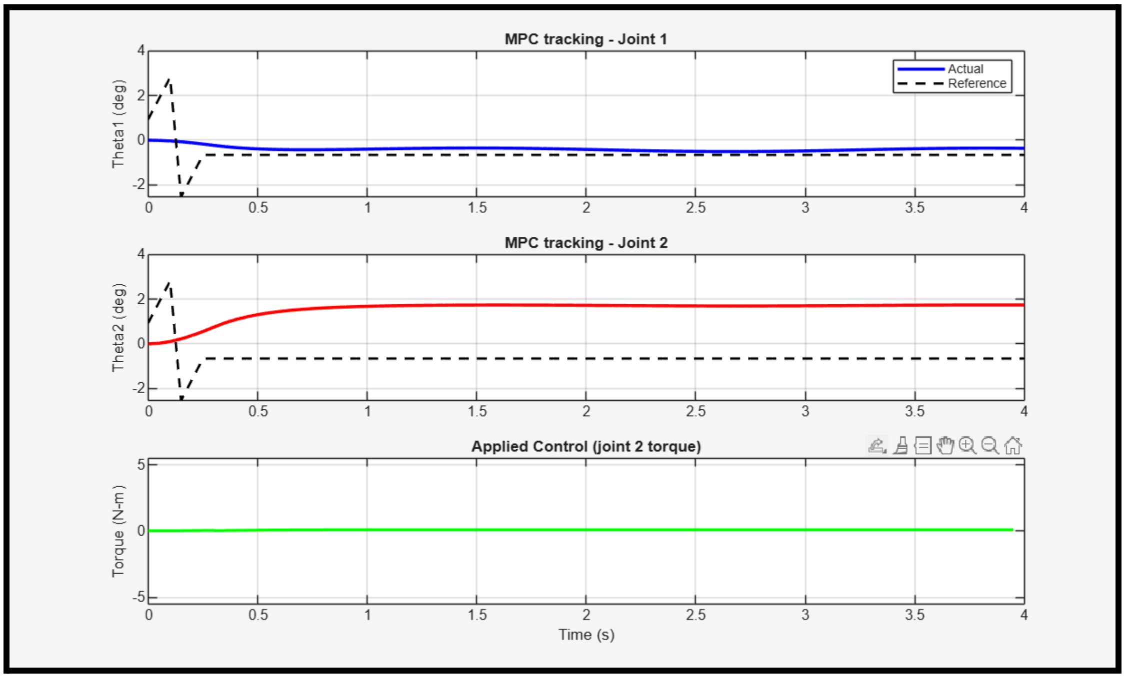


Fig4: MPC simulation and state changes with ODE45

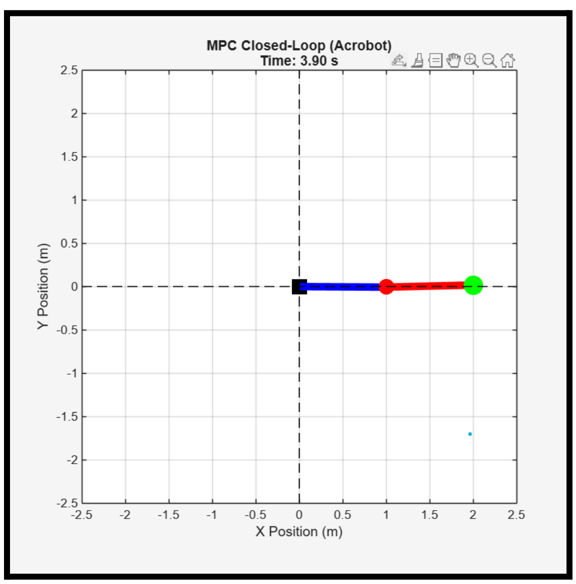


Fig5: New Equilibrium position with constantly applied torque (an additional torque was applied to the system other than the controller torque to maintain the horizontal equilibrium position intact)

Conclusion

This project demonstrates a complete workflow for solving complex nonlinear control tasks on underactuated systems. The Acrobot swing-up problem is addressed using trajectory optimization and stabilized using time-varying LQR, while local maneuvering near equilibrium is enabled by MPC using linearized discrete-time dynamics.This combination highlights the strengths of:

- TO for global motion generation
- TVLQR for robust trajectory stabilization
- MPC for local predictive maneuvering

Together, they provide a powerful framework for controlling nonlinear robotic systems and create a foundation for real-time implementations such as nonlinear MPC or hardware experiments.

REFERENCES

1. S. Kajita et al., "Biped walking pattern generation by using preview control of zero-moment point," 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), Taipei, Taiwan, 2003, pp. 1620-1626 vol.2, doi: 10.1109/ROBOT.2003.1241826. keywords: {Legged locomotion;Foot;Weight control;Humanoid robots;Centralized control;Industrial control;Servomechanisms;Control theory;Spirals;Control systems},
2. Katayama, Tohru & OHKI, TAKAHIRA & INOUE, TOSHIO & KATO, TOMOYUKI. (1985). Design of an optimal controller for a discrete-time system subject to previewable demand. International Journal of Control - INT J CONTR. 41. 677-699. 10.1080/0020718508961156.
3. Katayama, Sotaro & Murooka, Masaki & Tazaki, Yuichi. (2023). Model predictive control of legged and humanoid robots: models and algorithms. Advanced Robotics. 37. 1-18. 10.1080/01691864.2023.2168134.
4. Kelly, Matthew. “trajectoryOptimizationTutorials.” Trajectory optimization, 2016. https://www.matthewpeterkelly.com/tutorials/trajectoryOptimization/index.html.