Robotic Systems II: Homework III Report

Student: Vangelis Tsiatsianas

December 22, 2024

Contents

Introduction

This report focuses involves modeling and controlling a cart double-pendulum system by combining offline trajectory optimization and online Time-Variant LQR (TVLQR). The objectives are to model the system, solve the system's swing-up problem using non-linear trajectory optimization, and control it under disturbances with TVLQR. This report outlines the approach and methodology for solving these tasks in three distinct parts.

1 Modeling

The double-pendulum on a cart system is a non-linear system consisting of two connected links and a cart, each with a mass and a force applied at the cart along the horizontal axis. The system is modeled in the $\hat{x}$-$\hat{y}$ plane, under the influence of gravity g applied in $-\hat{y}$. The generalized coordinates x_1 represents the position of the cart on the $\hat{x}$ axis and θ_2 and θ_3 represent the angles of the two links measured from the positive $\hat{y}$ axis. The state of the system consists from the generalized coordinates and it's velocities. The parameters used are $g = 9.81\,\mathrm{m/s}^2$, $m_1 = m_2 = m_3 = 0.5\,\mathrm{kg}$, and $l_2 = l_3 = 0.6\,\mathrm{m}$, and $b = 0.1$.

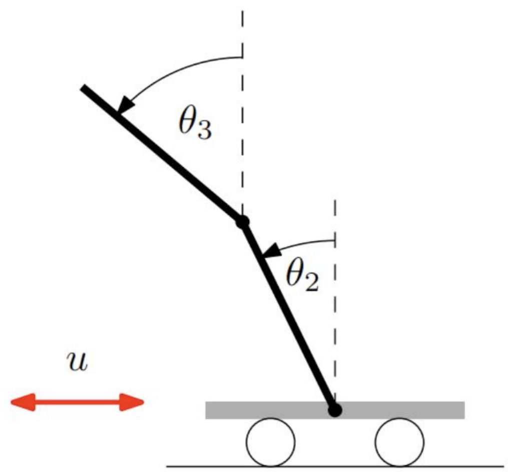


Figure 1: Cart double-pendulum. Figure & system adapted from Deisenroth, M.P., 2010. Efficient reinforcement learning using Gaussian processes (Vol. 9). KIT Scientific Publishing.

1.1 Dynamics Function

The dynamics of the double pendulum system are described by the continuous equations of motion:

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{x}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \ddot{x}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix},$$

where the system state and the input controls are:

$$x = \begin{bmatrix} x_1 \\ \theta_2 \\ \theta_3 \\ \dot{x}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \in \mathbb{R}^6, \quad u = \begin{bmatrix} u \end{bmatrix} \in \mathbb{R}.$$

The dynamics of the system are given by solving the following system:

$$\begin{bmatrix} (m_1 + m_2 + m_3) & -\frac{1}{2}m_2 l_2 \cos(\theta_2) & -\frac{1}{2}m_3 l_3 \cos(\theta_3) \\ -\frac{1}{2}m_2 l_2 \cos(\theta_2) & \frac{1}{2}(m_2 + 2m_3) l_2 \cos(\theta_2) - \frac{1}{2}m_3 l_3 \cos(\theta_3) & m_3 l_2 l_3 \cos(\theta_2 - \theta_3) \\ -\frac{1}{2}m_3 l_3 \cos(\theta_3) & m_3 l_2 l_3 \cos(\theta_2 - \theta_3) & \frac{1}{4}m_2 l_2 + I_2 + \frac{1}{4}m_3 l_3 + I_3 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

with:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} u - b\dot{x}_1 - \frac{1}{2}(m_2 + 2m_3) l_2 \theta_2^2 \sin(\theta_2) - \frac{1}{2}m_3 l_3 \theta_2^3 \sin(\theta_3) \\ -\frac{1}{2}(m_2 + m_3) l_2 g \sin(\theta_2) - \frac{1}{2}m_3 l_2 l_3 \sin(\theta_2 - \theta_3) \\ \frac{1}{2}m_3 l_3 g \sin(\theta_3) + l_2 \theta_2^2 \sin(\theta_2 - \theta_3) \end{bmatrix}$$

The dynamics of the system are implemented in the `dynamics(x, u)` function, which computes the time derivative of the state vector $\dot{x}$ based on the given control inputs u.

1.2 Discretization

To discretize the continuous system dynamics, Runge-Kutta 4th Order (RK4) integration was used, allowing precise integration over a fixed time step $\Delta T = 0.05$. This method iteratively calculates the state at each time step on the basis of intermediate evaluations of the continuous dynamics.

2 Trajectory Optimization

In this section, we formulate the swing-up problem as a nonlinear trajectory optimization problem. The goal is to find a trajectory that drives the system from the initial state to the final state while respecting the dynamics and minimizing a cost function. This is achieved by discretizing the system solving the optimization problem and performing implicit integration for "enforcing the dynamics".

2.1 Problem Formulation

The trajectory optimization problem is formulated as a finite-horizon optimal control problem. The result is a control trajectory $u(t)$ which satisfies our constraints and from which we can derive the state trajectory. The optimal control problem is formulated as:

$$\min_{x_{1:K}, u_{1:K-1}} \mathcal{J}(x_{1:K}, u_{1:K-1}) = \sum_{k=1}^{K-1} \ell(x_k, u_k) + \ell_f(x_K),$$
$$s.t. \quad \text{dynamics constraints}$$

In our implementation the dynamic constraints are specified as an implicit integration using direct collocation methods, we used the trapezoidal rule without extra collocation points. The dynamics constraints deriving from the trapezoidal rule are expressed as:

$$x_{k+1} - x_k = \frac{1}{2}(t_{k-1} - t_k)(f_{k+1} + f_k)$$
$$g(x_k, x_{k+1}, u_k, u_{k+1}) = 0$$

Finally, the solution comes when solving the problem using non-linear programming.

2.2 Cost Function and Constraints

The cost function is designed to penalize both state deviations and control efforts:

$$J = \sum_{k=0}^{N-1} \left(\|x_k - x_{\mathrm{f}}\|_Q^2 + \|u_k\|_R^2 \right),$$

where $\|\cdot\|_Q^2$ denotes the weighted Euclidean norm with matrix Q. The constraints include:

- $x_0 = \begin{bmatrix} 0 & \pi & \pi & 0 & 0 & 0 \end{bmatrix}^T$ is the initial state.
- $x_{\mathrm{f}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ is the desired final state.
- Equality constraints to enforce the discrete-time system dynamics.
- Control constraints $u_k \in \left[-20, 20\right]$.

Also an initialization trajectory was used by interpolating the initial and target states, helping the optimizer find a solution.

2.3 Results and Visualization

After solving the optimization problem using an interior point optimizer, we visualize the resulting trajectory of the system. The pendulum's angles, cart position, and control inputs over time are plotted to illustrate how the system behaves during the swing-up maneuver.

Figure 2: Optimal trajectory trace

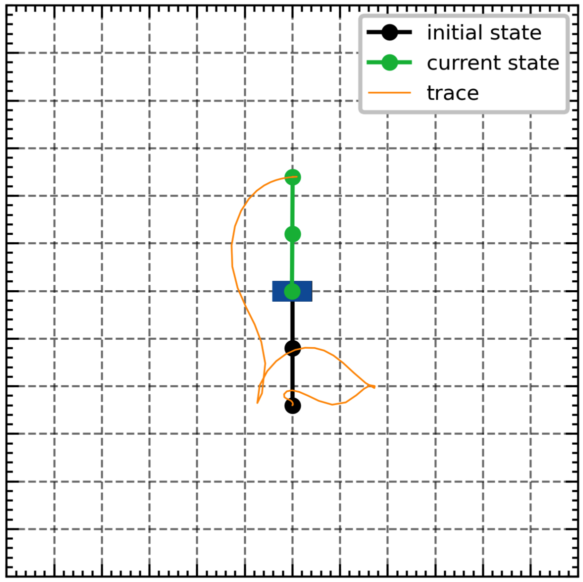
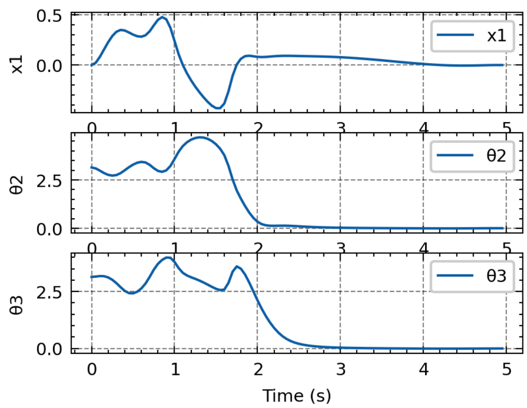
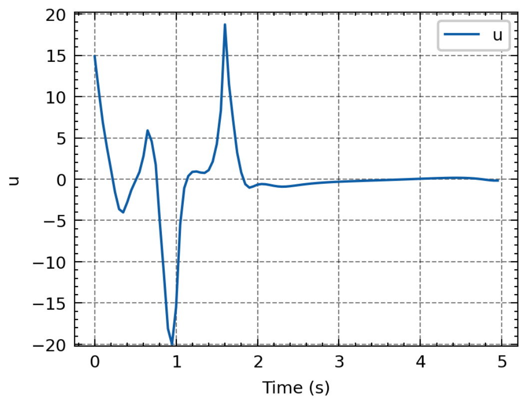


Figure 3: Optimal Trajectory



(a) State Trajectory



(b) Control Trajectory

3 Controlling via TVLQR

In this section, we implement a Time-Variant Linear Quadratic Regulator (TVLQR) controller to stabilize the cart double-pendulum system under disturbances. The TVLQR controller is designed to compute the control inputs based on a linear approximation of the system dynamics around the trajectory generated by the trajectory optimization in the previous section.

3.1 TVLQR Controller Implementation

The TVLQR controller operates by linearizing the nonlinear system dynamics around the trajectory that was obtained from the trajectory optimization problem. Given a trajectory $\{x_t[k], u_t[k]\}$, we first linearize the system dynamics at each time step. The system is then approximated by a linear time-varying system:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

where $A(t)$ and $B(t)$ are the time-varying state and input matrices, respectively, obtained by computing the Jacobians of the system dynamics with respect to the state and control input:

$$A(t) = \frac{\partial f_{discrete}(x, u)}{\partial x} \quad \text{and} \quad B(t) = \frac{\partial f_{discrete}(x, u)}{\partial u}$$

The goal of the TVLQR controller is to minimize the quadratic cost function:

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

where Q and R are positive definite weighting matrices that penalize deviations of the state from the desired trajectory and large control inputs, respectively.

At each time step, the TVLQR controller computes the optimal control input $u(t)$ by solving the following Riccati equation:

Using the Riccati recursion, we obtain:

$$K_k = \left(R_k + B_k^T P_{k+1} B_k\right)^{-1} B_k^T P_{k+1} A_k, \tag{3.1}$$

$$P_k = Q_k + A_k^T P_{k+1} \left(A_k - B_k K_k\right). \tag{3.2}$$

Using the optimal gains, the following optimal control law is inherited:

$$u = u_t[k] - K[k](x - x[k])$$

This feedback control law is applied to the system at each time step to compute the control input $u(t)$, which is then used to drive the system.

3.2 Adding Disturbances

To simulate the effect of noise in real-world applications, we add Gaussian noise to the observations. The noise is assumed to have zero mean and diagonal covariance. The noisy state is given by:

$$\hat{x}(t) = x(t) + \epsilon(t)$$

where $\epsilon(t)$ is the noise vector, with each component sampled from a Gaussian distribution with zero mean and a specified covariance matrix. The controller then operates on the noisy state estimate, which may cause the controller to deviate from the optimal trajectory.

3.3 Controller Performance

The performance of the TVLQR controller is evaluated by running simulations with different levels of noise. The resulting control inputs and system trajectories are compared to assess the robustness of the controller.

Figure 4 shows the optimal trajectory to be followed. Figures 5 and 6 show the system's behavior when the TVLQR controller is applied with different magnitudes of noise in the observation. As noise is added, the control inputs and trajectories become less accurate, highlighting the importance of accurate state estimation for optimal control.

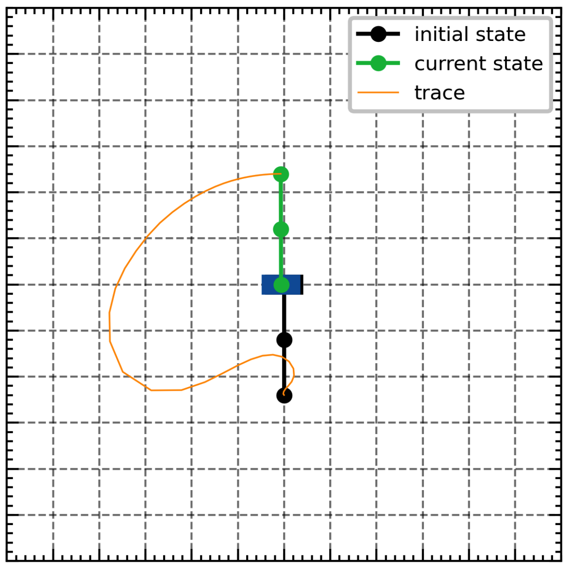
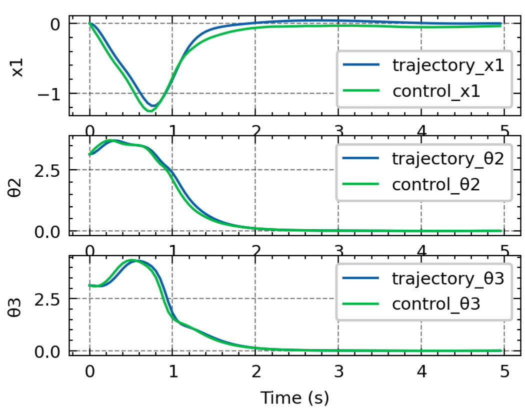
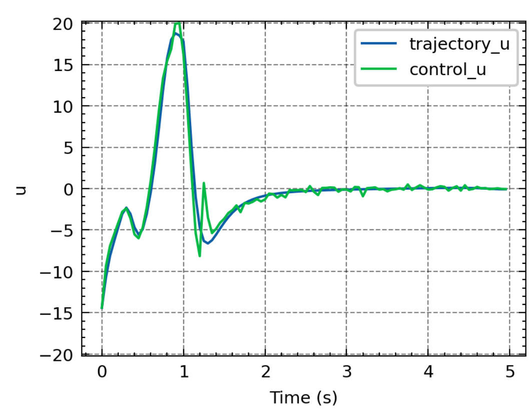


Figure 4: Trajectory trace

Figure 5: TV-LQR Control, noise covariance = 10^{-3}

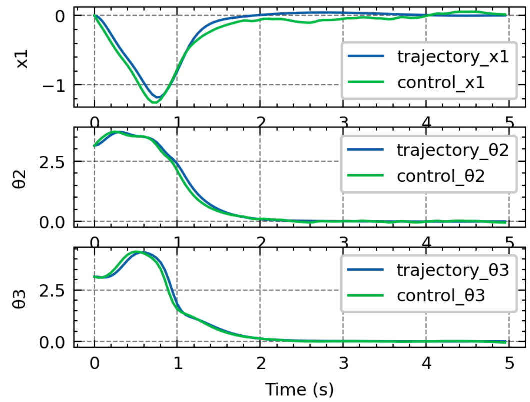


(a) State Trajectory tracking

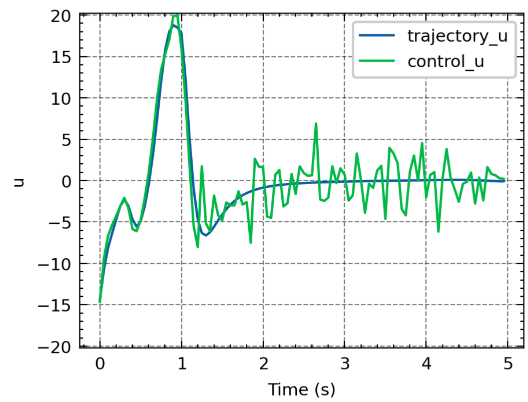


(b) Control Trajectory tracking

Figure 6: TV-LQR Control, noise covariance = 10^{-2}



(a) State Trajectory tracking



(b) Control Trajectory tracking

3.4 Conclusion

In this homework, we successfully implemented a TV-LQR controller for the cart double-pendulum system. The controller was able to stabilize the system and track the desired trajectory. However, the performance of the controller was affected by the presence of noise, demonstrating the challenges of controlling nonlinear systems with imperfect state measurements.