Optimal Control

Course Project #1
Optimal Control of a Gymnast Robot

November 25, 2025

In this project, you are required to design an optimal trajectory for the robot shown in Figure 1. The system is a planar *gymnast* robot that performs swings via actuation at the hip. In a planar setting, it is modeled as a double pendulum with torque applied on the second link.

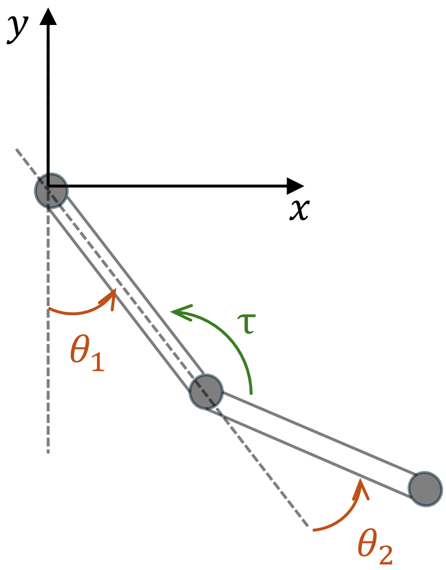

Figure 1: Model of an Acrobot

The state space consist in $x = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^\top$, where θ_1 represent the angle of the first link with respect to the vertical direction, θ_2 represents the angle of the second link with respect to the first link, $\dot{\theta}_1$ and $\dot{\theta}_2$ the angular rates of changes associate to θ_1 and θ_2, respectively.

The input is the torque τ on the second link.

$$M(\theta_1, \theta_2)\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{pmatrix} + C(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix} + F\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix} + G(\theta_1, \theta_2) = \begin{pmatrix}0\\ \tau\end{pmatrix}$$

where

$$M = \begin{pmatrix} I_1 + I_2 + l_{c1}^2 m_1 + m_2\left(l_1^2 + 2 l_1 l_{c2}\cos(\theta_2) + l_{c2}^2\right) & I_2 + l_{c2} m_2\left(l_1\cos(\theta_2) + l_{c2}\right) \\ I_2 + l_{c2} m_2\left(l_1\cos(\theta_2) + l_{c2}\right) & I_2 + l_{c2}^2 m_2 \end{pmatrix}$$

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

$$G = \begin{pmatrix} g l_{c1} m_1 \sin(\theta_1) + g m_2 (l_1 \sin(\theta_1) + l_{c2}\sin(\theta_1 + \theta_2)) \\ g m_2 l_{c2} \sin(\theta_1 + \theta_2) \end{pmatrix}$$

$$F = \begin{pmatrix} f_1 & 0 \\ 0 & f_2 \end{pmatrix}$$

Where l_{c1} and l_{c2} are the distances between the pivot points of the links and their centers of mass, m_1 and m_2 are the masses of the links, f_1 and f_2 are the viscous friction coefficients and g is the gravitational acceleration. l_1 and l_2 are the lengths of the links, while I_1 and I_2 are the moments of inertia of the links about their centers of mass. All the parameters of the robot are available in table 1.

Parameters: Set 1	
m_1	1
m_2	1
l_1	1
l_2	1
l_{c1}	0.5
l_{c2}	0.5
I_1	0.33
I_2	0.33
g	9.81
f_1	1.0
f_2	1.0

Parameters: Set 2	
m_1	2
m_2	2
l_1	1.5
l_2	1.5
l_{c1}	0.75
l_{c2}	0.75
I_1	1.5
I_2	1.5
g	9.81
f_1	1.0
f_2	1.0

Parameters: Set 3	
m_1	1.5
m_2	1.5
l_1	2
l_2	2
l_{c1}	1
l_{c2}	1
I_1	2
I_2	2
g	9.81
f_1	1.0
f_2	1.0

Table 1: Model parameters with variations.

Task 0 – Problem setup

Discretize the dynamics, write the discrete-time state-space equations and code the `dynamics` function.

Hint: Try different discretization schemes (e.g., Euler, Runge-Kutta, etc.). Some of them preserve stability and other properties. As an example, Runge-Kutta 4th order method has the following update rule:

$$k_1 = f(x_t,\, u_t)\,,$$

$$k_2 = f\left(x_t + \frac{d_t}{2}k_1,\, u_t\right)\,,$$

$$k_3 = f\left(x_t + \frac{d_t}{2}k_2,\, u_t\right)\,,$$

$$k_4 = f(x_t + d_t\, k_3,\, u_t)\,,$$

$$x_{t+1} = x_t + \frac{d_t}{6}\,(k_1 + 2k_2 + 2k_3 + k_4)\,.$$

Hint: You can use *symbolic* toolboxes to define the dynamics and compute the Jacobians.

Task 1 – Trajectory generation (I)

Compute two equilibria for your system and define a reference curve between the two. Compute the optimal transition to move from one equilibrium to another exploiting the Newton's-like algorithm (in closed-loop version) for optimal control.
Hint: you can exploit any numerical root-finding routine to compute the equilibria.
Hint: define two long constant parts between the two equilibria with a transition in between. Try to keep everything as symmetric as possible, see, e.g., Figure 2.

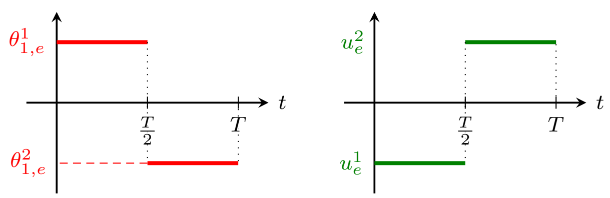

Figure 2: Example of a possible desired transition for one of the angles of the robot.

Task 2 – Trajectory generation (II)

Generate a desired (smooth) state-input curve and perform the trajectory generation task (Task 1) on this new desired curve.

Task 3 – Trajectory tracking via LQR

Linearizing the robot dynamics about the generated trajectory $(\mathbf{x}^{\text{gen}}, \mathbf{u}^{\text{gen}})$ computed in Task 2, exploit the LQR algorithm to define the optimal feedback controller to track this reference trajectory. In particular, you need to solve the LQ Problem

$$\begin{aligned}
\min_{\substack{\Delta x_1,\ldots,\Delta x_T \\ \Delta u_0,\ldots,\Delta u_{T-1}}} \quad & \sum_{t=0}^{T-1} \Delta x_t^\top Q^{\text{reg}} \Delta x_t + \Delta u_t^\top R^{\text{reg}} \Delta u_t + \Delta x_T^\top Q_T^{\text{reg}} \Delta x_T \\
\text{subj.to } & \Delta x_{t+1} = A_t^{\text{gen}} \Delta x_t + B_t^{\text{gen}} \Delta u_t \qquad t = 0, \ldots, T-1 \\
& x_0 = 0
\end{aligned}$$

where A_t^{gen}, B_t^{gen} represent the linearization of the (nonlinear) system about the optimal trajectory. The cost matrices of the regulator are a degree-of-freedom you have.
Hint: to showcase the tracking performances, consider a perturbed initial condition, i.e., different than x_0^{gen}.

Task 4 – Trajectory tracking via MPC

Linearizing the robot dynamics about the trajectory $(\mathbf{x}^{\text{gen}}, \mathbf{u}^{\text{gen}})$ computed in Task 2, exploit an MPC algorithm to track this reference trajectory.
Hint: to showcase the tracking performances, consider a perturbed initial condition, i.e., different than x_0^{gen}.

Task 5 – Animation

Produce a simple animation of the robot executing Task 3. You can use Python or any other visualization tool.

Required plots

For Tasks 1-2, you are required to attach to the report the following plots

- Optimal trajectory and desired curve.
- Optimal trajectory, desired curve and few intermediate trajectories.
- Armijo descent direction plot (at least of few initial and final iterations).
- Norm of the descent direction along iterations (semi-logarithmic scale).
- Cost along iterations (semi-logarithmic scale).

For the other tasks, you are required to attach to the report the following plots

- System trajectory and desired (optimal) trajectory.
- Tracking error for different initial conditions.

Guidelines and Hints

- As optimization algorithm, you can use the (regularized) Newton's method for optimal control introduced during the lectures based on the Hessians of the cost only.
- In the definition of the desired curve, you may try to calculate the desired trajectories using a simplified model, e.g., a simplified kinematic model.

Notes

1. Each group must be composed of 3 students (except for exceptional cases to be discussed with the instructor).
2. Any other information and material necessary for the project development will be given during project meetings.
3. The project report must be written in LaTeX and follow the main structure of the attached template.

4. Any email for project support must have the subject:

   "[OPTCON-RL]-Group X: rest of the subject".

5. **All** the emails exchanged **must be cc-ed** to professor Notarstefano, dr. Falotico and the other group members.

IMPORTANT: Instructions for the Final Submission

1. The final submission **deadline** is **one** week before the exam date.

2. One member of each group must send an email with subject "[OPTCON-RL]-Group X: Submission", with attached a link to a OneDrive folder shared with professor Notarstefano, dr. Falotico and the other group members.

3. The final submission folder must contain:

   - `report_group_XX.pdf`
   - `report` – a folder containing the LaTeX code and `figs` folder (if any)
   - `code` – a folder containing the code, including `README.txt`