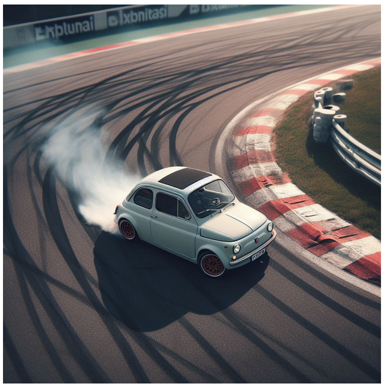
Optimal Control

Course Project #1
Optimal Control of a Vehicle

November 30, 2023

A fundamental task in high-performances autonomous driving is represented by the generation of optimal trajectories considering the whole vehicle dynamics.



This project deals with the design and implementation of an optimal control law for a simple bicycle model with static load transfer, schematically represented in Figure 1. The

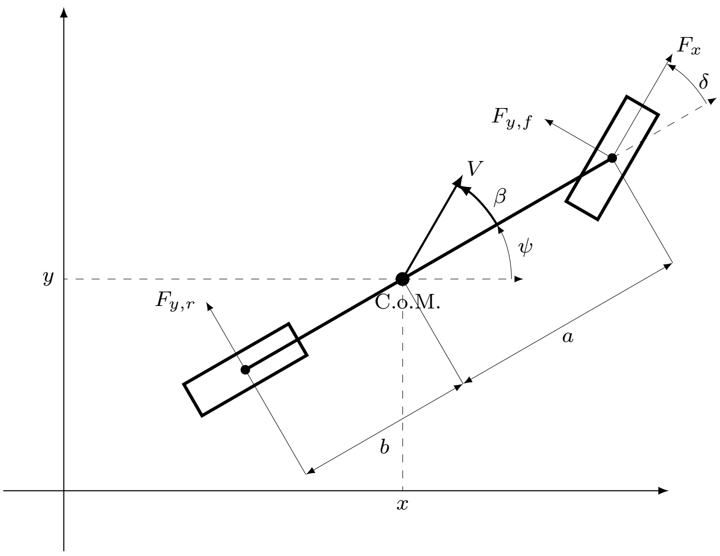


Figure 1: Bicycle model

state space consist in $\mathrm{x} = (x, y, \psi, V, \beta, \dot{\psi})$, where (x, y, ψ) are the cartesian coordinates of the center of mass of the vehicle and its yaw in a global reference frame, V is the speed modulus in body fixed reference frame, β is the angle between the speed and the body fixed reference frame and $\dot{\psi}$ is the yaw rate. The control input is $\mathrm{u} = (\delta, F_x)$ where δ is the steering angle of the vehicle, and F_x, which is the force applied by the front wheel.

The dynamic model is:

$$
\begin{aligned}
\dot{x} &= V\cos\beta\cos\psi - V\sin\beta\sin\psi \\
\dot{y} &= V\cos\beta\sin\psi + V\sin\beta\cos\psi \\
m\dot{V} &= F_{y,r}\sin\beta + F_x\cos(\beta-\delta) + F_{y,f}\sin(\beta-\delta) \\
\dot{\beta} &= \frac{1}{mV}\Big(F_{y,r}\cos\beta + F_{y,f}\cos(\beta-\delta) - F_x\sin(\beta-\delta)\Big) - \dot{\psi} \\
I_z\ddot{\psi} &= (F_x\sin\delta + F_{y,f}\cos\delta)a - F_{y,r}b.
\end{aligned}
$$

The lateral forces F_y can be found as

$$
\begin{aligned}
F_{y,f} &= \mu F_{z,f}\beta_f \\
F_{y,r} &= \mu F_{z,r}\beta_r
\end{aligned}
$$

where β_f, β_r are the front and rear sideslip angles,

$$
\begin{aligned}
\beta_f &= \delta - \frac{V\sin\beta + a\dot{\psi}}{V\cos\beta} \\
\beta_r &= -\frac{V\sin\beta - b\dot{\psi}}{V\cos\beta}.
\end{aligned}
$$

The vertical forces on the front and rear wheel can be found as:

$$
\begin{aligned}
F_{z,f} &= \frac{mgb}{a+b} \\
F_{z,r} &= \frac{mga}{a+b}
\end{aligned}
$$

The mechanical parameters of the car are available in Table 1.

Parameters:		
m	1480	$[Kg]$
I_z	1950	$[Kgm^2]$
a	1.421	$[m]$
b	1.029	$[m]$
μ	1	$[nodim]$
g	9.81	$[m/s^2]$

Table 1: Mechanical parameters of the vehicle

Task 0 – Problem setup

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

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 for optimal control.

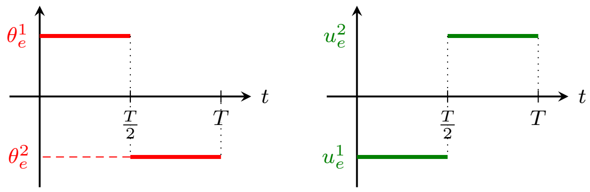


Figure 2: Example of possible desired trajectory for a pendulum system.

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.
Hint: As for the reference equilibria, it is convenient to consider the so-called *cornering equilibria*, i.e., those associated to the systems configurations where $\dot{\beta}$, $\dot{V}$ and $\ddot{\psi}$ are 0. The associated $x(t)$, $y(t)$ trajectory can then be obtained by forward integration of the dynamics with the values you just found.

Task 2 – Trajectory generation (II)

Generate a desired (smooth) state-input curve and perform the trajectory generation task (Task 1) on this new desired trajectory.
Hint: as initial guess you may need to compute a quasi-static trajectory, i.e., a collection of equilibria, and generate the first trajectory by tracking this quasi-static trajectory via the feedback matrix solution of an LQR problem computed on the linearization of the system about the quasi-static trajectory with a user-defined cost.

Task 3 – Trajectory tracking via LQR

Linearizing the vehicle dynamics about the (optimal) trajectory $(\mathbf{x}^{\text{opt}}, \mathbf{u}^{\text{opt}})$ 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

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

where A_t^{opt}, B_t^{opt} 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^{opt}.

Task 4 – Trajectory tracking via MPC

Linearizing the vehicle dynamics about the (optimal) trajectory $(\mathbf{x}^{\text{opt}}, \mathbf{u}^{\text{opt}})$ 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^{opt}.

Task 5 – Animation

Produce a simple animation of the vehicle 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]-Group X: rest of the subject".
5. **All** the emails exchanged **must be cc-ed** to professor Notarstefano, dr. Sforni 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]-Group X: Submission", with attached a link to a OneDrive folder shared with professor Notarstefano, dr. Sforni 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`