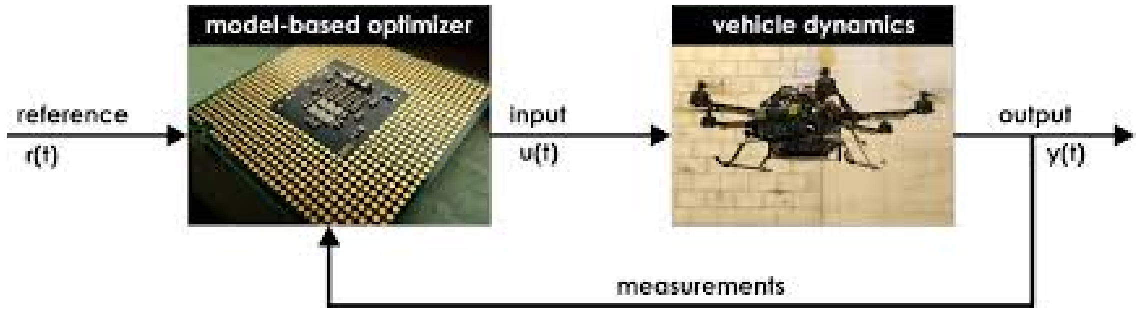
Control of Autonomous Systems – Autumn 2025

Jerome Jouffroy, PhD

Exercise Session 6



1. **A small Matlab primer on Quadratic Programming**.

   1.1. In a Matlab file, use the commands `meshgrid` and `surf` (you can also use the command `fsurf` if you prefer) to make a script to program and plot the two-dimensional parabola described by the function

   $$J(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 2)^2, \tag{1}$$

   where $J(x_1, x_2)$ should be plotted on intervals $x_1, x_2 \in [-10, 10] \times [-10, 10]$.

   1.2. Find the global minimum of function (1) by using quadratic programming through the command `quadprog` (hint: you need to rewrite (1) in order to find $\mathbf{H}$ and $\mathbf{F}$). Plot this point on top of your parabola (you can use `plot3` and `hold on` for that).

   1.3. Modify your program to find the local minimum on interval $[-5, 5] \times [-5, -3]$. Add this point on your plot.

2. **Model Predictive Control**.

   2.1. In a new Matlab script, make a loop to implement the discrete-time/digital system represented by the following state-space representation

   $$\begin{cases} \mathbf{x}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k), \qquad \mathbf{x}(0) = \mathbf{x}_0 = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \\ y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(k) \end{cases}, \tag{2}$$

   where you can decide for now what input $u(k)$ should be. Run this system and plot $\mathbf{x}(k)$ on 41 iterations (from $k = 0$ to $k = 40$). Check whether this system is stable by examining the eigenvalues of matrix $\mathbf{A}$.

2.2. We would like to stabilize system (2) with a linear MPC controller with matrices $\mathbf{Q} = \mathbf{C}^T\mathbf{C}$ and $R = 1/10$ with a receding horizon of $N = 3$, and the constraints $-1 \leq u(k) \leq 1$ for all $k \geq 0$. Find the corresponding quadratic programming terms $\mathbf{H}$ and $\mathbf{F}$.

2.3. Use the previously-computed terms $\mathbf{H}$ and $\mathbf{F}$ to implement your MPC controller within your script (use the command `quadprog`).

2.4. Plot the state $\mathbf{x}(k)$ and the control input $u(k)$ in two different figures. How long did it take to stabilize the system? Does the control input respect the given constraints?

2.5. Use the results of the previous questions to implement a linear MPC controller for system (2) in Simulink.

3. Feedback linearization of the controlled pendulum. Consider the actuated pendulum (robot with one degree of freedom)

$$ml^2\ddot{\theta} + d\dot{\theta} + mgl\sin\theta = u \tag{3}$$

where $m = 50$, $l = 1$, $g = 9.8$ and $d = 0.1$.

3.1. Re-use the Simulink model of system (3), which you made in exercise session 2.

3.2. In a feedback controller subsystem, implement a first control law allowing to cancel the system's nonlinear dynamics and obtain, after feedback, a simple double integrator.

3.3. Use the result to the previous question to stabilize the feedback-linearized system around the origin.

3.4. Modify your controller to stabilize the pendulum around any desired angular position.