Learning-Based Model Predictive Control

Tutorial session - Scenario Model Predictive Control of an electromechanical positioning system

Prof. Lorenzo Fagiano

Contact for this tutorial session:
Lorenzo Fagiano
lorenzo.fagiano@polimi.it

Control problem

The considered control problem is to manipulate the voltage applied to the motor of an electromechanical positioning system, in order to track a given reference position of the output shaft. The problem features both input and state constraints. Albeit the plant is single-input, single-output (SISO), the tutorial will be carried out in a general setup, so that the procedure and the developed functions will be easily re-usable for another problem of interest. On the other hand, considering a linear model allows us to work with linear-quadratic MPC, which is a class of MPC problems of particular interest, thanks to the availability of very efficient numerical solvers.

The learning goals are:

- To learn how to set up a model of the plant to be controlled;
- To learn how to set up a MPC strategy for a given control problem, defining cost and constraint functions and including slack variables to avoid feasibility problems;
- To learn how to implement a linear-quadratic MPC strategy;
- To learn how to employ scenario MPC results to tune the rate of constraint violations in presence of stochastic uncertainty.

Electromechanical positioning system

Consider the positioning system depicted in Figure 1. It consists of an electric gearmotor where we indicate with V the input voltage, θ_i and θ_o the angular positions of the input and output shafts respectively, T the torsional momentum applied to the output shaft, whose torsional stiffness is indicated with K_θ. R_{mot} is the motor electrical resistance, K_T is the motor constant, J_i and J_o the moments of inertia of the input and output shafts, β_i and β_o the friction coefficients of the shafts, finally τ_g is the transmission ratio of the gearbox. The numerical values of the system parameters are reported in Table 1.

The input variable is the voltage V. The control objective is to track a desired reference $\theta_{o,\text{ref}}$ of the output angular position θ_o, while keeping T below its yield point of $\pm\bar{T}$ Nm. The input voltage limits are equal to $\pm\bar{V}$. Finally, an external, unmeasured torque d can be also applied to the output shaft, e.g. by a load attached to it. Such a torque acts as a disturbance from the standpoint of the feedback control system.

The model equations in continuous time read ($\tau \in \mathbb{R}$ is the continuous time variable):

$$
\begin{array}{rcl}
J_i\ddot{\theta}_i(\tau) + \beta_i\dot{\theta}_i(\tau) & = & \dfrac{K_T V(\tau) - K_T^2\theta_i(\tau)}{R_{mot}} - \dfrac{T(\tau)}{\tau_g} \\
J_o\ddot{\theta}_o(\tau) + \beta_o\dot{\theta}_o(\tau) & = & T(\tau) \\
T(\tau) & = & K_\theta\left(\dfrac{\theta_i(\tau)}{\tau_g} - \theta_o(\tau)\right)
\end{array}
\qquad (1)
$$

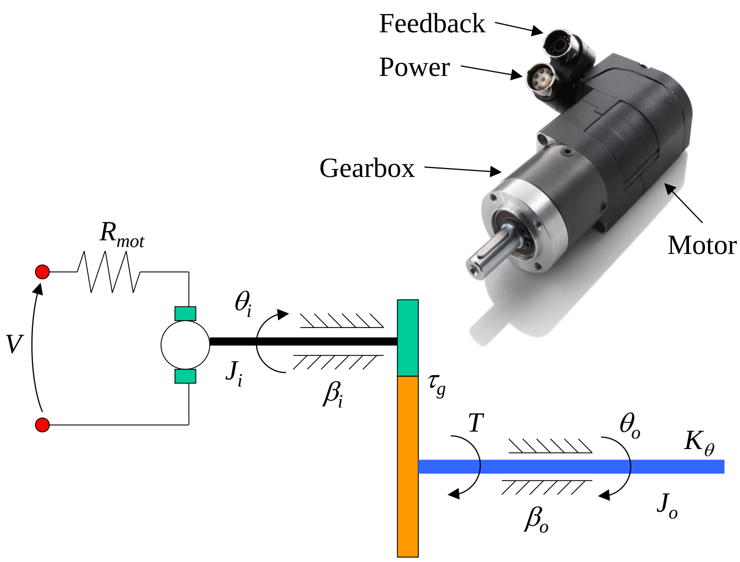

Figure 1: Example of servomotor with gearbox used in positioning systems, and related model.

Symbol	Description	Value
R_{mot}	Motor electrical resistance (Ω)	20
K_T	Motor constant (Nm/A)	10
K_θ	Output shaft torsional stiffness (Nm/rad)	1280
J_i	Input shaft moment of inertia (kg m^2)	0.5
J_o	Output shaft moment of inertia (kg m^2)	25
β_i	Input shaft friction coefficient (Nm s/rad)	0.1
β_o	Output shaft friction coefficient(Nm s/rad)	25
τ_g	Gear ratio (input/output)	20
$\bar{T}$	Maximal output shaft torsional moment (Nm)	25
$\bar{V}$	Maximal input voltage (V)	220

Table 1: Electromechanical positioning system - model parameters

We can arrange the equations in the standard form for LTI systems:

$$\begin{aligned}
\dot{\boldsymbol{x}}(\tau) &= \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{K_\theta}{J_i \tau_g^2} & \frac{K_\theta}{J_i \tau_g} & -\frac{\beta_i + \frac{K_T^2}{R_{mot}}}{J_i} & 0 \\ \frac{K_\theta}{J_o} & -\frac{K_\theta}{J_o} & 0 & -\frac{\beta_o}{J_o} \end{bmatrix}}_{A^{\mathrm{CT}}} \boldsymbol{x}(\tau) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{K_T}{J_i R_{mot}} \\ 0 \end{bmatrix}}_{B^{\mathrm{CT}}} u(\tau) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_o} \end{bmatrix}}_{B_d^{\mathrm{CT}}} d(\tau) \\
y(\tau) &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}}_{C^{\mathrm{CT}}} \boldsymbol{x}(\tau)
\end{aligned} \tag{2}$$

where $\boldsymbol{x} = [\theta_i, \theta_o, \dot{\theta}_i, \dot{\theta}_o]^T$, $u = V$, $y = \theta_o$. After choosing a suitable sampling time T_s, we obtain a discrete time LTI model of the form (3) (see the Appendix).

1 Tutorial session's assignments

This tutorial session includes three parts and one extra assignment.

I) Derive a continuous-time and a discrete-time model of the electromechanical positioning system.

The continuous-time model will be used to simulate the "real" system. Write in Matlab the model equations, and use a numerical integration method to be able to compute open-loop system trajectories, such as a fixed-step or variable-step solver. The discrete-time model will be used inside the MPC law. Derive the discrete-time LTI system matrices A, B, B_d, C, D using a method of choice and a suitable sampling time T_s, which will be the sampling period of the closed-loop system.

II) Formulate and solve a FHOCP for this problem.

The control objective is to track a desired reference $\theta_{o,\text{ref}}$ of the output angular position θ_o, while keeping $|T|$ below its yield point of $\bar{T}$ Nm. The input voltage limits are equal to $\pm\bar{V}$. This can be translated into a FHOCP with both input and state constraints (the torsional moment is in fact linear in the state), which can be written as a standard QP for improved efficiency, see the appendix for a review.

III) Formulate a MPC strategy, and simulate the positioning system with MPC.

Set up a MPC strategy where the FHOCP is re-formulated and solved at each time step, considering the current state as initial condition, and the first input of the obtained optimal sequence is applied to the plant. Simulate the feedback system. Try different prediction horizon values, and also include the disturbance. Note that the optimization problem may become unfeasible in certain situations: include soft constraints to solve this issue.

IV) Set up a scenario MPC strategy to control the rate of constraint violations.

Assume that the disturbance d has a non-zero average value and uniform distribution. Set up the problem with sampled disturbance predictions (scenarios) and the resulting constraints, and test what happens with different values of the nominal torque limit $\bar{T}$. Compute the empirical average-in-time number of constraint violations with and without the slack variable and analyze the results.

Appendix: Review of linear-quadratic MPC

The term linear-quadratic MPC refers to MPC for linear time invariant (LTI) systems with quadratic cost function and linear equality and/or inequality constraints. Linear-quadratic MPC entails a quadratic program (QP) to be solved at each time step: such a computation can be carried out in fractions of milliseconds with today's software and hardware. Even in such seemingly restricted settings (linear dynamics and constraints, quadratic cost), there are several variants of linear quadratic MPC, depending for example on the employed terminal cost and constraints, on the inclusion of integral action, on the use of soft constraints.

Consider the LTI, discrete-time system:

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) + B\boldsymbol{u}(k) + B_d\boldsymbol{d}(k) \\ \boldsymbol{y}(k) &= C\boldsymbol{x}(k) + D\boldsymbol{u}(k) \end{aligned} \tag{3}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{x}}}$, $\boldsymbol{u} \in \mathbb{R}^{n_{\boldsymbol{u}}}$, $\boldsymbol{d} \in \mathbb{R}^{n_{\boldsymbol{d}}}$, $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{y}}}$ and A, B, B_d, C, D are the system matrices. We assume that the system is fully reachable and observable. The FHOCP of interest takes the following form:

$$\min_U \sum_{i=0}^{N} (\boldsymbol{y}_{\text{ref}} - \boldsymbol{y}(i|k))^T Q (\boldsymbol{y}_{\text{ref}} - \boldsymbol{y}(i|k)) + \sum_{i=0}^{N-1} \boldsymbol{u}(i|k)^T R \boldsymbol{u}(i|k) \tag{4a}$$

subject to

$$\boldsymbol{x}(i+1|k) = A\boldsymbol{x}(i|k) + B\boldsymbol{u}(i|k) + B_d\boldsymbol{d}(i|k) \tag{4b}$$

$$\boldsymbol{y}(i|k) = C\boldsymbol{x}(i|k) + D\boldsymbol{u}(i|k) \tag{4c}$$

$$\boldsymbol{x}(0|k) = \boldsymbol{x}(k) \tag{4d}$$

$$C_{\boldsymbol{x}}\boldsymbol{x}(i|k) + \boldsymbol{d}_{\boldsymbol{x}} \geq 0,\ i = 0, \ldots, N \tag{4e}$$

$$C_{\boldsymbol{u}}\boldsymbol{u}(i|k) + \boldsymbol{d}_{\boldsymbol{u}} \geq 0,\ i = 0, \ldots, N-1 \tag{4f}$$

Notation $\boldsymbol{y}(i|k)$ indicates the value of $\boldsymbol{y}$ at time $k+i$, predicted at time k. N is the prediction horizon, $U = [\boldsymbol{u}(0|k)^T, \ldots, \boldsymbol{u}(N-1|k)^T]^T$, $Q > 0$, $R > 0$ are symmetric weighting matrices, and $C_{\boldsymbol{x}} \in \mathbb{R}^{q_{\boldsymbol{x}} \times n_{\boldsymbol{x}}}$, $C_{\boldsymbol{u}} \in \mathbb{R}^{q_{\boldsymbol{u}} \times n_{\boldsymbol{u}}}$, $\boldsymbol{d}_{\boldsymbol{x}} \in \mathbb{R}^{q_{\boldsymbol{x}}}$, $\boldsymbol{d}_{\boldsymbol{u}} \in \mathbb{R}^{q_{\boldsymbol{u}}}$ are known matrices and vectors defining the state and input inequality constraints. The initial state $\boldsymbol{x}(k)$ is assumed to be known.

Other formulations, involving mixed state-input constraints and additional equality constraints, can be considered with minor modifications.

We shall now re-write (4) as a QP in a standard form, such as:

$$\min_U \ \boldsymbol{c}^T U + \frac{1}{2} U^T H U \tag{5a}$$

subject to

$$GU + \boldsymbol{h} \geq 0 \tag{5b}$$

First, note that (4b)-(4d) can be eliminated by substitution, defining vector $\bar{x} = [\boldsymbol{x}(0|k)^T, \ldots, \boldsymbol{x}(N|k)^T]^T \in \mathbb{R}^{(N+1)n_{\boldsymbol{x}}}$ and $\bar{Y} = [\boldsymbol{y}(0|k)^T, \ldots, \boldsymbol{y}(N|k)^T]^T \in \mathbb{R}^{(N+1)n_{\boldsymbol{y}}}$:

$$\bar{X} = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\Lambda_{\boldsymbol{x}}} \boldsymbol{x}(k) + \underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ B & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ AB & B & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ A^{N-1}B & \cdots & \cdots & AB & B \end{bmatrix}}_{\Gamma_{\boldsymbol{x}}} U + \underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ B_d & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ AB_d & B_d & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ A^{N-1}B_d & \cdots & \cdots & AB_d & B_d \end{bmatrix}}_{E_{\boldsymbol{x}}} \Delta$$

$$\bar{Y} = \underbrace{\begin{bmatrix} C & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & C & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & C \end{bmatrix}}_{\bar{C}} \bar{X} + \underbrace{\begin{bmatrix} D & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & D & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & D \end{bmatrix}}_{\bar{D}} U = \underbrace{\bar{C}\Lambda_{\boldsymbol{x}}}_{\Lambda_{\boldsymbol{y}}} \boldsymbol{x}(k) + \underbrace{(\bar{C}\Gamma_{\boldsymbol{x}} + \Delta)}_{\Gamma_{\boldsymbol{y}}} U + \underbrace{(\bar{C}E_{\boldsymbol{x}})}_{E_{\boldsymbol{y}}} \Delta$$

Where $\Delta = [\boldsymbol{d}(0|k)^T, \ldots, \boldsymbol{d}(N-1|k)^T]^T$. Moreover, we define vector $\bar{Y}_{\text{ref}} = [\boldsymbol{y}_{\text{ref}}^T, \ldots, \boldsymbol{y}_{\text{ref}}^T]^T \in \mathbb{R}^{(N+1)n_{\boldsymbol{y}}}$, $\bar{D}_{\boldsymbol{x}} = [\boldsymbol{d}_{\boldsymbol{x}}^T, \ldots, \boldsymbol{d}_{\boldsymbol{x}}^T]^T \in \mathbb{R}^{(N+1)q_{\boldsymbol{x}}}$, $\bar{D}_{\boldsymbol{u}} = [\boldsymbol{d}_{\boldsymbol{u}}^T, \ldots, \boldsymbol{d}_{\boldsymbol{u}}^T]^T \in \mathbb{R}^{(N)q_{\boldsymbol{u}}}$, and matrices

$$\bar{C}_{\boldsymbol{x}} = \begin{bmatrix} C_{\boldsymbol{x}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & C_{\boldsymbol{x}} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & C_{\boldsymbol{x}} \end{bmatrix} \quad \bar{C}_{\boldsymbol{u}} = \begin{bmatrix} C_{\boldsymbol{u}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & C_{\boldsymbol{u}} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & C_{\boldsymbol{u}} \end{bmatrix}$$

$$\bar{Q} = \begin{bmatrix} Q & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & Q & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & Q \end{bmatrix} \quad \bar{R} = \begin{bmatrix} R & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & R & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & R \end{bmatrix}$$

We can now re-write the cost function in (4) as:

$$\begin{aligned} &\bar{Y}_{\text{ref}}^T \bar{Q} \bar{Y}_{\text{ref}} + (\Lambda_{\boldsymbol{y}}\boldsymbol{x}(k))^T \bar{Q} (\Lambda_{\boldsymbol{y}}\boldsymbol{x}(k)) + (\Gamma_{\boldsymbol{y}}U)^T \bar{Q} (\Gamma_{\boldsymbol{y}}U) + (E_{\boldsymbol{y}}\Delta)^T \bar{Q} (E_{\boldsymbol{y}}\Delta) \\ &-2\bar{Y}_{\text{ref}}^T \bar{Q} \Lambda_{\boldsymbol{y}}\boldsymbol{x}(k) - 2\bar{Y}_{\text{ref}}^T \bar{Q} \Gamma_{\boldsymbol{y}} U - 2\bar{Y}_{\text{ref}}^T \bar{Q} E_{\boldsymbol{y}} \Delta \\ &+2(\Lambda_{\boldsymbol{y}}\boldsymbol{x}(k))^T \bar{Q} \Gamma_{\boldsymbol{y}} U + 2(\Lambda_{\boldsymbol{y}}\boldsymbol{x}(k))^T \bar{Q} E_{\boldsymbol{y}} \Delta + 2(E_{\boldsymbol{y}}\Delta)^T \bar{Q} \Gamma_{\boldsymbol{y}} U + U^T \bar{R} U \end{aligned}$$

By eliminating terms in the cost function that do not depend on the decision variables U and dividing the cost by two, we obtain the following QP:

$$\min_U \underbrace{((\Lambda_{\boldsymbol{y}}\boldsymbol{x}(k))^T + (E_{\boldsymbol{y}}\Delta)^T - \bar{Y}_{\text{ref}}^T)\bar{Q}\Gamma_{\boldsymbol{y}}}_{\boldsymbol{c}^T} U + \frac{1}{2} U^T \underbrace{(\Gamma_{\boldsymbol{y}}^T \bar{Q} \Gamma_{\boldsymbol{y}} + \bar{R})}_{H} U \tag{6a}$$

subject to

$$\underbrace{\begin{bmatrix} \bar{C}_{\boldsymbol{x}}\Gamma_{\boldsymbol{x}} \\ \bar{C}_{\boldsymbol{u}} \end{bmatrix}}_{E} U + \underbrace{\begin{bmatrix} \bar{C}_{\boldsymbol{x}}\Lambda_{\boldsymbol{x}}\boldsymbol{x}(k) + \bar{C}_{\boldsymbol{x}}E_{\boldsymbol{x}}\Delta + \bar{D}_{\boldsymbol{x}} \\ \bar{D}_{\boldsymbol{u}} \end{bmatrix}}_{\boldsymbol{f}} \geq 0 \tag{6b}$$

$$\tag{6c}$$

where If matrices Q and R are positive definite, the QP is convex and its solution is a unique global minimum. The receding horizon implementation of the FHOCP leads to the following MPC strategy.

Linear-Quadratic Model Predictive Control

1. Acquire the current state $\boldsymbol{x}(k)$;
2. Solve the QP (6), let $U^* = [\boldsymbol{u}^*(0|k)^T, \ldots, \boldsymbol{u}^*(N-1|k)^T]^T$ be the optimal solution;
3. Apply the first input in the optimal sequence to the plant, i.e. $\boldsymbol{u}(k) = \boldsymbol{u}^*(0|k)$;
4. Set $k = k + 1$, go to 1.