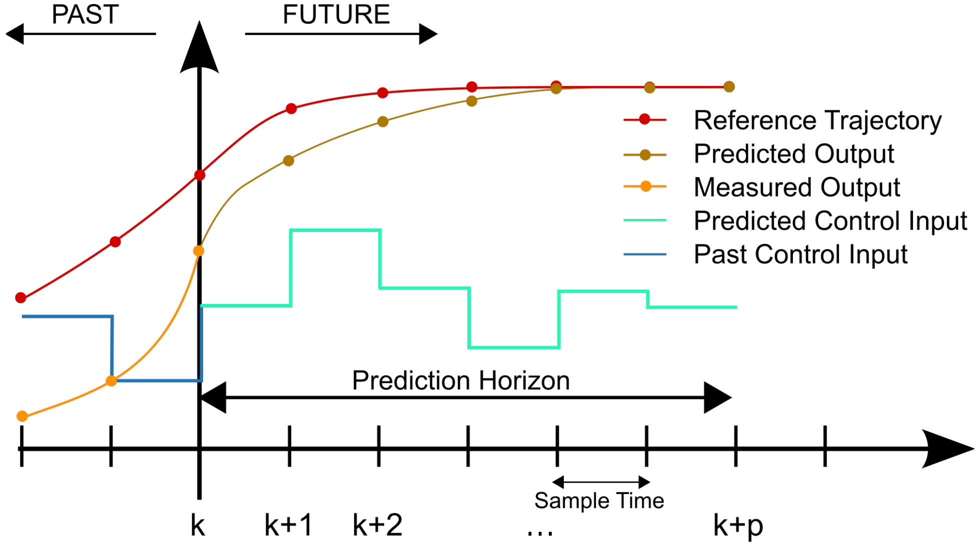
Instruction 1

5LMB0 - Model Predictive Control



1 Introduction

The following material is intended as a training exercise where students can apply techniques taught during the Model Predictive Control lectures on a real-life inspired control problem. The instructions are not subject to examination. However, it is highly recommended that students do the exercises in order to get familiar with course content and practice for the final examination. By following the systematic steps of the training exercises, students will gain valuable experience in developing Model Predictive Controllers and thus completing successfully the course homeworks.

The structure of the exercise is as follows: first two models will be presented, then a set of questions is given. Throughout the course, more questions will be published. It is advised to solve the questions first using the first-order model and subsequently generalize the solutions to be independent of the size of the model. The first-order model can be used to test if the solutions are correct for a second-order model.

2 Models

2.1 First-order model

Consider a 1st order model of a car moving in 1D:

$$\dot{p}(t) = v(t) \qquad t \in \mathbb{R}_+, \tag{1}$$

where $p(t)$ is the traveled distance and $\dot{p}(t) = v(t)$ is the velocity, which is the manipulated variable in this problem. The sampling period is $T_s = 0.1s$. To simulate the model you need to use an initial condition, for example $p(0) = 1$.

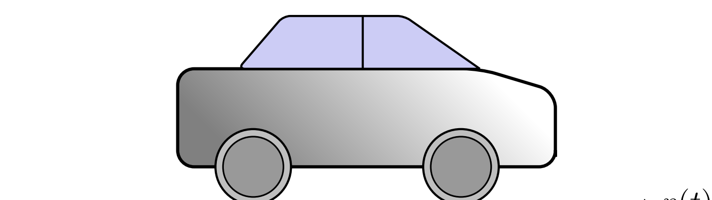


Figure 1: Simple car moving in 1D.

2.2 Second-order model

Consider two vehicles driving in a platoon formation. Figure 2 shows the inter-vehicle dynamics in a platoon formation. The variables $p_i(t)$, $v_i(t)$ and $a_i(t)$ for $i = 1, 2$, denote the position, the velocity and the acceleration of the corresponding vehicle.

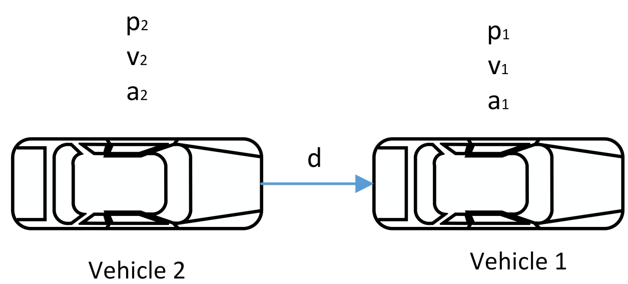


Figure 2: Two adjacent vehicles.

The goal is to control Vehicle 2 by applying an input (acceleration) $a_2(t)$ such that it maintains a safe distance d to Vehicle 1. This implies that the velocity of both vehicles must be equal, $v_1(t) = v_2(t)$. In this scenario, a human drives Vehicle 1 at a constant velocity, i.e. $v_1(t) = \bar{v}$ and $a_1(t) = 0$. The objective is to regulate the errors to zero. Hence we use the dynamics of the model for the errors:

$$
\begin{aligned}
\dot{e}_1(t) &= v_1(t) - v_2(t) = e_2(t), \\
\dot{e}_2(t) &= \dot{v}_1(t) - \dot{v}_2(t) = -a_2(t), \qquad t \in \mathbb{R}_+,
\end{aligned}
$$

where $e_1(t) = 0$ would mean that the distance equals the desired distance d and $e_2(t) = 0$ would mean that Vehicle 2 moves with the same velocity $v_1(t) = v_2(t)$. Thus, a controller must be designed such that the system is steered to the equilibrium point at the origin, with $e(t) := [e_1(t) \quad e_2(t)]^T = [0 \quad 0]^T$ and $a_2(t) = 0$. For simulation, initial conditions should be defined, for example $e(0) = [3 \quad 1]^T$. The sampling period is $T_s = 0.1s$.

3 Tasks

3.1 Plant model (State-space representation)

(a) Define the states $x(t)$, inputs $u(t)$ and outputs $y(t)$ for the system and rewrite it in a state-space form, defining matrices A, B, C and D:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{2}$$

Next, using the continuous model, derive the discrete-time state-space model by hand. You may use the equations for discretization of linear systems and the first-order Taylor approximation for the matrix exponential terms. What can you say about the discrete model? (You have used a Taylor approximation: how accurate is the model?)
Hint: more details on discretization can be found here: http://profjrwhite.com/sdyn_course/SDYNnotes_files/sdyn_s3.pdf, *page 24-26.*

Solution:
The first-order model in the state space form:

$$\dot{x} = u, \tag{3}$$

with position $p(t)$ as a state x and velocity $v(t)$ as a control variable u.

The second-order model is

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -u, \end{aligned} \tag{4}$$

where the state vector is $x := [x_1 \;\; x_2]^T = [e_1 \;\; e_2]^T$ corresponding to the position and velocity error and the acceleration is the manipulated variable u.

$A_d = e^{A_c T_s}, \quad B_d = \int_0^{T_s} e^{A_c \tau} d\tau B_c, \quad C_d = C_c, \quad D_d = D_c$
(from http://profjrwhite.com/sdyn_course/SDYNnotes_files/sdyn_s3.pdf, *page 24-26.*)
For the 2x2 matrix use
$e^{A\tau} = I + A\tau + \frac{1}{2}A^2\tau^2 + \ldots$
$\int_0^T e^{At} dt = T(I + \frac{AT}{2!} + \frac{(AT)^2}{3!} + \ldots)$

Note that you have obtained a discrete state-space model which completely resembles the continuous model in the case that the control input u(t) is constant between sampling time instants ($A = 0$ for the first-order model and $A^2 = 0$ for the second-order model).

(b) Find the discrete-time state-space model using MATLAB (use the `c2d.m` function). Compare your answer with the model obtained in item (a).

Solution:

$T_s = 0.1, \quad A_1 = 1,\ B_1 = 0.1,\ C_1 = 1,\ D_1 = 0$

$T_s = 0.1, \quad A_2 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix},\ B_2 = \begin{bmatrix} -0.005 \\ -0.1 \end{bmatrix},\ C_2 = I,\ D_2 = 0$

(c) Is the system stabilizable?

Solution:

$\mathrm{ctrb}(A_1, B_1) = 0.1$ is full rank, thus controllable and stabilizable.

$\mathrm{ctrb}(A_2, B_2) = \begin{bmatrix} -0.005 & -0.015 \\ -0.1 & -0.1 \end{bmatrix}$ is full rank, thus controllable and stabilizable.

(d) Design an LQR controller. Choose Q and R yourself. Simulate the closed-loop system for various initial conditions and plot the trajectories.

Solution:

$Q_1 = R_1 = 1,\ K_1 = \mathrm{dlqr}(A_1, B_1, Q_1, R_1) = 0.9512$

$Q_2 = I,\ R_2 = 1,\ K_2 = \mathrm{dlqr}(A_2, B_2, Q_2, R_2) = [0.9171 \quad 1.6356]$

(e) Is the closed-loop system with LQR controller stable?

Solution:

$\mathrm{eig}(A_1 - B_1 * K_1) = 0.9049$ is inside unit circle

$\mathrm{eig}(A_2 - B_2 * K_2) = 0.9159 \pm 0.0459i$ are inside unit circle

(f) What happens if you saturate the input to the system $u(t)$ between -1 and 1? And between -0.1 and 0.1?

Solution: The second-order model starts to oscillate with a saturation of [0.1,-0.1]. A saturation of [1,-1] only makes the system converge slower.

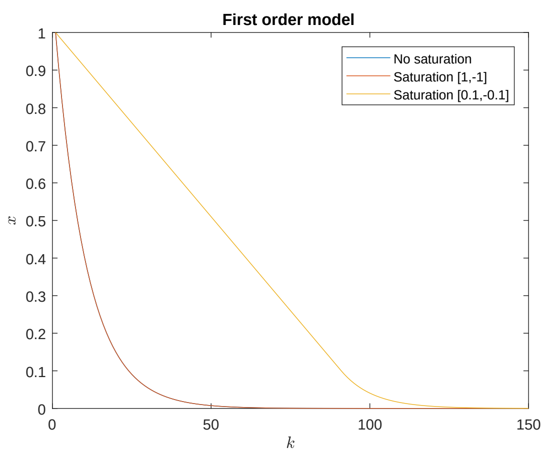


Figure 3: Trajectories of model 1

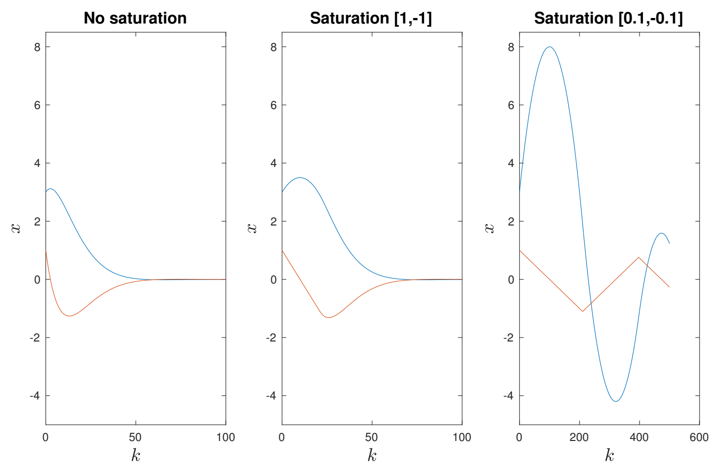


Figure 4: Trajectories of model 2

3.2 Prediction model and cost function: compact formulation

(a) Construct prediction matrices Φ and Γ. Write a MATLAB function with inputs A and B which outputs Φ and Γ for any prediction horizon $N \in \mathbb{N}$.

Solution: $$\Phi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \Gamma = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}$$

(b) Write a MATLAB function with inputs Q, P, R and N, which outputs the matrices Ω and Ψ. Construct the MPC cost function matrices G and F using the prediction model.

Solution: $$\Omega = \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & P \end{bmatrix}, \quad \Psi = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}$$

$$G = 2(\Psi + \Gamma^T \Omega \Gamma), \quad F = 2\Gamma^T \Omega \Phi$$

3.3 Unconstrained MPC

(a) Compute the unconstrained MPC feedback matrix K_{MPC} for different combinations of $Q/R/P/N$.

Solution: Check the provided MATLAB files

(b) Simulate the system in closed-loop with the obtained unconstrained MPC controller.

Solution: Use $\nabla_{Uk} J(x(k), U_k) = GU_k + Fx(k) = 0$, which results in $K_{MPC} = -(I_m \quad 0 \quad \dots \quad 0)G^{-1}F$

(c) What can you say about the closed-loop stability? Is the system closed-loop stable? (try various $Q/R/P/N$)

Solution: Systems stability is guaranteed if P is chosen to be a solution to the dARE. This way the resulting controller $K_{MPC} = K_{LQR}$. Details on stability will be treated later in the course.

Intuitively, it makes no sense to set $P \prec 0$, since the cost might not be positive definite. P should be chosen strictly positive definite to avoid non-uniqueness of the optimum and to guarantee stability, as it will be shown later on.

The weighting matrices Q and R have to be positive definite (or one of them may be positive semi-definite: $R \succ 0 \quad \& \quad Q \succeq 0$ or $R \succeq 0 \quad \& \quad Q \succ 0$) in order to have a unique solution (e.g. strictly convex quadratic cost). For Lyapunov stability, it must hold that $Q \succ 0$.

(d) Compare the unconstrained MPC with the LQR. Do you notice any similarities?

Solution: unconstrained MPC and LQR should give the same results when Q and R are chosen the same and the P from the LQR is used for the MPC. If P is smaller (in the positive definiteness sense) than the P corresponding to dARE solution and hence, LQR, the MPC control law will be less aggressive. If P is larger the MPC will still be similar to LQR.