FRTN15 Predictive Control—Home Work 3
Model Predictive Control

In this homework exercise, the purpose is for you to familiarise yourself with the concepts of model predictive control (MPC) and see how it may be implemented in practice using standard functions in Matlab. We will study a simple double integrator system, which has a particular appeal in that it is a reasonable approximation of several physical systems, including the translational dynamics of quadcopters.

To pass the homework, answer each of the five questions in a short report (no more than three pages) documenting your findings and implementation.

In continuous time, the double integrator takes the form,

$$\ddot{x}(t) = u(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0, \qquad \mathbf{x}(t) \in D_{\mathbf{x}} \subseteq \mathbb{R}^2, \qquad u(t) \in D_u \subseteq \mathbb{R}, \tag{1}$$

with a state vector $\mathbf{x}(t) \triangleq [x(t), \dot{x}(t)]^T$ and an initial condition $\mathbf{x}_0$ defined on a domain $D_{\mathbf{x}}$, and a control signal u defined on a domain D_u. This system may be written on many forms, one for instance a continuous form

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c u(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t). \tag{2}$$

For the purpose of controlling the system using MPC, we need to evaluate the dynamics at discrete points in time on a prediction horizon and control horizon. Assuming the points are located equitemporally on the horizon with a time-step h, we can let $t = hk$ and find the dynamics on the form,

$$\mathbf{x}(t+h) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t). \tag{3}$$

Exercise 1 *Define the matrices of the discrete time dynamics explicitly. A useful tool is the* `c2d()` *function.*

Now we are ready to formulate the MPC. Our goal is to follow a time-varying reference trajectory $\mathbf{x}_r(t) \triangleq [x_r(t), \dot{x}_r(t)]^T$, by computing a suitable control signal $u(t)$ on each time step hk. We will do this by solving a small optimisation problem on each such time-step using Matlabs `quadprog()` function, but any other QP-solver may be used to similar effect, for instance the `scipy` or `cvxopt` modules in Python, or `Convex.jl` package in Julia. At a time t, we set out to minimize a quadratic cost on the *prediction horizon* and *control horizon* respectively. The cost is defined as

$$J(t) = \sum_{k=0}^{H_p} \|\mathbf{x}_r(t+hk) - \mathbf{x}(t+hk)\|_{\mathbf{Q}} + \sum_{k=0}^{H_p-1} \|u(t+hk)\|_R, \tag{4a}$$

subject to

$$\ddot{x}(t) = u(t) \tag{4b}$$

$$\|\mathbf{x}(t)\|_1 \leq x_{max} = 10 \tag{4c}$$

$$|u(t)| \leq u_{max} = 2 \tag{4d}$$

for some $\mathbf{Q} \in \mathbb{R}^2, R \in \mathbb{R}$, with the conventional notation $\|\mathbf{u}\|_{\mathbf{Q}} = \mathbf{u}^T\mathbf{Q}\mathbf{u}$. For simplicity, we will let $H_u = H_p - 1$, and proceed by deriving an MPC in five steps.

Exercise 2

Define a vector $\bar{\mathbf{x}}(t)$ containing the N points on the prediction and control horizon, with associated matrices $\bar{\mathbf{H}}, \bar{\mathbf{f}}$ such that the cost function (4a) may is expressed

$$J(t) = \frac{1}{2}\bar{\mathbf{x}}(t)^T\mathbf{H}\bar{\mathbf{x}}(t) + \mathbf{f}^T\bar{\mathbf{x}}(t). \tag{5}$$

Hint: We will optimize over $\bar{\mathbf{x}}$ which contains $\{\mathbf{x}(t+hk), u(t+hk)\}_{k=0}^{H_p-1}$ on the entire prediction and control horizon. The matrix $\bar{\mathbf{H}}$ will only contain $\{\mathbf{Q}, R\}$ and $\bar{\mathbf{f}}$ will contain $\{\mathbf{Q}, R\}$ and $\{\mathbf{x}_r(t+hk)\}_{k=0}^{H_p-1}$ evaluated over the prediction horizon,

Now that we have defined a cost function, we need to find a way of writing the constraints in terms of the vector $\bar{\mathbf{x}}$. The dynamical constraint (4b) may be written in terms of an equality constraints $\mathbf{A}_{eq}\bar{\mathbf{x}} = \mathbf{B}_{eq}$ by defining a suitable $\mathbf{A}_{eq}$ matrix and $\mathbf{B}_{eq}$ vector. Similarly, the domain constraints, (4c) and (4d), on the form $\mathbf{A}_{ineq}\bar{\mathbf{x}} \leq \mathbf{B}_{ineq}$ by defining a suitable $\mathbf{A}_{ineq}$ matrix and $\mathbf{B}_{ineq}$ vector.

Exercise 3

Define suitable matrices and vectors $\mathbf{A}_{eq}, \mathbf{B}_{eq}, \mathbf{A}_{ineq}, \mathbf{B}_{ineq}$ explicitly.

Exercise 4

Complete the code in `MPC_example.m` which solves the optimisation problem on each time step by implementing `quadprog(H,f,Aineq,Bineq,Aeq,Beq)`. Simulate the system (1) with the controlled by the MPC over $T = 10$ seconds with $\mathbf{x}_0 = [2, 2]^T$, a reference trajectory $x_r = \sin(2t)$ and a prediction horizon of $H_p = 10$.

1. Are the state constraints satisfied?
2. How does the system behave if you bound the control signal to $u_{max} = 3$?
3. What happens if you change the length of the horizons?
4. What happens if you change the initial state to $\mathbf{x}_0 = [0, 0]^T$ and let $\mathbf{x}_{max} = 3$?

Exercise 5

Try to replace the double integrator with a critically damped second order system

$$\frac{1}{4}\ddot{x}(t) = -\dot{x}(t) - x(t) + u(t), \tag{6}$$

by writing the system on a discrete-time state-space form and replacing the double integrator model with the damped system in the code. What do you observe?