5LMB0 MPC: Graded Homework project assignment

Required material for grading:

- One report per group of maximum 2 students with no more than 5 Pages (A4, 10 - 11pt font). The report is preferably in pdf format built by Latex. It must provide the obtained results, as requested by each question; observe if a specific plot is requested; explain succinctly the adopted MPC design methodology, justify your design choices and comment on the obtained performance and difficulties encountered.
- The Matlab files for the controller synthesis and simulation.

Submission instructions: Please submit your results by e-mail to both s.shi@tue.nl and m.lazar@tue.nl, and attach 2 files: 1 PDF file of your report; and 1 7-zip archive of your Matlab files. Please write your full name and student number on the first page of the report. Do not use other archivers. Make sure all sub-functions are included so that your Matlab code can be run and make sure your code is compatible with previous versions of Matlab. Use commented lines in your Matlab code to explain the function (building up constraint matrices, simulation loop, plotting code, for example) of separate parts of your code.

Submission deadline: April 30, 2019, 23:59 hours (end of the working day).

Assignment description: The project assignment contains two main parts, one part related to the regulation control problem and another one related to the tracking control and disturbance rejection problems. The **deliverable** of the answers aims to give the students some hints about the necessary elements of the report but they are not sufficient for a perfect report. Besides the points mentioned in the guidelines, the students are encouraged to explore more subjects and using their creativity to come up with solutions.

Restriction on using MPT3: Apart from polytopic operations, computation of invariant sets, computation of the terminal cost using YALMIP, the MPT3 toolbox may also be used for constructing explicit MPC controllers and for reducing their complexity. YALMIP can also be used for designing nonlinear MPC controllers and disturbance-model based MPC controllers. However, the MPT3 toolbox, or YALMIP, **may not** be used for designing and simulating on-line QP-based constrained linear or quasi-LPV MPC controllers, either for constant set-point regulation or tracking problems.

Part 1: MPC design for constant set-point regulation

In this project, we consider the Van der Pol oscillator which was originally proposed by the Dutch engineer Balthasar van der Pol when he was working at Philips around 1922. The Van der Pol equation and corresponding oscillations have a long history of being used in several science domains, including electrical, physical and biological sciences. See Figure 1 for an example of using the Van der Pol oscillator to reproduce hip oscillations.

The corresponding continuous-time model of the oscillator is

$$\ddot{p}(t) = \mu(1 - p^2(t))\dot{p}(t) - p(t) + u(t), \tag{1}$$

where $p(t)$ denotes position and $u(t)$ is the input, $\mu = 2$ and

$$-2 \leqslant p(t) \leqslant 2,\ -5 \leqslant \dot{p}(t) \leqslant 5,\ -1 \leqslant u(t) \leqslant 1, \text{ for all } t.$$

The considered sampling time for the Van der Pol oscillator is $T_s = 0.1$.

The objective is to design MPC controllers to achieve $p(t) = 0$, $\dot{p}(t) = 0$ and $u(t) = 0$ when t approaches infinity. To prepare for the controller design, please complete the first task:

1. Based on (1), define position and velocity as the states of the oscillator and formulate a continuous-time state space model (only state equation) in the following form:

   $$\dot{x}(t) = f(x(t), u(t)). \tag{2}$$

   Simulate the nonlinear model with zero input and observe the trajectories of the system in state space, i.e. in a plot with x_1 as x axis and x_2 as y axis.

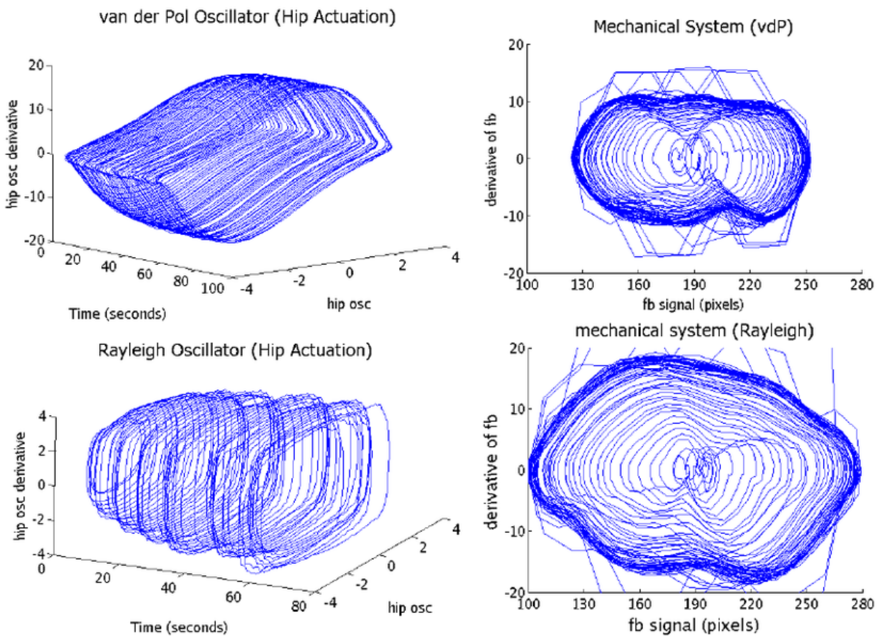

Figure 1: Phase portraits of different oscillators, including the Van der Pol oscillator.

Deliverable: Explain and show the formulation of state space model; Trajectories of the system in state space and analysis of its behavior.

Hint: Use the provided *Simulink* model for correctness of simulation and the Matlab function "sim". Do not change any of the parameters inside the Simulink model, i.e., variable step size, numerical integration method, initial conditions for integrators, etc.

2. This part concerns the design of linear MPC controllers:

   - Linearize the nonlinear system (2) at the state $x = [1\ 0]^T$ and input $u = 0$ and discretize the resulting linear system using the Matlab function "c2d". For the resulting discrete-time linear model design a linear MPC controller with the guarantee of stability and recursive feasibility with the following settings:

     $$N = 50,\ Q = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix},\ R = 0.1,$$

     where N is the prediction horizon, Q is the cost matrix for states, R is the cost for inputs. In addition, use the LQR controller as the terminal controller and the corresponding P from the discrete-time Riccati equation as the terminal cost. Given initial state $[1\ 0]^T$, simulate the linear MPC controller in closed-loop with the linear discrete-time system and then in closed-loop with the provided Simulink model of the nonlinear system (2). Compare and explain the results.

   - Tune the linear MPC controller cost matrices Q and R such that the states of the Simulink model of the nonlinear system (2) converge to the origin.

   - Linearize the nonlinear system (2) at the state $x = [0\ 0]^T$ and input $u = 0$ and discretize the resulting linear system using the Matlab function "c2d". For the resulting discrete-time linear model design a linear MPC controller with the guarantee of stability and recursive feasibility. Given initial state $[1\ 0]^T$, simulate the linear MPC controller in closed-loop with the provided Simulink model of the nonlinear system (2). Explain the results.

Deliverable: Time trajectories of the simulation tasks (input, states over time and states in state space); Explain and analyze the results; Plot of feasible set.

3. The second task concerns the design of a nonlinear MPC controller to regulate the system, which should be designed based on the following discrete-time approximation of (2):

$$x(k+1) = x(k) + T_s f(x(t), u(t)). \tag{3}$$

   - Simulate the above discrete-time model with zero input and compare the resulting trajectories with the open-loop simulation results of the Simulink model of (2).
   - Simulate the nonlinear MPC controller in closed loop with the Simulink model of the nonlinear system (2). Observe and compare the results with the ones obtained above for linear MPC in closed-loop with the same Simulink model. Test also the initial condition $[1\ 0]^T$. In this task you are free to choose N, P, Q and R as you think best to optimize closed-loop performance.

   Deliverable: Explain the difference of the open-loop simulations using the two different nonlinear models, i.e. the Simulink model (2) and the discrete-time model (3); Explain the nonlinear MPC design and plot the closed–loop trajectories. Observe and explain differences in the obtained closed–loop trajectories by using nonlinear MPC versus linear MPC.

4. Formulate model (3) in quasi linear parameter varying form and then design an iterative QP nonlinear MPC. Simulate the MPC controller based on quasi-LPV model in closed-loop with the nonlinear Simulink model and compare the results with the linear and the nonlinear MPC controller for the same N, P, Q and R.

5. In this third task, fix $N = 5$ and choose your "best" P, Q and R to be used for all linear, nonlinear and quasi-LPV MPC controllers. Then conduct the following comparisons among these MPC controllers:

   - Compare the average computation time of the linear, nonlinear and quasi-LPV based MPC controllers in closed-loop with the Simulink nonlinear system.
   - Make a uniform grid of the state-space box defined by the state constraints. Use the points of the grid as initial states and simulate all three MPC controllers in closed-loop with the Simulink model of the nonlinear system (2) for each initial state. Make a plot for each of the MPC controllers in which you show the grid points using "*" and plot a circle around each grid point for which the corresponding MPC problem was infeasible during simulation. These plots will provide you with an estimate of the feasible set of states for each MPC controller. Compare the results and explain them.
   - Compare the estimate (grid points) of the feasible set of states obtained above for the linear MPC controller with the actual feasible set of states (polytopic set) of the linear MPC controller computed as in Lecture 3 or the Instructions. Provide plots of both the grid points estimate and the feasible set of states in the same figure and comment the results.

   Hint: Useful Matlab functions: *meshgrid*-generation of grid points; *try* and *catch*-can be used to stop a simulation that has run infeasible and automatically start a new simulation with a different grid point as initial state.

Part 2: MPC design for reference tracking and disturbance rejection

Consider the longitudinal model of an aircraft flight dynamics, which has as states the angle of elevation, $x_1 = \alpha$ and the pitch angular rate, $x_2 = q$, while the control input $u = \delta$ is the elevator control surface deflection.
After linearization and discretization, the flight dynamics is given by the discrete-time 2D model:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.9719 & 0.0155 \\ 0.2097 & 0.9705 \end{bmatrix} x(k) + \begin{bmatrix} 0.0071 \\ 0.3263 \end{bmatrix} u(k), \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(k), \end{aligned}$$

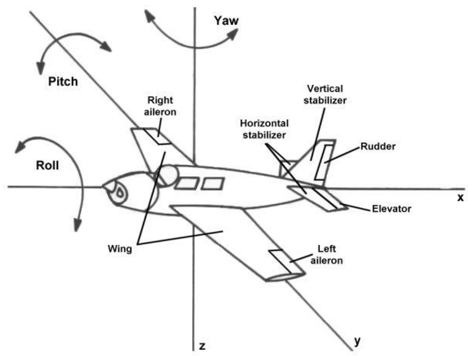

Figure 2: Graphical depiction of aircraft axis of motion.

where the state and the input should satisfy the following constraints:

$$x(k) \in \mathbb{X} = \left\{ \begin{bmatrix} \alpha \\ q \end{bmatrix} \mid -15 \leqslant \alpha \leqslant 30, -100 \leqslant q \leqslant 100 \right\},$$
$$u(k) \in \mathbb{U} = \{\delta \mid -25 \leqslant \delta \leqslant 25\}, \quad \forall k \in \mathbb{N}.$$

The considered initial state is $x(0) = [8\ 0]^T$ and we assume that only the output can be measured. For all the controllers which will be designed in Part 2, the guarantee of stability and recursive feasibility is not required.

Please complete the following tasks and answer questions using the given discrete-time model:

1. Design an MPC controller which is able to track the reference given in Figure 3.

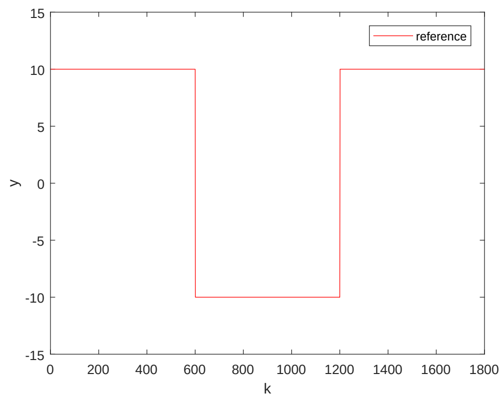

Figure 3: Output reference to be tracked.

The MPC controller should achieve good tracking performance: small overshoot, fast response and minimum tracking error (zero tracking error whenever the reference is constant).

Deliverable: Explain the MPC design and plot closed-loop output trajectories over the desired reference; Plot the estimated states from the observer and the real closed-loop states.

2. Due to air turbulence there is a disturbance acting on both states as follows:

$$x(k+1) = \begin{bmatrix} 0.9719 & 0.0155 \\ 0.2097 & 0.9705 \end{bmatrix} x(k) + \begin{bmatrix} 0.0071 \\ 0.3263 \end{bmatrix} u(k) + \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix} d(k),$$
$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k).$$

- Consider the following realization of the disturbance: $d(k) = 0.1$ for $k \in [0,\ 600]$, $d(k) = 0.5$ for $k \in [601,\ 1200]$, and $d(k) = -0.5$ for $k \in [1201,\ 1800]$. Design a linear MPC controller with disturbance model to track the reference in Figure 3 with zero offset.

Deliverable: Plot the estimated disturbance from the observer and the actual disturbance, as well as the closed-loop output trajectories over the desired reference.