OptReg 5

Problem 1

a) Stable if $|eig(A)| \leq 1$

$$\det(\lambda I - A) = \begin{vmatrix} \lambda & 0 & 0 \\ 0 & \lambda & -1 \\ -0.1 & 0.79 & \lambda - 1.78 \end{vmatrix} = \lambda\left(\lambda(\lambda - 1.78) + 0.79\right)$$

$$= \lambda^3 - 1.78\lambda^2 + 0.79\lambda = \lambda(\lambda^2 - 1.78\lambda + 0.79) = 0$$

$$\lambda_1 = 0$$

$$\lambda_{2,3} = \frac{1.78 \pm \sqrt{1.78^2 - 4 \cdot 0.79}}{2} \approx \frac{1.78 \pm 0.09}{2} = 0.845 \text{ ^ } 0.935$$

Since $|\lambda_i| \leq 1$ for all i, the system is stable.

b) $\dim(\underline{x}_t) = 3$, $\dim(u) = 1$

$$y_{t+1} = C\underline{x}_{t+1} = x_{3,t+1}$$

want (2) on the form

$$f(z) = \frac{1}{2}\sum_{t=0}^{N-1} 2x_{3,t+1}^2 + 2u_t$$

$$\Rightarrow f(z) = \frac{1}{2}\sum_{t=0}^{N-1}\left[\underline{x}_{t+1}^T C^T C\, \underline{x}_{t+1} + u_t \cdot 2 \cdot u_t\right]$$

$$Q = 2C^TC = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad R = 2$$

c) The minimization problem is convex since:

$Q \geq 0$ (positive semi-definite)

$R > 0$ (positive definite).

Thus the objective function is convex, and since there are only linear constraints, the feasible set is also convex. Thus the problem is convex.

Convexity only depends on Q and R.

d) Firstly, we have

$$z = [\underline{x}_1^T, \underline{x}_2^T, \dots, \underline{x}_N^T, U_0, U_1, \dots, U_{N-1}]^T$$

where

$$\underline{x}_{k+1} = A\underline{x}_k + BU_k$$

which gives

$$\left.\begin{aligned} \underline{x}_1 &= A\underline{x}_0 + BU_0 \\ \underline{x}_2 &= A\underline{x}_1 + BU_1 \\ &\vdots \\ x_N &= A\underline{x}_{N-1} + BU_{N-1} \end{aligned}\right\} \quad \begin{aligned} \underline{x}_1 - BU_0 &= A\underline{x}_0 \\ \underline{x}_2 - A\underline{x}_1 - BU_1 &= 0 \\ &\vdots \\ x_N - A\underline{x}_{N-1} - BU_{N-1} &= 0 \end{aligned}$$

This gives us the matrix

$$A_{eq} = \begin{bmatrix} I & 0 & \cdots & 0 & -B & 0 & \cdots & 0 \\ A & I & \cdots & 0 & 0 & -B & \cdots & 0 \\ 0 & -A & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & I & 0 & 0 & \cdots & -B \end{bmatrix} \quad \begin{matrix} N\cdot\dim(x)+N\cdot\dim(U) \\ \\ \\ \\ N\cdot\dim(x) \end{matrix}$$

$$b_{eq} = \begin{bmatrix} Ax_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad N\dim(x)$$

d) Since we have

$$f(z) = \frac{1}{2}\sum_{k=0}^{N-1}\left(x_{k+1}^T Q\, x_{k+1} + u_k^T R u_k\right)$$

where

$$z = \left[x_1^T, x_2^T, \dots, x_N^T, u_0, \dots, u_{N-1}\right]^T$$

we want a matrix G on the form

$$G = \begin{bmatrix} Q & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & & 0 \\ 0 & Q & 0 & \cdots & 0 & 0 & \cdots & 0 & & \\ \vdots & & \ddots & & & & & \vdots & & \\ 0 & 0 & 0 & & Q & 0 & \cdots & 0 & & \\ 0 & 0 & \cdots & 0 & R & & & & & \\ \vdots & & & & & \ddots & & & & \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & R & & \end{bmatrix}$$

Want to minimize a function on the form

$$\min f(x) = \frac{1}{2}x^T G x + x^T C \quad \text{s.t. } Ax = b$$

$$\begin{bmatrix} G & -A_{eq}^T \\ A_{eq} & 0 \end{bmatrix}\begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -C \\ b \end{bmatrix}$$

In our case

$$\begin{bmatrix} G & -A_{eq}^T \\ A_{eq} & 0 \end{bmatrix}\begin{bmatrix} z^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} 0 \\ b_{eq} \end{bmatrix}$$

Problem 1

d)

The plot is shown in Figure 1.
We call this form of control open-loop because it does not use feedback. This means that the system only uses the precalculated values for the input, and does not adjust based on it's current state.
By including feedback, the system can adjust for modelling errors and disturbances/noice, which makes it more robust.

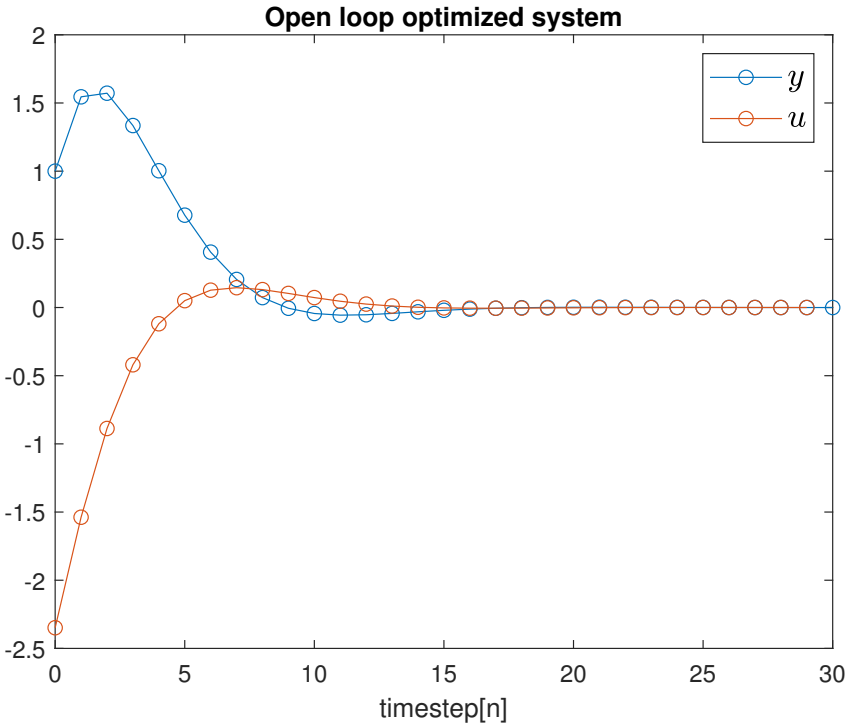

Figure 1: Optimal plot using KKT system

e)

The plot with the same r is very similar to the one in d), so this is not shown here. With small r, it becomes as shown in Figure 2, and with large r as in Figure 3.
Quadprog uses 1 iteration to solve the problem.

Exercise 5

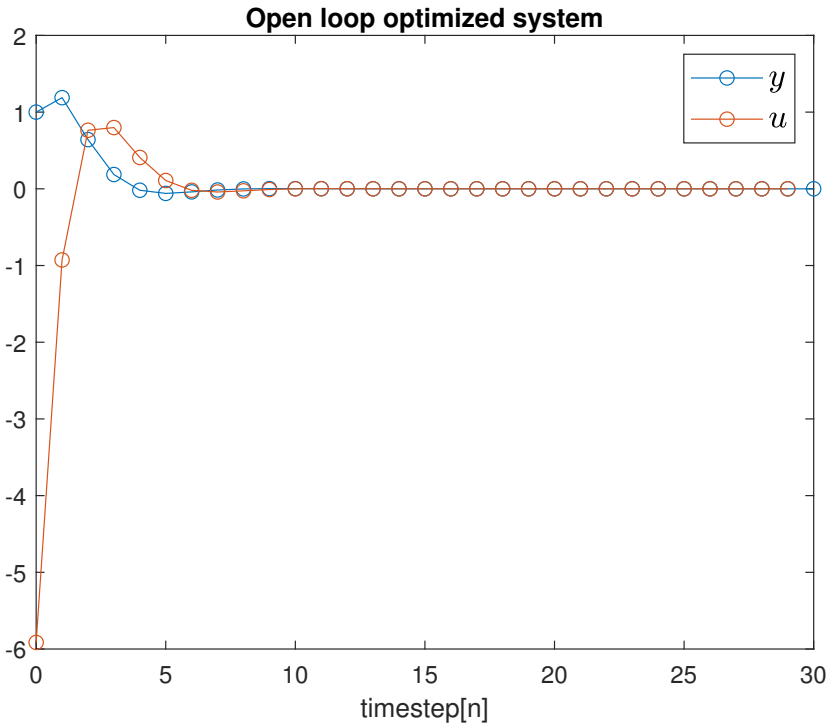

Figure 2: Simulation with small r

f)

With the inequality constraints, the plot is as shown in Figure 4.
Here, quadprog uses 5 iterations. This is due to the inequality constraints. As a result of this, quadprog must use the active set method, which takes more iterations to complete.

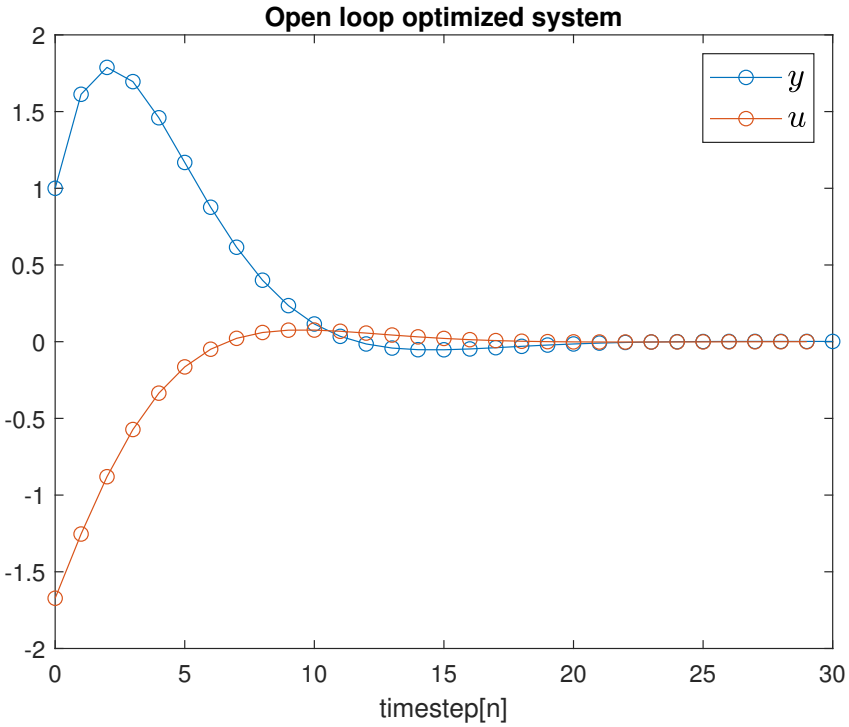

Figure 3: Simulation with large r

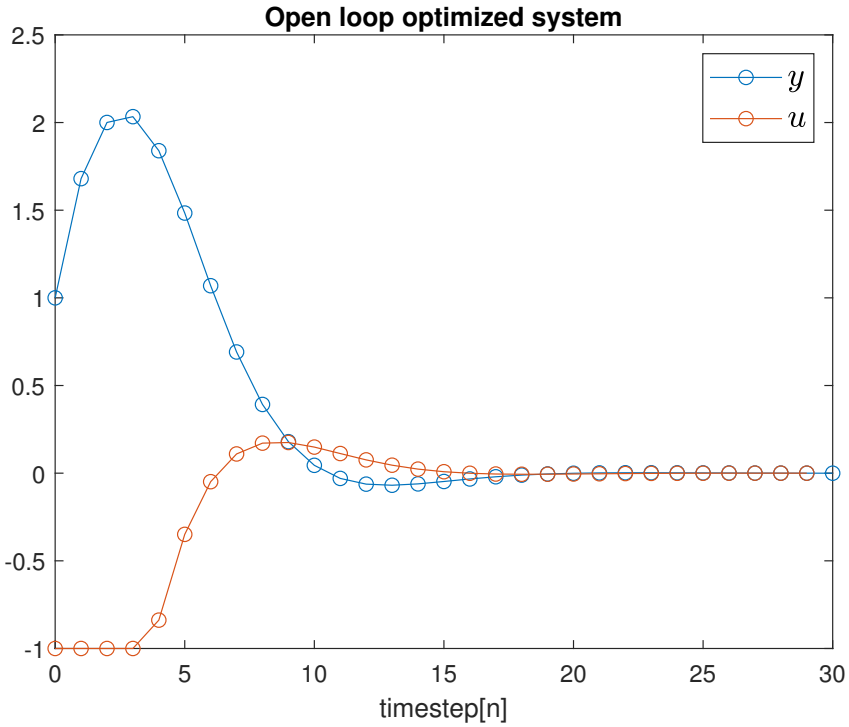

Figure 4: Simulation with inequality constraints

Problem 2

a)

A figure of a general MPC system is shown in Figure 5. The purpose of MPC is to be able to adjust the input such that the effects of modelling errors and disturbances are minimal. It will, from each timestep, calculate the optimal input for the next N timesteps to change it's current state to a desired state by solving a minimization problem. It also allows to weight your inputs and states to achieve desired behaviour/account for limitations.

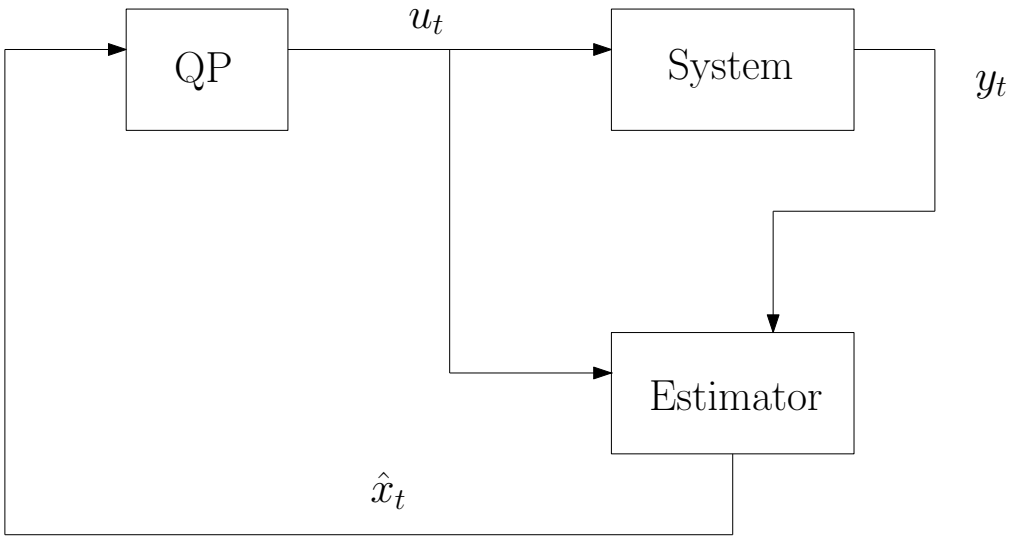

Figure 5: Figure of a general MPC system

b)

The plot is shown in Figure 6. Notice that the result does not differ much from 1f), which is because we are working with an assumed perfect model. Although, it is not exactly the same, since the MPC will calculate the input for the next for each iteration $N = 30$ timesteps, which will alter it's behaviour slightly. This is not visible in the figure.

c)

The MPC system is simulated with the imperfect system, as shown in Figure 7.
Notice how the system massively overshoots in the beginning, before slowly swinging toward the correct stationary value. Because the controller is built from the imperfect model of the system, the inputs, as expected, does not calculate an optimal trajectory for y. Also notice how the input is much larger than in Figure 6. This is because the output feedback is there to

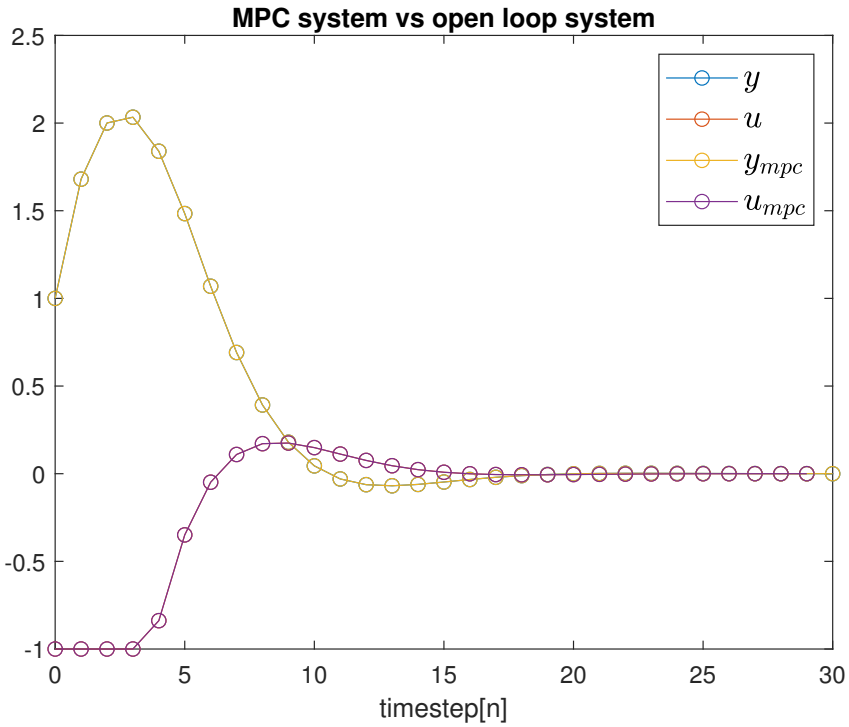

Figure 6: Comparison between 1f) and 2b)

help us correct the modelling errors, and in the end reach the desired reference (if simulated longer).

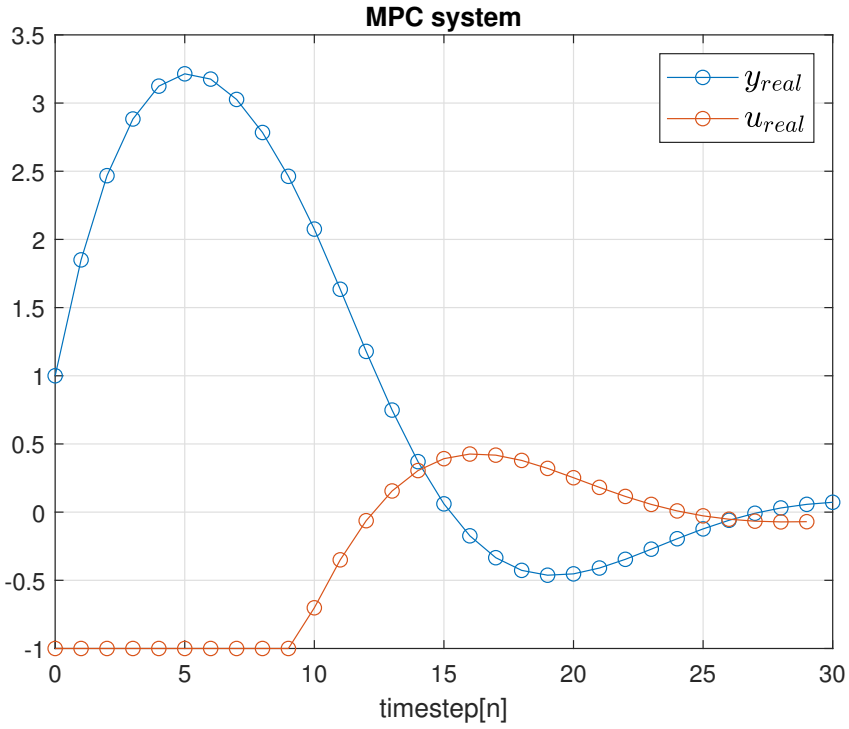

Figure 7: Simulated imperfect system