Optimal Control WS19/20 - **Homework exercise 2**
Value function iteration and MPC

Lucas Rath (3449668), Julius Hiller (3444375)

Problem 1 - Discrete-time Infinite-Horizon Optimal Control

A) The given problem can be described as a discrete-time infinite-horizon optimal control problem:

$$\begin{aligned}
\min \quad & \sum_{k=0}^{\infty} 0.9^k f_o(x_k, u_k) \\
s.t. \quad & x_{k+1} = f(x_k, u_k) \\
& u_k \in \mathbb{U}(x) = \{u_0, u_1, u_2\} \\
& x_0 = \xi_1,
\end{aligned} \tag{1}$$

where $f_o(x_k, u_k)$ and $f(x_k, u_k)$ are given in according to the provided transition costs and the graph.

Theorem 1. *Suppose a solution of (1) exists, then the value function V and the optimal feedback policy u solve the Bellman equation:*

$$V(x) = \min_{u \in \mathbb{U}(x)} \{f_o(x, u) + \alpha V(f(x, u))\}$$
$$u(x) = \arg\min_{u \in \mathbb{U}(x)} \{f_o(x, u) + \alpha V(f(x, u))\}$$

The value function iteration for this problem is then given by solving the following algorithm until convergence:

Algorithm 1. *Starting with $k = 0$ and some $V^0 = [V^0(\xi_1), \ldots, V^0(\xi_8)]$, do:*

$$V^{k+1}(x) = TV^k(x) = \min_{u \in \mathbb{U}(x)} \{f_o(x, u) + \alpha V(f(x, u))\}$$

for $x = \xi_1, \ldots, \xi_8$, until convergence, i.e., $\|V^{k+1}(x) - V^k(x)\| < \epsilon$, $\forall x$

Replacing the given transition dynamics f into the algorithm 1, we get the following iteration for each state ξ:

$$V^{k+1}(\xi_1) = \min \left\{ \begin{array}{c} f_o(\xi_1, u_{0/1}) + \alpha\, V^k(\xi_2)\ , \\ f_o(\xi_1, u_2) + \alpha\, V^k(\xi_3) \end{array} \right\} = \min \left\{ \begin{array}{c} 3 + 0.9\, V^k(\xi_2)\ , \\ 1 + 0.9\, V^k(\xi_3) \end{array} \right\}$$

$$V^{k+1}(\xi_1) = \min \left\{ \begin{array}{c} f_o(\xi_2, u_0) + \alpha\, V^k(\xi_7)\ , \\ f_o(\xi_2, u_1) + \alpha\, V^k(\xi_5)\ , \\ f_o(\xi_2, u_2) + \alpha\, V^k(\xi_4) \end{array} \right\} = \min \left\{ \begin{array}{c} 5 + 0.9\, V^k(\xi_7)\ , \\ 3 + 0.9\, V^k(\xi_5)\ , \\ 1 + 0.9\, V^k(\xi_4) \end{array} \right\}$$

$$V^{k+1}(\xi_3) = \min \left\{ \begin{array}{c} f_o(\xi_3, u_0) + \alpha\, V^k(\xi_4)\ , \\ f_o(\xi_3, u_1) + \alpha\, V^k(\xi_6)\ , \\ f_o(\xi_3, u_2) + \alpha\, V^k(\xi_5) \end{array} \right\} = \min \left\{ \begin{array}{c} 6 + 0.9\, V^k(\xi_4)\ , \\ 6 + 0.9\, V^k(\xi_6)\ , \\ 5 + 0.9\, V^k(\xi_5) \end{array} \right\}$$

$$V^{k+1}(\xi_4) = \min \left\{ \begin{array}{c} f_o(\xi_4, u_0) + \alpha\, V^k(\xi_7)\ , \\ f_o(\xi_4, u_1) + \alpha\, V^k(\xi_8)\ , \\ f_o(\xi_4, u_2) + \alpha\, V^k(\xi_6) \end{array} \right\} = \min \left\{ \begin{array}{c} 1 + 0.9\, V^k(\xi_7)\ , \\ 0 + 0.9\, V^k(\xi_8)\ , \\ 1 + 0.9\, V^k(\xi_6) \end{array} \right\}$$

$$V^{k+1}(\xi_5) = \min \left\{ \begin{array}{c} f_o(\xi_5, u_{0/1}) + \alpha\, V^k(\xi_4)\ , \\ f_o(\xi_5, u_2) + \alpha\, V^k(\xi_6) \end{array} \right\} = \min \left\{ \begin{array}{c} 3 + 0.9\, V^k(\xi_4)\ , \\ 2 + 0.9\, V^k(\xi_6) \end{array} \right\}$$

$$V^{k+1}(\xi_6) = \min \left\{ \begin{array}{c} f_o(\xi_6, u_0) + \alpha\, V^k(\xi_1)\ , \\ f_o(\xi_6, u_1) + \alpha\, V^k(\xi_7)\ , \\ f_o(\xi_6, u_2) + \alpha\, V^k(\xi_8) \end{array} \right\} = \min \left\{ \begin{array}{c} 2.5 + 0.9\, V^k(\xi_1)\ , \\ 2 + 0.9\, V^k(\xi_7)\ , \\ 4 + 0.9\, V^k(\xi_8) \end{array} \right\}$$

$$V^{k+1}(\xi_7) = \min \left\{ f_o(\xi_7, u_{0/1/2}) + \alpha\, V^k(\xi_8) \right\} = 1 + 0.9V^k(\xi_8)$$

$$V^{k+1}(\xi_8) = \min \left\{ f_o(\xi_8, u_{0/1/2}) + \alpha\, V^k(\xi_8) \right\} = 0.9V^k(\xi_8)$$

B) After implementing the value iteration algorithm on Matlab the value function and the corresponding optimal feedback converged to the following values:

Table 1: *Converged value function and optimal policy*

	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7	ξ_8
V(x)	3.9	1.0	6.0	0	3.0	2.9	1.0	0.0
u(x)	0	2	0	1	0	1	0	0

with the converged optimal policy $u(x)$ we are then able to calculate optimal input sequence when starting in the state ξ_1.

C)

Code 1: *Optimal input sequence*

```
State:1, Optimal input:0, Cost:3.0
State:2, Optimal input:2, Cost:1.0
State:4, Optimal input:1, Cost:0.0
State:8, Optimal input:0, Cost:0.0
```

Proposition 1. *Let V^* be the optimal value function and V^0 some initial guess such that:*

$$V^0(x) \leq TV^0(x) \quad \forall x \in \mathcal{X} \tag{2}$$

then $V^k \to V^*$ and $V^0(x) \leq V^*(x)$, $\forall x \in \mathcal{X}$

Proof. According to the following properties of the Bellman operator T:

(i) $TV^* = V^*$,

(ii) $\|TV^1 - TV^2\|_\infty \leq \alpha \|V^1 - V^2\|$

and making use of the value iteration definition $V^{k+1} = TV^k$ we can write:

$$\begin{aligned} \left\|V^{k+1} - V^*\right\|_\infty = \left\|TV^k - TV^*\right\|_\infty &\leq \alpha \left\|V^k - V^*\right\|_\infty \\ &\leq \dots \\ &\leq \alpha^{k+1} \left\|V^0 - V^*\right\|_\infty \end{aligned} \tag{3}$$

which implies that for $\alpha \in [0, 1)$ and $k \to \infty$, we have:

$$\begin{aligned} &\left\|V^{k+1} - V^*\right\|_\infty \leq \ \alpha^{k+1} \left\|V^0 - V^*\right\|_\infty \to 0 \\ \Rightarrow &\boxed{\lim_{k\to\infty} V^k = V^*}. \end{aligned} \tag{4}$$

Moreover, the monotonicity property:

(iii) $\forall x : V^1(x) \geq V^2(x) \ \Rightarrow \ TV^1(x) \geq TV^2(x)$

can be used with equation (2), yielding

$$\begin{aligned} TTV^0(x) \quad &\geq TV^0(x) \\ \Rightarrow V^2(x) \quad &\geq V^1(x) \quad \geq V^0(x) \end{aligned} \tag{5}$$

which can be used recursively to obtain:

$$\forall x, i: \ V^*(x) = T^\infty V(x) \geq T^i V(x). \tag{6}$$

For $i = 0$, we finally get $\boxed{V^*(x) \geq V^0(x), \forall x}$. □

D)

Proposition 2. *Let V be the value function, then V^* is also the solution to the problem:*

$$\begin{aligned} \max_{V(\xi_1),\dots,V(\xi_n)} \quad & \sum_{i=1}^{n} V(\xi_i) \\ s.t. \quad & V(\xi_i) \leq TV(\xi_i) \quad i = 1, \dots, n \end{aligned} \tag{7}$$

Proof. Using the proof from last item, we have that the constraint equation in (7) yields:

$$V^0(x) \leq V^1(x) \leq \cdots \leq T^{\infty}V(x) = V^*(x) \quad \forall x \in \mathcal{X}, \tag{8}$$

which means that, in this case, the value function iteration sequence $V^k(x)$ is monotonically increasing and bounded by the optimal value function for all states. Hence, finding the sum of the largest possible value function at each state, when the mentioned constraint is satisfied, must result in the optimal value function V^*. □

E)

Proposition 3. *Let V be the value function, then the optimization problem (7) can be transformed into the linear program:*

$$\begin{aligned} \max_{V(\xi_1),\dots,V(\xi_n)} \quad & \sum_{i=1}^{n} V(\xi_i) \\ s.t. \quad & V(\xi_i) \leq T_u V(\xi_i) \quad \forall u \in \mathbb{U}(\xi_i),\ i = 1, \dots, n \end{aligned} \tag{9}$$

where $T_u V(x) = f_0(x, u) + \alpha V(f(x, u))$

Proof. In (9), the maximization of $V(\xi_i)$ is upper bounded by the minimum value that $T_u V(\xi_i)$ can assume w.r.t. $u \in \mathbb{U}(\xi_i)$. Therefore, we verify that:

$$\begin{aligned} V(\xi_i) \ &\leq \ \{T_u V(\xi_i) : u \in \mathbb{U}(\xi_i)\} \\ \Rightarrow V(\xi_i) \ &\leq \ \min_{u \in \mathbb{U}(\xi_i)} \{T_u V(\xi_i)\} = TV(\xi_i) \end{aligned} \tag{10}$$

□

Remark. *The optimization problem (9) can be easily rewritten as a Linear Program (LP) of the form:*

$$\begin{aligned} V^* = \ & \arg\min_{V} \quad c^{\mathsf{T}} V \\ & \quad s.t. \quad AV \leq b \end{aligned} \tag{11}$$

Given the dimensions of the state space $\mathcal{X} \in \mathbb{R}^n$ and input space $\mathbb{U} \in \mathbb{R}^m$, the optimization problem will habe $n = 8$ optimization variables and $n \times m = 8 \times 3$ constraints.

The matrices A, b and c of 11 are given bellow:

$$A = \begin{bmatrix} 1.0 & -0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0 & -0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0 & 0 & -0.9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & -0.9 & 0 \\ 0 & 1.0 & 0 & 0 & -0.9 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & -0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & -0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & -0.9 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & -0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & -0.9 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & -0.9 \\ 0 & -0.9 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.9 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.9 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & -0.9 & 0 & 0 \\ -0.9 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & -0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & -0.9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & -0.9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & -0.9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & -0.9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \quad b = \begin{bmatrix} 3.0 \\ 3.0 \\ 1.0 \\ 5.0 \\ 3.0 \\ 1.0 \\ 6.0 \\ 6.0 \\ 5.0 \\ 1.0 \\ 0 \\ 1.0 \\ 3.0 \\ 3.0 \\ 2.0 \\ 2.5 \\ 2.0 \\ 4.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad c = \begin{bmatrix} -1.0 \\ -1.0 \\ -1.0 \\ -1.0 \\ -1.0 \\ -1.0 \\ -1.0 \\ -1.0 \end{bmatrix} \tag{12}$$

F) Solving the LP in (9) results in the value function shown in table 2. The obtained results are exactly equal to the value iteration approach, displayed in table 1.

After obtaining the optimal value function, the optimal policy can be obtained from:

$$u(\xi_i) = \arg\min_{u \in \mathbb{U}(\xi_i)} TuV(\xi_i), \quad i = 1, \cdots, n \tag{13}$$

Again, the calculated optimal policy, shown in table 2 matches exactly the one obtained with the value iteration approach.

Table 2: *Optimal value function using Linear Programming approach*

	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7	ξ_8
V(x)	3.9	1.0	6.0	0	3.0	2.9	1.0	0.0
u(x)	0	2	0	1	0	1	0	0

Problem 2 - Model Predictive Control

Consider the discrete-time system:

$$x_{k+1} = Ax_k + Bu_k, \tag{14}$$

where

$$A = \begin{bmatrix} 1 & 2 \\ -1 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 0.3 \\ -0.25 \end{bmatrix} \tag{15}$$

A) The equilibrium of the unforced system (14) is given by:

$$\begin{aligned} x_{ss} &= Ax_{ss} \\ \Rightarrow (I-A)x_{ss} &= 0 \end{aligned} \tag{16}$$

whose unique solution is:

$$x_{ss} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{17}$$

since $Null(I - A) = \emptyset$

In addition, the system is not stable, since

$$|eig(A)| = \{2.2361, 2.2361\} > 1 \tag{18}$$

B) For the given dynamics and specifications, the discrete-time system can be formulated as an MPC problem as follows:

$$\begin{aligned} \min_{u_{0:N-1}} \quad & \sum_{k=0}^{N-1} \underbrace{\|x_k\|_Q^2 + \|u_k\|_R^2}_{f_o(x_k,u_k)} + \underbrace{\|x_N\|_P^2}_{\phi(x_N)} \\ s.t. \quad & x_{k+1} = Ax_k + Bu_k \\ & u_k \in \mathbb{U} = \{u \in \mathbb{R} : u \leq 1, \; -u \leq 1\} \\ & x_0 = \bar{x} \\ & x_N \in \mathcal{X}_f = \{x \in \mathbb{R} : x^\mathsf{T} P x \leq c\} \end{aligned} \tag{19}$$

where $N = 2$, $Q = I_2$, $R = 1.5$, $c = \frac{\lambda_{min}(P)}{|K|^2}$, $K = [-0.91, 2.85]$ and $P = \begin{bmatrix} 10.65 & 16.02 \\ 16.02 & 67.01 \end{bmatrix}$

C) In order to show that the given MPC scheme of 19 is stable we have to show that there exists a feasible input u such that:

1. The control u is feasible for all $x \in \mathcal{X}_f$:

Proof. First, we use the property of quadratic forms: $\|x\|^2_{\lambda_{min}(P)} \le \|x\|^2_P \le \|x\|^2_{\lambda_{max}(P)}$ to lower bound the terminal set $\mathcal{X}_f = \{x \in \mathbb{R} : x^\mathsf{T} P x \le c\}$:

$$\begin{aligned} x^\mathsf{T}\lambda_{min}(P)x &\le x^\mathsf{T}Px \le c = \frac{\lambda_{min}(P)}{\|K\|^2} \\ \Rightarrow \quad x^\mathsf{T}\lambda_{min}(P)x &\le \frac{\lambda_{min}(P)}{\|K\|^2} \\ \Rightarrow \quad x^\mathsf{T}xK^\mathsf{T}K &\le 1 \end{aligned} \tag{20}$$

Next, we use the norm of product inequality rule $\|A.B\| \le \|A\|\|B\|$ in (20):

$$\|Kx\|^2 \le \|x\|^2\|K\|^2 \le 1 \tag{21}$$

In the equation above we verify that, if there is a local controller $u_{MPC}(x) = -Kx$, then the input can be bounded by:

$$\begin{aligned} \|Kx\|^2 = \|u\|^2 &\le 1 \\ \Rightarrow &\boxed{\|u\| \le 1} \end{aligned} \tag{22}$$

Equation (22) proves that the controller input is feasible whenever $x \in \mathcal{X}_f$ as long as the MPC controller is based on a local feedback, i.e.:

$$u_{MPC}(x) = -Kx \in \mathbb{U}, \quad \forall x \in \mathcal{X}_f. \tag{23}$$

□

2. For all $x \in \mathcal{X}_f$ it is $\phi(x_{k+1}) - \phi(x_k) \le -f_0(x_k, u_k)$

   Proof. Lets start replacing the cost functions into the inequality:

$$\begin{aligned} \phi(x_{k+1}) - \phi(x_k) &\le -f_0(x_k, u_k) \\ x^\mathsf{T}_{k+1}Px_{k+1} - x^\mathsf{T}_kPx_k &\le -x^\mathsf{T}_kQx_k - u^\mathsf{T}_kRu_k \\ (Ax_k + Bu_k)^\mathsf{T}P(Ax_k + Bu_k) - x^\mathsf{T}_kPx_k &\le -x^\mathsf{T}_kQx_k - u^\mathsf{T}_kRu_k \end{aligned} \tag{24}$$

   Next, considering again that if $x \in \mathcal{X}_f$ then the MPC controller is based on the local feedback $u_k = u_{MPC}(x_k) = -Kx_k$, we have:

$$x^\mathsf{T}_k\left((A - BK)^\mathsf{T}P(A - BK) - P + Q + K^\mathsf{T}RK\right)x_k \le 0 \tag{25}$$

$$\Rightarrow \quad \underbrace{(A - BK)^\mathsf{T}P(A - BK) - P + Q + K^\mathsf{T}RK}_{M_{dare}} \preccurlyeq 0 \tag{26}$$

   The left hand side of equation (26) is the well known discrete-time algebraic Ricatti equation, which is equal zero and therefore seminegative-definite, if P and K are chosen accordingly.

   In the case of the proposed MPC scheme, the given K and P matrices do not satisfy the *dare* equation. However, the eigenvalues of M_{dare} in (26) are $\{-0.0966, -0.1086\}$, which are indeed negative, satisfying (26). □

3. The terminal region $\mathcal{X}_f$ is invariant

Proof. By definition, a set $\mathcal{S}$ is control-invariant if:

$$\forall x \in \mathcal{S} \Rightarrow \exists u \in \mathbb{U} : x_{k+1} = f(x, u) \in \mathcal{S} \tag{27}$$

From *2.*, we proved that for all $x \in \mathcal{X}_f$ it is $\phi(x_{k+1}) - \phi(x_k) \leq -f_0(x_k, u_k)$. Since $f_o(x_k, u_k) \geq 0, \forall x_k, u_k$ then:

$$\begin{aligned} & \phi(x_{k+1}) - \phi(x_k) \leq -f_0(x_k, u_k) \leq 0 \\ \Rightarrow \quad & \phi(x_{k+1}) \leq \phi(x_k) \\ \Rightarrow \quad & x_{k+1}^{\mathsf{T}} P\, x_{k+1} \leq x_k^{\mathsf{T}} P\, x_k \end{aligned} \tag{28}$$

Moreover, $x_k \in \mathcal{X}_f = \{x \in \mathbb{R} : x^{\mathsf{T}} P x \leq c\}$ implies that:

$$x_{k+1}^{\mathsf{T}} P\, x_{k+1} \leq x_k^{\mathsf{T}} P\, x_k \leq c \tag{29}$$

and therefore:

$$x_k \in X_f \;\Rightarrow\; \exists u \in \mathbb{U} : x_{k+1} \in \mathcal{X}_f \tag{30}$$

□

D) The task now is to define suitable optimization variable $\boldsymbol{z}$ to write (19) as a quadratically constrained quadratic program.

Lets start with the dynamics constraints. Repeated use of $x_{k+1} = Ax_k + Bu_k$ with initial condition $x(0) = \bar{x}$ gives us:

$$\underbrace{\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \\ x(N) \end{bmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^N \end{bmatrix}}_{\Omega} \bar{x} + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ B & 0 & \cdots & 0 & 0 \\ AB & B & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ A^{N-2}B_D & A^{N-3}B & \cdots & B & 0 \\ A^{N-1}B & A^{N-2}B & \cdots & AB & B \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-2) \\ u(N-1) \end{bmatrix}}_{\boldsymbol{u}} \tag{31}$$

which we can write in a more compact notation:

$$\boldsymbol{x} = \Omega\, \bar{x} + \Gamma \boldsymbol{u} \tag{32}$$

Defining the a new variable vector as $\boldsymbol{z} = [\boldsymbol{x}, \boldsymbol{u}]^T = [x(0), ..., x(N), u(0), ..., u(N-1)]^T$, we get the redefined constraint equations as:

$$\underbrace{[I \quad -\Gamma]}_{A_{eq}}\, \boldsymbol{z} \;= \underbrace{\Omega\, \bar{x}}_{b_{eq}} \tag{33}$$

For $N = 2$:

$$\underbrace{\begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & -B & 0 \\ 0 & 0 & I & -AB & -B \end{bmatrix}}_{A_{eq}} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ u(0) \\ u(1) \end{bmatrix} = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \end{bmatrix}}_{b_{eq}} \bar{x} \tag{34}$$

Now lets redefine the cost function in terms of the new variable $\boldsymbol{z}$

$$
\begin{aligned}
J &= \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k + x_N^\top P x_N && (35)\\
&= (\boldsymbol{x}^T \bar{Q} \boldsymbol{x} + \boldsymbol{u}^T \bar{R} \boldsymbol{u}) && (36)\\
&= \boldsymbol{z}^T \underbrace{\begin{bmatrix} \bar{Q} & 0 \\ 0 & \bar{R} \end{bmatrix}}_{H \succcurlyeq 0} \boldsymbol{z} && (37)
\end{aligned}
$$

with $\bar{Q} = diag(Q, ..., Q, P)$ and $\bar{R} = diag(R, R, ..., R)$ block diagonal matrices with N+1 and N blocks respectively.

Next, we specify the inequality constraints concerning the input constraints $u_k \in \mathbb{U} = \{u \in \mathbb{R} : u \leq 1,\ -u \leq 1\}$:

$$
\underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{A_{ineq}} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ u(0) \\ u(1) \end{bmatrix} \leq \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}_{b_{ineq}} \tag{38}
$$

Lastly, we define the final state constraint $\quad x_N \in \mathcal{X}_f = \{x \in \mathbb{R} : x^\top P x \leq c\}$:

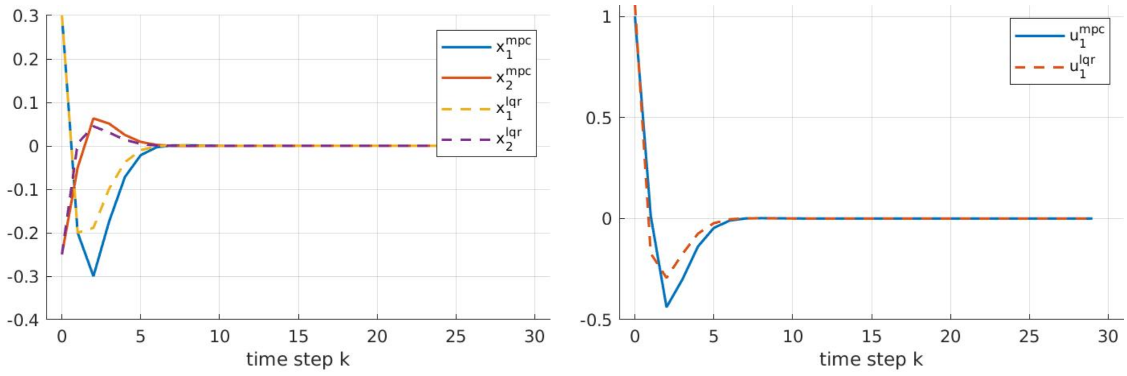
$$
\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ u(0) \\ u(1) \end{bmatrix}^\top \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{T \succcurlyeq 0} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ u(0) \\ u(1) \end{bmatrix} \leq \underbrace{\frac{\lambda_{min}(P)}{|K|^2}}_{c=d} \tag{39}
$$

The new discretized optimal control problem can be then rewritten as:

$$
\begin{aligned}
\min_{\boldsymbol{z}} \quad & z^\top H z \\
s.t. \quad & A_{eq} z = B_{eq} \\
& A_{ineq} z \leq B_{ineq} \\
& z^\top T z \leq d
\end{aligned} \tag{40}
$$

We then observe that this problem is convex because: the cost function is quadratic and H is positive definite; the constraints are either linear or quadratic with semipositive definite matrix T.

E) The figure bellow shows the evolution of the states and the first optimal MPC control input at each time step for the initial condition $x_0 = [0.3, -0.25]^\top$



(a) optimal state squence x_k *(b) optimal input sequence u_k*

Figure 1: *Plots of the resulting evolution of the state as well as of the determined input*

As can be verified, the input constraints are always satisfied.

Another look into a x-y plot allows us to visualize the terminal set constraint $\mathcal{X}_f$ and the progress of the states into that region.

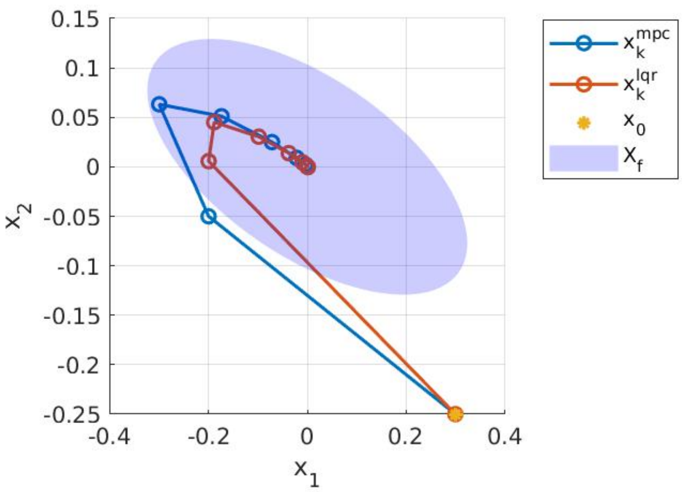


Figure 2: *State sequence for the MPC and LQR controllers and terminal constraint set $\mathcal{X}_f$*

As expected, the system entered in the terminal constraint zone in N=2 steps.

F) If we choose another initial condition $x_0 = [1, -0.9]^\mathsf{T}$ that is too far away from the terminal set $\mathcal{X}_f$, the problem become infeasible and the solver can not find any solution for the optimization problem.

Formally, this happened because $x_o \notin \mathcal{X}_N$, where $\mathcal{X}_N$ is the feasibility set of initial states, defined by:

$$\mathcal{X}_N = \{x \in \mathbb{X} : \mathbb{U}_N(x_o) \neq \emptyset\} \tag{41}$$

and $\mathbb{U}_N$ is the set of all control sequences driving the initial state x_0, to the terminal set $\mathcal{X}_f$ in N steps, while satisfying state and input constraints al all times.

Usually, the feasibility set is chosen as the N-step controllable set $\mathcal{X}_N = \mathcal{K}_N(\mathcal{X}_f)$, such that the optimization problem will be feasible whenever $x_0 \in \mathcal{X}_N$.

G) As the last experiment, we calculate the LQR gain corresponding to the following problem:

$$\begin{aligned} \min \quad & \sum_{k=0}^{\infty} x_k^\top Q x_k + u_k^\top R u_k \\ \text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k \end{aligned} \tag{42}$$

whose LQR input $u_{lqr}(x_k) = -K_{lqr}x_k$, calculated in Matlab with the function `dlqr` has gain:

$$K_{lqr} = [-0.7660, 3.3038] \tag{43}$$

The simulation results, i.e. state and input sequence for the LQR controller can be seen in Figures 1 and 2. As expected, the states are driven faster to the origin with the LQR controller than with MPC. On the other hand, the input constraints are violated in the first time step ($u_1^{lqr} = 1.06$. This happens because LQR is an optimal controller, but is not able to consider any kind of state or input constraint.