$$A = \begin{bmatrix} 1 & T \\ (-k/m)T & (1 - \frac{c}{m}T) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ T/m \end{bmatrix}, \quad C = [1 \; 0]$$

$m = 1$, $c = 0.4$, $k = 1$, $T = 0.1$

$$A = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 0.96 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}, \quad C = [1 \; 0]$$

a) $Y = Fx_k + HU$ $\qquad$ F is 5×2 Matrix

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{bmatrix} \Rightarrow F = \begin{bmatrix} 1 & 0.1 \\ 0.904 & 0.192 \\ 0.818 & 0.271 \\ 0.742 & 0.339 \\ 0.674 & 0.398 \end{bmatrix}$$

→ (used python for calculations)

$$H = \begin{bmatrix} CB & 0 & 0 & 0 & 0 \\ CAB & CB & 0 & 0 & 0 \\ CA^2B & CAB & CB & 0 & 0 \\ CA^3B & CA^2B & CAB & CB & 0 \\ CA^4B & CA^3B & CA^2B & CAB & CB \end{bmatrix}$$

$CB = 0$
$CAB = 0.1$
$CA^2B = 0.096$
$CA^3B = 0.091$
$CA^4B = 0.087$

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 \\ 0.096 & 0.1 & 0 & 0 & 0 \\ 0.091 & 0.096 & 0.1 & 0 & 0 \\ 0.087 & 0.091 & 0.096 & 0.1 & 0 \end{bmatrix}$$

b) $N_p = 5$, $N_c = 3$. We assume the last two columns of H is the same ($N_c = 3$).

$$H = \begin{bmatrix} CB & 0 & 0 \\ CAB & CB & 0 \\ CA^2B & CAB & CB \\ CA^3B & CA^2B & CB + CAB \\ CA^4B & CA^3B & CB + CAB + CA^2B \end{bmatrix}$$

the system works less free with the updated version of H but also with a less power of computation

$$H_{ij} = \begin{cases} CA^{i-j}B & j \leq \min(i, N_c - 1) \\ \sum_{l=0}^{i-N_c} CA^l B & j = N_c,\ i \geq N_c \\ 0 & j > i \text{ or } (i < N_c \text{ and } j = N_c \end{cases}$$

less decision variables while maintaining smoothness in the pred. control seq.

c) $U = S\Delta U + U_{k-1}$ when $N_p = 5$ and $N_c = 3$

S is a 5×3 right-side-cut identity mat.

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ & & \end{bmatrix}$$

] control inputs stay the same.

$$\Delta U_k = U_k - U_{k-1}, \quad U = \begin{bmatrix} U_k \\ U_{k+1} \\ U_{k+2} \end{bmatrix} = S\Delta U + \begin{bmatrix} U_{k-1} \\ U_{k-1} \\ U_{k-1} \end{bmatrix}$$

$$\Delta U = [\Delta U_k, \Delta U_{k+1}, \Delta U_{k+2}]^T$$

d) $Y = Fx_k + HU \longrightarrow U = S\Delta U + U_{k-1} \longrightarrow Y = Fx_k + HU_{k-1} + HS\Delta U$

$Y = b_k + \Phi \Delta U$, $b_k = Fx_k + HU_{k-1}$, $\Phi = HS$

$$\Phi = HS = \begin{bmatrix} CB & 0 & 0 \\ CAB & CB & 0 \\ CA^2B & CAB & CB \\ CA^3B & CA^2B & CB + CAB \\ CA^4B & CA^3B & CB + CAB + CA^2B \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 0 & 0 & 0 \\ 0.1 & 0 & 0 \\ 0.0296 & 0.01 & 0 \\ 0.058 & 0.029 & 0.01 \\ 0.095 & 0.058 & 0.029 \end{bmatrix}$$

(Computations that takes time or are cumbersome are done with python, matlab and etc.)

$$Y = b_k + \Phi\Delta U = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ y_{k+4} \\ y_{k+5} \end{bmatrix} = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{bmatrix} x_k + HU_{k-1} + \Phi \begin{bmatrix} \Delta U_k \\ \Delta U_{k+1} \\ \Delta U_{k+2} \end{bmatrix}$$

```
close all hidden; clear; clc;
rng(418);                                  % seed for reproducibility,
```

1.e Unconstrained MPC simulation

```
clear; clc; close all;

% System parameters
m = 1; c = 0.4; k = 1; T = 0.1;
A = [1 T; -k/m*T 1 - (c/m)*T];
B = [0; T/m];
C = [1 0];

% Horizons and weights
Np = 200; Nc = 100;
Qbar = eye(Np);
Rbar = 0.1 * eye(Nc);

% F and H matrices
F = zeros(Np, size(A,1));
Ap = A;
for i = 1:Np
    F(i,:) = C * Ap;
    Ap = Ap * A;
end

H = zeros(Np, Nc);
for i = 1:Np
    for j = 1:Nc
        if j <= i
            H(i,j) = C * (A^(i-j)) * B;
        else
            H(i,j) = 0;
        end
    end
end

% S and Phi matrices
S = tril(ones(Nc));
Phi = H * S;

% Step pattern for ΔU
DeltaU = 0.3 * ones(Nc, 1);
% step change, saturating at ±0.3

% Initial conditions
xk = [0; 0];
u_prev = 0;
u_b = u_prev * ones(Nc,1);
```

```
bk = F*xk + H*u_b;

% Predicted output (Y = b_k + ΦΔU)
Ypred = bk + Phi * DeltaU;

% Plot predicted trajectory
figure('Color','w');
stairs(1:Np, Ypred, 'b-o','LineWidth',1.2); hold on; grid on;
xlabel('Prediction step (k+i)');
ylabel('Predicted output y_{k+i}');
title('Predicted output trajectory under step ΔU');
legend('Prediction','Location','best');
```