CHALMERS

UNIVERSITY OF TECHNOLOGY

SSY281 - MODEL PREDICTIVE CONTROL

NIKOLCE MURGOVSKI

Division of Systems and Control
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden

2021-01-21

Lecture 2: Review and preliminaries

Goals for today:

- To master different discrete-time state space models used for MPC.
- To understand conditions for setpoint tracking and disturbance rejection.
- To refresh basic concepts used to solve systems of linear equations.
- To refresh and learn a new way to test for controllability and observability.
- To know how to use quadratic forms for stability investigations of linear systems.
- To refresh how to handle LTI systems in Matlab.

Lecture 2: Review and preliminaries

Goals for today:

- To master different discrete-time state space models used for MPC.
- To understand conditions for setpoint tracking and disturbance rejection.
- To refresh basic concepts used to solve systems of linear equations.
- To refresh and learn a new way to test for controllability and observability.
- To know how to use quadratic forms for stability investigations of linear systems.
- To refresh how to handle LTI systems in Matlab.

Learning objectives:

- Understand and explain the basic principles of model predictive control, its pros and cons, and the challenges met in implementation and applications.
- Use software tools for analysis and synthesis of MPC controllers.

Continuous State space model

We will study continues state space models of the form

$$
\begin{aligned}
\dot{x}(t) &= A_c x(t) + B_c u(t) \\
y(t) &= C_y x(t) \\
z(t) &= C_z x(t)
\end{aligned}
\tag{4}
$$

where

$x = (x_1, \ldots, x_n)$ is the state vector

$u = (u_1, \ldots, u_m)$ is the control input vector

$y = (y_1, \ldots, y_{p_y})$ is the vector of **measured outputs**

$z = (z_1, \ldots, z_{p_z})$ is the vector of **controlled outputs**.

Continuous State space model

We will study continues state space models of the form

$$\begin{aligned}\dot{x}(t) &= A_c x(t) + B_c u(t)\\ y(t) &= C_y x(t)\\ z(t) &= C_z x(t)\end{aligned} \tag{4}$$

where

$x = (x_1, \dots, x_n)$ is the state vector

$u = (u_1, \dots, u_m)$ is the control input vector

$y = (y_1, \dots, y_{p_y})$ is the vector of **measured outputs**

$z = (z_1, \dots, z_{p_z})$ is the vector of **controlled outputs**.

Often, $z = y = (y_1, \dots, y_p)$, and then we simply write $y(t) = Cx(t)$.

Discrete state space model

The solution of (4) with initial condition $x(t_0)$ is

$$x(t) = e^{A_c(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A_c(t-s)}B_c u(s)ds. \tag{5}$$

Discrete state space model

The solution of (4) with initial condition $x(t_0)$ is

$$x(t) = e^{A_c(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A_c(t-s)}B_c u(s)ds. \tag{5}$$

Assume that the control signal u is piecewise constant (h is the sampling interval),

$$u(t) = u(kh), \quad kh \leq t < (k+1)h.$$

Discrete state space model

The solution of (4) with initial condition $x(t_0)$ is

$$x(t) = e^{A_c(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A_c(t-s)}B_c u(s)ds. \tag{5}$$

Assume that the control signal u is piecewise constant (h is the sampling interval),

$$u(t) = u(kh), \quad kh \leq t < (k+1)h.$$

By using this in (5) with $t = (k+1)h$ and $t_0 = kh$, we get the discrete time state equation

$$x(k+1) = e^{A_c h}x(k) + \left(\int_0^h e^{A_c s}B_c\, ds\right)u(k) = Ax(k) + Bu(k),$$

where, for simplicity of notation, h has been omitted from the time argument.

Discrete state space model

The solution of (4) with initial condition $x(t_0)$ is

$$x(t) = e^{A_c(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A_c(t-s)}B_c u(s)ds. \tag{5}$$

Assume that the control signal u is piecewise constant (h is the sampling interval),

$$u(t) = u(kh), \quad kh \leq t < (k+1)h.$$

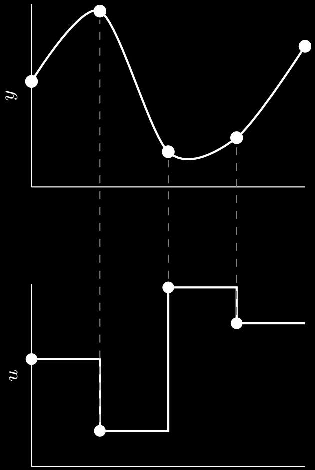
By using this in (5) with $t = (k+1)h$ and $t_0 = kh$, we get the discrete time state equation

$$x(k+1) = e^{A_c h}x(k) + \left(\int_0^h e^{A_c s}B_c\, ds\right)u(k) = Ax(k) + Bu(k),$$

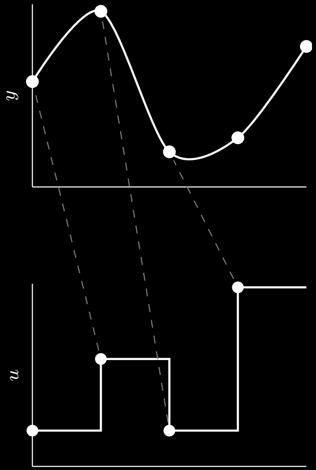
where, for simplicity of notation, h has been omitted from the time argument. The output equation of (4) is the same (but with t replaced by k). A compact version of the **discrete state-space model** in the case $z = y$ is

$$\begin{aligned} x^+ &= Ax + Bu \\ y &= Cx. \end{aligned}$$

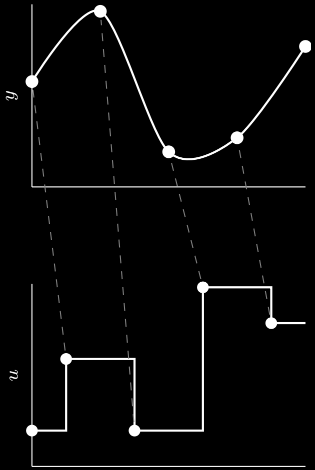
Sampling and computational delay



(a) Instantaneous computation.



(b) Full sampling interval delay.



(c) Fractional delay.

Figure 5: Control action and computational delay.

Computational delay

Allowing for a computational delay τ, the control signal is still piecewise constant but now given by

$$u(t) = \begin{cases} u(k-1), & kh \leq t < kh + \tau \\ u(k), & kh + \tau \leq t < (k+1)h. \end{cases}$$

Computational delay

Allowing for a computational delay τ, the control signal is still piecewise constant but now given by

$$u(t) = \begin{cases} u(k-1), & kh \leq t < kh+\tau \\ u(k), & kh+\tau \leq t < (k+1)h. \end{cases}$$

The solution (5) is now obtained in two steps (corresponding to the two different input signal levels) as

$$x(k+1) = e^{A_c(h-\tau)}x(kh+\tau) + \int_0^{h-\tau} e^{A_c s}B_c\,ds \cdot u(k)$$

Computational delay

Allowing for a computational delay τ, the control signal is still piecewise constant but now given by

$$u(t) = \begin{cases} u(k-1), & kh \leq t < kh + \tau \\ u(k), & kh + \tau \leq t < (k+1)h. \end{cases}$$

The solution (5) is now obtained in two steps (corresponding to the two different input signal levels) as

$$\begin{aligned} x(k+1) &= e^{A_c(h-\tau)} x(kh+\tau) + \int_0^{h-\tau} e^{A_c s} B_c \, ds \cdot u(k) \\ &= e^{A_c(h-\tau)} \left(e^{A_c \tau} x(kh) + \int_0^{\tau} e^{A_c s} B_c \, ds \cdot u(k-1) \right) + \int_0^{h-\tau} e^{A_c s} B_c \, ds \cdot u(k) \end{aligned}$$

.

Computational delay

Allowing for a computational delay τ, the control signal is still piecewise constant but now given by

$$u(t) = \begin{cases} u(k-1), & kh \leq t < kh + \tau \\ u(k), & kh + \tau \leq t < (k+1)h. \end{cases}$$

The solution (5) is now obtained in two steps (corresponding to the two different input signal levels) as

$$\begin{aligned} x(k+1) &= e^{A_c(h-\tau)} x(kh+\tau) + \int_0^{h-\tau} e^{A_c s} B_c \, ds \cdot u(k) \\ &= e^{A_c(h-\tau)} \left(e^{A_c \tau} x(kh) + \int_0^{\tau} e^{A_c s} B_c \, ds \cdot u(k-1) \right) + \int_0^{h-\tau} e^{A_c s} B_c \, ds \cdot u(k) \\ &= e^{A_c h} x(kh) + \underbrace{e^{A_c(h-\tau)} \int_0^{\tau} e^{A_c s} B_c \, ds}_{B_1} \cdot u(k-1) + \underbrace{\int_0^{h-\tau} e^{A_c s} B_c \, ds}_{B_2} \cdot u(k) \end{aligned}$$

.

Computational delay

Allowing for a computational delay τ, the control signal is still piecewise constant but now given by

$$u(t) = \begin{cases} u(k-1), & kh \leq t < kh + \tau \\ u(k), & kh + \tau \leq t < (k+1)h. \end{cases}$$

The solution (5) is now obtained in two steps (corresponding to the two different input signal levels) as

$$\begin{aligned} x(k+1) &= e^{A_c(h-\tau)} x(kh+\tau) + \int_0^{h-\tau} e^{A_c s} B_c \, ds \cdot u(k) \\ &= e^{A_c(h-\tau)} \left(e^{A_c \tau} x(kh) + \int_0^{\tau} e^{A_c s} B_c \, ds \cdot u(k-1) \right) + \int_0^{h-\tau} e^{A_c s} B_c \, ds \cdot u(k) \\ &= e^{A_c h} x(kh) + \underbrace{e^{A_c(h-\tau)} \int_0^{\tau} e^{A_c s} B_c \, ds}_{B_1} \cdot u(k-1) + \underbrace{\int_0^{h-\tau} e^{A_c s} B_c \, ds}_{B_2} \cdot u(k) \\ &= A x(k) + B_1 u(k-1) + B_2 u(k). \end{aligned}$$

This can be put in a standard form by introducing the augmented state vector

$$\xi(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}.$$

The new state space model becomes

$$\xi(k+1) = \begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} A & B_1 \\ 0 & 0 \end{bmatrix} \xi(k) + \begin{bmatrix} B_2 \\ I \end{bmatrix} u(k).$$

State space model with incremental control

$$
\begin{aligned}
x^+ &= Ax + Bu \\
y &= Cx.
\end{aligned}
$$

State space model with incremental control

$$
\begin{aligned}
x^+ &= Ax + Bu \\
y &= Cx.
\end{aligned}
$$

Introduce the incremental control (or *control move*)

$$\Delta u(k) = u(k) - u(k-1)$$

and the augmented state vector

$$\xi(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}.$$

State space model with incremental control

$$
\begin{aligned}
x^+ &= Ax + Bu \\
y &= Cx.
\end{aligned}
$$

Introduce the incremental control (or *control move*)

$$\Delta u(k) = u(k) - u(k-1)$$

and the augmented state vector

$$\xi(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}.$$

This leads to the new model

$$
\begin{aligned}
\xi^+ &= \mathcal{A}\xi + \mathcal{B}\Delta u \\
y &= \mathcal{C}\xi
\end{aligned}
$$

with

$$\mathcal{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \quad \mathcal{B} = \begin{bmatrix} B \\ I \end{bmatrix} \quad \mathcal{C} = \begin{bmatrix} C & 0 \end{bmatrix}.$$

Setpoint tracking

Consider the system

$$x^{+} = Ax + Bu \tag{6}$$

$$y = Cx. \tag{7}$$

Setpoint tracking

Consider the system

$$x^+ = Ax + Bu \tag{6}$$

$$y = Cx. \tag{7}$$

A steady state (x_s, u_s) of the system satisfies the equation

$$\begin{bmatrix} I - A & -B \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = 0.$$

Setpoint tracking

Consider the system

$$x^+ = Ax + Bu \tag{6}$$

$$y = Cx. \tag{7}$$

A steady state (x_s, u_s) of the system satisfies the equation

$$\begin{bmatrix} I - A & -B \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = 0.$$

The task to bring the system output y to a desired, constant setpoint y_{sp} is termed **setpoint tracking**. At steady state, this requires $Cx_s = y_{sp}$ and the condition for setpoint tracking becomes

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_{sp} \end{bmatrix}. \tag{8}$$

This is a system of $n + p$ equations with $n + m$ unknowns.

Regulation problem for deviation variables

Assume that equation (8) holds. Then the deviation variables

$$
\begin{aligned}
\delta x(k) &= x(k) - x_s \\
\delta u(k) &= u(k) - u_s
\end{aligned}
$$

satisfy the following dynamics

$$
\begin{aligned}
\delta x(k+1) &= Ax(k) + Bu(k) - x_s = Ax(k) + Bu(k) - (Ax_s + Bu_s) = A\delta x(k) + B\delta u(k) \\
\delta y(k) &= y(k) - y_{sp} = Cx(k) - Cx_s = C\delta x(k).
\end{aligned}
$$

Thus, the deviation variables satisfy the same state equation as the original variables.

Systems of linear equations

For any $m \times n$ matrix $A \in \mathbb{R}^{m \times n}$ we have,

Rank of A :	$\operatorname{rank}(A) = \operatorname{rank}(A^\top) = r.$	
Range of A :	$\mathcal{R}(A) = \{Ax \mid x \in \mathbb{R}^n\}$	$\dim \mathcal{R}(A) = r.$
Nullspace of A :	$\mathcal{N}(A) = \{x \mid Ax = 0\}$	$\dim \mathcal{N}(A) = n - r.$
Orthogonal subspaces:	$\mathcal{R}(A^\top) \perp \mathcal{N}(A)$	$\mathcal{R}(A) \perp \mathcal{N}(A^\top).$

Systems of linear equations

For any $m \times n$ matrix $A \in \mathbb{R}^{m\times n}$ we have,

Rank of A :	$\mathrm{rank}(A) = \mathrm{rank}(A^\top) = r.$	
Range of A :	$\mathcal{R}(A) = \{Ax \mid x \in \mathbb{R}^n\}$	$\dim \mathcal{R}(A) = r.$
Nullspace of A :	$\mathcal{N}(A) = \{x \mid Ax = 0\}$	$\dim \mathcal{N}(A) = n - r.$
Orthogonal subspaces:	$\mathcal{R}(A^\top) \perp \mathcal{N}(A)$	$\mathcal{R}(A) \perp \mathcal{N}(A^\top).$

For the system of linear equations

$$Ax = b, \quad A \in \mathbb{R}^{m\times n}$$

we have:

- There exists a solution x for every b if and only if $r = m$, i.e. $\mathcal{R}(A) = \mathbb{R}^m$.
- The solution is unique if and only if $r = n$, i.e. $\mathcal{N}(A) = \{0\}$.

Solution to overdetermined linear system

Consider the *overdetermined* system of linear equations

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}$$

with $m > n$. Assume that A has maximal rank $r = n$. Then the solution to the minimization problem

$$\min_{x \in \mathbb{R}^n} |Ax - b|$$

is given by

$$x^* = A^\dagger b$$

where the *pseudo-inverse* $A^\dagger$ is defined by

$$A^\dagger = (A^\top A)^{-1} A^\top.$$

Solution to overdetermined linear system

Consider the *overdetermined* system of linear equations

$$Ax = b, \quad A \in \mathbb{R}^{m\times n}$$

with $m > n$. Assume that A has maximal rank $r = n$. Then the solution to the minimization problem

$$\min_{x\in\mathbb{R}^n} |Ax - b|$$

is given by

$$x^* = A^\dagger b$$

where the *pseudo-inverse* $A^\dagger$ is defined by

$$A^\dagger = (A^\top A)^{-1} A^\top.$$

Remark

$Ax^* = AA^\dagger b$ is the orthogonal projection of b onto $\mathcal{R}(A)$, i.e. x^* is mapped to the vector in $\mathcal{R}(A)$ that is closest to b. In Matlab notation, $x^* = A\backslash b$.

Solutions to undetermined linear system

Consider, e.g., the *undetermined* system of linear equations

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}$$

with $m < n$. Assume that A has maximal rank $r = m$. Since the system has many solutions, a common choice is to pick a representative solution, e.g. the one which minimises the least-norm

$$\min_{x \in \mathbb{R}^n,\ Ax=b} x^\top x. \tag{9}$$

Solutions to undetermined linear system

Consider, e.g., the *undetermined* system of linear equations

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}$$

with $m < n$. Assume that A has maximal rank $r = m$. Since the system has many solutions, a common choice is to pick a representative solution, e.g. the one which minimises the least-norm

$$\min_{x \in \mathbb{R}^n,\ Ax=b} x^\top x. \tag{9}$$

Proposition

The solution to (9)*, where A is a full row rank, is*

$$x^* = A^\top (AA^\top)^{-1} b.$$

Solutions to undetermined linear system

Consider, e.g., the *undetermined* system of linear equations

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}$$

with $m < n$. Assume that A has maximal rank $r = m$. Since the system has many solutions, a common choice is to pick a representative solution, e.g. the one which minimises the least-norm

$$\min_{x \in \mathbb{R}^n,\ Ax=b} x^\top x. \tag{9}$$

Proposition

The solution to (9), *where A is a full row rank, is*

$$x^* = A^\top (AA^\top)^{-1} b.$$

Remark

The matrix $A^\top (AA^\top)^{-1}$ is also known as right pseudo-inverse.

Observability

Definition (Observability)

A linear, discrete time system

$$
\begin{aligned}
x^+ &= Ax \\
y &= Cx
\end{aligned}
$$

is observable if for some N, any $x(0)$ can be determined from $\{y(0), y(1), \dots, y(N-1)\}$.

Lemma (Observability)

The system is observable if and only if any of the following, equivalent, conditions hold:

- *The matrix* $\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$ *has full rank* n.
- *The matrix* $\begin{bmatrix} \lambda I - A \\ C \end{bmatrix}$ *has rank* n *for all* $\lambda \in \mathbb{C}$.

Lemma (Observability)

The system is observable if and only if any of the following, equivalent, conditions hold:

- *The matrix* $\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$ *has full rank* n.
- *The matrix* $\begin{bmatrix} \lambda I - A \\ C \end{bmatrix}$ *has rank* n *for all* $\lambda \in \mathbb{C}$.

Remark

A weaker condition is ***detectability****, which requires that any unobservable modes are strictly stable.*

Controllability

Definition (Controllability)

A linear, discrete time system

$$x^+ = Ax + Bu$$

is controllable if it is possible to steer the system from any state x_0 to any state x_f in finite time.

Controllability

Definition (Controllability)

A linear, discrete time system

$$x^+ = Ax + Bu$$

is controllable if it is possible to steer the system from any state x_0 to any state x_f in finite time.

Lemma (Controllability)

The system is controllable if and only if any of the following, equivalent conditions hold:

- *The matrix* $\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$ *has full rank* n.
- *The matrix* $\begin{bmatrix} \lambda I - A & B \end{bmatrix}$ *has rank* n *for all* $\lambda \in \mathbb{C}$.

Controllability

Definition (Controllability)

A linear, discrete time system

$$x^+ = Ax + Bu$$

is controllable if it is possible to steer the system from any state x_0 to any state x_f in finite time.

Lemma (Controllability)

The system is controllable if and only if any of the following, equivalent conditions hold:

- *The matrix* $\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$ *has full rank* n.
- *The matrix* $\begin{bmatrix} \lambda I - A & B \end{bmatrix}$ *has rank* n *for all* $\lambda \in \mathbb{C}$.

Remark

A weaker condition is ***stabilisability*** *which requires that any uncontrollable modes are strictly stable.*

Constant disturbance modelling

A constant disturbance d of dimension n_d can be modelled by the state equation

$$d^+ = d.$$

Constant disturbance modelling

A constant disturbance d of dimension n_d can be modelled by the state equation

$$d^+ = d.$$

Augmenting the state model (6) with this disturbance gives the model

$$\begin{aligned} \begin{bmatrix} x \\ d \end{bmatrix}^+ &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} \end{aligned} \tag{10}$$

where the disturbance affects both state and output equations through the matrices B_d and C_d.

Proposition (Detectability of the augmented system)

The augmented system (10) *is detectable if and only if the original system is detectable and the following condition holds:*

$$\operatorname{rank}\left(\begin{bmatrix} I - A & -B_d \\ C & C_d \end{bmatrix}\right) = n + n_d \tag{11}$$

Proposition (Detectability of the augmented system)

The augmented system (10) *is detectable if and only if the original system is detectable and the following condition holds:*

$$\operatorname{rank}\left(\begin{bmatrix} I-A & -B_d \\ C & C_d \end{bmatrix}\right) = n + n_d \tag{11}$$

Remark

In order for the rank condition to be satisfied, we must have

$$n_d \leq p$$

i.e. we must have at least as many measured outputs as the dimension of the disturbance vector.

Quadratic forms

A *quadratic form* is given by the expression

$$x^\top S x = |x|_S^2$$

with $x \in \mathbb{R}^n$ and S a real, *symmetric* matrix. The matrix S is *positive definite* ($S \succ 0$) if and only if

$$x^\top S x > 0, \quad \forall \text{ nonzero } x \in \mathbb{R}^n$$

and the matrix S is *positive semidefinite* ($S \succeq 0$) if and only if

$$x^\top S x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

Stability

Definition (Stability)

The linear, discrete time system

$$x^+ = Ax \tag{12}$$

is asymptotically stable (solutions converge to the origin) if and only if **the magnitudes of the eigenvalues of A are strictly less than 1 (A is stable)**.

Stability

Definition (Stability)

The linear, discrete time system

$$x^+ = Ax \tag{12}$$

is asymptotically stable (solutions converge to the origin) if and only if **the magnitudes of the eigenvalues of A are strictly less than 1 (A is stable)**.

The evolution of the state trajectory can be studied via the quadratic form

$$V(x) = x^\top Sx = |x|_S^2, \quad S \succeq 0$$

which can be interpreted as a generalization of the Euclidean norm $|x|$ (squared). Evaluated along solutions to (12), V changes according to

$$V(x^+) - V(x) = -x^\top(S - A^\top SA)x \equiv -x^\top Qx.$$

Hence, the matrix Q determines whether V will decay or not.

Lemma (Stability)

For a stable system, the following conditions are equivalent:

(a) *The matrix A is stable.*

(b) *For each $Q \succ 0$ there is a unique solution $S \succ 0$ of the discrete Lyapunov equation*

$$S - A^\top S A = Q. \tag{13}$$

LTI systems in Matlab

Example: Paper machine headbox – Ex 2.4 in [4].

```
% Create continuous time LTI object
Ac = [-1.93    0        0        0;
       0.394  -0.426    0        0;
       0       0       -0.63     0;
       0.82   -0.784    0.413   -0.426 ];
Bc = [ 1.274   1.274;
       0       0;
       1.34   -0.65;
       0       0 ];
Cc = [ 0       1        0        0;
       0       0        1        0;
       0       0        0        1 ];
Dc = zeros(3,2);
csys = ss(Ac,Bc,Cc,Dc);   %create state space model

% Assign variable names
set(csys,'InputName',{'Stock flowrate';'WW flowrate'},...
    'OutputName',{'Headbox level';'Feed tank conc';'Headbox conc'},...
    'StateName',{'Feed tank level';'Headbox level';...
    'Feed tank conc';'Headbox conc'},...
    'TimeUnit','minutes');
get(csys);
```

```
% Compute eigenvalues
eig(Ac);

% Create discrete time model
Ts=2;
dsys=c2d(csys,Ts);

% Compute eigenvalues
eig(dsys.A);
exp(eig(Ac)*Ts);

% Controllability
rank(ctrb(dsys));
rank(ctrb(dsys.A,dsys.B(:,1)));
rank(ctrb(dsys.A,dsys.B(:,2)));

% Observability
rank(obsv(dsys))
rank(obsv(dsys.A,dsys.C(1,:)));
rank(obsv(dsys.A,dsys.C(2,:)));
rank(obsv(dsys.A,dsys.C(3,:)));

% Step response
step(dsys);
```

References

[1] J.B. Rawlings, D.Q. Mayne, and M.M. Diehl. *Model Predictive Control: Theory, Computation, and Design, 2nd edition*. Nob Hill Publishing 2017.
Available online at `https://sites.engineering.ucsb.edu/~jbraw/mpc`

[2] G. Goodwin, M.M. Seron, and J.A. De Don·. *Constrained Control and Estimation*. Springer 2004.
Available online via Chalmers Library.

[3] F Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*.
Available online at http://www.mpc.berkeley.edu/mpc-course-material

[4] J. Maciejowski. *Predictive Control with Constraints*. Prentice Hall 2002.

[5] S. Boyd and L. Vandenberghe. *Convex optimisation*. Cambridge University Press 2004.

[6] Diehl, M. *Real-Time Optimization for Large Scale Nonlinear Processes.* PhD thesis, University of Heidelberg, 2001.

CHALMERS
UNIVERSITY OF TECHNOLOGY