Introduction to Model Predictive Control

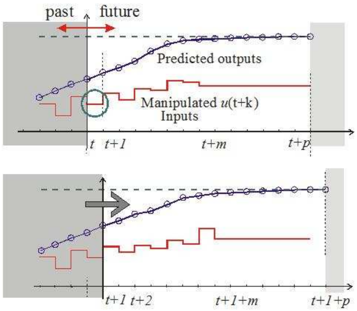
Lectures 2-3: Modeling

Francesco Borrelli

University of California at Berkeley,
Mechanical Engineering Department

ME190M-Fall 2009

Introduction



- At each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon (top diagram).
- The computed optimal manipulated input signal is applied to the process only during the following sampling interval $[t, t+1]$.
- At the next time step $t+1$ a new optimal control problem based on new measurements of the state is solved over a shifted horizon (bottom diagram).
- The resultant controller is referred to as *Model Predictive Control*

Reminder - MPC Algorithm

Consider the finite time optimal control problem

$$\begin{array}{lll} J^*(x(t)) = & \min_{u_t,u_{t+1},\ldots,u_{t+N-1}} & \sum_{k=0}^{N-1} l(x_{t+k},u_{t+k}) + p(x_{t+N}) \\ & \text{such that} & x_{k+1} = f(x_k,u_k),\ k=t,\ldots,t+N-1 \\ & & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k=t,\ldots,t+N-1 \\ & & x_{t+N} \in \mathcal{X}_f \\ & & x_t = x(t) \end{array} \tag{1}$$

At time t

- Measure (or estimate) the current state $x(t)$
- Find the optimal input sequence $U^* = \{u_t^*, u_{t+1}^*, u_{t+2}^*, \ldots, u_{t+N-1}^*\}$.
- Apply only $u(t) = u_t^*$, and discard $u_{t+1}^*, u_{t+2}^*, \ldots, u_{t+N-1}^*$

Repeat the same procedure at time $t+1$

Summarizing...

Need

1. A discrete-time model of the system (Matlab, Simulink)
2. A state observer
3. Set up an Optimization Problem (Matlab, MPT toolbox/Yalmip)
4. Solve an optimization problem (Matlab/Optimization Toolbox, NPSOL)
5. Verify that the closed-loop system performs as desired (avoid infeasibility/stability)
6. Make sure it runs in real-time and code/download for the embedded platform

Summarizing...

Need

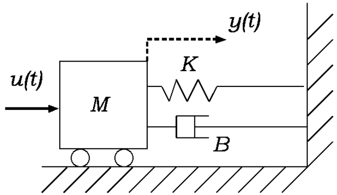
1. A discrete-time model of the system (Matlab, Simulink)
2. A state observer
3. Set up an Optimization Problem (Matlab, MPT toolbox/Yalmip)
4. Solve an optimization problem (Matlab/Optimization Toolbox, NPSOL)
5. Verify that the closed-loop system performs as desired (avoid infeasibility/stability)
6. Make sure it runs in real-time and code/download for the embedded platform

Continuous Time (CT) Systems

- Inputs and outputs of a CT system are defined for all time $t \in [0, +\infty)$, $t \in \mathbb{R}^+$
- Continuous linear dynamic systems are described by linear differential equations.

Example: The spring mass system

$$M\,\frac{d^2y(t)}{dt^2} + B\,\frac{dy(t)}{dt} + Ky(t) = u(t)$$



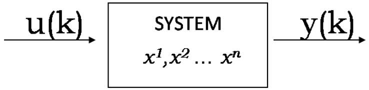
Discrete Time (DT) Systems

- Inputs and outputs of a DT system are defined at discrete time points. I.e. its inputs and outputs are sequences defined for $k = 0, 1, 2, \ldots,$ i.e. $k \in \mathbb{Z}^+$
- Discrete linear dynamic systems are described by linear difference equations.
  1. Bank savings account balance at the k-th month
  $$y(k+1) = (1+\alpha)y(k) + u(k)$$
  2. "Transformed" CT system

Discrete Time Systems

State space description of a system

Consider a discrete-time dynamic systems with an input $u(k)$ and an output $y(k)$.



- Given j and any $k > j$
- Given the input sequence $u(j), u(j+1), \ldots, u(k)$
- The state of the system at any instance j is the set of numbers $x^1(j), x^2(j), x^n(j)$ required to compute $y(k)$

Nonlinear DT State Space Description

$$\begin{aligned} x(k+1) &= f_d(x(k), u(k), k) \\ y(k) &= h_d(x(k), u(k), k) \end{aligned}$$

- Input vector $u(k) = [u^1(k),\ u^2(k), \cdots,\ u^m(k)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(k) = [y^1(k),\ y^2(k), \cdots,\ y^p(k)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(k) = [x^1(k),\ x^2(k), \cdots,\ x^n(k)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $x(k+1) = f_d(\ldots)$
- Output Equation: $y(t) = h_d(\ldots)$

DT LTI State Space Description

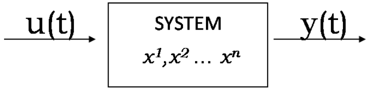
$$\begin{aligned} x(k+1) &= A_d\, x(k) + B_d\, u(k) \\ y(k) &= C_d\, x(k) + D_d\, u(k) \end{aligned}$$

- Input vector $u(k) = [u^1(k),\ u^2(k), \cdots,\ u^m(k)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(k) = [y^1(k),\ y^2(k), \cdots,\ y^p(k)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(k) = [x^1(k),\ x^2(k), \cdots,\ x^n(k)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $A_d \in \mathbb{R}^{n\times n},\ B_d \in \mathbb{R}^{n\times m}$
- Output Equation: $C_d \in \mathbb{R}^{p\times n},\ D_d \in \mathbb{R}^{p\times m}$

Continuous Time Systems

State Space Description of a CT System

Consider a continuous time dynamic systems with an input $u(t)$ and an output $y(t)$.



- Given t_0 and any $t > t_0$
- Given the input $u(\tau)$ for $t_0 \leq \tau \leq t$
- The state of the system at any instance t is the set of numbers $x^1(t_0), x^2(t_0), x^n(t_0)$ required to compute $y(t)$

Nonlinear CT State Space Description

$$\begin{aligned} \frac{d}{dt}x(t) &= f(x(t),u(t),t) \\ y(t) &= h(x(t),u(t),t) \end{aligned}$$

- Input vector $u(t) = [u^1(t),\ u^2(t), \cdots ,\ u^m(t)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(t) = [y^1(t),\ y^2(t), \cdots ,\ y^p(t)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(t) = [x^1(t),\ x^2(t), \cdots ,\ x^n(t)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $\dot{x}(t) = f(\dots)$
- Output Equation: $y(t) = h(\dots)$

CT LTI State Space Description

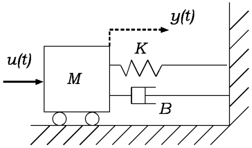
$$\begin{aligned} \frac{d}{dt}x(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned}$$

- Input vector $u(t) = [u^1(t),\ u^2(t), \cdots,\ u^m(t)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(t) = [y^1(t),\ y^2(t), \cdots,\ y^p(t)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(t) = [x^1(t),\ x^2(t), \cdots,\ x^n(t)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $A \in \mathbb{R}^{n\times n},\ \ B \in \mathbb{R}^{n\times m}$
- Output Equation: $C \in \mathbb{R}^{p\times n},\ \ D \in \mathbb{R}^{p\times m}$

Example

The spring mass system

$$M\,\frac{d^2y(t)}{dt^2} + B\,\frac{dy(t)}{dt} + Ky(t) = u(t)$$



$$\frac{d}{dt}\underbrace{\begin{bmatrix} p(t) \\ v(t) \end{bmatrix}}_{x(t)} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} p(t) \\ v(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B} u(t)$$

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C}\underbrace{\begin{bmatrix} p(t) \\ v(t) \end{bmatrix}}_{x(t)}$$

CT LTI Systems: Transfer Function Description

$$\begin{aligned} \tfrac{d}{dt}x(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned}$$

- assume zero initial conditions, i.e. $x(0) = 0$
- Use the Laplace transforms

$$Y(s) = \mathcal{L}\{y(t)\} \;\; U(s) = \mathcal{L}\{u(t)\}$$

- Then

LTI System in the Laplace Domain

$$Y(s) = G(s)\,U(s)$$

- $G(s)$ is called “Transfer Function”
- $G(s) = C(sI - A)^{-1}B + D$

From CT Systems to DT Systems

We Work With Discrete Time Models

We will use:

- Nonlinear Discrete Time

$$\begin{aligned} x(k+1) &= f_d(x(k), u(k), k) \\ y(k) &= h_d(x(k), u(k), k) \end{aligned}$$

- or LTI Discrete Time

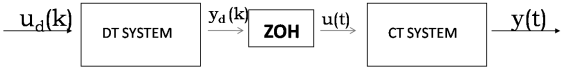
$$\begin{aligned} x(k+1) &= A_d\, x(k) + B_d\, u(k) \\ y(k) &= C_d\, x(k) + D_d\, u(k) \end{aligned}$$

Discretization

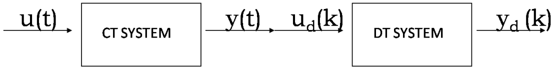
We call discretization the procedure of obtaining an "equivalent" DT model from a CT one.

CT and DT in Simulink

1. Connecting DT and CT systems



2. Connecting CT and DT systems



3. Note that sampling time of many blocks can be changed directly

Euler Discretization of Nonlinear Models

1. Given CT model

$$\begin{aligned}\frac{d}{dt}x(t) &= f(x(t), u(t), t)\\ y(t) &= h(x(t), u(t), t)\end{aligned}$$

2. Approximate $\frac{d}{dt}x(t) \simeq \frac{x(t+\Delta T)-x(t)}{\Delta T}$
3. ΔT is the sampling time
4. With abuse of notation $x(k) \triangleq x(t)$, $x(k+1) \triangleq x(t+\Delta T)$
5. Then DT model is

$$\begin{aligned}x(k+1) &= x(k) + \Delta T f(x(k), u(k), k) = f_d(x(k), u(k), k)\\ y(k) &= h(x(k), u(k), k) = h_d(x(k), u(k), k)\end{aligned}$$

Under regularity assumptions, if ΔT is small and CT and DT have “same” initial conditions and inputs, then outputs of CT and DT systems “will be close”

Euler Discretization of Linear Models

1. Given CT model

$$\begin{aligned}\tfrac{d}{dt}x(t) &= A\,x(t) + B\,u(t)\\ y(t) &= C\,x(t) + D\,u(t)\end{aligned}$$

2. the DT model obtained with Euler discretization is

$$\begin{aligned}x(k+1) &= A_d\,x(k) + B_d\,u(k)\\ y(k) &= C_d\,x(k) + D_d\,u(k)\end{aligned}$$

   with $A_d = I + \Delta T A$, $B_d = \Delta T B$, $C_d = C$, $D_d = D$.

- There are a variety of "better" discretization approaches (matlab: help c2d)
- will get rid of subscript d in the next lectures

ZOH ('Perfect') Discretization of Linear Models

1. Given CT model

$$\begin{array}{lcl} \frac{d}{dt}x(t) & = & A\,x(t) + B\,u(t) \\ y(t) & = & C\,x(t) + D\,u(t) \end{array}$$

2. the DT model obtained with ZOH discretization is

$$\begin{array}{lcl} x(k+1) & = & A_d\,x(k) + B_d\,u(k) \\ y(k) & = & C_d\,x(k) + D_d\,u(k) \end{array}$$

with $A_d = e^{A\Delta T}$, $B_d = \int_0^{\Delta T} e^{A}\eta B\ d\eta$, $C_d = C$, $D_d = D$
ΔT is the sampling time.

ZOH ('Perfect') Discretization of Linear Models

1. the DT model obtained with ZOH discretization is

$$\begin{aligned} x(k+1) &= A_d\, x(k) + B_d\, u(k) \\ y(k) &= C_d\, x(k) + D_d\, u(k) \end{aligned}$$

with $A_d = e^{A\Delta T}$, $B_d = \int_0^{\Delta T} e^{A\eta} B\, d\eta$, $C_d = C$, $D_d = D$

Main property:

- Simulate the CT model starting from x_0 and applying $\hat{u}(t) = ZOH(u(t), \Delta T)$:

$$\hat{u}(t) = u(k\Delta T) \;\text{ if }\; k\Delta T \le t < (k+1)\Delta T$$

- Simulate the DT model starting from x_0 and applying $u(0),\ u(\Delta T),\ u(2\Delta T), \ldots$
- Then

$$y(k) = y(k\Delta T)$$

where $y(k)$ is the output of the DT model at the k-th discrete instant and $y(k\Delta T)$ is the output of the CT model at time $k\Delta T$

From CT Systems in Transfer Function Form
to
CT Systems in State Space Form

Discretization of Linear Models: Transfer Functions

1. Consider the system described by the transfer Function

$$Y(s) = G(s)\,U(s)$$

2. Obtain state space equivalent model (matlab command tf2ss)

$$\begin{array}{rcl} \frac{d}{dt}x(t) & = & A\,x(t) + B\,u(t) \\ y(t) & = & C\,x(t) + D\,u(t) \end{array}$$

3. Apply discretization to state space model in 2

Note: There are infinite sets of $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ that produce the same $G(s)$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

Result:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u$$

$$y = \underbrace{\begin{bmatrix} b_o & b_1 & b_2 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

1. Define $X(s) = \frac{1}{A(s)} U(s)$
2. So that $Y(s) = B(s)\, X(s)$
3. We obtain

$$\begin{aligned} A(s)X(s) &= U(s) \\ (s^3 + a_2 s^2 + a_1 s + a_0)X(s) &= U(s) \end{aligned}$$

4. Recall

$$\mathcal{L}\left\{\frac{d^n}{dt^n} x(t)\right\} = s^n\, X(s)$$

5. Then

$$\frac{d^3\, x(t)}{dt^3} + a_2 \frac{d^2\, x(t)}{dt^2} + a_1 \frac{d\, x(t)}{dt} + a_o\, x(t) = u(t)$$

From TF to SS

Example

$$\frac{d^3\, x(t)}{dt^3} + a_2 \frac{d^2\, x(t)}{dt^2} + a_1 \frac{d\, x(t)}{dt} + a_o\, x(t) = u(t)$$

1. Define state Variables

$$x^1(t) = x(t),\ \ x^2(t) = \frac{d\, x(t)}{dt},\ \ x^3(t) = \frac{d^2\, x(t)}{dt^2}$$

2. Notice that $\frac{d^3\, x(t)}{dt^3} = \frac{d\, x^3(t)}{dt}$
3. Then

$$\begin{aligned}
\frac{d\, x^1(t)}{dt} &= x^2(t) \\
\frac{d\, x^2(t)}{dt} &= x^3(t) \\
\frac{d\, x^3(t)}{dt} &= -a_0\, x^1(t) - a_1\, x^2(t) - a_2\, x^3(t) + u(t)
\end{aligned}$$

From TF to SS

Example

$$X(s) = \frac{1}{A(s)} U(s), \quad Y(s) = B(s)\, X(s)$$

1. $Y(s) = (b_0 + b_1 s + b_2 s^2) X(s)$

2. $y(t) = b_o\, x(t) + b_1 \frac{d\,x(t)}{dt} + b_2 \frac{d^2\, x(t)}{dt^2}$

3. Therefore:

$$y(t) = b_o\, x^1(t) + b_1\, x^2(t) + b_2\, x^3(t)$$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

1. Is Equivalent to

$$\begin{aligned}
\frac{d\,x^1(t)}{dt} &= x^2(t)\\
\frac{d\,x^2(t)}{dt} &= x^3(t)\\
\frac{d\,x^3(t)}{dt} &= -a_0\,x^1(t) - a_1\,x^2(t) - a_2\,x^3(t) + u(t)\\
y(t) &= b_o\,x^1(t) + b_1\,x^2(t) + b_2\,x^3(t)
\end{aligned}$$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

1. Is Equivalent to

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u$$

$$y = \underbrace{\begin{bmatrix} b_o & b_1 & b_2 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Why Not Continuous Time Optimization?

$$
\begin{array}{rll}
J^*(x(t)) = & \min_{u_{[t:t+N]}} & \int_t^{t+N} l(x(\tau),u(\tau))d\tau \\
& \text{such that} & \dot{x}(\tau) = f(x(\tau),u(\tau)), \quad \tau \in [t,t+N] \\
& & x(\tau) \in \mathcal{X}, \quad \tau \in [t,t+N] \\
& & u(\tau) \in \mathcal{U}, \quad \tau \in [t,t+N]
\end{array} \qquad (2)
$$

- Choice of this course
- Issues are the same
- Soon or later need to discretize
- Software exists (Example: http://tomdyn.com/)

Homework 1/3

1. Download the "Ball and Plate" system description from bSpace. In the rest focus only on the x-axis and assume we measure all four states: $y_\beta(t) = x_\beta(t)$. Where sampling time is required use $\Delta T = 30ms$. (Note: we denoted the state vector x_β and the output $y_\beta(t)$ to distinguish them from the x and y positions of the ball).
2. Compute the transfer function of the system by using Matlab (hint: use the "ss2tf" command).
3. Compute a DT system by using Euler discretization, Call this system: DT system 1.
4. Compute a DT system the matlab command "c2d" with the method 'zoh'. Call this system: DT system 2.

Homework 2/3

1. Use Simulink to simulate the CT LTI State Space system starting from zero initial condition and with the following input profile: $u(t) = 0$ if $0 \leq t < 5$, $u(t) = 0.2V$ if $5 \leq t < 5.5$ and $u(t) = 0$ if $t \geq 5.5$.
2. Use Simulink to simulate the DT system 1 starting from zero initial conditions and with input obtained by sampling the continuous time input $u(t)$ of the previous point.
3. Use Simulink to simulate the DT system 2 starting from zero initial condition and with same input of the previous point.
4. Plot and compare the results of points 1,2,3

Homework 3/3

1. The control goal is to bring the ball from any admissible position to the origin of the state space (position and speed of the plate equal to zero and position and speed of the ball equal to zero).
   The state feedback discrete time controller $u(k) = Kx(k)$ with $K = [21.1 \;\; 4.0 \;\; -266.3 \;\; -5.8]$ and sampling $\Delta T = 30ms$ should do the job.
2. Simulate for 3 seconds the closed loop system (DT controller and CT LTI system in state space form) when the system starts from the initial condition $x(0) = [20, 0, 0, 0]$ (plate and ball at rest and ball at distance 20cm from the origin)
   1. Does the controller bring the ball to the origin?
   2. Are state and input constraints satisfied?
   3. How would you modify the controller in order to satisfy the constraints?

Check List

You should know how to:

1. Discretize nonlinear systems using Euler approach
2. Discretize linear systems using Euler or ZOH approaches
3. Discretize a linear system in TF form
4. Simulate a linear system in State Space form with Simulink (and/or Matlab)
5. Simulate a closed loop system with a CT model and a DT linear state feedback controller in Simulink (and/or Matlab)