Lecture 2: System Theory Basics

I. Model of Dynamic System

- **Initial Remarks**

Goal: Introduce mathematical models to be used in Model Predictive Control (MPC) describing the behavior of dynamic systems

Model classification: state space/transfer function, linear/nonlinear, time-varying/time-invariant, continuous-time/discrete-time, deterministic/stochastic

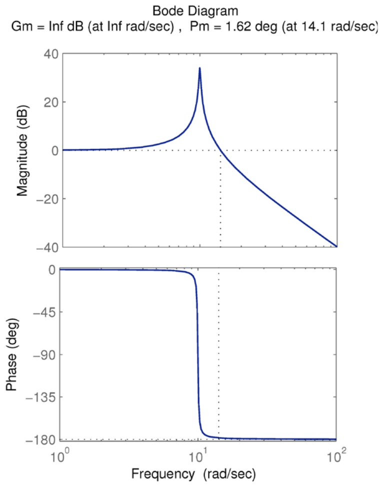
Note: If not stated differently, we use deterministic models.

Abbreviations used: LTI = linear, time-invariant; DT = discrete time; CT = continuous time

Models of physical systems derived from first principles are mainly: nonlinear, time-invariant, continuous-time, state space models (*)

Target models for MPC are mainly: linear, time-invariant, discrete-time, state space models (†)

The focus of this section is on how to 'transform' (*) to (†)



In classical controller design: Use of **transfer functions**

Represent input-output behavior of continuous time LTI system in **frequency domain**

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

Offer theory and tools for the analysis of the closed-loop behavior, e.g. Bode plots

Tools generally do not extend to nonlinear systems

(Recall: **constrained systems have nonlinear dynamics**)

Focus on state-space representation in our course

- **Nonlinear, Time-Invariant, Continuous-Time, State Space Models**

$$\dot{x} = g(x,u)$$

$$y = h(x,u)$$

$x \in \mathbb{R}^n$ state vector, $g(x,u)$: $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ system dynamics

$u \in \mathbb{R}^m$ input vector, $h(x,u)$: $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ output function $y \in \mathbb{R}^p$ output vector

A very general class of models

Higher-order ODEs can be easily brought to this form

Analysis and control synthesis generally is hard $\to$ linearization to bring it to be linear, time-invariant (LTI), continuous-time, state-space form

Recall: Equivalence of one n-th order ODE and n 1-st order ODEs

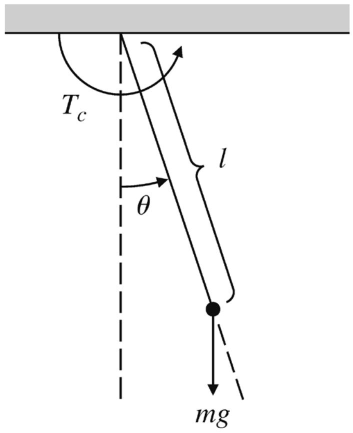
$$x^{(n)} + g_n(x, \dot{x}, \ddot{x}, \ldots, x^{(n-1)}) = 0$$

Define

$$x_{i+1} = x^{(i)},\ i = 0, \dots, n-1$$

Transformed system

$$\begin{aligned} x &= x_1 \\ \dot{x} &= \dot{x}_1 = x_2 \\ \ddot{x} &= \dot{x}_2 = x_3 \\ \vdots \quad & \quad \vdots \quad \quad \vdots \\ x^{(n-1)} &= \dot{x}_{n-1} = x_n \\ \dot{x}_n &= -\, g_n(x_1, x_2, \dots, x_n) \end{aligned}$$



Example: Pendulum

Moment of inertia wrt. rotational axis ml^2

Torque caused by the external force T_c

Torque caused by gravity $mgl\sin\theta$

System Equation: $ml^2\ddot{\theta} = T_c - mgl\sin\theta$

Using $x_1 = \theta,\ x_2 = \dot{\theta}$ and $u = T_c/(ml^2)$ the system can be brought to standard form:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\dfrac{g}{l}\sin x_1 + u \end{bmatrix} = g(x, u)$$

The output equation depends on the measurement, i.e. if θ is measured then $y = h(x, u) = x_1$

- **LTI Continuous-Time State Space Models**

$$\begin{aligned} \dot{x} &= A^c x + B^c u \\ y &= C^c x + D^c u \end{aligned}$$

$x \in \mathbb{R}^n$ state vector, $u \in \mathbb{R}^m$ input vector, $y \in \mathbb{R}^p$ output vector

$A^c \in \mathbb{R}^{n\times n}$ system matrix, $B^c \in \mathbb{R}^{n\times m}$ input matrix,

$C^c \in \mathbb{R}^{p\times n}$ output matrix, $D^c \in \mathbb{R}^{p\times m}$ throughput matrix

Vast theories exist for the analysis and control synthesis of linear systems

Recall: Solution to linear ODEs

Consider the ODE (written with explicit time dependence) $\dot{x}(t) = A^c x(t) + B^c u(t)$ with the initial condition $x_0 = x(t_0)$, then its solution is given by

$$x(t) = e^{A^c(t-t_0)}x_0 + \int_{t_0}^{t} e^{A^c(t-\tau)}Bu(\tau)\,\mathrm{d}\tau \quad \text{where} \quad e^{A^c t} = \sum_{n=0}^{\infty}\frac{(A^c t)^n}{n!}$$

Note: this is the superposition of the initial response and input response

Problem: Most physical systems are nonlinear but linear systems are much better understood. Nonlinear systems can be well approximated by a linear system in a 'small' neighborhood around a point in state space

Idea: Control keeps the system around some operating point! replace nonlinear by a linearized system around the operating point

Recall: First order Taylor expansion of $f(\cdot)$ around $\bar{x}$

$$f(x) \approx f(\bar{x}) + \frac{\partial f}{\partial x^{\mathrm{T}}}(x - \bar{x}) \quad \text{where} \quad \frac{\partial f}{\partial x^{\mathrm{T}}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad \text{is the Jacobian}$$

Note: x^{T} indicate that we use the numerator layout when doing matrix calculus in this course

Linearization

u_s keeps the system around the stationary operating point $x_s \rightarrow \dot{x}_s = g(x_s, u_s) = 0, y_s = h(x_s, u_s)$

$$\dot{x} = \underbrace{g(x_s, u_s)}_{=0} + \underbrace{\left.\frac{\partial g}{\partial x^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}}}_{=A^c} \underbrace{(x - x_s)}_{=\Delta x} + \underbrace{\left.\frac{\partial g}{\partial u^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}}}_{=B^c} \underbrace{(u - u_s)}_{=\Delta u} \Rightarrow \dot{x} - \underbrace{\dot{x}_s}_{=0} = \Delta\dot{x} = A^c \Delta x + B^c \Delta u$$

$$y = \underbrace{h(x_s, u_s)}_{=y_s} + \underbrace{\left.\frac{\partial h}{\partial x^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}}}_{=C} \underbrace{(x - x_s)}_{=\Delta x} + \underbrace{\left.\frac{\partial h}{\partial u^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}}}_{=D} \underbrace{(u - u_s)}_{=\Delta u} \Rightarrow y - \underbrace{y_s}_{=0} = \Delta y = C\Delta x + D\Delta u$$

Note:

The linearized system is written in terms of **deviation variables** $\Delta x, \Delta u, \Delta y$

The linearized system is only a good approximation for "small" $\Delta x, \Delta u$

Subsequently, instead of $\Delta x, \Delta u, \Delta y$, x, y, u are used for brevity.

Example: Linearization of pendulum equations

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 + u \end{bmatrix} = g(x, u)$$

$$y = x_1 = h(x, u)$$

Case 1: Want to keep the pendulum around $x_s = [\pi/4 \quad 0]^{\mathrm{T}} \rightarrow u_s = g/l \sin(\pi/4)$

$$A^c = \left.\frac{\partial g}{\partial x^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos\left(\frac{\pi}{4}\right) & 0 \end{bmatrix}, \quad B^c = \left.\frac{\partial g}{\partial u^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$C = \left.\frac{\partial h}{\partial x^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = [1 \quad 0], \quad D = \left.\frac{\partial h}{\partial u^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = 0$$

Case 2: Want to keep the pendulum around $x_s = [0 \quad 0]^{\mathrm{T}} \rightarrow u_s = 0$

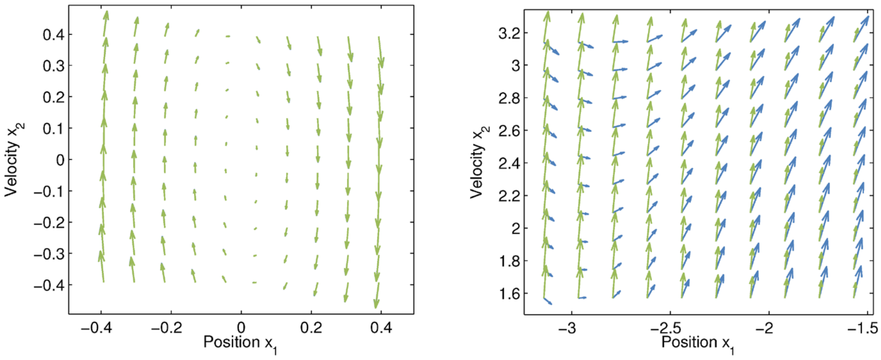
$$A^c = \left.\frac{\partial g}{\partial x^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = \begin{bmatrix} 0 & 1 \\ -\dfrac{g}{l} & 0 \end{bmatrix},\ B^c = \left.\frac{\partial g}{\partial u^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$C = \left.\frac{\partial h}{\partial x^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = [1 \quad 0],\ D = \left.\frac{\partial h}{\partial u^{\mathrm{T}}}\right|_{\substack{x=x_s \\ u=u_s}} = 0$$

If feedback control input $u = -x_2$ is applied:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \dfrac{-g}{l} & -1 \end{bmatrix} x$$

Note: always remember that linearization provides good approximations for small angles and velocities.



Gradient of
nonlinear system: blue
linearized system: green

Is the approximation far from the operating point really a bad approximation or not? It depends! If you demand the system to move really slowly, you can use feedback to correct it even if the model is inaccurate. If you demand the system to move in a fast manner, the model should be as accurate as possible. It depends on the robustness (ability to afford the wrong model) requirement of your task.

- **Time-Invariant, Discrete-Time, State Space Models**

Discrete-time systems are described by the difference equations:

$$x(k+1) = g(x(k), u(k))$$
$$y(k) = h(x(k), u(k))$$

$x \in \mathbb{R}^n$ state vector, $g(x,u)$: $\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ system dynamics

$u \in \mathbb{R}^m$ input vector, $h(x,u)$: $\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ output function $y \in \mathbb{R}^p$ output vector

Inputs and outputs of a discrete-time system are defined only at discrete time points, i.e. its inputs and outputs are sequences defined for $k \in \mathbb{Z}^+$

Discrete time systems describe either:

1. Inherently discrete systems, e.g. bank savings account balance at the k-th month

$$x(k+1) = (1-\alpha)x(k) + u(k)$$

2. 'Transformed' continuous-time system

Vast majority of controlled systems are not inherently discrete-time systems, but controllers are almost always implemented using microprocessors

$\rightarrow$ Finite computation time must be considered in the control system design, so we should **discretize the continuous-time system. Discretization is the procedure of obtaining an "equivalent" discrete-time system from a continuous-time system**

$\rightarrow$ The discrete-time model describes the state of the continuous-time system **only at particular instances** t_k, $k \in \mathbb{Z}^+$ **in time, where** $t_{k+1} = t_k + T_s$ **and** T_s **is called the sampling time**

Note: When we transform the continuous time system, usually, we would assume the **zeroth order hold** for the intermediate input. i.e. $u(t) = u(t_k), \forall t \in [t_k, t_{k+1})$, and it is also implemented in practice.

Recall: Euler Discretization of Nonlinear, Time-Invariant Models

Given CT model:

$$\dot{x}^c(t) = g^c(x^c(t), u^c(t))$$
$$y^c(t) = h^c(x^c(t), u^c(t))$$

(Finite difference) Approximation:

$$\dot{x}^c(t) \approx \frac{x^c(t+T_s) - x^c(t)}{T_s} = \frac{x(k+1) - x(k)}{T_s}$$

Where T_s is the sampling time and with $u(k) = u^c(t_0 + kT_s)$, the DT model is:

$$x(k+1) = x(k) + T_s g^c(x(k), u(k)) = g(x(k), u(k))$$
$$y(k) = h^c(x(k), u(k)) = h(x(k), u(k))$$

Note: for the time period between t and $t + T_s$, inherently the system dynamics g^c **is not constant (since both $x(t), u(t)$, especially $x(t)$ must change in the sampling period)**. However, in Euler discretization, we assume it to be unchanged, so **it is an approximation.** If T_s is small and CT and DT have "same" initial conditions and inputs, then outputs of CT and DT systems will be "close".

Example: Euler Discretization of the Nonlinear Pendulum Equations

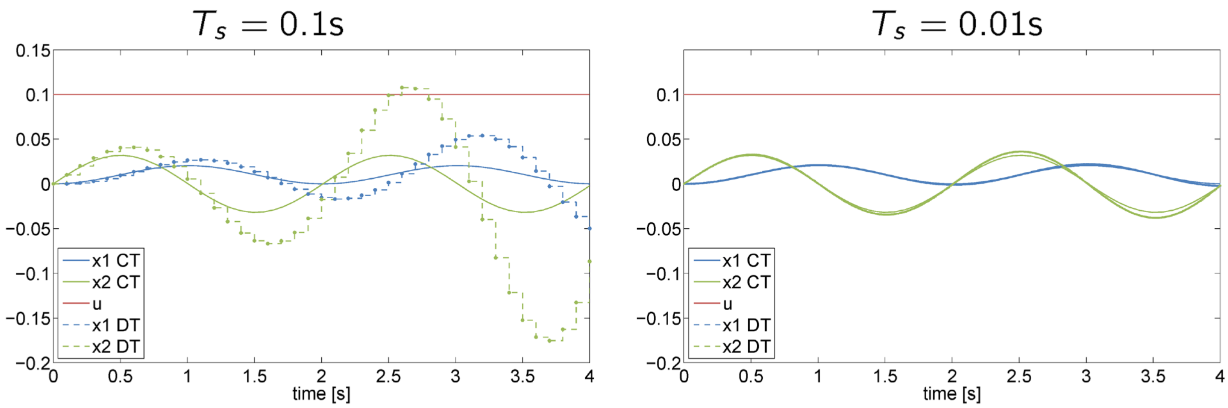
Using $g/l = 10\,[\mathrm{s}^{-2}]$, the nonlinear pendulum equations are given by

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ 10 \sin x_1 + u \end{bmatrix}$$

Discretizing the continuous time system using Euler we get the following discrete time system:

$$x(k+1) = x(k) + T_s \begin{bmatrix} x_2(k) \\ -10 \sin(x_1(k)) + u(k) \end{bmatrix}$$

Results for two different sampling times (left: too large sampling time, big error; right: good sampling):



Recall: Euler Discretization of Linear, Time-Invariant Models

Given CT model:

$$\dot{x}(t) = A^c x(t) + B^c u(t)$$
$$y(t) = C^c x(t) + D^c u(t)$$

The DT model obtained with Euler discretization is:

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

With $A = I + T_s A^c, B = T_s B^c, C = C^c, D = D^c$

Note: There are a variety of discretization approaches (matlab: help c2d)

Problem: Euler discretization always has errors. F**or linear systems, we have ways to calculate the exact discretization**

Exact Discretization of Linear Time-Invariant Models

We know the solution to the ODE:

$$x(t) = e^{A^c(t-t_0)} x_0 + \int_{t_0}^{t} e^{A(t-\tau)} Bu(\tau) \mathrm{d}\tau$$

Choose $t_0 = t_k$ hence $x_0 = x(t_0) = x(t_k)$

when $t = t_{k+1}$, use $t_{k+1} - t_k = T_s$, and $u(t) = u(t_k)\ \ \forall t \in [t_k, t_{k+1})$ we can get:

$$\begin{aligned} x(t_{k+1}) &= e^{A^c T_s} x(t_k) + \int_{t_k}^{t_{k+1}} e^{A^c(t_{k+1}-\tau)} B^c \mathrm{d}\tau\, u(t_k) \\ &= \underbrace{e^{A^c T_s}}_{\triangleq A} x(t_k) + \underbrace{\int_0^{T_s} e^{A^c(T_s-\tau')} B^c \mathrm{d}\tau'}_{\triangleq B} u(t_k) \\ &= Ax(t_k) + Bu(t_k) \end{aligned}$$

We found the exact discrete-time model predicting the state of the continuous-time system at time at time t_{k+1} given $x(t_k), k \in \mathbb{Z}^+$ under the assumption of a constant $u(t)$ during a sampling interval

Note: For continuous LTI system, we should always use exact discretization instead of forward Euler! Compared with forward Euler, it only requires constant $u(t)$ during the sampling time while forward Euler assumes constant dynamics $(x(t), u(t))$, which is impossible since $x(t)$ can never be constant. Also, recall that B can be calculated in an explicit way: $B = (A^c)^{-1}(A - I)B^c$ if A^c is invertible.

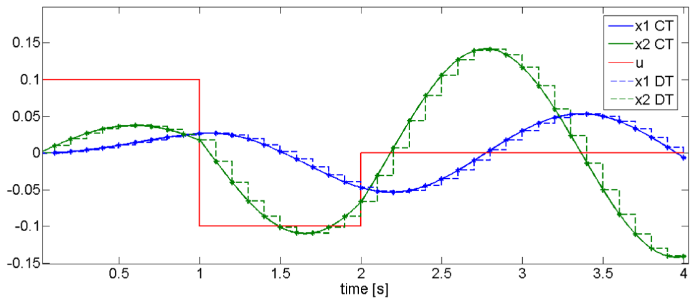
Example: Exact Discretization of Linear Time-Invariant Model (linearized pendulum)

Using $g/l = 10\,[\mathrm{s}^{-2}]$, the pendulum equations linearized about $x_s = [\pi/4, 0]^{\mathrm{T}}$ are given by:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -10\sqrt{2} & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

Discretizing the continuous-time system using the definitions of A and B, and $T_s = 0.1$ [s], we get the following discrete-time system:

$$x(k+1) = \begin{bmatrix} 0.965 & 0.099 \\ -0.699 & 0.965 \end{bmatrix} x(k) + \begin{bmatrix} 0.005 \\ 0.100 \end{bmatrix} u(k)$$



Note: we can observe that the discretization at each of the sample time is accurate!

Solution of Linear Time-Invariant Models

Find the solution of the discrete time system at time, given:

The discrete time linear system $x(k+1) = Ax(k) + Bu(k)$

An initial state $x(k)$ at discrete time k

An input sequence $\{u(k), \dots, u(k+N-1)\}$

$$\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \\
x(k+2) &= Ax(k+1) + Bu(k+1) = A(Ax(k) + Bu(k)) + Bu(k+1) \\
&= A^2 x(k) + ABu(k) + Bu(k+1) \\
x(k+3) &= Ax(k+2) + Bu(k+2) \\
&= A^3 x(k) + A^2 Bu(k) + ABu(k+1) + Bu(k+2) \\
&\vdots \\
x(k+N) &= A^N x(k) + A^{N-1} Bu(k) + \cdots + ABu(k+N-2) + Bu(k+N-1)
\end{aligned}$$

and therefore $x(k+N) = A^N x(k) + \sum_{i=0}^{N-1} A^i Bu(k+N-i-1)$

$\Rightarrow$ **A linear function of the initial state and the input sequence**

- **Summary and Recap**

Summary:

All in all, we work with discrete time models.

We will use: Nonlinear Discrete Time Models:

$$x(k+1) = g(x(k), u(k))$$
$$y(k) = h(x(k), u(k))$$

or Linear Time-Invariant Discrete Time Models:

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

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

Always remember to use the exact method for linear systems instead of Euler discretization

Recap:

Discrete time models are most suited for MPC

Solution of the discrete time model is a linear function of the initial state and the input sequence

Zero-order-hold provides exact discretization for linear systems

For general nonlinear systems, only approximate discretization methods exist, such as Euler discretization. Quality depends on sampling time

II. Analysis of LTI Discrete-Time Systems

- **Initial Remarks**

Goal: Introduce the concepts of stability, controllability, and observability

From this point on, we consider only discrete-time LTI systems for the rest of the section, so the notation is simplified, all matrices are now for discrete systems.

Recall: Coordinate Transform

Consider again the system

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

Input-output behavior, the sequence $\{y(k)\}_{k=0,1,2\ldots}$ entirely defined by $x(0)$ and $\{u(k)\}_{k=1,2,\ldots}$

$\Rightarrow$ **Infinitely many choices** of the state that yield the same input-output behavior, and certain choices facilitate system analysis

Consider the nonsingular (invertible) linear transformation $\tilde{x}(k) = Tx(k)$ $(\det(T) \neq 0)$, we have:

$$T^{-1}\tilde{x}(k+1) = AT^{-1}\tilde{x}(k) + Bu(k)$$
$$y(k) = CT^{-1}\tilde{x}(k) + Du(k)$$

or equivalently written:

$$\tilde{x}(k+1) = \underbrace{TAT^{-1}}_{\tilde{A}}\tilde{x}(k) + \underbrace{TB}_{\tilde{B}}u(k)$$
$$y(k) = \underbrace{CT^{-1}}_{\tilde{C}}\tilde{x}(k) + \underbrace{D}_{\tilde{D}}u(k)$$

Note: $u(k)$ and $y(k)$ are unchanged.

- **Stability of Linear Systems**

Theorem: Asymptotic Stability of Linear Systems

The autonomous (i.e. no control input) LTI system:

$$x(k+1) = Ax(k)$$

is globally asymptotically stable:

$$\lim_{k\to\infty} x(k) = 0, \ \ \forall x(0) \in \mathbb{R}^n$$

if and only if $|\lambda_i| < 1, \ \forall i = 1, \cdots, n$ where λ_i are the eigenvalues of A

Note:

$\max |\lambda_i|$ is also called the spectral radius of A. Stability means all the eigenvalues are strictly inside the unit circle in the complex plane, i.e. $\max |\lambda_i| < 1$.

Proof: through coordinate transformation (a simplified version under certain assumption)

Assume that A has n linearly independent eigenvectors $e_1, e_2, \ldots, e_n$, then the coordinate transformation $\tilde{x}(k) = [e_1 \ \ e_2 \ \ \ldots \ \ e_n]^{-1}x(k) = Tx(k)$ transforms an LTI discrete-time system to:

$$\tilde{x}(k+1) = TAT^{-1}\tilde{x}(k) = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}\tilde{x}(k) = \Lambda\tilde{x}(k)$$

The state $\tilde{x}(k)$ can be explicitly formulated as a function of $\tilde{x}(0) = Tx(0)$

$$\tilde{x}(k) = \Lambda\tilde{x}(0) = \begin{bmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \lambda_2^k & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n^k \end{bmatrix}\tilde{x}(0)$$

We thus have that:

$$\begin{aligned} \tilde{x}(k) = \ &\Rightarrow |\tilde{x}(k)| = |\Lambda^k\tilde{x}(0)| \\ &\Rightarrow |\tilde{x}(k)| = |\Lambda^k|\,|\tilde{x}(0)| \\ &\Rightarrow |\tilde{x}_i(k)| = |\lambda_i^k|\,|\tilde{x}(0)| = |\lambda_i|^k|\tilde{x}(0)| \end{aligned}$$

Therefore, if any $|\lambda_i| \geq 1$ then $\lim_{k\to\infty}\tilde{x}(k) \neq 0$ for $\tilde{x}(0) \neq 0$. On the other hand, if $|\lambda_i| < 1$,

$\forall i \in 1, \cdots, n$, then $\lim_{k\to\infty}\tilde{x}(k)=0$ and we have asymptotic stability.

Note:

1. Always remember that this proof is only suitable for linearly independent eigenvectors, i.e. the eigenvalues are distinct. For repeated eigenvalues, the eigenvectors might not be linearly independent and it can not be brought into diagonal form. Still, the generalized form of proof can be found and assertions still hold but Jordan matrices have to be used.
2. The new coordinates move/change independently. They are also called modes.

- **Reachability, Controllability and Stabilizability of Linear System**

Definition (Reachability): A system $x(k+1)=Ax(k)+Bu(k)$ is reachable if for any pair of states $x(0)$, x^* there exists a finite time N and a control sequence $\{u(0),\cdots,u(N-1)\}$ such that $x(N)=x^*$, i.e.:

$$x^*=x(N)=A^N x(0)+[B \quad AB \quad \cdots \quad A^{N-1}B]\begin{bmatrix} u(N-1) \\ u(N-2) \\ \vdots \\ u(0) \end{bmatrix}$$

Definition (Controllability): A system is controllable if $x^*=0$ mentioned above.

It can be observed that the key to reachability/controllability is the span of $[B \quad AB \quad \cdots \quad A^{N-1}B]$. It follows from the Cayley-Hamilton theorem that A^k can be expressed as linear combinations of A_i, $i \in 0, 1, \cdots, n-1$ for $k \geq n$. Hence for all $N \geq n$:

$$\text{range}[B \quad AB \quad \cdots \quad A^{N-1}B]=\text{range}[B \quad AB \quad \cdots \quad A^{n-1}B]$$

That also means, if the system cannot be controlled to x^* in n steps, where n is the dimension of the system, then it can not be controlled in arbitrary number of steps.

We also define the reachability (most of the time also called controllability) matrix $\mathcal{R}$ (or $\mathcal{C}$):

$$\mathcal{R}=[B \quad AB \quad \cdots \quad A^{n-1}B]$$

Theorem (Condition for reachability):

The system is reachable if

$$\mathcal{R}\begin{bmatrix} u(n-1) \\ u(n-2) \\ \vdots \\ u(0) \end{bmatrix}=x^*-A^n x(0)$$

Has a solution for all right hand sides (RHS). From linear algebra: solution exists for all RHS if and only if n columns of $\mathcal{R}$ are linearly independent

$\Rightarrow$ The necessary and sufficient condition for reachability is:

$$\text{rank}(\mathcal{R})=n$$

Note:

1. We call the space spanned by columns of $\mathcal{R}$ reachable subspace, reachable if it is the whole space.
2. To be more specific, reachability is arbitrary (initial) states to arbitrary (final) states (or, it focuses on the problem that given any arbitrary initial states, whether or not the system can go to arbitrary final states under certain control input, the logic direction is initial to final); controllability is some (initial) states to single (final) state (or, it focus on the problem that given a specific final state, whether or not exist initial states so that the system can start from those initial values to move to the target final value under certain control input, the logic direction is final to initial.)
3. Why do we separate the definition of controllability and reachability? Conceptually, controllability only means $A^n x(0)$ is in $\mathrm{Im}(\mathcal{R})$ since its target is single ($x^* = 0$), but does not mean that $\mathrm{Im}(\mathcal{R})$ is the whole space. Mathematically, if A **is a nilpotent, i.e.** $\exists k,\ A^k = 0$, then we do not need any input to make the state go to zero. Thus it is controllable, but not necessarily reachable. This case would not exist in the continuous time system because in continuous time, we would have e^{At} which would never be zero!

Definition (Stabilizability):

A system is called stabilizable if there exists an input sequence that returns the state to the origin asymptotically, starting from an arbitrary initial state.

Theorem (Condition for stabilizability):

A system is stabilizable if and only if all of its uncontrollable (i.e. native, not under certain control input) modes are stable. Stabilizability can be checked using the following condition

$$\text{if } \operatorname{rank}([\lambda_i I - A|\ B]) = n,\ \forall \lambda_i \in \Lambda_A^+ \ \Leftrightarrow\ (A,B) \text{ is stabilizable}$$

where Λ_A^+ is the set of all eigenvalues of A lying on or outside the unit circle in the complex plane. This test is also called the Hautus Lemma (PBH Test).

Note:

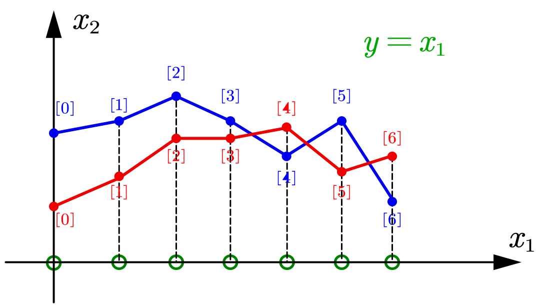
1. We only care about modes that are unstable. For stable modes, they will go to zero in the end even without control input.
2. Controllability implies stabilizability, and stabilization is not necessarily to be done in finite time.

- **Observability and Detectability**

Definition (Observability): Consider the following system **with zero input:**

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$

A system is said to be observable if there exists a finite N such that for every $x(0)$ the measurements $y(0), y(1), \cdots y(N-1)$ uniquely distinguish the initial state $x(0)$.



Note: Simple illustration of observability. See the two discrete trajectories (red and blue) given in the phase portrait above. The two trajectories has identical x_1 components. If we choose the output to be $y = x_1$, we can not uniquely distinguish the initial state. Hence this system is not observable.

Theorem (Condition for observability):

From our definition of observability, we can see that observability is asking the question of the uniqueness of the linear equations:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} x(0)$$

As previously mentioned, we can replace N by n (system's dimension) w.l.o.g. (Cayley-Hamilton)

We define $\mathcal{O} = [C^{\mathrm{T}} \ \ (CA)^{\mathrm{T}} \ \ \cdots \ \ (CA^{n-1})^{\mathrm{T}}]^{\mathrm{T}}$ to be the observability matrix

From linear algebra: solution is unique if and only if the n columns of $\mathcal{O}$ are linearly independent

$\Rightarrow$ The necessary and and sufficient condition for observability is:

$$\operatorname{rank}(\mathcal{O}) = n$$

Definition (Detectability):

A system is called detectable if it possible to construct from the measurement sequence a sequence of state estimates that converges to the true state asymptotically, starting from an arbitrary initial estimate.

Theorem (Condition for detectability):

A system is detectable if and only if all of its unobservable modes are stable. Detectability can be checked using the following condition. Likewise, this is also called the Hautus Lemma (PBH Test):

$$\text{if } \operatorname{rank}([A^{\mathrm{T}} - \lambda_i I \mid C^{\mathrm{T}}]) = n, \ \forall \lambda_i \in \Lambda_A^+ \ \Leftrightarrow \ (A, C) \text{ is detectable}$$

where Λ_A^+ is the set of all eigenvalues of A lying on or outside the unit circle in the complex plane.

Note:

1. Observability implies detectability
2. We call the space spanned by columns of $\mathcal{O}$ observable subspace, observable if this subspace is the whole space.

III. Stability of Nonlinear Systems

- **Initial Remarks**

Motivation: MPC has to deal with complex real world systems, which are always nonlinear!

For nonlinear systems there are many definitions of stability.

$\rightarrow$ Informally, we define a system to be stable **in the sense of Lyapunov**, if it stays in any arbitrarily small neighborhood of the origin when it is disturbed slightly. In the following, we always mean "stability" in the sense of Lyapunov.

We consider first the stability of a nonlinear, time-invariant, discrete-time system:

$$x(k+1)=g(x(k))$$

with an equilibrium point at 0, i.e. $g(0)=0$.

Note:

1. The above encompasses any open-loop or closed-loop autonomous system.
2. Always, stability is **a property of an equilibrium point of a system!**

- **Stability of Nonlinear Systems**

Definition (Stability for nonlinear systems):

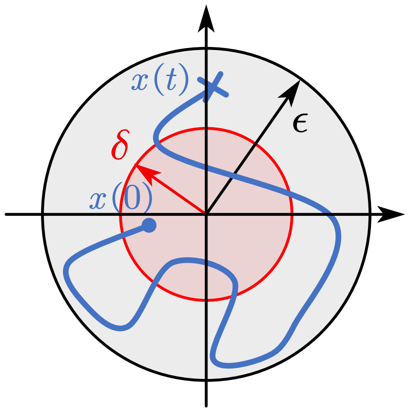
Formally, the equilibrium point $x=0$ of a system is:

Lyapunov stable if for every $\epsilon>0$ there exists a $\delta(\epsilon)$ such that (unstable otherwise)

$$\|x(0)\|<\delta(\epsilon)\rightarrow\|x(k)\|<\epsilon, \forall k\geq 0$$

Asymptotically stable in $\Omega\subseteq\mathbb{R}^n$ if it is Lyapunov stable and attractive, i.e.

$$\lim_{k\rightarrow\infty}x(k)=0,\ \forall x(0)\in\Omega$$



Globally asymptotically stable if it is asymptotically stable and $\Omega=\mathbb{R}^n$.

Note: Lyapunov stable is whether we can return (find) a $\delta(\epsilon)$ given ϵ as small as possible, since for large ϵ the condition is trivial and could always hold if the condition is satisfied for smaller ϵ.

How can we judge the stability of a nonlinear system? $\rightarrow$ constructing a Lyapunov function

What is Lyapunov function?

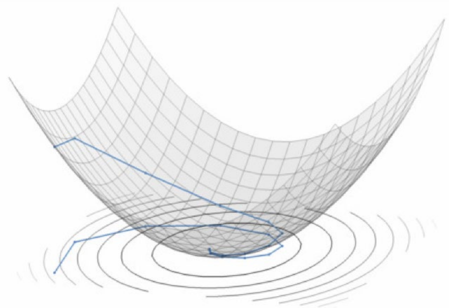
$\rightarrow$ Idea: A mechanical system is asymptotically stable when the total mechanical energy is decreasing over time (friction losses). A Lyapunov function is a system theoretic generalization of energy

Definition (Lyapunov functions):

Consider the equilibrium point $x_{eq}=0$ of the system. Let $\Omega\subset\mathbb{R}^n$ be a closed and bounded set containing the origin. A function $V:\mathbb{R}^n\rightarrow\mathbb{R}$, continuous at the origin, finite for every $x\in\Omega$ and such that

$$V(0)=0 \text{ and } V(x)>0,\ \forall x\in\Omega\backslash\{0\}$$

$$V(g(x))-V(x)\leq-\alpha(x),\ \forall x\backslash\{0\}$$



where $\alpha:\mathbb{R}^n\rightarrow\mathbb{R}$ is continuous positive definite, is called a Lyapunov function.

Theorem (Lyapunov theorem):

Lyapunov stability (asymptotic stability): If a system admits a Lyapunov function $V(x)$, then $x_{eq}=0$ is asymptotically stable in Ω

Lyapunov stability (global asymptotic stability): If a system (1) admits a Lyapunov function $V(x)$

that additionally satisfies:

$$\|x\| \to \infty \Rightarrow V(x) \to \infty$$

then $x=0$ is globally asymptotically stable.

Note:

1. The Lyapunov theorems only provide sufficient conditions
2. For general nonlinear systems it is usually difficult to find a Lyapunov function. It can sometimes be derived from physical considerations. Another approach is: First decide on the form of the Lyapunov function (e.g., quadratic), then search for parameter values e.g., via optimization so that the required properties hold.
3. For linear systems, there exist constructive theoretical results on the existence of a quadratic Lyapunov function

Global Lyapunov Stability of Linear Systems

Consider the linear system:

$$x(k+1) = Ax(k)$$

Take $V(x) = x^{\mathrm{T}}Px$ with $P \succ 0$ (positive definite) as a candidate Lyapunov function. It satisfies $V(0)=0, V(x)>0$ and $\|x\| \to \infty \Rightarrow V(x) \to \infty$

Check 'energy decrease' condition:

$$\begin{aligned} V(Ax(k)) - V(x(k)) &= x(k)^{\mathrm{T}}A^{\mathrm{T}}PAx(k) - x(k)^{\mathrm{T}}Px(k) \\ &= x(k)^{\mathrm{T}}(A^{\mathrm{T}}PA - P)x(k) \leq -\alpha(x(k)) \end{aligned}$$

We can choose $\alpha(x(k)) = x(k)^{\mathrm{T}}Qx(k),\ Q \succ 0$ Hence, the condition can be satisfied if a $P \succ 0$ can be found that solves the discrete-time Lyapunov equation:

$$A^{\mathrm{T}}PA - P = -Q,\ Q \succ 0$$

Theorem (Existence of solution to the DT Lyapunov equation):

The discrete-time Lyapunov equation has a unique solution $P \succ 0$ if and only if A has all eigenvalues inside the unit circle, i.e. if the system $x(k+1) = Ax(k)$ is stable.

Therefore, for LTI systems global asymptotic Lyapunov stability is not only sufficient but also necessary, and it agrees with the notion of stability based on eigenvalue location

Note (comments on theorem of existence of solution to the DT Lyapunov equation):

1. From definition, we first choose P then solve the discrete-time Lyapunov equation. If for some P , the equation can not be satisfied ($A^{\mathrm{T}}PA - P$ is not negative definite), what does this imply? It implies nothing. It only means for the current choice of P, we can not guarantee the condition, but for other P, it might work, because it is a necessary condition.

   However, for linear cases, we actually first specify Q , then we solve the equation for P, then from the definition, the system is stable iff $P \succ 0$, which means now it is necessary and sufficient judgment.

2. The Lyapunov equation should be easy to solve since it is a linear equation in the unknown P.
3. Note that stability is always global for linear systems.

Property of P

The matrix P used in the definition Lyapunov function can also be used to determine the infinite horizon cost-to-go for an asymptotically stable autonomous system $x(k+1)=Ax(k)$ with a quadratic cost function determined by Q. More precisely. Defining $\Psi(x(0))$ as:

$$(A^k)^{\mathrm{T}} Q A^k$$

Then we have that:

$$\Psi(x(0)) = x(0)^{\mathrm{T}} P x(0)$$

Proof sketch:

Define $H_k = (A^k)^{\mathrm{T}} Q A^k$ and $P = \sum_{k=0}^{\infty} H_k$ (limit of the sum exists because the system is assumed asymptotically stable). We have $A^{\mathrm{T}} H_k A = (A^{k+1})^{\mathrm{T}} Q A^{k+1} = H_{k+1}$, thus:

$$A^{\mathrm{T}} P A = \sum_{k=0}^{\infty} A^{\mathrm{T}} H_k A = \sum_{k=0}^{\infty} H_{k+1} = \sum_{k=1}^{\infty} H_k = P - H_0 = P - Q$$

Example:

Consider the discrete time system:

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$$

with $u(k) = Kx(k) = -[0.1160 \quad 0.5385]\, x(k)$

We choose $Q = I$ and solve the discrete time Lyapunov equation:

$$(A+BK)^{\mathrm{T}} P (A+BK) - P = -Q \Rightarrow P = \begin{bmatrix} 0.3135 & 0.0129 \\ 0.0129 & 0.2257 \end{bmatrix}$$

Closed-loop simulation: