NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Modeling and Optimal control

Cyber-Physical Control Systems

B. Guerreiro

2021

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Outline

Outline

Outline

State-space models

Properties of discrete-time systems

Unconstrained optimization

Constrained optimization

Optimal Control

Outline

State-space models

Properties of discrete-time systems

Unconstrained optimization

Constrained optimization

Optimal Control

Outline

Outline

State-space models

Properties of discrete-time systems

Unconstrained optimization

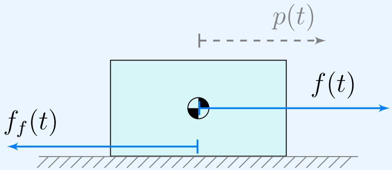
Constrained optimization

Optimal Control

A simple car model

Problem (Example 1.1: Free-body or simple car model)

Consider a car evolving on an horizontal plane, where the only forces acting on the vehicle are the one applied by the engine and the air friction.



What is the dynamic model of the car that describes the variation of its position, $p(t)$*, and velocity,* $v(t)$*?*

Simple car model

Example 1.1: Simple car model (motion)

- Newton's 2nd law: $\frac{d}{dt}(mv(t)) = \sum f_{ext} = f(t) + f_f(t)$.
- We control the force acting on the car, $f(t)$.
- The friction force is proportional to its velocity: $f_f(t) = -\beta v(t)$.
- Other effects are negligible.
- Equation of motion in velocity:

$$\frac{dv(t)}{dt} = \dot{v}(t) = \frac{1}{m}(-\beta v(t) + f(t))$$

Simple car model

Example 1.1: Simple car model (input/output)

- Equation of motion in position (noting that $v(t) = \frac{dp(t)}{dt}$):

$$\frac{d^2p(t)}{dt} = \ddot{p}(t) = -\frac{\beta}{m}\dot{p}(t) + \frac{1}{m}f(t)$$

- Apply the Laplace transform to both sides.
- Obtain the input/output transfer function model:

$$H(s) = \frac{P(s)}{F(s)} = \frac{\frac{1}{m}}{s^2 + \frac{\beta}{m}s} = \frac{\frac{1}{m}}{s(s + \frac{\beta}{m})}$$

Simple car model

Example 1.1: Simple car model (state-space)

- What we know:

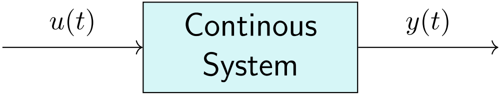
$$\begin{bmatrix} \dot{p}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ -\frac{\beta}{m} v(t) + \frac{1}{m} f(t) \end{bmatrix}$$

- Define state vector $\mathbf{x}(t) = \begin{bmatrix} p(t) \\ v(t) \end{bmatrix}$, input $u(t) = f(t)$, and output $y(t) = p(t)$.
- State-space representation of the car model:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \qquad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{\beta}{m} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$$

$$y(t) = \mathbf{C}\mathbf{x}(t) \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Continuous-time nonlinear systems



- Nonlinear state-space form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}_c(\mathbf{x}(t), \mathbf{u}(t), t) \quad , \mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{y}(t) = \mathbf{h}_c(\mathbf{x}(t), \mathbf{u}(t), t)$$

  where $\dot{\mathbf{x}}(t) := \frac{d\mathbf{x}}{dt}(t)$ and
  - $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector,
  - $\mathbf{u}(t) \in \mathbb{R}^m$ the control or input vector, and
  - $\mathbf{y}(t) \in \mathbb{R}^l$ the output vector.
- Input-output description of the system:

$$\mathbf{y}^{(k)}(t) = \mathbf{f}_y(\mathbf{y}(t), \mathbf{u}(t))$$

Linearization problem

Problem

Approximate a nonlinear system by a linear representation, near a given equilibrium (or operation) point of the original system.

- Equilibrium: system remains static without perturbations.

$$\mathbf{x}(t) = \text{const.} = \mathbf{x}_e \qquad \Rightarrow \qquad \dot{\mathbf{x}}(t) = 0 \qquad \Rightarrow \qquad \mathbf{f}_c(\mathbf{x}_e, \mathbf{u}_e) = 0$$

- Perturbed system:

$$\mathbf{x}(t) = \mathbf{x}_e + \delta\mathbf{x}(t) \qquad , \qquad \mathbf{u}(t) = \mathbf{u}_e + \delta\mathbf{u}(t)$$

- Perturbed system dynamics:

$$\frac{\mathbf{x}_e + \delta\mathbf{x}(t)}{dt} = \delta\dot{\mathbf{x}}(t) = \mathbf{f}_c(\mathbf{x}_e + \delta\mathbf{x}(t), \mathbf{u}_e + \delta\mathbf{u}(t))$$

Linearized system

- Taylor series expansion:

$$\begin{aligned}\mathbf{f}_c(\mathbf{x}_e + \delta\mathbf{x}(t), \mathbf{u}_e + \delta\mathbf{u}(t)) =& \mathbf{f}_c(\mathbf{x}_e, \mathbf{u}_e) + \frac{d\mathbf{f}_c}{d\mathbf{x}'}(\mathbf{x}_e, \mathbf{u}_e)\delta\mathbf{x}(t) \\ &+ \frac{d\mathbf{f}_c}{d\mathbf{u}'}(\mathbf{x}_e, \mathbf{u}_e)\delta\mathbf{u}(t) + h.o.t\end{aligned}$$

- As $\mathbf{f}_c(\mathbf{x}_e, \mathbf{u}_e) = 0$ and neglecting the 2nd and higher order terms, we get

$$\delta\dot{\mathbf{x}}(t) = \mathbf{A}\delta\mathbf{x}(t) + \mathbf{B}\delta\mathbf{u}(t) \quad , \quad \mathbf{A} = \frac{d\mathbf{f}_c}{d\mathbf{x}'}(\mathbf{x}_e, \mathbf{u}_e) \quad , \quad \mathbf{B} = \frac{d\mathbf{f}_c}{d\mathbf{u}'}(\mathbf{x}_e, \mathbf{u}_e)$$

- Similar procedure to the output equation.

Nonlinear car model

Example 1.2: Simple nonlinear car model (Exercise 1.3)

- The friction force is proportional to the squared velocity: $f_f(t) = -\beta v^2(t)$:

$$\dot{v}(t) = -\frac{\beta}{m} v^2(t) + \frac{1}{m} f(t)$$

- In equilibrium, $v(t) = v_e$ is constant ($\dot{v}(t) = 0$), then $f_e = \beta v_e^2$.
- Consider that $v(t) = v_e + \delta v(t)$ and $f(t) = f_e + \delta f(t)$
- Consider the Taylor expansion of the function $v^2(t)$ at v_e:

$$v^2 = v_e^2 + 2 v_e \delta v + \mathrm{h.o.t}$$

Linearized car model

Example 1.2: Simple linearized car model (Exercise 1.3)

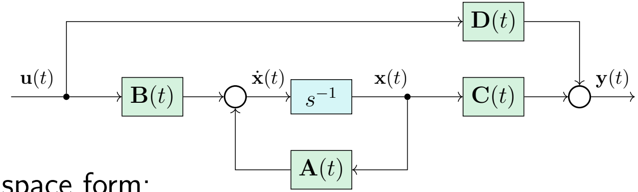
- The linearized equation of motion becomes

$$\begin{aligned}\frac{d(v_e+\delta v(t))}{dt} &= -\frac{\beta}{m}(v_e^2+2v_e\delta v(t))+\frac{1}{m}(f_e+\delta f(t))\\ &= -\frac{\beta v_e^2}{m}-\frac{2v_e\beta}{m}\delta v(t))+\frac{f_e}{m}+\frac{1}{m}\delta f(t))\end{aligned}$$

- Note that $\frac{f_e}{m}=\frac{\beta v_e^2}{m}$ and that $\frac{d(v_e+\delta v(t))}{dt}=\dot{\delta v}(t)$.
- Then, the simplified linearized equation of motion becomes

$$\dot{\delta v}(t)=-\frac{2v_e\beta}{m}\delta v(t)+\frac{1}{m}\delta f(t)$$

Continuous-time linear systems



- LTV state-space form:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad , \mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t)$$

  where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{l\times n}$, and $\mathbf{D} \in \mathbb{R}^{l\times m}$.

- Transfer function using the Laplace-transform of an LTI system:

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad \text{(general MIMO case)}$$

$$H(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} \quad \text{(SISO case)}$$

Solution of a continuous-time LTI system

- LTI system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad , \mathbf{x}(t_0) = \mathbf{x}_0$$

- Solution:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$

- Proof: differentiate both sides of the solution, and see that we obtain the LTI system equation, noting that the matrix exponential satisfies:

$$e^{\mathbf{A}t} = I + \mathbf{A}t + \frac{1}{2}\mathbf{A}^2t^2 + \frac{1}{3!}\mathbf{A}^3t^3 + \ldots = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k t^k$$

$$\frac{d}{dt}e^{\mathbf{A}t} = 0 + \mathbf{A} + \mathbf{A}^2t + \frac{1}{2}\mathbf{A}^3t^2 + \frac{1}{3!}\mathbf{A}^4t^3 + \ldots = \sum_{k=0}^{\infty} \mathbf{A}\frac{1}{k!}\mathbf{A}^k t^k = \mathbf{A}e^{\mathbf{A}t}$$

Transition matrix of a continuous-time LTI system

- Define the transition matrix (for LTI systems):

$$\mathbf{\Phi}(t, t_0) = e^{\mathbf{A}(t-t_0)}$$

- Important properties:

$$\dot{\mathbf{\Phi}}(t, t_0) = \mathbf{A}\mathbf{\Phi}(t, t_0)$$
$$\mathbf{\Phi}(t, t) = e^{\mathbf{A}0} = \mathbf{I}$$

- Solution of the continuous-time LTI system:

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{B}\mathbf{u}(\tau)d\tau$$

- Expression is also valid for LTV systems.

Incremental solution of an LTI

- Solution of an LTI system from time $t_0 = 0T$ to time $t_k = kT$:

$$\mathbf{x}(t_k) = e^{\mathbf{A}(t_k - t_0)}\mathbf{x}(0) + \int_{t_0}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$
$$= e^{\mathbf{A}kT}\mathbf{x}(0) + \int_{0}^{kT} e^{\mathbf{A}(kT-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$

- Next discrete-time instant (T is the sampling time):

$$\mathbf{x}(t_{k+1}) = e^{\mathbf{A}(k+1)T}\mathbf{x}(0) + \int_{0}^{kT} e^{\mathbf{A}((k+1)T-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau = \ldots$$
$$= e^{\mathbf{A}T}x(t_k) + \int_{kT}^{(k+1)T} e^{\mathbf{A}((k+1)T-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$

Discretization of state-space systems

- Assume $\mathbf{u}(t)$ is approximately constant during one sample interval, T.
- Let $\mathbf{x}(k) := \mathbf{x}(t_k) = \mathbf{x}(kT)$ and $\mathbf{u}(k) := \mathbf{u}(t_k) = \mathbf{u}(kT)$, then

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \quad \text{with}$$

$$\mathbf{A}_d = e^{\mathbf{A}T} \quad , \quad \mathbf{B}_d = \int_{kT}^{(k+1)T} e^{\mathbf{A}((k+1)T-\tau)}\mathbf{B}d\tau$$

- First-order approximation (Euler method):

$$\mathbf{A}_d \approx I + \mathbf{A}T \quad , \quad \mathbf{B}_d \approx \mathbf{B}T$$

- Forward Euler discretization of nonlinear systems:

$$\mathbf{x}(k+1) \approx \mathbf{x}(k) + T\mathbf{f}_c(\mathbf{x}(k), \mathbf{u}(k)) := \mathbf{f}_d(\mathbf{x}(k), \mathbf{u}(k))$$

- Higher-order methods: bilinear, Runge-Kutta, geometric, etc.

Discretized car model

Example 1.3: Simple discrete-time car model (Exercise 1.3)

- Recall the original car continuous-time model:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \qquad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{\beta}{m} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$$

$$y(t) = \mathbf{C}\mathbf{x}(t) \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

- Using the forward Euler approximation, with sample time T, the equivalent discrete-time state-space model is

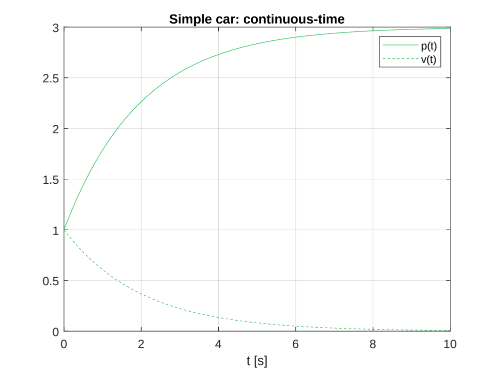
$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d u(k) \qquad \mathbf{A}_d = \begin{bmatrix} 1 & T \\ 0 & 1 - \frac{\beta T}{m} \end{bmatrix} \qquad \mathbf{B}_d = \begin{bmatrix} 0 \\ \frac{T}{m} \end{bmatrix}$$

$$y(k) = \mathbf{C}_d\mathbf{x}(k) \qquad \mathbf{C}_d = \mathbf{C}$$

Discretized car model

Example 1.3: Simple discrete-time car model (Exercise 1.3)

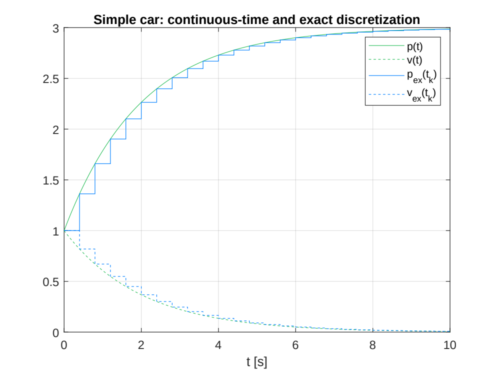
- Simulation results for $\beta = 0.5$, $m = 1$, $T = 0.4$ s, and $\mathbf{x}_0 = \mathbf{1} = \begin{bmatrix} 1 & 1 \end{bmatrix}'$:



Discretized car model

Example 1.3: Simple discrete-time car model (Exercise 1.3)

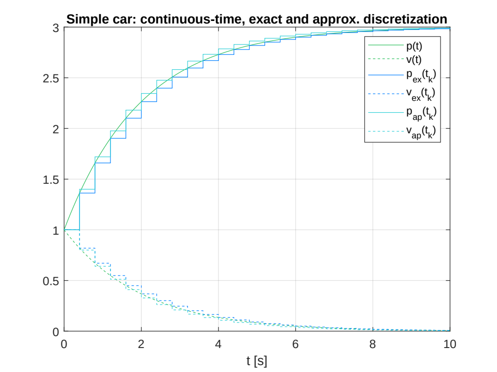
- Simulation results for $\beta = 0.5$, $m = 1$, $T = 0.4$ s, and $\mathbf{x}_0 = \mathbf{1} = \begin{bmatrix} 1 & 1 \end{bmatrix}'$:



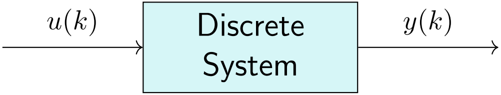
Discretized car model

Example 1.3: Simple discrete-time car model (Exercise 1.3)

- Simulation results for $\beta = 0.5$, $m = 1$, $T = 0.4$ s, and $\mathbf{x}_0 = \mathbf{1} = \begin{bmatrix}1 & 1\end{bmatrix}'$:



Discrete-time nonlinear dynamic systems



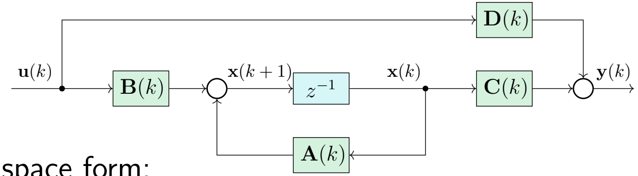
- Nonlinear state-space form:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}_d(\mathbf{x}(k), \mathbf{u}(k), k) \quad , \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(k) &= \mathbf{h}_d(\mathbf{x}(k), \mathbf{u}(k), k) \end{aligned}$$

  where
  - $\mathbf{x}(k) \in \mathbb{R}^n$ is the state vector,
  - $\mathbf{u}(k) \in \mathbb{R}^m$ the control or input vector, and
  - $\mathbf{y}(k) \in \mathbb{R}^l$ the output vector.
- Input-output description of the system:

$$\mathbf{y}(k) = \mathbf{f}_y(\mathbf{y}(k-1), \mathbf{u}(k))$$

Discrete-time linear dynamic systems



- LTV state-space form:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad , \mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k)$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{l\times n}$, and $\mathbf{D} \in \mathbb{R}^{l\times m}$.

- Transfer function using the z-transform of an LTI system:

$$\mathbf{H}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad \text{(general MIMO case)}$$

$$H(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \ldots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0} \quad \text{(SISO case)}$$

Transition matrix of a discrete-time LTI system

- Homogeneous LTI system (with no input, $\mathbf{u} = 0$):

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) \quad , \mathbf{x}(0) = \mathbf{x}_0$$

- Iterations from the initial condition:

$$\begin{aligned}
\mathbf{x}(1) &= \mathbf{A}\mathbf{x}(0) \\
\mathbf{x}(2) &= \mathbf{A}\mathbf{x}(1) = \mathbf{A}\mathbf{A}\mathbf{x}(0) = \mathbf{A}^2\mathbf{x}(0) \\
\mathbf{x}(3) &= \mathbf{A}\mathbf{x}(2) = \mathbf{A}\mathbf{A}^2\mathbf{x}(0) = \mathbf{A}^3\mathbf{x}(0) \\
&\dots \\
\mathbf{x}(k) &= \mathbf{A}\mathbf{x}(k-1) = \mathbf{A}^k\mathbf{x}(0)
\end{aligned}$$

- Define the LTI transition matrix $\mathbf{\Phi}(n) = \mathbf{A}^n$, noting that $\mathbf{\Phi}(0) = \mathbf{A}^0 = \mathbf{I}$.
- In general $\mathbf{x}(k+n) = \mathbf{A}^n\mathbf{x}(k) = \mathbf{\Phi}(n)\mathbf{x}(k)$.

Solution of a discrete-time LTI system

- What if we consider the input, $\mathbf{u} \neq 0$? (forced response)
- Iterations from the initial condition:

$$
\begin{aligned}
\mathbf{x}(1) &= \mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{u}(0) \\
\mathbf{x}(2) &= \mathbf{A}\mathbf{x}(1) + \mathbf{B}\mathbf{u}(1) = \mathbf{A}^2\mathbf{x}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(0) + \mathbf{B}\mathbf{u}(1) \\
\mathbf{x}(3) &= \mathbf{A}\mathbf{x}(2) + \mathbf{B}\mathbf{u}(2) = \mathbf{A}^3\mathbf{x}(0) + \mathbf{A}^2\mathbf{B}\mathbf{u}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(1) + \mathbf{B}\mathbf{u}(2) \\
&\cdots \\
\mathbf{x}(k) &= \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}(k-1) = \mathbf{A}^k\mathbf{x}(0) + \textstyle\sum_{m=0}^{k-1} \mathbf{A}^{k-1-m}\mathbf{B}\mathbf{u}(m) \\
\mathbf{x}(k) &= \mathbf{\Phi}(k)\mathbf{x}(0) + \textstyle\sum_{m=0}^{k-1} \mathbf{\Phi}(k-1-m)\mathbf{B}\mathbf{u}(m)
\end{aligned}
$$

- In general: $\mathbf{x}(k+n) = \mathbf{\Phi}(n)\mathbf{x}(k) + \sum_{m=0}^{n-1} \mathbf{\Phi}(n-1-m)\mathbf{B}\mathbf{u}(k+m)$

Solution of a discrete-time LTV system

- Iterations from the initial condition:

$$
\begin{aligned}
\mathbf{x}(1) &= \mathbf{A}(0)\mathbf{x}(0) + \mathbf{B}(0)\mathbf{u}(0) \\
\mathbf{x}(2) &= \mathbf{A}(1)\mathbf{x}(1) + \mathbf{B}(1)\mathbf{u}(1) = \mathbf{A}(1)\mathbf{A}(0)\mathbf{x}(0) + \mathbf{A}(1)\mathbf{B}(0)\mathbf{u}(0) + \mathbf{B}(1)\mathbf{u}(1) \\
\mathbf{x}(3) &= \mathbf{A}(2)\mathbf{x}(2) + \mathbf{B}(2)\mathbf{u}(2) = \mathbf{A}(2)\mathbf{A}(1)\mathbf{A}(0)\mathbf{x}(0) + \mathbf{A}(2)\mathbf{A}(1)\mathbf{B}(0)\mathbf{u}(0) \\
&\qquad + \mathbf{A}(2)\mathbf{B}(1)\mathbf{u}(1) + \mathbf{A}(2)\mathbf{B}(2)\mathbf{u}(2) + \mathbf{B}(3)\mathbf{u}(3) \\
&\dots
\end{aligned}
$$

- Define the LTV transition matrix $\mathbf{\Phi}(k,l) = \prod_{m=l}^{k-1} \mathbf{A}(m)$, with $\mathbf{\Phi}(k,k) = \mathbf{I}$.
- In general:

$$
\mathbf{x}(k) = \mathbf{\Phi}(k,k_0)\mathbf{x}(k_0) + \sum_{m=k_0}^{k-1} \mathbf{\Phi}(k,m+1)\mathbf{B}(m)\mathbf{u}(m)
$$

Incremental state-space model

- Models for controller design can account for integral action. How?
- Augment the state to include: disturbance estimation or input integrator.
- Recall the discrete-time LTI state-space model:

$$\begin{aligned} \mathbf{x}_d(k+1) &= \mathbf{A}_d\mathbf{x}_d(k) + \mathbf{B}_d\mathbf{u}(k) \quad , \mathbf{x}_d(0) = \mathbf{x}_0 \\ \mathbf{y}(k) &= \mathbf{C}_d\mathbf{x}_d(k) \end{aligned}$$

- Define the incremental state and input vectors:

$$\Delta\mathbf{x}_d(k) = \mathbf{x}_d(k) - \mathbf{x}_d(k-1) \qquad \text{and} \qquad \Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$$

- The incremental state equation becomes

$$\Delta\mathbf{x}(k+1) = \mathbf{A}_d\Delta\mathbf{x}(k) + \mathbf{B}_d\Delta\mathbf{u}(k)$$

Integral action state-space model

- Note that the output satisfies

$$\mathbf{y}(k+1) - \mathbf{y}(k) = \mathbf{C}_d\mathbf{A}_d\Delta\mathbf{x}(k) + \mathbf{C}_d\mathbf{B}_d\Delta\mathbf{u}(k)$$

- Define the augmented state vector $\mathbf{x}(k) = \begin{bmatrix} \Delta\mathbf{x}(k) \\ \mathbf{y}(k) \end{bmatrix}$
- Then, the augmented state-space model with integral action is

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \quad , \mathbf{x}(0) = \begin{bmatrix} \mathbf{0} \\ \mathbf{C}_d\mathbf{x}_0 \end{bmatrix}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_d & \mathbf{0} \\ \mathbf{C}_d\mathbf{A}_d & \mathbf{I} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{B}_d \\ \mathbf{C}_d\mathbf{B}_d \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}$$

Outline

State-space models

Properties of discrete-time systems

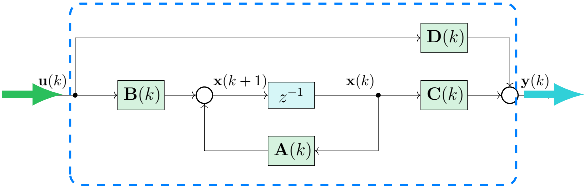
Unconstrained optimization

Constrained optimization

Optimal Control

Stability concepts

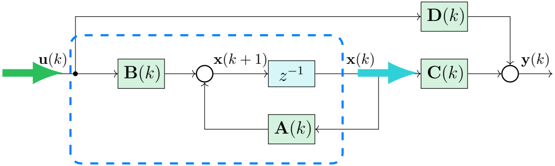
- Different definitions:
  - input-output stability
  - input-to-state stability
  - internal stability
  - Lyapunov stability



- Degree of stability:
  - unstable
  - marginal stability
  - assymptotic stability
  - exponencial stability

Stability concepts

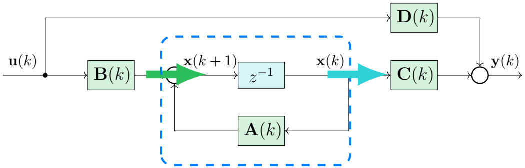
- Different definitions:
  - input-output stability
  - input-to-state stability
  - internal stability
  - Lyapunov stability



- Degree of stability:
  - unstable
  - marginal stability
  - assymptotic stability
  - exponencial stability

Stability concepts

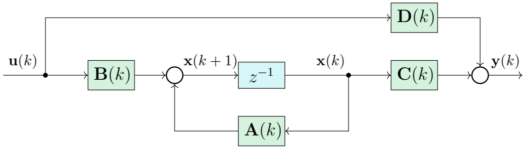
- Different definitions:
  - input-output stability
  - input-to-state stability
  - internal stability
  - Lyapunov stability



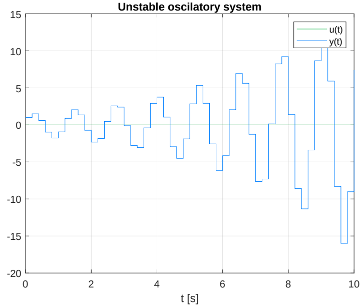
- Degree of stability:
  - unstable
  - marginal stability
  - assymptotic stability
  - exponencial stability

Stability concepts

- Different definitions:
  - input-output stability
  - input-to-state stability
  - internal stability
  - Lyapunov stability

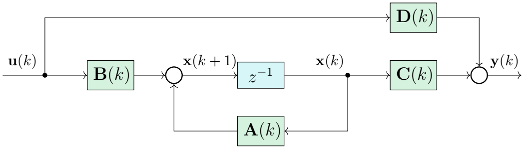


- Degree of stability:
  - unstable
  - marginal stability
  - assymptotic stability
  - exponencial stability

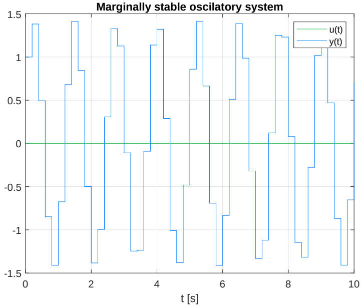


Stability concepts

- Different definitions:
  - input-output stability
  - input-to-state stability
  - internal stability
  - Lyapunov stability



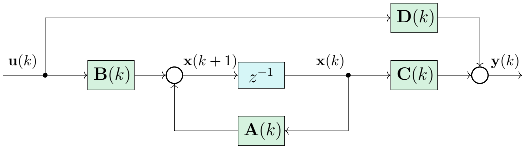
- Degree of stability:
  - unstable
  - marginal stability
  - assymptotic stability
  - exponencial stability

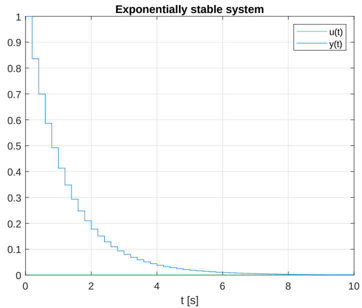


Stability concepts

- Different definitions:
  - input-output stability
  - input-to-state stability
  - internal stability
  - Lyapunov stability

- Degree of stability:
  - unstable
  - marginal stability
  - assyptotic stability
  - exponencial stability





Car model stability analysis (1)

Example 1.4: stability analysis (1)

- Recall the discrete-time LTI model ($\beta = 0.5$, $m = 1$, $T = 0.4$ s, and $\mathbf{x}_0 = \mathbf{1}$):

$$\mathbf{x}(k+1) = \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d u(k) \qquad \mathbf{A}_d = \begin{bmatrix} 1 & 0.4 \\ 0 & 0.8 \end{bmatrix} \qquad \mathbf{B}_d = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix}$$

$$y(k) = \mathbf{C}_d \mathbf{x}(k) + D_d u(k) \qquad \mathbf{C}_d = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D_d = 0$$

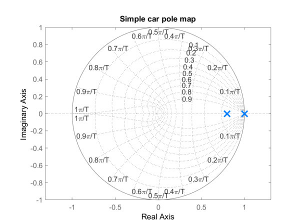
- z-transform transfer function is $H(z) = \mathbf{C}_d (z\mathbf{I} - \mathbf{A}_d)^{-1} \mathbf{B}_d + D_d$
- Characteristic Polynomial of $\mathbf{A}_d$ is $|z\mathbf{I} - \mathbf{A}_d| = (z-1)(z-0.8)$, thus

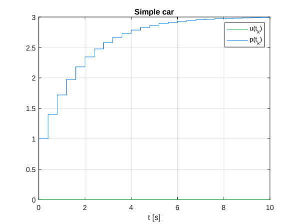
$$(z\mathbf{I} - \mathbf{A}_d)^{-1} = \frac{1}{|z\mathbf{I} - \mathbf{A}_d|} \operatorname{adj}(z\mathbf{I} - \mathbf{A}_d) = \begin{bmatrix} \frac{1}{z-1} & \frac{0.4}{(z-1)(z-0.8)} \\ 0 & \frac{1}{z-0.8} \end{bmatrix}$$

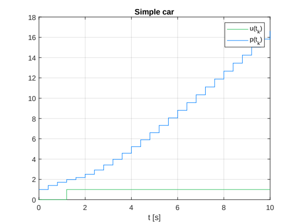
Car model stability analysis (1)

Example 1.4: stability analysis (1)

- Transfer function: $H(z) = \frac{0.16}{(z-1)(z-0.8)}$







- Bounded-input bounded-ouput (BIBO) unstable: one pole on unit sphere and another inside.
- The poles are actually the eigenvalues of matrix $\mathbf{A}_d$.

Eigenvalues of a matrix

- For a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, an eigenvalue λ and respective eigenvector $\mathbf{v}$ satisfy $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, or equivalently, $|\mathbf{A} - \lambda\mathbf{I}| = 0$ for $\mathbf{v} \neq 0$.
- Jordan normal form, $\mathbf{J}$, of matrix $\mathbf{A}$ is defined by $\mathbf{J} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$, with

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_3 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{J}_l \end{bmatrix} \quad \text{with} \quad \mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \lambda_i & 1 & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_i & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \lambda_i \end{bmatrix}$$

- If all Jordan blocks, $\mathbf{J}_i$, are 1×1, the Jordan normal form is diagonal.
- Thus, $\mathbf{A}$ is diagonalizable using $\mathbf{A} = \mathbf{T}^{-1}\mathbf{J}\mathbf{T}$, and $e^{\mathbf{A}t} = \mathbf{T}^{-1}e^{\mathbf{J}}\mathbf{T}$
- Exponential of a diagonal matrix is the diagonal of the exponentials.

Internal stability of LTI systems

Definition ([1])

The discrete-time LTI system described above is said to be:

1. marginally stable iff all the eigen values of $\mathbf{A}$ have magnitude smaller than of equal to 1 and all the Jordan blocks corresponding to eigenvalues with magnitude equal to 1 are 1×1;
2. asymptotically and exponentially stable iff all the eigenvalues of $\mathbf{A}$ have magnitude strictly smaller than 1; or
3. unstable iff at least one eigenvalue of $\mathbf{A}$ has magnitude larger than 1 or magnitude equal to 1, but the corresponding Jordan block is larger than 1×1.

Car model stability analysis (2)

Example 1.5: stability analysis (2)

- Eigenvalues of $\mathbf{A}_d$ are $\lambda_1 = 1$ and $\lambda_2 = 0.8$.
- Respective eigenvectors are $\mathbf{v}_1 = \begin{bmatrix}1 & 0\end{bmatrix}'$ and $\mathbf{v}_1 = \begin{bmatrix}-2 & 1\end{bmatrix}'$.
- Jordan normal form decomposition:

$$\mathbf{J} = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix} \qquad \text{and} \qquad \mathbf{T} = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}$$

- Internal stability:

  Marginally stable: all eigenvalues have magnitude ≤ 1, and the Jordan block of the eigenvalue with magnitude 1 is 1×1.

Lyapunov stability of LTV systems

Definition ([1])

The discrete-time LTV system described above, for every initial condition $\mathbf{x}(k_0) = \mathbf{x}_0 \in \mathbb{R}^n$, $k_0 \geq 0$, is said to be:

1. marginally stable in the Lyapunov/internal sense if the homogeneous state response $\mathbf{x}(k) = \mathbf{\Phi}(k, k_0)\mathbf{x}_0$, for all $k_0 \geq 0$, is uniformly bounded;
2. asymptotically stable (in the Lyapunov sense) if, in addition, the homogeneous state response satisfies $\mathbf{x}(k) \to 0$ as $l \to \infty$;
3. exponentially stable if, in addition, there exist constants $c > 0$, $\lambda < 1$ such that the homogeneous state response satisfies $\|\mathbf{x}(k)\| \leq c\lambda^{k-k_0}\|\mathbf{x}_0\|$, for all $k_0 \geq 0$;
4. unstable if it is not marginally stable in the Lyapunov sense.

Controlability and Reachability

- Basic idea: there is an input that drives the system from any initial state, $\mathbf{x}(k_0) = \mathbf{x}_0$ to any final state, $\mathbf{x}(k_1) = \mathbf{x}_1$ in finite time.

Definition (Controlability)

A discrete-time linear system is controllable on $[k_0, k_1]$ if given any state $\mathbf{x}_0$, there exists an input signal can take system from $\mathbf{x}(k_0) = \mathbf{x}_0$ to $\mathbf{x}(k_1) = \mathbf{0}$.

Definition (Reachability)

A discrete-time linear system is reachable on $[k_0, k_1]$ if given any state $\mathbf{x}_1$, there exists an input signal can take system from $\mathbf{x}(k_0) = \mathbf{0}$ to $\mathbf{x}(k_1) = \mathbf{x}_1$.

- **Note:** While these concepts are not equivalent, even for LTI systems when the state matrix $\mathbf{A}$ is singular, many texts use just the term controlability.

Controlability matrix for LTV systems

- Recall the solution of the LTV system, with initial state $\mathbf{x}(k_0) = \mathbf{0}$:

$$
\begin{aligned}
\mathbf{x}(k_1) =& \mathbf{\Phi}(k_1, k_0)\mathbf{0} + \sum_{m=k_0}^{k_1-1} \mathbf{\Phi}(k_1, m+1)\mathbf{B}(m)\mathbf{u}(m) \\
=& \mathbf{B}(k_1-1)\mathbf{u}(k_1-1) + \mathbf{\Phi}(k_1, k_1-1)\mathbf{B}(k_1-2)\mathbf{u}(k_1-2) + \dots \\
& + \mathbf{\Phi}(k_1, k_0+2)\mathbf{B}(k_0+1)\mathbf{u}(k_0+1) + \mathbf{\Phi}(k_1, k_0+1)\mathbf{B}(k_0)\mathbf{u}(k_0)
\end{aligned}
$$

- Define the controlability matrix, $\mathbf{M}_c(k_1,k_0)$, and stacked control $\mathbf{U}(k_1,k_0)$:

$$
\mathbf{M}_c(k_1, k_0) = \begin{bmatrix} \mathbf{B}(k_1-1) & \mathbf{\Phi}(k_1, k_1-1)\mathbf{B}(k_1-2) & \cdots & \mathbf{\Phi}(k_1, k_0+1)\mathbf{B}(k_0) \end{bmatrix}
$$

$$
\mathbf{U}(k_1, k_0) = \begin{bmatrix} \mathbf{u}(k_1)' & \mathbf{u}(k_1-1)' & \cdots & \mathbf{u}(k_0)' \end{bmatrix}'
$$

- Then the solution becomes: $\mathbf{x}(k_1) = \mathbf{M}_c(k_1, k_0)\mathbf{U}(k_1-1, k_0)$.

Controlability matrix for LTI systems

- Recall the solution of the LTI system, with initial state $\mathbf{x}(k_0) = \mathbf{0}$:

$$\begin{aligned}\mathbf{x}(k_1) =& \mathbf{A}^{k_1-k_0}\mathbf{0} + \sum_{m=0}^{k_1-k_0-1} \mathbf{A}^{k_1-k_0-1-m}\mathbf{B}\mathbf{u}(m) \\ =& \mathbf{B}\mathbf{u}(k_1-1) + \mathbf{A}\mathbf{B}\mathbf{u}(k_1-2) + \ldots + \mathbf{A}^{k_1-k_0-1}\mathbf{B}\mathbf{u}(k_0)\end{aligned}$$

- Define the LTI controlability matrix, $\mathbf{M}_c(k_1, k_0)$:

$$\begin{aligned}\mathbf{M}_c(k_1, k_0) &= \mathbf{M}_c(k_1 - k_0, 0) = \mathbf{M}_c(k_1 - k_0) \\ &= \begin{bmatrix}\mathbf{B} & \mathbf{A}\mathbf{B} & \cdots & \mathbf{A}^{k_1-k_0-1}\mathbf{B}\end{bmatrix}\end{aligned}$$

- Then the solution becomes: $\mathbf{x}(k_1) = \mathbf{M}_c(k_1 - k_0)\mathbf{U}(k_1 - 1, k_0)$.

Controlability of linear systems

Theorem (LTV systems [2])

The LTV system defined above (with $\mathbf{x}(k) \in \mathbb{R}^n$) is controllable on $[k_1, k_0]$ iff

$$\operatorname{rank} \mathbf{M}_c(k_1, k_0) = n$$

Theorem (LTI systems [2])

The LTI system defined above (with $\mathbf{x}(k) \in \mathbb{R}^n$) is controllable iff

$$\operatorname{rank} \mathbf{M}_c(n) = n$$

Car model controlability analysis

Example 1.6: controlability analysis

- Recall the discrete-time LTI model ($\beta = 0.5$, $m = 1$, and $T = 0.4$ s):

$$\mathbf{x}(k+1) = \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d u(k) \qquad \mathbf{A}_d = \begin{bmatrix} 1 & 0.4 \\ 0 & 0.8 \end{bmatrix} \qquad \mathbf{B}_d = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix}$$

- The controlability matrix of the pair $(\mathbf{A}_d, \mathbf{B}_d)$, with $n = 2$ is

$$\mathbf{M}_c(2) = \begin{bmatrix} \mathbf{B}_d & \mathbf{A}_d \mathbf{B}_d \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0.4 \end{bmatrix} & \begin{bmatrix} 1 & 0.4 \\ 0 & 0.8 \end{bmatrix} \begin{bmatrix} 0 \\ 0.4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0.16 \\ 0.4 & 0.32 \end{bmatrix}$$

- As $\operatorname{rank} \mathbf{M}_c(2) = 2$ the system is controllable.

Observability

Definition ([2])

A linear system is observable on $[k_0, k_1]$ if any initial state $\mathbf{x}(k_0) = \mathbf{x}_0$ is uniquely determined by the corresponding zero-input response $y(k)$ for $k = k_0, \ldots, k_1 - 1$.

- Consider the zero-input response of the LTV system, with $\mathbf{u}(k) = \mathbf{0}$:

$$\begin{bmatrix} \mathbf{y}(k_0) \\ \mathbf{y}(k_0+1) \\ \cdots \\ \mathbf{y}(k_1-1) \end{bmatrix} = \begin{bmatrix} C(k_0)\mathbf{x}(k_0) \\ C(k_0+1)\mathbf{x}(k_0+1) \\ \cdots \\ C(k_1-1)\mathbf{x}(k_1-1) \end{bmatrix} = \begin{bmatrix} C(k_0)\mathbf{x}_0 \\ C(k_0+1)\mathbf{\Phi}(k_0+1, k_0)\mathbf{x}_0 \\ \cdots \\ C(k_1-1)\mathbf{\Phi}(k_1-1, k_0)\mathbf{x}_0 \end{bmatrix}$$

Observability of LTV systems

- Define the observability matrix, $\mathbf{M}_o(k_0,k_1)$, and stacked output $\mathbf{Y}(k_0,k_1)$:

$$\mathbf{M}_o(k_0,k_1) = \begin{bmatrix} \mathbf{C}(k_0) \\ \mathbf{C}(k_0+1)\mathbf{\Phi}(k_0+1,k_0) \\ \cdots \\ \mathbf{C}(k_1-1)\mathbf{\Phi}(k_1-1,k_0) \end{bmatrix} \quad , \quad \mathbf{Y}(k_0,k_1) = \begin{bmatrix} \mathbf{y}(k_0) \\ \mathbf{y}(k_0+1) \\ \cdots \\ \mathbf{y}(k_1) \end{bmatrix}$$

- Then the output becomes $\mathbf{Y}(k_0,k_1-1) = \mathbf{M}_o(k_0,k_1)\mathbf{x}_0$.

Theorem (LTV systems [2])

The LTV system defined above (with $\mathbf{x}(k) \in \mathbb{R}^n$) is observable on $[k_0,k_1]$ iff

$$\operatorname{rank} \mathbf{M}_o(k_0,k_1) = n$$

Observability of LTI systems

- Define the LTI observability matrix, $\mathbf{M}_o(k_0, k_1)$,:

$$\mathbf{M}_o(k_0, k_1) = \mathbf{M}_o(0, k_1 - k_0) = \mathbf{M}_o(k_0 - k_1) = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \cdots \\ \mathbf{CA}^{k_1 - k_0 - 1} \end{bmatrix}$$

- Then the output becomes $\mathbf{Y}(k_0, k_1 - 1) = \mathbf{M}_o(k_1 - k_0)\mathbf{x}_0$.

Theorem (LTI systems [2])

The LTI system defined above (with $\mathbf{x}(k) \in \mathbb{R}^n$) is observable iff

$$\operatorname{rank} \mathbf{M}_o(n) = n$$

Observability of LTI systems

Sketch of the proof:

- If $\operatorname{rank} \mathbf{M}_o(n) = n$, then the matrix $\mathbf{M}_o(n)'\mathbf{M}_o(n)$ is invertible.
- Therefore, we can compute the initial state from the sequence of outputs:

$$\mathbf{x}_0 = (\mathbf{M}_o(n)'\mathbf{M}_o(n))^{-1}\mathbf{M}_o(n)'\mathbf{Y}(k_0, k_1 - 1)$$

- which means that the system is observable.

Car model observability analysis

Example 1.7: observability analysis

- Recall the discrete-time LTI model ($\beta = 0.5$, $m = 1$, and $T = 0.4$ s):

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d u(k) \qquad \mathbf{A}_d = \begin{bmatrix} 1 & 0.4 \\ 0 & 0.8 \end{bmatrix} \qquad \mathbf{B}_d = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix}$$

$$y(k) = \mathbf{C}_d\mathbf{x}(k) + D_d u(k) \qquad \mathbf{C}_d = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D_d = 0$$

- The observability matrix of the pair $(\mathbf{A}_d, \mathbf{C}_d)$, with $n = 2$, is

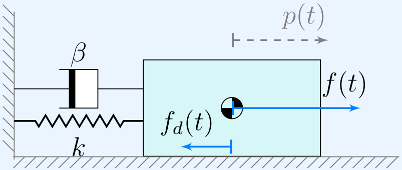
$$\mathbf{M}_o(2) = \begin{bmatrix} \mathbf{C}_d \\ \mathbf{C}_d\mathbf{A}_d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0.4 \end{bmatrix}$$

- As $\operatorname{rank} \mathbf{M}_o(2) = 2$ the system is observable.

Mass-spring-damper system

Problem (Exercise 1.4: Mass-spring-damper system)

Consider a mass-spring-damper system shown below, where the position and velocity of the cart are denoted by $p(t)$ *and* $v(t)$*, respectively.*



Model the system, obtain its linearized and discretized versions. Analyze the stability, controlability and observability of the system.

Summary 1.1

The main learning outcomes of this class are:

- Describe a physical system with continuous-time state-space model.
- Linearize a nonlinear continuous-time state-space model.
- Discretize a system to obtain a discrete-time state-space model.
- Compute the solution of an LTV or LTI discrete-time model.
- Analyze the stability of linear systems.
- Test a linear system for controlability and observability.

Modeling and Optimal control

Cyber-Physical Control Systems

B. Guerreiro

2021

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Course plan

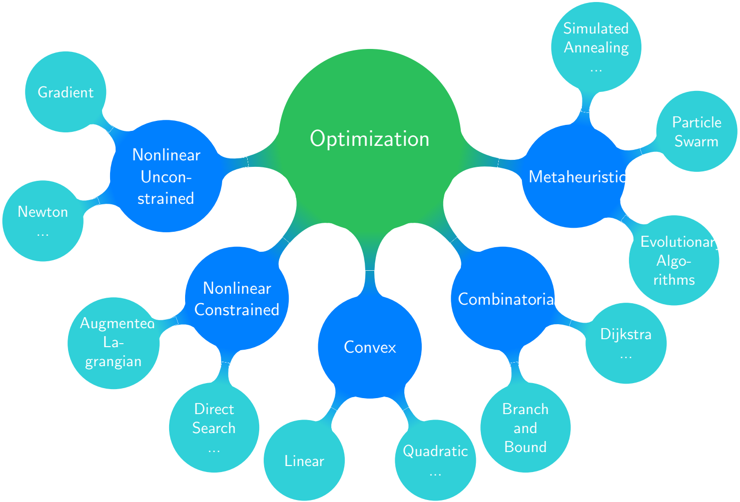
Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Course plan

Week	Subject	Assignment
1	M0 Course details and introduction	
2	M1.1 Discrete-time systems state model representation	P1 start
3	M1.2 Optimization and optimal control	HW1
4	M2.1 Introduction to model predictive control (MPC)	
5	M2.2 Constrained MPC design	HW2
6	M3.1 Nonlinear MPC design	
7	M3.2 Feasibility and Stability analysis of MPC design	HW3, P1 due
8	M4.1 Decentralized MPC design	P2 start
9	M4.2 Distributed MPC design	
10	M4.3 Networked control systems	HW4
11	Discussion and feedback of draft P2 paper	Draft P2 due
12	M5.1 Hybrid dynamic systems	
13	M5.2 MPC for hybrid dynamic systems	HW5
14	Final project 2 presentations and discussion	P2 due

Outline

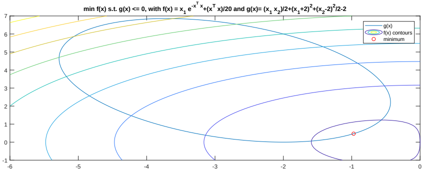
Optimization problems and methods



Mathematical optimization

- (mathematical) optimization problem in standard form:

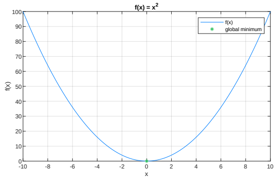
$$
\begin{array}{rl}
\text{minimize} & f(x) \\
\text{subject to} & g(x) \leq 0 \\
& h(x) = 0
\end{array}
$$

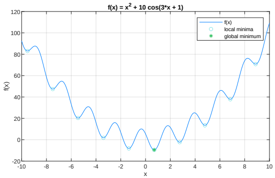


  - $x \in \mathbb{R}^n$ is the optimization (vector) variable
  - $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function
  - $g : \mathbb{R}^n \to \mathbb{R}^m$ are inequality constraint functions
  - $h : \mathbb{R}^n \to \mathbb{R}^l$ are equality constraint functions

Local and global solutions

- Local optimal solution:
  - x^* is a local minimum if it satisfies the constraints and is no worse than its neighbors, that is, there is $\epsilon > 0$ such that $f(x^*) \leq f(x)$ for all x such that $\|x - x^*\| < \epsilon$.
- Global optimal solution:
  - x^* is a global minimum if it satisfies the constraints and is no worse than all others, that is, $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.





The Taylor series

- Taylor series of a scalar function with scalar argument, $f : \mathbb{R} \to \mathbb{R}$:

$$f(x) = f(x_0) + \left.\frac{df}{dx}\right|_{x=a} (x - x_0) + \frac{1}{2} \left.\frac{d^2 f}{dx^2}\right|_{x=x_0} (x - x_0)^2 + O(3)$$

- Taylor series of a scalar function with vector argument, $f : \mathbb{R}^n \to \mathbb{R}$:

$$f(x) = f(a) + \left.\frac{df}{dx^T}\right|_{x=x_0} (x - x_0) + \frac{1}{2}(x - x_0)^T \left.\frac{d^2 f}{dx dx^T}\right|_{x=x_0} (x - x_0) + O(3)$$

- Taylor series at $x = x_0 + dx$:

$$f(x) = f(x_0) + \left.\frac{df}{dx^T}\right|_{x=x_0} dx + \frac{1}{2} dx^T \left.\frac{d^2 f}{dx dx^T}\right|_{x=x_0} dx + O(3)$$

- Taylor series for an infinitesimal increment of f:

$$df = f_x dx + dx^T f_{xx} dx + O(3)$$

The gradient and Hessian

- The gradient vector of a scalar function with vector arguments:

$$\frac{df}{dx} = \nabla f(x) = f_x^T == \begin{bmatrix} \frac{d}{dx_1} \\ \cdots \\ \frac{d}{dx_n} \end{bmatrix} f(x)$$

$$\frac{df}{dx^T} = \nabla^T f(x) = f_x == \begin{bmatrix} \frac{d}{dx_1} & \cdots & \frac{d}{dx_n} \end{bmatrix} f(x)$$

- The Hessian matrix of a scalar function with vector arguments:

$$\frac{d^2 f}{dx dx^T} = \mathbf{H}_f(x, x) = \nabla^2 f(x) = f_{xx} == \begin{bmatrix} \frac{d^2}{dx_1 dx_1} & \cdots & \frac{d^2}{dx_1 dx_n} \\ \cdots & \cdots & \cdots \\ \frac{d^2}{dx_n dx_1} & \cdots & \frac{d^2}{dx_n dx_n} \end{bmatrix} f(x)$$

Help in vector calculus (1)

- Some basic derivatives of scalar functions with vector arguments (consider a constant $a \in \mathbb{R}^n$ and vector functions $g(x), f(x) \in R^n$) [6]:

$$\frac{da^Tx}{dx} = a \qquad \frac{dg(x)^T}{dx} = \left(\frac{dg(x)}{dx^T}\right)^T$$

$$\frac{da^Tg(x)}{dx} = \frac{dg(x)^T}{dx}a$$

$$\frac{df(x)^Tg(x)}{dx} = f(x)^T\frac{dg(x)}{dx} + g(x)^T\frac{df(x)}{dx}$$

$$\frac{dg(x)^TAg(x)}{dx} = \frac{dg(x)^T}{dx}(A + A^T)g(x)$$

Help in vector calculus (2)

- Some basic derivatives of scalar functions with vector arguments (consider a constant $a \in \mathbb{R}^n$ and vector functions $g(x), f(x) \in R^n$) [6]:

$$\frac{da^T x}{dx^T} = a^T$$

$$\frac{da^T g(x)}{dx^T} = a^T \frac{dg(x)}{dx^T}$$

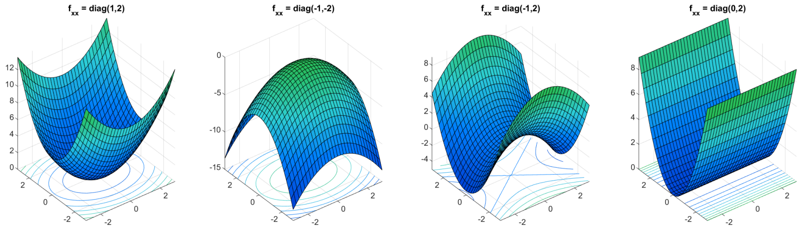
$$\frac{df(x)^T g(x)}{dx^T} = f(x)^T \frac{dg(x)}{dx^T} + g(x)^T \frac{df(x)}{dx^T}$$

$$\frac{dg(x)^T A g(x)}{dx^T} = g(x)^T (A + A^T) \frac{dg(x)}{dx^T}$$

$$\frac{dg(x)}{dx^T} = \left(\frac{dg(x)^T}{dx} \right)^T$$

Critical points of a function

- A critical point of a function $f(x)$ implies that $\nabla f(x) = 0$, and:
  - is a local minimum if $\mathbf{H}_f(x,x)$ is positive definite, $\mathbf{H}_f(x,x) > 0$;
  - is a local maximum if $\mathbf{H}_f(x,x)$ is negative definite, $\mathbf{H}_f(x,x) < 0$;
  - is a saddle point if $\mathbf{H}_f(x,x)$ is indefinite;
  - need further characterization if $\mathbf{H}_f(x,x)$ is semidefinite.



Unconstrained Example 1

Problem (Critical points of functions (Exercise 1.5))

Find the critical points of the following functions and characterize them in terms of their second order derivative:

$$f_1(x) = 9 - 2x_1 + 4x_2 - x_1^2 - 4x_2^2$$
$$f_2(x) = 2x_1^3 + x_1 x_2^2 + 5x_1^2 + x_2^2$$

Example (Critical points of functions - part 1)

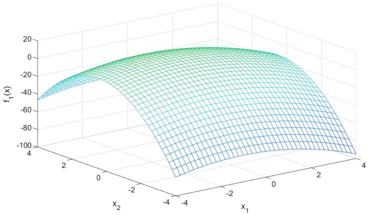
Analyzing the gradient and Hessian of $f_1(x)$, we have that

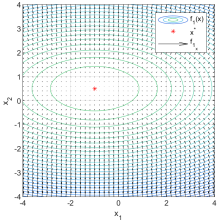
$$\nabla f_1(x) = \begin{bmatrix} -2 - 2x_1 \\ 4 - 8x_2 \end{bmatrix} \qquad \mathbf{H}_{f_1}(x, x) = \begin{bmatrix} -2 & 0 \\ 0 & -8 \end{bmatrix}$$

Unconstrained Example 1

Example (Critical points of functions - part 1)

From the gradient we see that there is a critical point at $\nabla f(x) = 0$ given by $x = [-1\ \frac{1}{2}]^T$. Looking at the Hessian matrix, we can easily see that it is negative definite, which implies that the critical point is a global maximum of the function.





Unconstrained Example 1

Example (Critical points of functions - part 2)

Analyzing the gradient and Hessian of $f_2(x)$, we have that

$$\nabla f_2(x) = \begin{bmatrix} 6x_1^2 + 10x_1 + x_2^2 \\ 2x_1x_2 + 2x_2 \end{bmatrix} \qquad \mathbf{H}_{f_2}(x,x) = \begin{bmatrix} 12x_1 + 10 & 2x_2 \\ 2x_2 & 2x_1 + 2 \end{bmatrix}$$

from where we see that there are 4 critical points at $\nabla f(x) = 0$ given by $x = [0\ 0]^T$, $x = [-\frac{5}{3}\ 0]^T$, $x = [-1\ 2]^T$, and $x = [-1\ -2]^T$. To characterize all these critical points we have to evaluate the Hessian at each of them, yielding

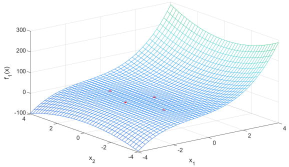
$$\mathbf{H}_{f_2}(0,0) = \begin{bmatrix} 10 & 0 \\ 0 & 2 \end{bmatrix} \qquad \mathbf{H}_{f_2}(-\frac{5}{3},0) = \begin{bmatrix} -10 & 0 \\ 0 & -4/3 \end{bmatrix}$$

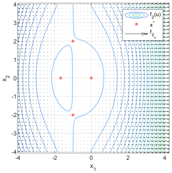
$$\mathbf{H}_{f_2}(-1,2) = \begin{bmatrix} -2 & 4 \\ 4 & 0 \end{bmatrix} \qquad \mathbf{H}_{f_2}(-1,-2) = \begin{bmatrix} -2 & -4 \\ -4 & 0 \end{bmatrix}$$

Unconstrained Example 1

Example (Critical points of functions - part 2)

A matrix is positive definite or negative definite iff its eigenvalues are, respectively, all positive or all negative. This means that the point $x = [0\ 0]^T$ is a local minimum as $H_{f_2}(0,0) > 0$, the point $x = [-\frac{5}{3}\ 0]^T$ is a local maximum as $H_{f_2}(-\frac{5}{3}, 0) < 0$, while the remaining to critical points are saddle points, as the eigen values of their Hessians have both positive and negative eigenvalues. An indefinite point would have at least one zero eigenvalue.





Unconstrained Optimization

- Unconstrained optimization problem:

$$\text{minimize} \quad f(x)$$

  - $x \in \mathbb{R}^n$ is the optimization (vector) variable
  - $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function

- Optimal solution x^*:
  - the objective function f attains the smallest value at x^*, among all the values of x.

- Optimality conditions:
  - $\nabla f(x) = 0$: zero gradient
  - $\mathbf{H}_f(x, x) > 0$: positive definite Hessian matrix

Gradient methods

Unconstrained optimization

- The steepest descent algorithm:
  1. Select initial value for x
  2. Determine the steepest descent direction, $\Delta x = -\nabla f(x)$
  3. Choose the step size α using a line search method.
  4. Update de control vector by $x := x + \alpha \Delta x$
  5. Determine the change in the cost function, $\Delta f = \alpha \nabla f(x)^T \Delta x$. If it is sufficiently small, stop, otherwise go to step 2.
- The Newton algorithm:
  - Same steps as the steepest descent method.
  - Use $\Delta x = -\mathbf{H}_f(x, x)^{-1} \nabla f(x)$ for direction.
  - And $\Delta f = \alpha \nabla f(x)^T \Delta x$ for stopping.

Unconstrained Example 2

Problem (Minimizing on quadratic surfaces (Exercise 1.6))

Find the optimal value of the optimization variable $u \in \mathbb{R}^2$, for the optimization problem

$$\min_u \quad q(u) := \frac{1}{2}u^T Q u + s^T u$$

where $Q \in \mathbb{R}^{2\times 2}$ is symmetric constant matrix and $s \in \mathbb{R}^2$ is a constant vector.

Example (Minimizing on quadratic surfaces - part 1)

Analyzing the gradient and Hessian of $q(u)$, we have that

$$q_u^T = \frac{dq}{du} = Qu + s \qquad\qquad q_{uu} = \frac{d^2q}{dudu} = Q\ .$$

Unconstrained Example 2

Example (Minimizing on quadratic surfaces - part 2)

The optimality conditions state that for the optimal value, u^*, to be a minimizing critical point of the function $q(u)$, the gradient must be zero and the Hessian must be positive definite, yielding

$$u^* = -Q^{-1}s \,, \quad Q > 0 \,.$$

The optimal cost function is then

$$q^* := q(u^*) = \frac{1}{2}s^T Q^{-T} Q Q^{-1} s - s^T Q^{-1} s = -\frac{1}{2} s^T Q^{-1} s \,.$$

Unconstrained Example 2

Example (Minimizing on quadratic surfaces - part 3)

Considering $Q = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$, $s = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and noting that $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$, the cost function can also be written as

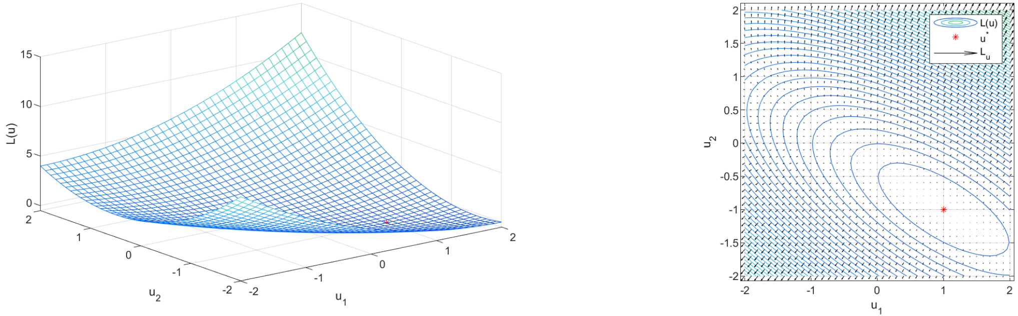
$$q(u) = \frac{1}{2}(u_1^2 + 2u_2^2 + 2u_1u_2) + u_2$$

In this case, the gradient and Hessian are

$$q_u^T = Qu + s = \begin{bmatrix} u_1 + u_2 \\ u_1 + 2u_2 + 1 \end{bmatrix} \qquad q_{uu} = Q = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} .$$

which yields the optimal input, $u^* = [1 \ \ -1]^T$ and cost value $q^* = -0.5$.

Unconstrained Example 2



Outline

State-space models

Properties of discrete-time systems

Unconstrained optimization

Constrained optimization

Optimal Control

Equality constrained Optimization

- Equality constrained optimization problem:

$$\min_u \quad q(x,u)$$
$$\text{s.t.} \quad f(x,u) = 0$$

  - $x \in \mathbb{R}^n$ is the optimization (control) variable
  - $u \in \mathbb{R}^m$ is the auxiliary (state) variable
  - $q : \mathbb{R}^{n\times m} \to \mathbb{R}$ is the objective function
  - $f : \mathbb{R}^{n\times m} \to \mathbb{R}^n$ is the equality constraint function

- Optimal solution u^*:
  - the objective function q attains the smallest value at u^*, among all the values of u that satisfy the constraint $f(x,u) = 0$.

- Optimality conditions...??

The Hamiltonian

- **Idea:** what if we include the equality constraint in the cost function?
- **Simplest method:** use the internal product of some nonzero vector $\lambda \in \mathbb{R}^n$ by the equality constraint function
- Resulting unconstrained optimization problem:

$$\min_u \quad q(x,u) + \lambda^T f(x,u)$$

- Define the new cost functional – the Hamiltonian:

$$H(x,u) = q(x,u) + \lambda^T f(x,u)$$

- First-order Taylor series for increments of H:

$$dH = H_u du + H_x dx + H_\lambda d\lambda + O(2)$$

- Gradients of the Hamiltonian:

$$H_u = q_u + \lambda^T f_u \qquad H_x = q_x + \lambda^T f_x \qquad H_\lambda = f(x,u)^T$$

Optimality conditions

- Necessary conditions (gradients must be zero at the optimal point):

$$H_u = q_u + \lambda^T f_u = 0 \qquad H_x = q_x + \lambda^T f_x = 0 \qquad H_\lambda = f(x,u)^T = 0$$

from where we obtain the conditions

$$q_u + \lambda^T f_u = 0 \qquad q_x + \lambda^T f_x = 0 \qquad f(x,u) = 0$$

- Sufficient conditions are obtained noting that

$$[1\ \lambda^T] \begin{bmatrix} dq \\ df \end{bmatrix} = [H_x^T\ H_u^T] \begin{bmatrix} dx \\ du \end{bmatrix} + \frac{1}{2}[dx^T\ du^T] \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} dx \\ du \end{bmatrix} + O(3)$$

and that at the critical point $df = 0$, yielding $dx = -f_x^{-1} f_u du$, and therefore, to have a minimizing critical point

$$q_{uu}^f := du^T [-f_u^T f_x^{-T}\ 1] \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} -f_x^{-1} f_u \\ 1 \end{bmatrix} du > 0$$

Constrained example 1

Problem (Quadratic surface with linear constraint (Exercise 1.10))

Find the optimal value of the optimization variable $u \in \mathbb{R}^2$, for the optimization problem

$$\begin{aligned} \min \quad & q(x,u) := \frac{1}{2}[x\ u]Q\begin{bmatrix} x \\ u \end{bmatrix} + s^T\begin{bmatrix} x \\ u \end{bmatrix} \\ s.t. \quad & f(x,u) = 0 \end{aligned}$$

where $Q = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$, $s = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ *and* $f(x,u) = x - u - 3$.

Constrained example 1

Example (Quadratic surface with linear constraint - part 1)

Considering the Lagrange multiplier $\lambda \in \mathbb{R}$, define the Hamiltonian

$$H(x,u) = q(x,u) + \lambda f(x,u) = (1/2\, x^2 + xu + u^2 + u) + \lambda(x - u - 3)$$

Find the first order conditions of optimality

$$\begin{aligned}
H_\lambda &= f(x,u) = 0 && \Leftrightarrow x - u - 3 = 0\\
H_x &= q_x + \lambda f_x = 0 && \Leftrightarrow x + u + \lambda = 0\\
H_u &= q_u + \lambda f_u = 0 && \Leftrightarrow x + 2u + 1 - \lambda = 0
\end{aligned}$$

Constrained example 1

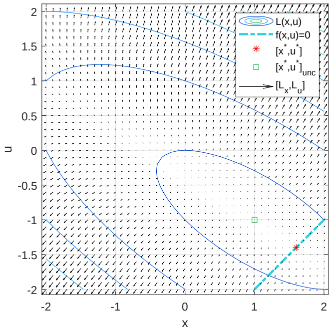
Example (Quadratic surface with linear constraint - part 2)

Solving for x, λ and u yields a linear system

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ u \\ \lambda \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ -1 \end{bmatrix}$$

for which the solution (optimal) is

$$\begin{bmatrix} x^* \\ u^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} 1.6 \\ -1.4 \\ -0.2 \end{bmatrix}$$



General Constrained Optimization

- General constrained optimization problem:

$$
\begin{aligned}
\min \quad & f(x) \\
\text{subject to} \quad & g(x) \leq 0 \\
& h(x) = 0
\end{aligned}
$$

  - $x \in \mathbb{R}^n$ is the optimization (vector) variable
  - $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function
  - $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are inequality constraint functions
  - $h : \mathbb{R}^n \rightarrow \mathbb{R}^l$ are equality constraint functions

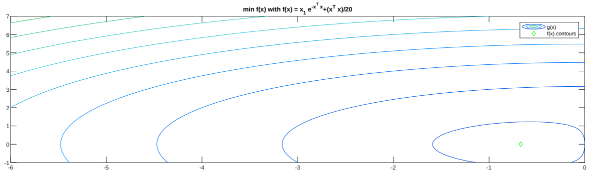
- Lagrangian method of multipliers defines a new cost function:

$$L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x)$$

- Other methods: Augmented Lagrangian method of multipliers; Alternating direction method of multipliers, etc.

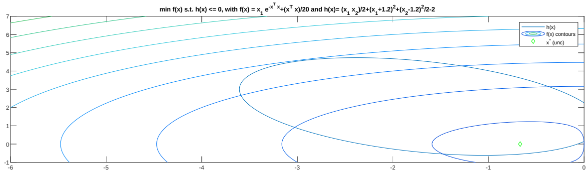
Solving inequality constraints

- How to deal with inequality constraints?
  - Check which inequalities are active: being violated in the unconstrained problem.
  - Consider them as equality constraints.
  - Proceed as before for equality constraints



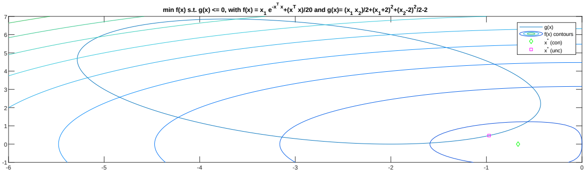
Solving inequality constraints

- How to deal with inequality constraints?
  - Check which inequalities are active: being violated in the unconstrained problem.
  - Consider them as equality constraints.
  - Proceed as before for equality constraints



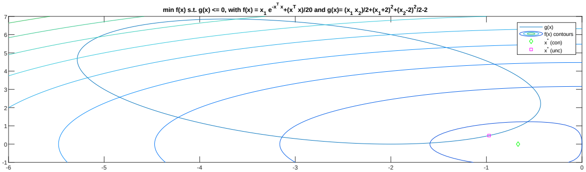
Solving inequality constraints

- How to deal with inequality constraints?
  - Check which inequalities are active: being violated in the unconstrained problem.
  - Consider them as equality constraints.
  - Proceed as before for equality constraints



Solving inequality constraints

- How to deal with inequality constraints?
  - Check which inequalities are active: being violated in the unconstrained problem.
  - Consider them as equality constraints.
  - Proceed as before for equality constraints



KKT conditions of optimality

Theorem (Karush-Kuhn-Tucker sufficient conditions)

Consider the constrained optimization problem defined above where f, g, and h are twice differentiable, and that x^*, λ^* and μ^* are such that they satisfy the problem constraints and

$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0$$
$$y^T \mathbf{H}_L(x^*, \lambda^*, \mu^*) y > 0$$

with $\mu^* = 0$ or $\mu* \geq 0$ for the inactive or active constraints, respectively, for all $y \neq 0$, such that $\nabla h(x^*)^T y = 0$ and $\nabla g(x^*)^T y = 0$. Then, x^* is a strict local minimum of the problem.

Outline

State-space models

Properties of discrete-time systems

Unconstrained optimization

Constrained optimization

Optimal Control

Linear state-space model

- Linear discrete-time dynamic system:

$$x(k+1) = Ax(k) + Bu(k), \quad x(0) = r_0$$

- Simplified notation:

$$x_{k+1} = Ax_k + Bu_k, \quad x_0 = r_0$$

  - $x_k \in \mathbb{R}^n$ is the state variable
  - $u_k \in \mathbb{R}^m$ is the control variable
  - $A \in \mathbb{R}^{n \times n}$ and $B_k \in \mathbb{R}^{n \times m}$ are the system matrices
  - Initial condition, $x_0 = r_0$.

The cost functional

- LQ performance index:

$$J_i := J_{i\to N}(x_0, u) = p(x_N) + \sum_{k=i}^{N-1} q(x_k, u_k)$$

  - time interval: $\{i, \ldots, N\}$
  - final time cost function:

$$p(x_N) = \frac{1}{2} x_N^T P x_N$$

  - intermediate cost function at time k:

$$q(x_k, u_k) = \frac{1}{2} x_k^T Q x_k + \frac{1}{2} u_k^T R u_k$$

  - Assume $P = P^T \geq 0$, $Q = Q^T \geq 0$, and $R = R^T > 0$.

Problem definition

Problem (LQR optimal control problem)

Find the optimal control sequence u_k^* *for* $k \in \{0, \ldots, N-1\}$ *that drives the system along a state trajectory* x_k^* *for* $k \in \{0, \ldots, N\}$ *according to the linear system dynamics, such that the specified performance index* J_0 *is minimized. That is,*

$$\begin{aligned} \min_{u_0,\ldots,u_{N-1}} \quad & J_0 = \frac{1}{2} x_N^T P x_N + \frac{1}{2} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) \\ s.t. \quad & x_{k+1} = A x_k + B u_k \\ & x_0 = r_0 \end{aligned}$$

Reformulating the optimization problem

- **Idea:** what if we include the equality constraint in the cost function?
- **Simple method:** for each k, use the inner product of the Lagrange multiplier $\lambda_k \in \mathbb{R}^n$ by the respective equality constraint.
- Resulting unconstrained optimization problem:

$$\min_u \quad J_0' := p(x_N) + \sum_{k=0}^{N-1} [q(x_k, u_k) + \lambda_{k+1}^T (Ax_k + Bu_k - x_{k+1})]$$

- Define the Hamiltonian:

$$\begin{aligned} H(x_k, u_k) &= q(x_k, u_k) + \lambda_{k+1}^T (Ax_k + Bu_k) \\ &= \frac{1}{2} x_k^T Q x_k + \frac{1}{2} u_k^T R u_k + \lambda_{k+1}^T (Ax_k + Bu_k) \end{aligned}$$

Reformulating the cost functional

- New cost functional

$$J_0' = p(x_N) + \sum_{k=0}^{N-1} [H(x_k, u_k) - \lambda_{k+1}^T x_{k+1}]$$

$$= p(x_N) - \lambda_N^T x_N + \sum_{k=1}^{N-1} [H(x_k, u_k) - \lambda_k^T x_k] + H(x_0, u_0)$$

- First-order approximation for increments of J_0':

$$dJ_0' = (p_{x_N} - \lambda_N^T) dx_N + H_{u_0}^T du_0$$
$$+ \sum_{k=1}^{N-1} [(H_{x_k} - \lambda_k^T) dx_k + H_{u_k}^T du_k] + \sum_{k=1}^{N} (H_{\lambda_k} - x_k^T) d\lambda_k$$

Optimality conditions

- Relevant derivatives:

$$H_{x_k} := \frac{dH(x_k, u_k)}{dx_k^T} = q_{x_k} + \lambda_{k+1}^T f_{x_k} \qquad p_{x_N} := \frac{dp(x_N)}{dx_N^T}$$

$$H_{u_k} := \frac{dH(x_k, u_k)}{du_k^T} = q_{u_k} + \lambda_{k+1}^T f_{u_k} \qquad H_{\lambda_{k+1}} := \frac{dH(x_k, u_k)}{d\lambda_{k+1}^T}$$

- Necessary conditions of optimality ($dJ_0' = 0$):

$$(H_{\lambda_{k+1}} - x_{k+1}^T)d\lambda_{k+1} = 0 \ , \ \forall_{k=0,\ldots,N-1}$$

$$(H_{x_k} - \lambda_k^T)dx_k = 0 \ , \ \forall_{k=1,\ldots,N-1} \qquad (p_{x_N} - \lambda_N^T)dx_N = 0$$

$$H_{u_k} du_k = 0 \ , \ \forall_{k=0,\ldots,N-1} \qquad H_{x_0} dx_0 = 0$$

Optimal controller

- State equation, obtained from $(H_{\lambda_{k+1}} - x_{k+1}^T)d\lambda_{k+1} = 0$:

$$x_{k+1} = H_{\lambda_{k+1}}^T = Ax_k + Bu_k$$

- Co-state equation, obtained from $(H_{x_k} - \lambda_k^T)dx_k = 0$:

$$\lambda_k = H_{x_k}^T = Qx_k + A^T\lambda_{k+1}$$

- Stationary condition, obtained from $H_{u_k}du_k = 0$:

$$0 = H_{u_k} = Ru_k + B^T\lambda_{k+1} \qquad u_k = -R^{-1}B^T\lambda_{k+1}$$

- Boundary conditions:

$$(P_N x_N - \lambda_N)^T dx_N = 0 \qquad (Q_0 x_0)^T dx_0 = 0$$

Finding the closed-form solution

- The recursive form of the optimal controller is

$$\begin{aligned} u_k &= -R^{-1}B^T\lambda_{k+1} \\ x_{k+1} &= Ax_k - BR^{-1}B^T\lambda_{k+1} \\ \lambda_k &= Qx_k + A^T\lambda_{k+1} \quad , \ \lambda_N = Px_N \end{aligned}$$

- Assume that we can also find P_k such that $\lambda_k = P_k x_k$.
- Then, the state equation is given by

$$x_{k+1} = Ax_k - BR^{-1}B^TP_{k+1}x_{k+1} = (I + BR^{-1}B^TP_{k+1})^{-1}Ax_k$$

- while the control law becomes

$$u_k = -R^{-1}B^TP_{k+1}x_{k+1}$$

- **Missing:** how to compute P_k for all $k = 0, \ldots, N$.

The discrete-time Riccati equation

- Use $\lambda_k = P_k x_k$ the costate equation:

$$P_k x_k = Q x_k + A^T P_{k+1} x_{k+1}$$
$$P_k x_k = Q x_k + A^T P_{k+1} (I + B R^{-1} B^T P_{k+1})^{-1} A x_k$$

yielding the matrix equation

$$\begin{aligned} P_k &= A^T P_{k+1} (I + B R^{-1} B^T P_{k+1})^{-1} A + Q \\ &= A^T (P_{k+1}^{-1} + B R^{-1} B^T)^{-1} A + Q \end{aligned}$$

- The discrete-time Riccati equation (DRE) is

$$P_k = A^T (P_{k+1} - P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1}) A + Q$$

Jacopo Francesco Riccati (1676-1754)

- Born in Venice, Italy
- Doctor in Law, Università degli Studi di Padova
- Studied and teached mathematics by himself
- Contributions in quadratic differential equations:
  - Given an equation of the form

    $$\dot{y}(t) = a_0(t) + a_1(t)y(t) + a_2(t)y^2(t)$$

  - Change of variable: $z(t) = q_2(t)y(t)$
  - Showed how to reduce it into an ODE

    $$\ddot{z}(t) - R\dot{z}(t) + Sz(t) = 0$$

    for which a solution is easy to find.
- Other contributions in Law, Philosophy, and Poetry


;

The closed-form LQR controller

- Replacing the costate expression in the controller yields

$$u_k = -R^{-1}B^T P_{k+1} x_{k+1} = -R^{-1}B^T P_{k+1}(Ax_k + Bu_k)$$

$$u_k = -(B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A x$$

- Defining the (Kalman) gain:

$$K_k = (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A$$

- The optimal LQR control law is simply

$$u_k^* = -K_k x_k$$

- While the closed-loop system and Riccati equation become

$$x_{k+1} = (A - BK_k)x_k$$

$$P_k = A^T P_{k+1}(A - BK_k) + Q$$

LQR optimal control example 1

Problem (Exercise 1.12 – Computing a control sequence)

Consider the time-invariant discrete linear system described by

$$x(k+1) = 0.5x(k) + 2u(k)$$

with $x(0) = 1$, *and the cost functional is*

$$J_0 = 4x^2(3) + \frac{1}{2}\sum_{k=0}^{2}[x^2(k) + 0.5u^2(k)] \ .$$

Compute the optimal control sequence $u^*(k)$ *for all* $k \in \{0, 1, 2\}$.

LQR optimal control example 1

Exercise 1.12 – Computing a control sequence - part 1

The optimal control law is $u_k = -K_k x_k$, so lets compute the optimal gain and the Riccati equation.
Noting that we have $P_N = 8$, $Q = 1$, $R = 0.5$, $A = 0.5$, $B = 2$, and $N = 3$, the Riccati equation is given by

$$P_k = A^T(P_{k+1} - P_{k+1}B(B^T P_{k+1} B + R)^{-1} B^T P_{k+1})A + Q$$

$$P_k = A^2 P_{k+1} - \frac{A^2 B^2 P_{k+1}^2}{B^2 P_{k+1} + R} + Q = \frac{A^2 B^2 P_{k+1}^2 + RA^2 P_{k+1} - A^2 B^2 P_{k+1}^2}{B^2 P_{k+1} + R} + Q$$

$$= \frac{RA^2 P_{k+1}}{B^2 P_{k+1} + R} + Q = \frac{0.5\, 0.5^2 P_{k+1}}{2^2 P_{k+1} + 0.5} + 1 = \frac{1}{8} \frac{P_{k+1}}{4P_{k+1} + 0.5} + 1$$

LQR optimal control example 1

Exercise 1.12 – Computing a control sequence - part 2

The recursion on P_k yields

$$P_N = P_3 = 8$$

$$P_2 = \frac{1}{8}\frac{P_3}{4P_3 + 0.5} + 1 = \frac{1}{8}\frac{8}{32 + 0.5} + 1 = \frac{1}{32.5} + 1 = \frac{33.5}{32.5} = 1.030769$$

$$P_1 = \frac{1}{8}\frac{P_2}{4P_2 + 0.5} + 1 = 1.027870$$

$$P_0 = \frac{1}{8}\frac{P_1}{4P_1 + 0.5} + 1 = 1.027861$$

LQR optimal control example 1

Exercise 1.12 – Computing a control sequence - part 3

Having the values for each $P(k)$ the Kalman gain is

$$K_k = (B_k^T P_{k+1} B_k + R_k)^{-1} B_k^T P_{k+1} A_k = \frac{a\, b\, p_{k+1}}{b^2 p_{k+1} + r} = \frac{p_{k+1}}{4p_{k+1} + 1}$$

which yields

$$K_2 = \frac{p_3}{4p_3 + 1} = \frac{8}{33} = 0.246153$$

$$K_1 = \frac{p_2}{4p_2 + 1} = 0.222961$$

$$K_0 = \frac{p_1}{4p_1 + 1} = 0.222893$$

LQR optimal control example 1

Exercise 1.12 – Computing a control sequence - part 4

We can now find the optimal control and state sequence using

$$x_{k+1} = (a - bK_k)x_k = (0.5 - 2K_k)x_k$$

which recursively yields

$$\begin{aligned}
x_0 &= 1\\
x_1 &= (0.5 - 2K_0)x_0 = (0.5 - 2 \times 0.222893) \times 1 = 0.054214\\
x_2 &= (0.5 - 2K_1)x_1 = (0.5 - 2 \times 0.222961) \times 0.054214 = 0.002931\\
x_3 &= (0.5 - 2K_2)x_2 = (0.5 - 2 \times 0.246153) \times 0.002931 = 2.25571 \times 10^{-5}
\end{aligned}$$

LQR optimal control example 1

Exercise 1.12 – Computing a control sequence - part 5

whereas the optimal control sequence is

$$
\begin{aligned}
u_0 &= -K_0 x_0 = -0.222893 \times 1 = -0.222893 \\
u_1 &= -K_1 x_1 = -0.222961 \times 0.054214 = -0.012087 \\
u_2 &= -K_2 x_2 = -0.246153 \times 0.002931 = -7.21474 \times 10^{-4}
\end{aligned}
$$

Unconstrained Example 3

Problem (Exercise 1.7 – Minimizing on general surfaces)

Find the optimal value of the optimization variable $u \in \mathbb{R}^2$, for the optimization problem

$$\min_{u} \quad q(u) := 100(u_2 - u_1^2)^2 + (1 - u_1)^2 \,.$$

Constrained example 2

Problem (Exercise 1.11 – Quadratic index with linear constraint)

Find the optimal value of the optimization variable $u \in \mathbb{R}^2$, for the optimization problem

$$\begin{aligned} \min \quad & q(x,u) := \frac{1}{2}x^T Q x + \frac{1}{2}u^T R u \\ s.t. \quad & f(x,u) = 0 \end{aligned}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $f \in \mathbb{R}^n$, the constraint function is

$$f(x,u) = Ax + Bu + c$$

while $Q > 0$, $R > 0$, A and B are constant matrices with appropriate dimensions and $c \in \mathbb{R}^n$ is a constant vector.

LQR example 2

Problem (Exercise 1.13 – Computing a control sequence)

Consider the time-invariant discrete linear system described by

$$x(k+1) = 0.5x(k) + 3u(k)$$

and the cost functional

$$J_0 = 6x^2(4) + \frac{1}{2}\sum_{k=0}^{3}[2x^2(k) + 3u^2(k)] \, .$$

Assuming $x(0) = 1$*, compute the optimal control sequence* $u^*(k)$ *for all* $k \in \{0, 1, 2, 3\}$.

Summary 1.2

The main learning outcomes of this class are:

- Define an optimization problem (OP)
- Classify OPs relative to basic characteristics
- Compute optimality conditions for unconstrained and constrained OPs
- Obtain closed-form or iterative solutions for OPs
- Define an LQR optimal control problem (OCP)
- Obtain the solution of the LQR OCP

Bibliography

João P. Hespanha.
Linear System Theory
Princeton, 2nd Edition, 2018.

Rugh and Kailath
Linear System Theory
Pearson, 2nd Edition, 1995.

Aström, Murray.
Feedback Systems
Princeton, 3rd Ed., 2019.

Jorge Nocedal and Stephen Wright
Numerical Optimization
Springer, 2006.

Frank L. Lewis, Draguna Vrabie, and Vassilis L. Syrmos.
Optimal control (Chapter 2)
John Wiley & Sons, 3rd edition, 2012.

Helmut Lütkepohl
Handbook of matrices
Wiley, 1996.

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Modeling and Optimal control

Cyber-Physical Control Systems

B. Guerreiro

2021