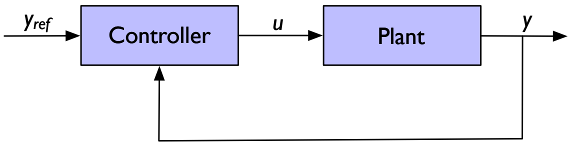
Lecture 1

Introduction to Networked Control Systems

Giancarlo Ferrari Trecate[1]

[1]Dependable Control and Decision Group
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
giancarlo.ferraritrecate@epfl.ch

Classic feedback control



y_{ref} : setpoint

u : input

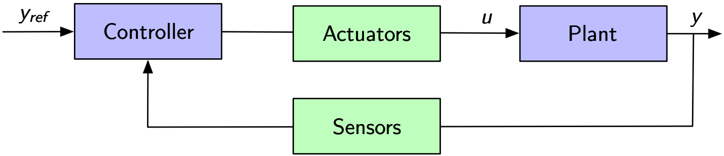
y : output

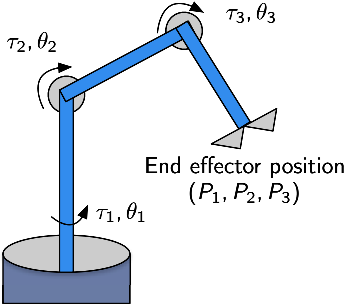
The block diagram

- Summarizes relations between variables
- Abstracts away from details

? How variables are measured/transmitted/generated

Classic feedback control - with sensor/actuators



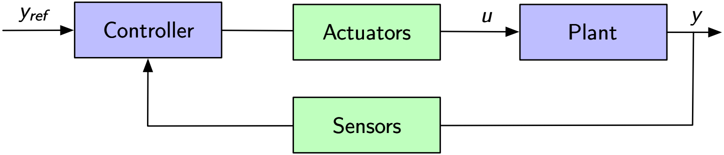


Example: robotic arm

Make $y(t) = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}(t)$ track $y_{ref}(t) = \begin{bmatrix} P_1^r \\ P_2^r \\ P_3^r \end{bmatrix}(t)$

- Sensors: encoders $\rightarrow \theta_1, \theta_2, \theta_3$
- Actuators: electric motors $\rightarrow \tau_1, \tau_2, \tau_3$
- Microcontroller

Classic feedback control - with sensor/actuators

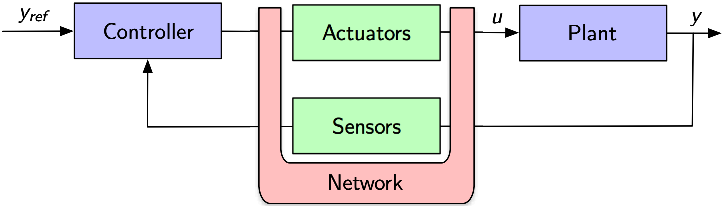


Remarks

- Variable measurements depend on the specific plants and technology of controller, sensors, and actuators
- Standard control technologies: microcontrollers, control stations, etc.
  - Receive/send electric signals!

? How variables are transmitted between devices

Abstract view: Networked Control System (NCS)



Multipurpose shared network

- Motivated by progresses in communication networks (computers, wireless, etc.) over the last 20 years

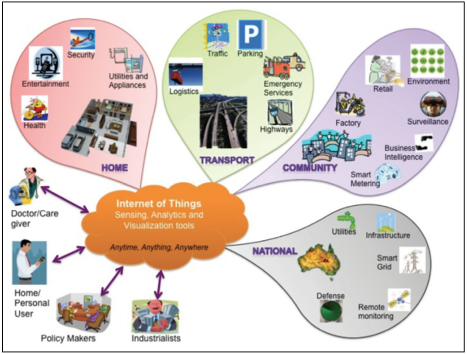
Goals of the course

Analyze

- opportunities offered by NCSs
- challenges: how the network non-idealities impact the system behavior

Motivations for NCSs

Motivations for NCSs: the Internet of Things



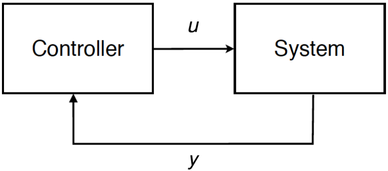
[Gubbi et al. '13]

Also known as CyberPhysical Systems (CPSs), Industry 4.0, Industrial Internet,...

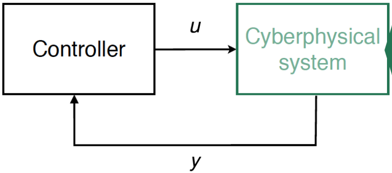
Ubiquitous sensing and actuation

- Fueled by wireless sensor networks, MEMS, cloud computing

Implication for modeling and control



Implication for modeling and control



Services through networks of interactions

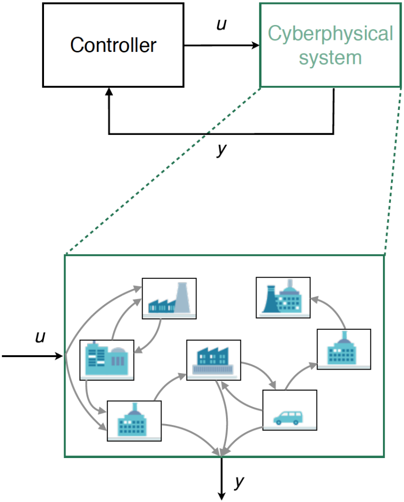
swarm of robots

traffic control

green districts

smart grids

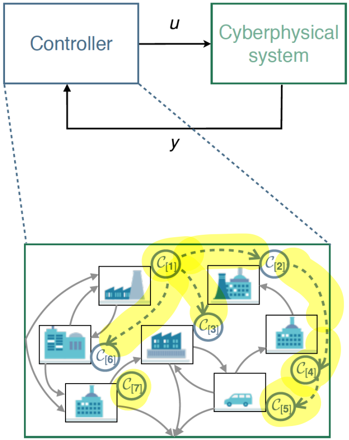
Implication for modeling and control



Modeling

- Multiple coupled subsystems
- Spatially distributed

Implication for modeling and control



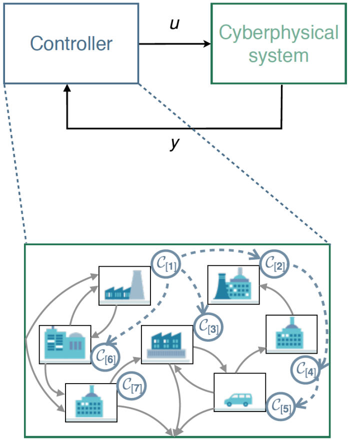
Modeling

- Multiple coupled subsystems
- Spatially distributed

Control architecture

- Seldom centralized
- Most likely distributed

Implication for modeling and control



Modeling

- Multiple coupled subsystems
- Spatially distributed

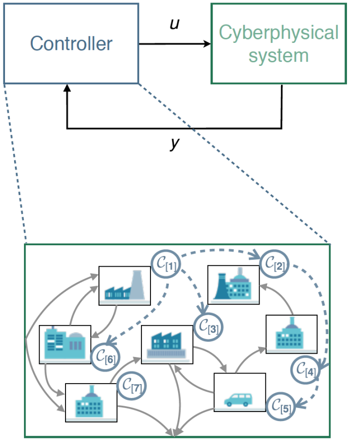
Control architecture

- Seldom centralized
- Most likely distributed

Communication

- Subsystems $\leftrightarrow$ controller(s)
- Between controllers

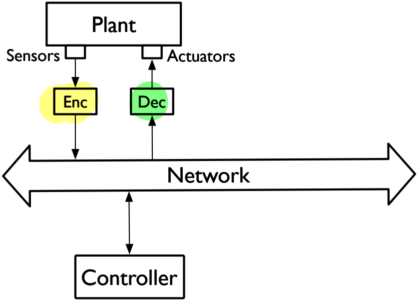
Implication for modeling and control



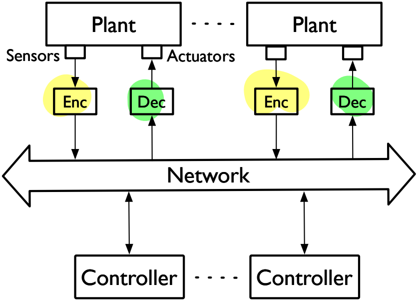
Control networks

- are fundamental for CPSs !
- allow for flexible architectures
- reduce installation and maintenance cost, compared to point-to-point links

Abstract view: Networked Control Systems (NCS)



Centralized control



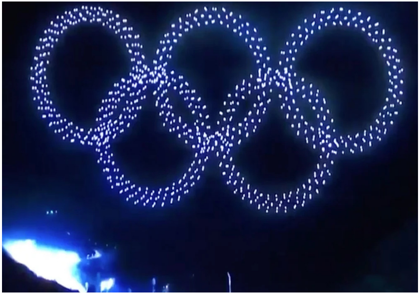
Decentralized/distributed control

Digital networks call for

- Encoders: when to sample continuous-time signals, what to send
- Decoders: map symbols into continuous-time signals

Opportunities offered by NCSs

Opportunities: coordination among agents



Drone show at the 2020 ~~olympic~~ olympic games

Swarm of mobile robots

Wishes

- Partial communication (limited transmission power)
- Distributed control
- Self-organizing for performing tasks

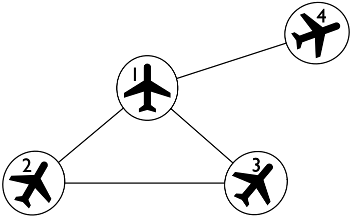
Toy example

Set of N moving agents

- Dynamics of agent i:

$$\dot{v}_i = u_i$$

- Velocity : $v_i(t) \in \mathbb{R}^2$
  Control input : $u_i(t) \in \mathbb{R}^2$



Communication network

- Graph with agents as nodes and communication links as edges
- Neighboring relation: $i \sim j$. Meaning: v_j is available to agent v_i.
  Partial communication $\Leftrightarrow$ the graph is not complete

Coordination goal

- **Alignment:** velocity becomes the same for all agents asymptotically

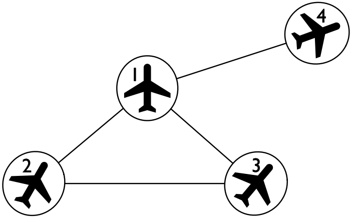
? How to compute the input u_i such that alignment is achieved

Laplacian control

Control law

Consider the input

$$u_i = \sum_{j \sim i} (v_j - v_i)$$



- Networked control law
- **Alignment** is achieved, independently of the number of agents (we will provide a formal proof in the course!)
- The basis for many other coordination algorithms

Simulation example

Alignment of agent velocities with time

Coordination in nature

Social behavior: creatures cluster in large moving formations

School of fish

Swarm of flying birds

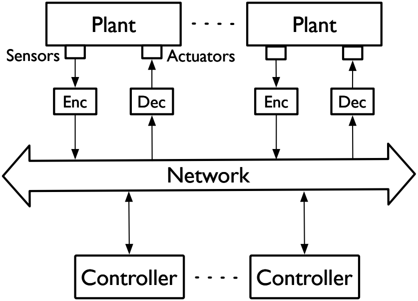
- Partial communication
- No centralized control
- Global emergent behavior

Challenges of NCSs

Challenges

Network nonidealities:

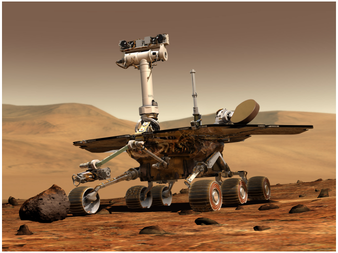
- Band-limited channels
- Sampling and delay
- Packet dropout

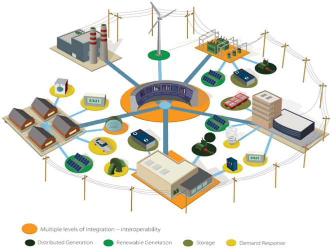


Band-limited channels

Any communication network can only carry a finite amount of information per unit time. Significant constraint in several applications ,e.g.,

- power-starved vehicles such as planetary rovers
- long-endurance, energy-limited systems, e.g. sensor networks





Impact on

- Stability of the closed-loop system
- Performance

Band-limited channels

Packet networks

Header (H)	Payload (P)

Example: protocols

- Asynchronous transfer Mode — H:40 bits P:384 bits
- Ethernet — H:22 bytes P:46-1500 bytes
- Bluetooth — H:126 bits P:2744 bits

Band-limited channels

Packet networks

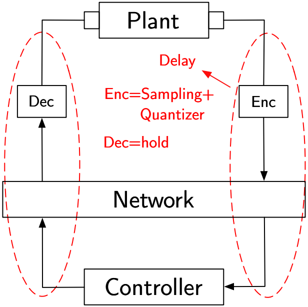
Header (H)	Payload (P)

- Channels are characterized by a packet rate (n packets per second) $\rightarrow$ The bigger the packets the lower the rate
- Sending just a $0/1$ or a much bigger number has the same cost (1 packet)

Simplifying (often realistic) assumption: finite packet rate but each packet can carry any number.

$\rightarrow$ If the assumption is not fulfilled, quantization effects can substantially impact on stability and performance

Sampling and Delay



Remote Controller

The delay between encoding and decoding essentially depends on:

- the *network access protocol* influencing the time it takes for a shared network to accept a packet
- the *transmission delays*: the time packets spend inside the network
  - $\rightarrow$ Variable delays (depend on congestion and channel quality)

Packet Dropout

Loss of packets during transmission

Causes:

- errors in the physical network links
- buffer overflow due to congestion
- long transmission delay $\rightarrow$ packet reordering and re-transmission $\rightarrow$ dropout if the receiver discards the old data. Common in real-time control as re-transmission of old data is not useful

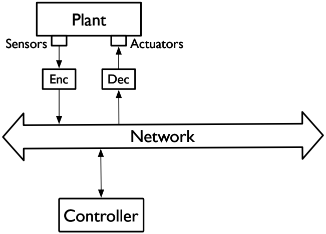
1h41

Course organization, supporting material, exams

Timetable and Course Schedule (tentative)

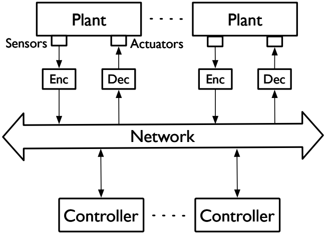
Part 1: Challenges (Week 1-6) - mostly 1 plant, 1 controller setting

- Review of LTI systems
- Linear Matrix Inequalities (LMIs)
- Control networks and NCS
- Impact of delays
- Impact of packet drops



Part 2: Opportunities (Week 7-14) - multiple systems

- Coordination: motivating examples
- Elements of graph and matrix theory
- Discrete-time consensus
- Continuous-time consensus



Course information

- Professor: Giancarlo Ferrari Trecate, Room ME C2 398, `giancarlo.ferraritrecate@epfl.ch`
- Lectures: Wed 13:00-15:00 ELA 1
  - **Course slides on Moodle, videos of 2021 available**
  - Probably, a couple lectures will be exceptionally pre-recorded. This will be properly notified on Moodle in advance
- Exercises: Wed 15:00-16:00 ELA 1 Laptops+ Matlab required!

Course Information

- Assistants:

Mahrokh Ghoddousi, Riccardo Cescon, Nicolas Kirsch, Daniele Martinelli

- Forums
  - Students can post questions anytime on the 'Discussions' forum. Students can also (and are encouraged to!) answer their colleagues. The TAs will check once a week.

… and the teaching team can be always contacted via email!

Exams and grades

- Written exam: 2 hours - example copy on Moodle.
  - 5/6 sections, 1 multiple choice
  - Closed book, closed notes, no computers. Bring with you a pen, an eraser, an ID and a non-programmable calculator
  - You are also permitted to bring **one crib sheet**, formatted on A4 paper. The sheet must be **handwritten only** (no tablet-generated content or copies of the slides), and you may use both sides
- Each problem will give a maximal number of points, clearly indicated. The total is 100 points. Example (NOT the real numbers):

Problem:	1	2	3	4	5	6	Total
Value:	20	20	15	15	15	15	100
Grade:							

- Final grade

Points	96-100	91-95	$\cdots$	56-60	51-55	$\cdots$	6-10	1-5	0
Grade	6.00	5.75	$\cdots$	4.00	3.75	$\cdots$	1.50	1.25	1.00

Literature

- No textbooks required!
- Challenges in NCSs
  - J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," in Proceedings of the IEEE, vol. 95, no. 1, pp. 138-162, Jan. 2007.
  - W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems" in IEEE Control Systems Magazine, vol. 21, no.1, pp. 84-99, 2001.
  - Feng-Li Lian, J. R. Moyne and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," in IEEE Control Systems Magazine, vol. 21, no. 1, pp. 66-83, 2001.
- Opportunities in NCSs
  - Francesco Bullo, Lecture notes on network systems, 2017. Available on moodle. New 2020 version available online at: `http://motion.me.ucsb.edu/book-lns/`
  - F. Garin and L. Schenato, "A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms," in Networked Control Systems, Springer London, pp.75-107, 2010.

Software for exercises

- Matlab with Yalmip and Mosek for solving optimization problems. Required next week!
- For installing Yalmip and Mosek, follow the document "Steps for Matlab configuration for convex optimization" available on Moodle
  - For activating Mosek you have to submit a license request using your EPFL student mail

Matlab code for testing the installation

```
ops = sdpsettings('solver','mosek'); P = sdpvar(2,2);
Q = eye(2,2);
CONS = [P > = Q]; %Constraint
infosolve=solvesdp(CONS,[],ops);
infosolve.info;
% The last command should give "successfully solved"
```

Review of System Theory

Dynamical systems

Linear time-varying (LTV) system

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \qquad (1) \qquad x(t) \in \mathbb{R}^n \text{ state}$$

$$y(t) = C(t)x(t) + D(t)u(t) \qquad (2) \qquad u(t) \in \mathbb{R}^m \text{ input}$$

$$x(t_0) = x_0 \qquad (3) \qquad y(t) \in \mathbb{R}^p \text{ output}$$

- (1): state equation
- (2): output equation
- n: system order
- $t \in \mathbb{R}$: Continuous-Time (CT) system
- $A(t)$, $B(t)$, $C(t)$, $D(t)$ matrices

Definition

$u(t) \quad t_0 \leq t \leq t$

A state trajectory is a function $x(t), t \geq t_0$ verifying (1) and (3). For highlighting the dependence on the input, initial time and initial states, we write $x(t) = \phi(t, t_0, x_0, u)$ and ϕ is called *transition map*

Review - invariant systems

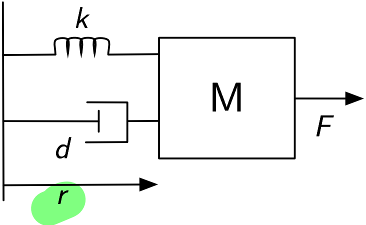
A linear system is invariant if $A(t)$, $B(t)$, $C(t)$, $D(t)$ do not depend on time

Linear Time-Invariant (LTI) system

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

$A,\ B,\ C,\ D$ matrices

Example - mass/spring/damper



- $k > 0$: elastic coefficient
- $d > 0$: damping coefficient
- F: external force (input)
- r: position (output)

Set $x_1 = r$, $x_2 = \dot{r}$, $u = F$, $y = x_1 \rightarrow M\ddot{x}_1 = -kx_1 - d\dot{x}_1 + u$

For M=1, d=1, k=1

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_1 - x_2 + u$$
$$y = x_1$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = 0$$

Linear systems: superposition principle

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

For $\alpha, \beta \in \mathbb{R}$, let

- $x_a(t) = \phi(t, t_0, x_{0,a}, u_a)$ and $y_a(t)$ the corresponding output
- $x_b(t) = \phi(t, t_0, x_{0,b}, u_b)$ and $y_b(t)$ the corresponding output
- $x(t) = \phi(t, t_0, \alpha x_{0,a} + \beta x_{0,b}, \alpha u_a + \beta u_b)$ and $y(t)$ the corresponding output

Then, $\forall t \geq t_0$

- $x(t) = \alpha x_a(t) + \beta x_b(t)$
- $y(t) = \alpha y_a(t) + \beta y_b(t)$

The same holds for linear time-varying systems

LTI systems: Lagrange formula

How the transition map looks like for an LTI system?

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \\ x(t_0) &= x_0 \end{aligned} \tag{4}$$

Matrix exponential

$$e^{At} = exp(At) = I + At + \frac{A^2t^2}{2} + \frac{A^3t^3}{3!} + \ldots + \frac{A^kt^k}{k!} + \ldots$$

- Always convergent
- Generalizes the power series of $e^{\alpha}, \alpha \in \mathbb{R}$
- Can be difficult to compute for all $t \geq 0$. MatLab `expm(A*t)`

LTI systems: Lagrange formula

Theorem

For (4)

- $x(t) = \phi(t, t_0, x_0, u) = \underbrace{e^{A(t-t_0)}x_0}_{\phi(t,t_0,x_0,0)=\textit{free state}} + \underbrace{\int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau}_{\phi(t,t_0,0,u)=\textit{forced state}}$
- $y(t) = \underbrace{Ce^{A(t-t_0)}x_0}_{\textit{free output}} + \underbrace{C\int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau + Du(t)}_{\textit{forced output}}$

$\hookrightarrow Cx + Du$

Equilibria of LTI systems

Given $u(t) = \bar{u}, t \geq 0$, the state $\bar{x} \in \mathbb{R}^n$ is an equilibrium state for $\dot{x} = Ax + Bu$ if

$$A\bar{x} + B\bar{u} = 0$$

and the pair $(\bar{x}, \bar{u})$ is called equilibrium.

- $\bar{u} = 0$, $\bar{x} = 0$ is always an equilibrium
- if $\bar{u} \in \mathbb{R}^m$, there might be one/none/infinitely many equilibria

Example: $\dot{x} = u$, $u(t) \in \mathbb{R}$

$0 = \bar{u}$ → $\bar{u} \neq 0$ → no equilibrium $\bar{x}$

↳ $\bar{u} = 0$ → any $\bar{x} \in \mathbb{R}$ is an equilibrium state.

LTI system: stability of equilibria

Let $(\bar{x}, \bar{u})$ be an equilibrium for $\dot{x} = Ax + Bu$, $x(0) = x_0$. How uncertainty on $x_0 = \bar{x}$ propagates to $x(t)$?

- Perturbed experiment : $\tilde{x}(t) = \phi(t, 0, \tilde{x}_0, \bar{u})$

Definitions (Lyapunov stability)

The equilibrium state $\bar{x}$ is

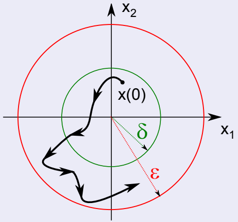
- stable if $\forall \epsilon > 0\ \exists \delta > 0 : \|\tilde{x}_0 - \bar{x}\| \leq \delta \Rightarrow \|\tilde{x}(t) - \bar{x}\| < \epsilon, \forall t \geq 0$
- (globally) asymptotically stable (AS) if it is stable and attractive, i.e.,

$$\lim_{t\to\infty} \|\tilde{x}(t) - \bar{x}\| = 0,\ \forall \tilde{x}_0 \in \mathbb{R}^n$$

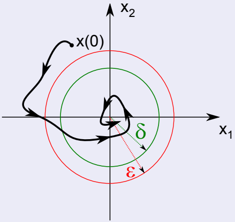
- unstable, if not stable

LTI system: stability of equilibria

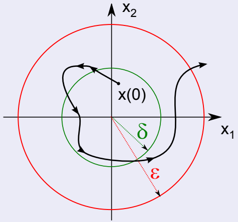
$\bar{x} = 0$ stable



$\bar{x} = 0$ AS



$\bar{x} = 0$ unstable



2h 27

Definition

$\bar{x}$ is (globally) exponentially stable (ES) if there are $\alpha, \lambda > 0$ such that

$$\|\tilde{x}(t) - \bar{x}\| \leq \alpha e^{-\lambda t} \|\tilde{x}_0 - \bar{x}\|, \; \forall \tilde{x}_0 \in \mathbb{R}^n$$

The parameter λ is called convergence rate

Key result for LTI systems

ES $\Leftrightarrow$ AS $\Rightarrow$ stability

Stability of LTI systems - relevant properties

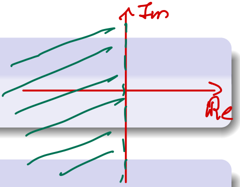
$(\bar{x}, \bar{u})$

$$\dot{x} = Ax + Bu$$

For an LTI system, **all equilibria have the same stability properties**

- study stability of the origin, i.e., $(\bar{x}, \bar{u}) = (0, 0)$
- the whole system can be termed stable/AS/unstable/ES

Stability test through the eigenvalues of A



Spec = set of eigenvalues

Definition

A is Hurwitz if all $\lambda \in \text{Spec}(A)$ verify $\text{Re}(\lambda) < 0$

Theorem (stability test)

An LTI system is

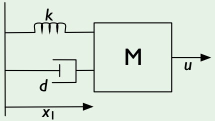
- AS $\Leftrightarrow$ A is Hurwitz
- unstable if A has at least one eigenvalue λ with $\text{Re}(\lambda) > 0$
- stable if all eigenvalues λ of A verify $\text{Re}(\lambda) \leq 0$ and those verifying $\text{Re}(\lambda) = 0$ are simple

Remark

Multiple eigenvalues on the imaginary axis can lead either to stability or instability (more complex, in textbooks, related to the Jordan form of A)

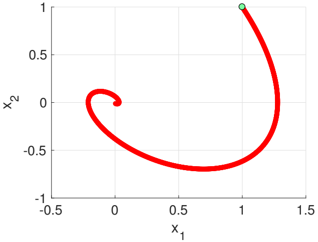
Example

Mass spring damper with $M = 1, \bar{u} = 0$



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- $k = 0$, $d > 0 \Rightarrow \text{Spec}(A) = \{0, -d\} \Rightarrow$ stable but not AS
  - Equilibrium states: $\bar{x} = [\alpha, 0], \alpha \in \mathbb{R}$
- $k > 0$, $d > 0$, $\det(\lambda I - A) = \det(\begin{bmatrix} \lambda & -1 \\ k & \lambda + d \end{bmatrix}) = \lambda^2 + d\lambda + k$
  - roots with real part $< 0 \Rightarrow$ AS
  - just one equilibrium (even if $\bar{u} \neq 0$)



State evolution for $k = 1$