# Control Theory

## Course ID: 322MI – Spring 2023

Felice Andrea Pellegrino

University of Trieste
Department of Engineering and Architecture

322MI –Spring 2023

# Lecture 1: Generalities: systems, models, and control

# Introduction

# Control theory

According to the Merriam-Webster's Dictionary, "to control" means "to exercise directing influence over" (synonyms: "to regulate", "to keep", "to restrain").

> *Control theory [is a] field of applied mathematics that is relevant to the control of certain physical processes and systems. Although control theory has deep connections with classical areas of mathematics, such as the calculus of variations and the theory of differential equations, it did not become a field in its own right until the late 1950s and early 1960s. At that time, problems arising in engineering and economics were recognized as variants of problems in differential equations and in the calculus of variations, though they were not covered by existing theories. At first, special modifications of classical techniques and theories were devised to solve individual problems. It was then recognized that these seemingly diverse problems all had the same mathematical structure, and control theory emerged.*
>
> (Rudolf E. Kalman for the Encyclopædia Britannica)

The "physical processes and systems" mentioned by Kalman are dynamical systems, thus "control theory" is a shorthand for "theory of the control of dynamical systems".

# Control theory (cont.)

*There is nothing more practical than a good theory.* (Ludwig Boltzmann)

The control theory is of paramount practical importance. It led to the design and construction of control systems in

- aerospace,
- robotics,
- manufacturing,
- power industry,
- automotive,
- electronics and communications,
- ...

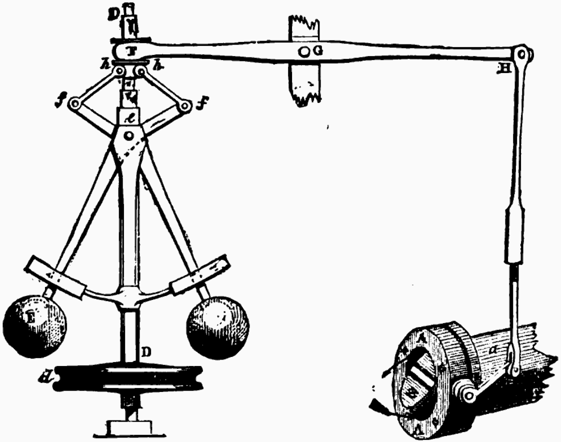
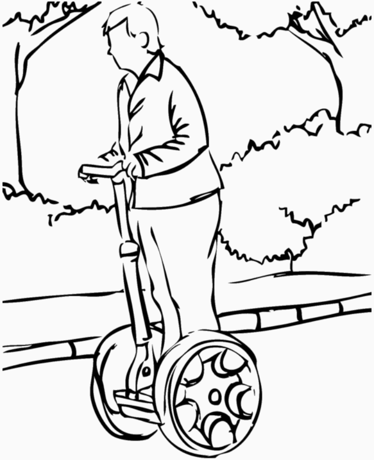
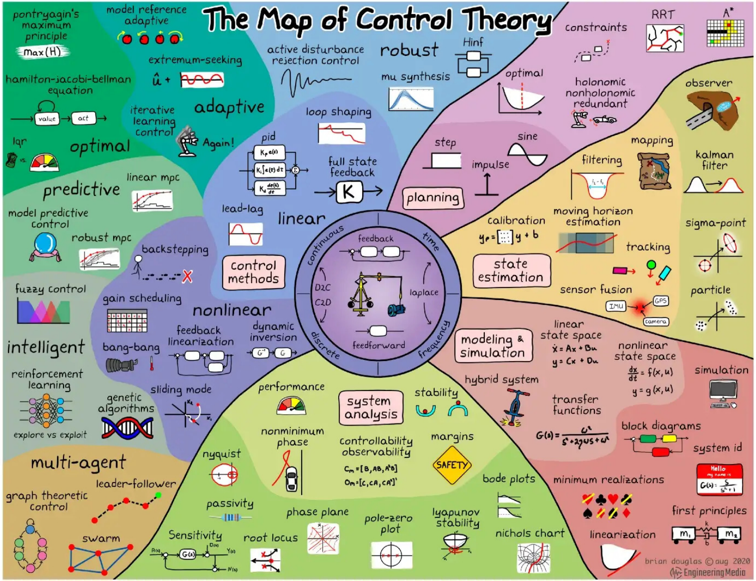


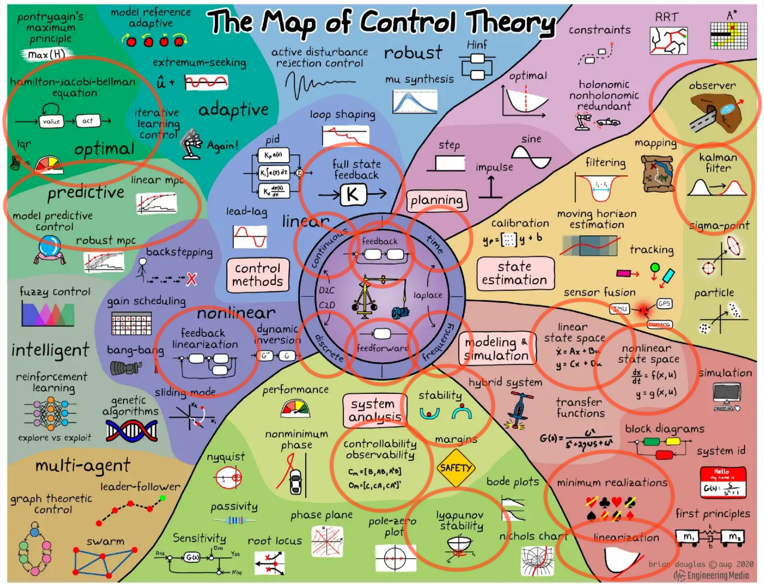
FIG. 4.—*Governor and Throttle-Valve.*



# A map



The Map of Control Theory
pontryagin's maximum principle
max (H)
model reference adaptive
extremum-seeking
hamilton-jacobi-bellman equation
value
act
iterative learning control
adaptive
Again!
lqr
optimal
predictive
linear mpc
model predictive control
robust mpc
fuzzy control
intelligent
reinforcement learning
explore vs exploit
genetic algorithms
multi-agent
leader-follower
graph theoretic control
swarm
active disturbance rejection control
robust
Hinf
mu synthesis
loop shaping
pid
full state feedback
K
lead-lag
linear
control methods
backstepping
gain scheduling
nonlinear
feedback linearization
dynamic inversion
bang-bang
sliding mode
continuous
time
feedback
D2C
C2D
laplace
feedforward
discrete
frequency
performance
system analysis
stability
nonminimum phase
controllability observability
margins
SAFETY
nyquist
passivity
phase plane
pole-zero plot
lyapunov stability
bode plots
nichols chart
Sensitivity
root locus
constraints
RRT
A*
optimal
holonomic nonholonomic redundant
step
sine
impulse
planning
observer
mapping
filtering
kalman filter
moving horizon estimation
calibration
sigma-point
tracking
state estimation
sensor fusion
IMU
GPS
camera
particle
modeling & simulation
linear state space
nonlinear state space
simulation
hybrid system
transfer functions
block diagrams
system id
minimum realizations
first principles
linearization
brian douglas © aug 2020
Engineering Media


The Map of Control Theory
pontryagin's maximum principle
max(H)
model reference adaptive
extremum-seeking
active disturbance rejection control
robust
Hinf
mu synthesis
constraints
RRT
A*
hamilton-jacobi-bellman equation
value
act
iterative learning control
adaptive
Again!
loop shaping
optimal
holonomic nonholonomic redundant
observer
lqr
optimal
pid
full state feedback
K
step
sine
impulse
mapping
filtering
kalman filter
linear mpc
predictive
model predictive control
robust mpc
lead-lag
linear
planning
calibration
moving horizon estimation
sigma-point
backstepping
control methods
continuous
feedback
time
D2C
C2D
laplace
discrete
feedforward
frequency
state estimation
tracking
sensor fusion
IMU
GPS
camera
particle
fuzzy control
gain scheduling
nonlinear
feedback linearization
dynamic inversion
intelligent
bang-bang
sliding mode
modeling & simulation
linear state space
nonlinear state space
simulation
reinforcement learning
genetic algorithms
explore vs exploit
performance
system analysis
stability
hybrid system
transfer functions
nonminimum phase
controllability observability
margins
block diagrams
system id
multi-agent
leader-follower
nyquist
SAFETY
bode plots
minimum realizations
graph theoretic control
passivity
phase plane
pole-zero plot
lyapunov stability
nichols chart
first principles
linearization
swarm
Sensitivity
root locus
brian douglas © aug 2020
Engineering Media

# Systems

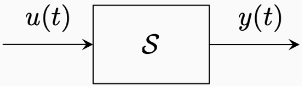
With the expression "dynamical system", or "dynamic system", we refer to **any entity that evolves over time, interacting with the environment according to the cause-effect principle**.

Inputs ("causes")

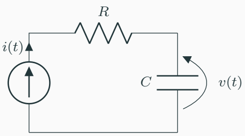
$u(t)$

Outputs ("effects")

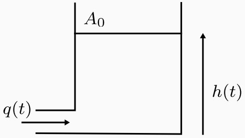
$y(t)$



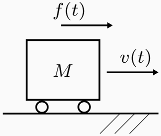
Typically, but not exclusively, dynamic systems are described by ordinary differential or difference equations (ODE) where **time** is the independent variable.



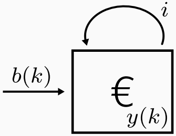

Circuit.




Reservoir.




Body moving on a straight line.




Bank deposits.

$$i(t) = C\frac{dv(t)}{dt}$$

$$q(t) = A_0\frac{dh(t)}{dt}$$

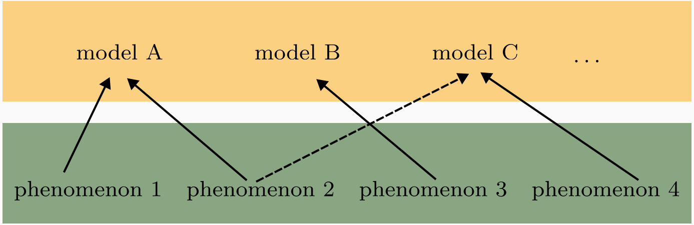
$$f(t) = M\frac{dv(t)}{dt}$$

$$y(k+1) = (1+i)y(k)+b(k)$$

**Note.** For convenience, in the following we will often employ $\dot{f}(t)$ and $\ddot{f}(t)$ as a shorthand for $\frac{df(t)}{dt}$ and $\frac{d^2f(t)}{dt^2}$, respectively.

- In control theory, we manipulate **models** (mathematical descriptions of real phenomena).
- Modelling is an entire subject[1] in itself (Ljung and Glad (1994)).
- **Abstraction** is essential to control theory, because models are abstractions.
- Advantage of operating at an abstract level: different phenomena may be described by the same model, e.g.:

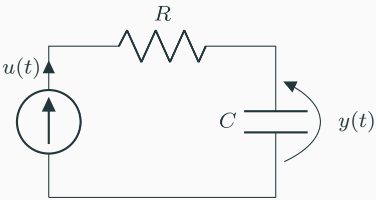
$$i(t) = C\frac{dv(t)}{dt}, \qquad q(t) = A_0\frac{dh(t)}{dt}, \qquad f(t) = M\frac{dv(t)}{dt}.$$

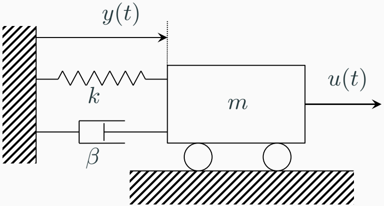



[1] and, to some extent, an art.

# Initial conditions and state

Generally, to determine uniquely the output $y(t)$ in a certain time interval $[t_0, t_1]$, the knowledge of the sole input $u(t)$ in the same interval, in not sufficient. Some **initial conditions** must be known.



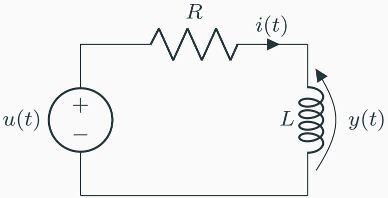




The initial condition $y(t_0)$ is required to determine $y(t)$, $t \in [t_0, t_1]$.

The initial conditions $y(t_0)$ and $\dot{y}(t_0)$ are required to determine $y(t)$, $t \in [t_0, t_1]$

# Initial conditions and state (cont.)

Often the initial conditions are the value of $y(t)$ and, possibly, its derivatives $\dot{y}(t), \ddot{y}(t)\dots$ at a certain initial time $t_0$. Other times, the initial conditions involve other variables that the output and its derivatives.




The circuit is described by:

$$u(t) = Ri(t) + L\frac{di(t)}{dt}$$

The output is:

$$y(t) = u(t) - Ri(t)$$

The initial condition is the value $i(t_0)$ of the current $i(t)$ at time $t_0$.

Generally, the initial conditions depend on the past history of the system, that determines its current “state”.

# State variables

### State variables

Variables to be known at time $t = t_0$ in order to be able to determine the output $y(t)$, $t \geq t_0$ from the knowledge of the input $u(t)$, $t \geq t_0$:

$$x_i(t),\ i = 1, 2, \dots, n \qquad \text{(state variables)}$$

The concept of state leads to the *state-space representation* of dynamical systems (as opposite to the *input-output representation*).

# Dynamic systems: formal definition

Dynamic systems are a rather general concept, that extends far beyond the systems described by ODE.

Indeed, a dynamic system is an abstract entity defined in axiomatic way:

$$\mathcal{S} = \{T, U, \Omega, X, Y, \Gamma, \varphi, \eta\}$$

- $T$ : a set of times, which is totally ordered (i.e. if $t_1 \neq t_2$ then either $t_1 < t_2$ or $t_1 > t_2$). Possible choices are $T = \mathbb{R}$ and $T = \mathbb{Z}$.
- $U$ : set of input values. It may be finite or not; frequently $U = \mathbb{R}^m$.
- $\Omega$ : set of admissible input functions, $\Omega = \{u(\cdot) : T \longrightarrow U | \text{condition}\}$.
- $X$ : set of states. It may be finite or not; frequently $X = \mathbb{R}^n$.
- $Y$ : set of output values; frequently $Y = \mathbb{R}^p$.
- $\Gamma$ : set of admissible output functions, $\Gamma = \{y(\cdot) : T \longrightarrow Y | \text{condition}\}$.

# Dynamic systems: formal definition (cont.)

Let define the *state transition function*:

$$\varphi : T \times T \times X \times \Omega \mapsto X \quad \Longrightarrow \quad x(t) = \varphi(t, t_0, x_0, u(\cdot))$$

where $t_0$ is the initial time and $x_0 = x(t_0)$ is the initial state. The state transition function must satisfy the following axioms:

1. **Consistency**: $\varphi(t_0, t_0, x_0, u(\cdot)) = x_0\,,\ \forall\,(t_0, x_0, u(\cdot)) \in T \times X \times \Omega$
2. **Forward definition**: $\varphi$ is defined $\forall\, t \geq t_0,\ t \in T$
3. **Composition**:

$$\varphi(t_2, t_0, x_0, u(\cdot)) = \varphi(t_2, t_1, \varphi(t_1, t_0, x_0, u(\cdot)), u(\cdot))$$
$$\forall\,(t_0, u(\cdot)) \in T \times \Omega\,,\ \forall\, t_0, t_1, t_2 \in T :\ t_0 \leq t_1 \leq t_2$$

4. **Causality**: given two (possibly different) input functions $u'(\cdot)$ and $u''(\cdot)$

$$u'_{[t_0,t)}(\cdot) = u''_{[t_0,t)}(\cdot) \implies \varphi(t, t_0, x_0, u'(\cdot)) = \varphi(t, t_0, x_0, u''(\cdot)),$$
$$\forall(t, t_0, x_0) \in T \times T \times X, \quad t \geq t_0$$

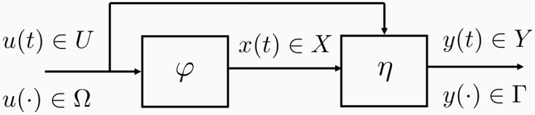
# Dynamic systems: formal definition (cont.)

Referring to the output, we can define a function

$$y(t) = \eta(t, x(t), u(t))$$

where

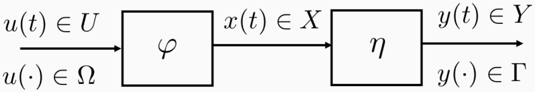
$$\eta : T \times X \times U \mapsto Y$$




In the particular case of no explicit dependence of the output on $u(t)$, i.e.

$$y(t) = \eta(t, x(t))$$

the system is said *strictly proper*:

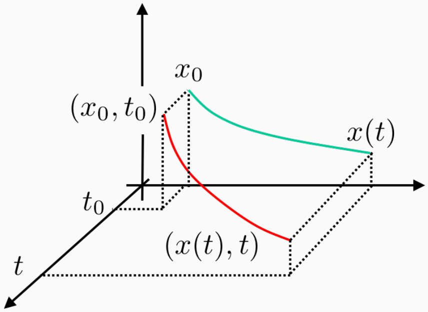


$(x,t) \in X \times T$ is defined as *event*

Given:

- $(x_0, t_0)$ initial event
- $u(\cdot)$ input function

the system evolves in time along a *trajectory* (set of crossed states). The whole function of time describing the evolution is called the *movement*.




$\varphi(\cdot, t_0, x_0, u(\cdot))$ state movement

$\varphi(t, t_0, x_0, u(\cdot))$, $t \geq t_0$ state trajectory

$\eta(\cdot, \varphi(\cdot, t_0, x_0, u(\cdot)))$ output movement

$\eta(t, \varphi(\cdot, t_0, x_0, u(\cdot)))$ output trajectory

$\bar{x} \in X$ is an *equilibrium state* if $\forall\, t_0 \in T$, $\exists\, u(\cdot) \in \Omega$ such that

$$\varphi(t, t_0, \bar{x}, u(\cdot)) = \bar{x}\,,\ \forall\, t \geq t_0,\, t \in T$$

$\bar{y} \in Y$ is an *equilibrium output* if $\forall\, t_0 \in T$, $\exists \bar{x} \in X$, $\exists\, u(\cdot) \in \Omega$ such that

$$\eta(t, \varphi(t, t_0, \bar{x}, u(\cdot))) = \bar{y}\,,\ \forall\, t \geq t_0,\, t \in T$$

Notice that, in general:

- the specific input function $u(\cdot) \in \Omega$ depends on the choice of the initial time-instant $t_0 \in T$
- the state of a dynamic system being is at equilibrium does not imply that the output is at equilibrium as well, unless $\eta(t, x(t))$ does not depend explicitly on time (in which case, the output function takes on the form $\eta(x(t))$ )

# Dynamic systems: formal definition (cont.)

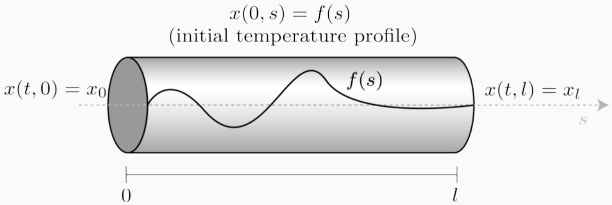
- A dynamic system is *time-invariant* if $T$ is an additive algebraic group and $\forall\, u(\cdot) \in \Omega$, $\forall\, \tau \in T$, letting $u^{\tau}(t) \doteq u(t-\tau) \in \Omega$, it follows that

$$\begin{cases} \varphi(t, t_0, x_0, u(\cdot)) = \varphi(t+\tau, t_0+\tau, x_0, u^{\tau}(\cdot))\,, \ \forall\, t, \tau \in T \\ y(t) = \eta(t, x(t)) \end{cases}$$

- A dynamic system is *discrete-time* if $T$ is isomorphic to $\mathbb{Z}$
- A dynamic system is *continuous-time* if $T$ is isomorphic to $\mathbb{R}$
- A dynamic system is *finite-dimensional* (*lumped-parameter*) if $U, X, Y$ are finite-dimensional vector spaces
- A dynamic system is *infinite-dimensional* (*distributed-parameter*) if $U$ and/or $X$ and/or $Y$ are infinite-dimensional vector spaces

# Example of infinite-dimensional system

The model of heat conduction in a rod leads to an infinite dimensional dynamic system.




The heat conduction is governed by

$$\frac{\partial x(t,s)}{\partial t} = \alpha \frac{\partial^2 x(t,s)}{\partial s^2}$$

where $x(t, s)$ is temperature at time $t$ of the section having abscissa $s$.

Here, the state at time $t$ is **the whole temperature profile** $x(t, s)$, $s \in [0, l]$.

# Automata as dynamical systems

A class of dynamical systems of interest are the *finite automata*, in which the sets $U$, $Y$, $X$ are finite:

$$U = \{u_1, u_2, ..., u_m\}$$
$$Y = \{y_1, y_2, ..., y_p\}$$
$$X = \{x_1, x_2, ..., x_n\}$$

An automaton is defined by a function:

$$x(k+1) = F(k, x(k), u(k)), \tag{1}$$

that, in case of time-invariant automata, becomes:

$$x(k+1) = F(x(k), u(k)). \tag{2}$$

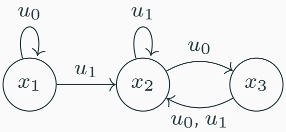
# Representation

Two possible ways to represent an automaton are:

- a *transition table*: an $m \times n$ table in which the rows denote the inputs and the columns the states. Each entry $F_{ij}$ represents the future state of the system when the input $i$ is applied while the system is in state $j$.
- a *flow graph*: in which each possible state is represented by a circle. Each transition from a state to another is described by an arc whose label denotes the input that determines that transition.

| | $x_1$ | $x_2$ | $\cdots$ | $x_n$ |
|---|---|---|---|---|
| $u_1$ | $F_{11}$ | $F_{12}$ | $\cdots$ | $F_{1n}$ |
| $u_2$ | $F_{21}$ | $F_{22}$ | $\cdots$ | $F_{2n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $u_m$ | $F_{m1}$ | $F_{m2}$ | $\cdots$ | $F_{mn}$ |

Transition table




Flow graph

$\updownarrow$

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $u_0$ | $x_1$ | $x_3$ | $x_2$ |
| $u_1$ | $x_2$ | $x_2$ | $x_2$ |

Transition table

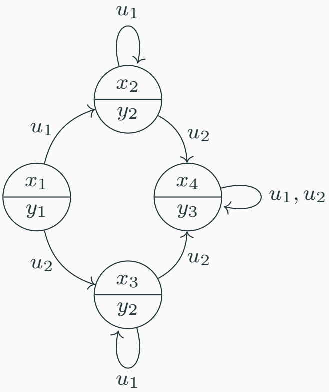
Two classes of automata are the following:

- *Moore automaton*: whose output is function only of the state. In this case the circle that represents the state is divided in two parts; the upper part denotes the state index and the lower part the output associated to it.

$$y = g(x) \qquad \text{(strictly proper system)} \tag{3}$$

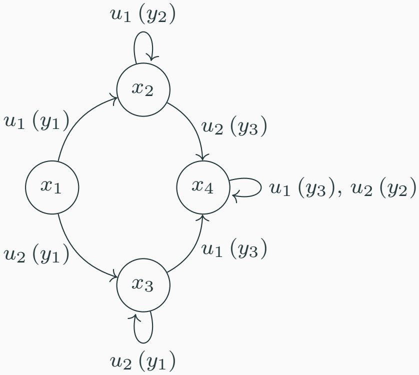
- *Mealy automaton*: whose output is function of both the state and the input. In this case, an arc is assigned two values; the first represents the input, and the second the output.

$$y = g(x, u) \tag{4}$$




Moore automaton

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $u_1$ | $x_2$ | $x_2$ | $x_3$ | $x_4$ |
| $u_2$ | $x_3$ | $x_4$ | $x_4$ | $x_4$ |



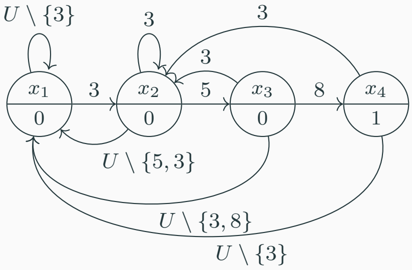

Mealy automaton

| | $x_1$ | | $x_2$ | | $x_3$ | | $x_4$ | |
|---|---|---|---|---|---|---|---|---|
| | $x_*$ | $y$ | $x_*$ | $y_*$ | $x_*$ | $y$ | $x_*$ | $y$ |
| $u_1$ | $x_2$ | $y_1$ | $x_2$ | $y_2$ | $x_4$ | $y_3$ | $x_4$ | $y_3$ |
| $u_2$ | $x_3$ | $y_1$ | $x_4$ | $y_3$ | $x_3$ | $y_1$ | $x_4$ | $y_2$ |

Goal: Design a finite state machine that recognises the string "358" within a longer string. The sets $U, Y, X$ are:

$$U = \{0, 1, 2, ...., 9\}$$
$$Y = \{0, 1\}$$
$$X = \{x_1, x_2, x_3, x_4\}$$

where the output is $0$ if the string is not recognised and 1 otherwise. The flow graph is:




Example: recognition of the string "358"

# Regular dynamic systems

In the following we will mainly consider the class of *regular* dynamic systems i.e. systems that can be represented by differential of difference equations.

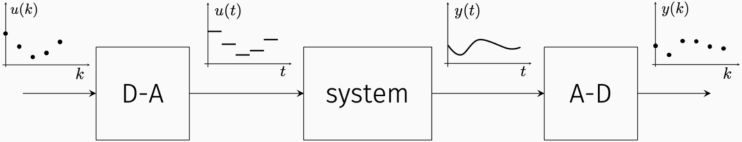
Regular finite-dimensional continuous-time dynamical systems can be represented as

$$\begin{cases} \dot{x}(t) & = & f(x(t), u(t), t) \\ x(t_0) & = & x_0 \\ y(t) & = & g(x(t), u(t), t) \end{cases}$$

In the following we will assume that for any $t_0$, $x(t_0)$, $u(\cdot)$, the solution $x(t) = \varphi(t, t_0, x_0, u(\cdot))\ t \geq t_0$ **exists and is unique**. This is true if the function $f$ has some regularity property (for instance, it is a Lipschitz function) and $u$ is sufficiently regular (e.g., piecewise-continuous).
For discrete-time systems, we have

$$\begin{cases} x(k+1) & = & f(x(k), u(k), k) \\ x(k_0) & = & x_0 \\ y(k) & = & g(x(k), u(k), k) \end{cases}$$

In the latter case, the solution $x(k) = \varphi(k, k_0, x_0, u(\cdot))\ k \geq k_0$ does always exist and is unique.




Sampled-data systems are discrete-time dynamic systems obtained by sampling a continuous-time regular system. The figure above represents a continuous-time system connected to a digital-analog converter (D-A) and an analog-digital converter (A-D).

- $U, X, Y$ finite-dimensional normed vector spaces
- $\Omega = \{u(\cdot) :$ piecewise constant $u_i(\cdot)\,,\ i = 1, \ldots, m\}$
- Let the sampling time be $\Delta T$. Then, denoting (with slight abuse of notation) by $u(k)$ and $y(k)$, respectively, the input and the output of the sampled-data system, we have:

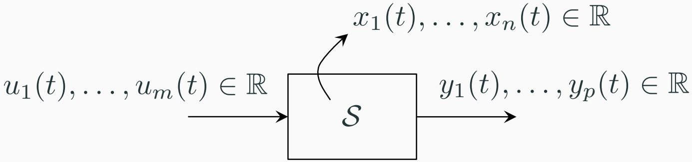
$$u(k) = u(t)\,,\ t_0 + k\Delta T \leq t < t_0 + (k+1)\Delta T,\ k = 0, 1, \ldots$$
$$y(k) = y(t_0 + k\Delta T),\ k = 0, 1, \ldots$$

Then we can write the following state-space representation of a discrete-time system:

$$\begin{cases} x(k+1) = f_d(x(k), u(k), k) \\ y(k) = g_d(x(k), u(k), k) \end{cases}$$

where, letting $\bar{u}(t) = u(k), \forall t \geq t_0 k + \Delta T$, and recalling the composition property of the state transition function $\varphi$:

$$f_d(x(k), u(k), k) = \varphi(t_0 + (k+1)\Delta T, t_0 + k\Delta T, x(k), \bar{u}(\cdot))$$
$$g_d(x(k), u(k), k) = \eta(x(k), u(k), t_0 + k\Delta T)$$




State equations (differential)

$$\begin{cases} \dot{x}_1(t) = f_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \\ \quad \vdots \\ \dot{x}_n(t) = f_n(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \end{cases}$$

Output equations (algebraic)

$$\begin{cases} y_1(t) = g_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \\ \quad \vdots \\ y_p(t) = g_p(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t) \end{cases}$$

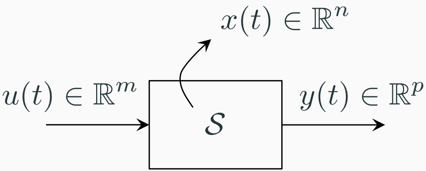
# Continuous-time state equations (cont.)

$$u(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix} \in \mathbb{R}^m, \quad y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_p(t) \end{bmatrix} \in \mathbb{R}^p$$

$$x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} \in \mathbb{R}^n$$

$$f(x,u,t) = \begin{bmatrix} f_1(x,u,t) \\ \vdots \\ f_n(x,u,t) \end{bmatrix} \in \mathbb{R}^n$$

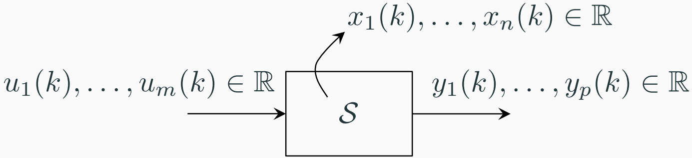
$$g(x,u,t) = \begin{bmatrix} g_1(x,u,t) \\ \vdots \\ g_p(x,u,t) \end{bmatrix} \in \mathbb{R}^p$$




**Compact form**

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases}$$

# Discrete-time state equations




State equations
(difference)

$$\begin{cases} x_1(k+1) = f_1(x_1(k), \dots, x_n(k), u_1(k), \dots, u_m(k), k) \\ \quad \vdots \\ x_n(k+1) = f_n(x_1(k), \dots, x_n(k), u_1(k), \dots, u_m(k), k) \end{cases}$$

Output equations
(algebraic)

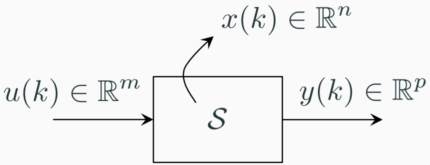
$$\begin{cases} y_1(k) = g_1(x_1(k), \dots, x_n(k), u_1(k), \dots, u_m(k), k) \\ \quad \vdots \\ y_p(k) = g_p(x_1(k), \dots, x_n(k), u_1(k), \dots, u_m(k), k) \end{cases}$$

$$u(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_m(k) \end{bmatrix} \in \mathbb{R}^m, \quad y(k) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_p(k) \end{bmatrix} \in \mathbb{R}^p$$

$$x(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} \in \mathbb{R}^n$$

$$f(x,u,k) = \begin{bmatrix} f_1(x,u,k) \\ \vdots \\ f_n(x,u,k) \end{bmatrix} \in \mathbb{R}^n$$

$$g(x,u,k) = \begin{bmatrix} g_1(x,u,k) \\ \vdots \\ g_p(x,u,k) \end{bmatrix} \in \mathbb{R}^p$$




**Compact form**

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

# More definitions and properties

- When $u(t) \in \mathbb{R}$, the system is called *single-input (SI)*, otherwise it is called *multiple-input (MI)*;
- When $y(t) \in \mathbb{R}$, the system is called *single-output (SO)*, otherwise it is called *multiple-output (MO)*;
- The acronyms can be combined: for instance, *SISO* denotes a single-input, single-output system;
- When the state and output equations do not depend explicitly on time, the system is called *time-invariant*:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), \cancel{t}) \\ y(t) = g(x(t), u(t), \cancel{t}) \end{cases} \implies \begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), u(k), \cancel{k}) \\ y(k) = g(x(k), u(k), \cancel{k}) \end{cases} \implies \begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = g(x(k), u(k)) \end{cases}$$

- When the output equation does not depend explicitly on the input, the system is called *strictly proper*:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), \cancel{u(t)}, t) \end{cases} \implies \begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), t) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), \cancel{u(k)}, k) \end{cases} \implies \begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), k) \end{cases}$$

# More definitions and properties (cont.)

- When there is no input, the system is called *unforced*:

$$\begin{cases} \dot{x}(t) = f(x(t), \cancel{u(t)}, t) \\ y(t) = g(x(t), \cancel{u(t)}, t) \end{cases} \Longrightarrow \begin{cases} \dot{x}(t) = f(x(t), t) \\ y(t) = g(x(t), t) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), \cancel{u(k)}, k) \\ y(k) = g(x(k), \cancel{u(k)}, k) \end{cases} \Longrightarrow \begin{cases} x(k+1) = f(x(k), k) \\ y(k) = g(x(k), k) \end{cases}$$

It is worth noting that in case the input function $u(t)$, $\forall\, t$ or input sequence $u(k)$, $\forall\, k$ are **known beforehand**, the dynamic system can be re-written as an unforced one:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) = \widetilde{f}(x(t), t) \\ y(t) = g(x(t), u(t), t) = \widetilde{g}(x(t), t) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), u(k), k) = \widetilde{f}(x(k), k) \\ y(k) = g(x(k), u(k), k) = \widetilde{g}(x(k), k) \end{cases}$$

# More definitions and properties (cont.)

- *Natural response* (the response to zero input):

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= g(x(t), u(t), t) \end{aligned}$$

with:

$$x(t_0) = x_0\,; \quad u(t) = 0\,, \ \forall\, t$$

$\Longrightarrow$ $x_N(t),\ t \geq t_0$ natural response

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

with:

$$x(k_0) = x_0\,; \quad u(k) = 0\,, \ \forall\, k$$

$\Longrightarrow$ $x_N(k),\ k \geq k_0$ natural response

- *Forced response* (the response from zero initial state):

$$\dot{x}(t) = f(x(t), u(t), t)$$
$$y(t) = g(x(t), u(t), t) \quad \Longrightarrow \quad x_F(t),\ t \geq t_0$$

with: $x(t_0) = 0$ $\quad$ forced response

$$x(k+1) = f(x(k), u(k), k)$$
$$y(k) = g(x(k), u(k), k) \quad \Longrightarrow \quad x_F(k),\ k \in\geq k_0$$

with: $x(k_0) = 0$ $\quad$ forced response

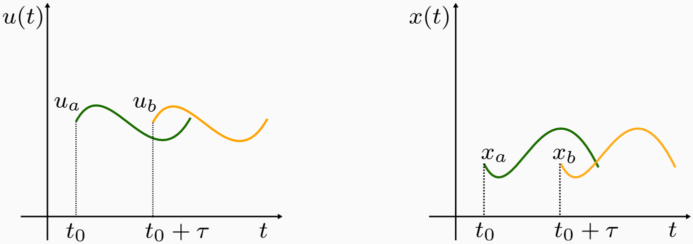
# Time-invariant continuous-time systems

Consider the system $\dot{x}(t) = f(x(t), u(t))$, and apply the input $u_a(\cdot)$:

$$\begin{aligned} x(t_0) &= x_0 \\ u_a(t) &= u(t), \quad t \in [t_0, t_1) \end{aligned} \implies x_a(t), \quad t \in [t_0, t_1)$$

Now shift the initial time by $\tau$, and the input as well:

$$\begin{aligned} x(t_0 + \tau) &= x_0 \\ u_b(t) &= u_a(t - \tau),\ t \in [t_0 + \tau, t_1 + \tau) \end{aligned} \implies \begin{aligned} x_b(t) &= x_a(t - \tau), \\ & t \in [t_0 + \tau, t_1 + \tau) \end{aligned}$$




Since the state movement is shifted as well, without loss of generality, we set $t_0 = 0$.

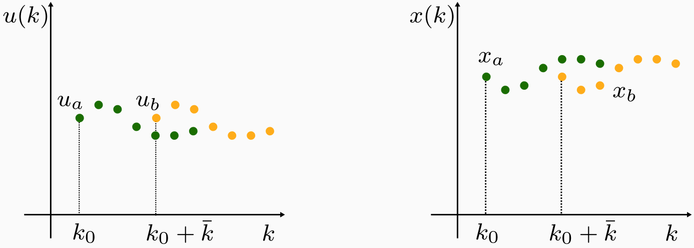
# Time-invariant discrete-time systems

Consider the system $x(k+1) = f(x(k), u(k))$, and apply the input $u_a(\cdot)$:

$$\begin{aligned} x(k_0) &= x_0 \\ u_a(k) &= u(k), k \in \{k_0, \dots, k_1\} \end{aligned} \implies x_a(k),\ k \in \{k_0, \dots, k_1\}$$

Now shift the initial time by $\bar{k}$ and the input sequence as well:

$$\begin{aligned} x(k_0+\bar{k}) &= x_0 \\ u_b(k) &= u_a(k-\bar{k}),\ k \in \{k_0+\bar{k}, \dots, k_1+\bar{k}\} \end{aligned} \implies \begin{aligned} &x_b(k) = x_a(k-\bar{k}), \\ &\quad k \in \{k_0+\bar{k}, \dots, k_1+\bar{k}\} \end{aligned}$$




Since the state movement is shifted as well, without loss of generality, we set $k_0 = 0$.

- A state $\bar{x} \in \mathbb{R}^n$ is an *equilibrium state* if $\forall\, t_0$, $\exists\, \bar{u}(\cdot)$ such that

$$\begin{matrix} x(t_0) = \bar{x} \\ u(t) = \bar{u}(t),\, \forall\, t \geq t_0 \end{matrix} \implies x(t) = \bar{x},\, \forall\, t > t_0$$

- An output $\bar{y} \in \mathbb{R}^p$ is an *equilibrium output* if $\forall\, t_0$, $\exists\, x_0, \bar{u}(\cdot)$ such that

$$\begin{matrix} x(t_0) = x_0 \\ u(t) = \bar{u}(t),\, \forall\, t \geq t_0 \end{matrix} \implies y(t) = \bar{y},\, \forall\, t > t_0$$

In general:

- the input function depends on the initial time $t_0$;
- the system being in an equilibrium state does not imply that the output is at equilibrium too.

In the time-invariant case, all equilibrium states are obtained by imposing constant input.

A state $\bar{x} \in \mathbb{R}^n$ is an equilibrium state if $\exists\, \bar{u}$ such that

$$\begin{aligned} x(t_0) &= \bar{x} \\ u(t) &= \bar{u},\, \forall\, t \geq t_0 \end{aligned} \implies x(t) = \bar{x},\, \forall\, t > t_0$$

All equilibrium states $\bar{x} \in \mathbb{R}^n$ can thus be obtained by solving the equations

$$f(\bar{x}, \bar{u}) = 0\,, \quad \forall\, \bar{u} \in \mathbb{R}^m$$

The following sets can also be defined:

$$\begin{aligned} \bar{X}_{\bar{u}} &= \{\bar{x} \in \mathbb{R}^n :\ 0 = f(\bar{x}, \bar{u})\} \\ \bar{X} &= \{\bar{x} \in \mathbb{R}^n :\ \exists\, \bar{u} \in \mathbb{R}^m \text{ such that } 0 = f(\bar{x}, \bar{u})\} \end{aligned}$$

which are, respectively, the set of all the equilibrium states corresponding to the constant input $\bar{u}$, and the set of the equilibrium states corresponding to at least one constant input.

# Equilibrium states and outputs, discrete-time

- A state $\bar{x} \in \mathbb{R}^n$ is an *equilibrium state* if $\forall\, k_0\,,\ \exists\, \{\bar{u}(k) \in \mathbb{R}^m, k \geq k_0\}$ such that

$$\begin{aligned} x(k_0) &= \bar{x} \\ u(k) &= \bar{u}(k),\ \forall\, k \geq k_0 \end{aligned} \quad \Longrightarrow \quad x(k) = \bar{x},\ \forall\, k > k_0$$

- An output $\bar{y} \in \mathbb{R}^p$ is an *equilibrium output* if $\forall\, k_0\,,\ \exists\, \{\bar{u}(k) \in \mathbb{R}^m, k \geq k_0\}$ such that

$$\begin{aligned} x(k_0) &= \bar{x} \\ u(k) &= \bar{u}(k),\ \forall\, k \geq k_0 \end{aligned} \quad \Longrightarrow \quad y(k) = \bar{y},\ \forall\, k > k_0$$

In general:

- the input sequence $\{\bar{u}(k) \in \mathbb{R}^m, k \geq k_0\}$ depends on the initial time $k_0$;
- the system being in an equilibrium state does not imply that the output is at equilibrium too.

# Equilibrium states and outputs, discrete-time (cont.)

In the time-invariant case, all equilibrium states are obtained by imposing constant input sequences.

A state $\bar{x} \in \mathbb{R}^n$ is an equilibrium state if $\exists\, \bar{u} \in \mathbb{R}^m$ such that

$$\begin{aligned} x(k_0) &= \bar{x} \\ u(k) &= \bar{u},\, \forall\, k \geq k_0 \end{aligned} \quad \Longrightarrow \quad x(k) = \bar{x},\, \forall\, k > k_0$$

All equilibrium states $\bar{x} \in \mathbb{R}^n$ can thus be obtained by solving the equations

$$\bar{x} = f(\bar{x}, \bar{u})\,, \quad \forall\, \bar{u} \in \mathbb{R}^m$$

The following sets can also be defined:

$$\begin{aligned} \bar{X}_{\bar{u}} &= \{\bar{x} \in \mathbb{R}^n : \; \bar{x} = f(\bar{x}, \bar{u})\} \\ \bar{X} &= \{\bar{x} \in \mathbb{R}^n : \; \exists\, \bar{u} \in \mathbb{R}^m \text{ such that } \bar{x} = f(\bar{x}, \bar{u})\} \end{aligned}$$

which are, respectively, the set of all the equilibrium states corresponding to the constant input $\bar{u}$, and the set of the equilibrium states corresponding to at least one constant input.

State-space descriptions are often derived from first principles.

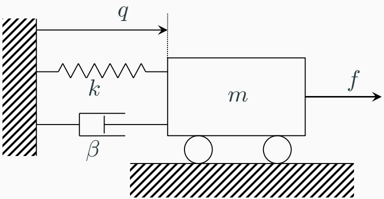
In that case, choosing the state variables is part of the modeling process.

State variables should be chosen as quantities associated with **storage** of mass, energy, momentum.... . .

For example:

- **Passive electrical systems**: voltage on capacitors, current on inductors
- **Translational mechanical systems**: linear displacement and velocity of each moving body
- **Rotational mechanical systems**: angular displacement and velocity of each rotating body
- **Hydraulic systems**: pressure or level of fluids in tanks
- **Thermal systems**: temperature and enthalpy
- . . .

A mechanical system

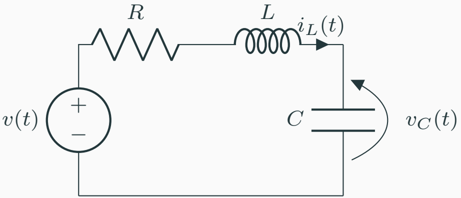


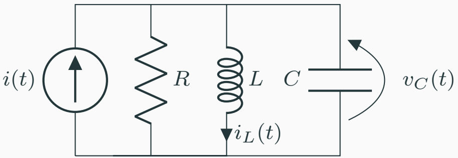

$$m\ddot{q}(t)+\beta\dot{q}(t)+kq(t) = f(t)$$

$$\begin{aligned} x_1(t) &\doteq q(t) \\ x_2(t) &\doteq \dot{q}(t) \end{aligned} \implies x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \ddot{q}(t) = -\dfrac{k}{m}x_1(t) - \dfrac{\beta}{m}x_2(t) + \dfrac{1}{m}f(t) \end{cases}$$

## Electrical systems




$$L\frac{di_L(t)}{dt} = v - Ri_L(t) - v_C(t)$$

$$C\frac{dv_C(t)}{dt} = i_L(t)$$

$$C\frac{dv_C}{dt} = i(t) - \frac{1}{R}v_C(t) - i_L(t)$$

$$L\frac{di_L(t)}{dt} = v_C(t)$$

$$x_1(t) \doteq i_L(t)\,;\;\; x_2(t) \doteq v_C(t)$$

$$\begin{cases} \dot{x}_1(t) = -\frac{R}{L}x_1(t) - \frac{1}{L}x_2(t) + \frac{1}{L}v(t) \\ \dot{x}_2(t) = \frac{1}{C}x_1(t) \end{cases}$$

$$\begin{cases} \dot{x}_1(t) = \frac{1}{L}x_2(t) \\ \dot{x}_2(t) = -\frac{1}{C}x_1(t) - \frac{1}{RC}x_2(t) + \frac{1}{C}i(t)v(t) \end{cases}$$

# State-space description from input-output description

If an input-output description is available (irrespective of how it has been derived), the state-space description can be obtained directly from the ODE, as shown in the following examples.

The basic idea is to **choose the state variables in such a way to obtain a first order differential or difference equation**.

Note that the state variables, in that case, do not necessarily have a physical meaning.

Continuous-time case:

$$\frac{d^n y}{dt^n} = \varphi\left(\frac{d^{n-1}y}{dt^{n-1}}, \ldots, \frac{dy}{dt}, y, u, t\right)$$

Letting:

$$\begin{cases} x_1(t) & \doteq & y(t) \\ x_2(t) & \doteq & \dfrac{dy}{dt} \\ \vdots & & \\ x_n(t) & \doteq & \dfrac{d^{n-1}y}{dt^{n-1}} \end{cases} \implies x \doteq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

we get:

$$\begin{cases} \dot{x}_1(t) & = & x_2(t) \\ \dot{x}_2(t) & = & x_3(t) \\ \vdots & & \\ \dot{x}_n(t) & = & \varphi(x(t), u(t), t) \\ y(t) & = & x_1(t) \end{cases}$$

Discrete-time case:

$$y(k+n) = \varphi\left(y(k+n-1), y(k+n-2), \dots, y(k), u(k), k\right)$$

Letting:

$$\begin{cases} x_1(k) & \doteq & y(k) \\ x_2(k) & \doteq & y(k+1) \\ \vdots & & \\ x_n(k) & \doteq & y(k+n-1) \end{cases} \implies x \doteq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

we get:

$$\begin{cases} x_1(k+1) & = & x_2(k) \\ x_2(k+1) & = & x_3(k) \\ \vdots & & \\ x_n(k) & = & \varphi(x(k), u(k), k) \\ y(k) & = & x_1(k) \end{cases}$$

Example (discrete-time):

$$w(k) - 3w(k-1) + 2w(k-2) - w(k-3) = 6u(k)$$

Letting:

$$\begin{cases} x_1(k) \doteq w(k-3) \\ x_2(k) \doteq w(k-2) \\ x_3(k) \doteq w(k-1) \end{cases} \quad \Longrightarrow \quad x \doteq \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

we get:

$$\begin{cases} x_1(k+1) = x_2(k) \\ x_2(k+1) = x_3(k) \\ x_3(k+1) = 3x_3(k) - 2x_2(k) + x_1(k) + 6u(k) \\ y(k) = x_3(k) \end{cases}$$

# Equivalent state-space representations

The state-space description is not unique. Equivalent descriptions can be obtained by change of basis in the space $X$.

Consider the continuous-time dynamic system:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases}$$

Let

$$\hat{x} \doteq T^{-1} x$$

where $T \in \mathbb{R}^{n \times n}$ is a generic non-singular $n \times n$ matrix ( $\det(T) \neq 0$ ).

Then, an equivalent state-space description is given by:

$$\begin{cases} \dot{\hat{x}}(t) = T^{-1}\dot{x}(t) = T^{-1} f(T\hat{x}(t), u(t), t) = \hat{f}(\hat{x}(t), u(t), t) \\ y(t) = g(T\hat{x}(t), u(t), t) = \hat{g}(\hat{x}(t), u(t), t) \end{cases}$$

where $\hat{f}$ and $\hat{g}$ are suitably defined functions.

# Equivalent state-space representations

Similarly, consider the discrete-time dynamic system:

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

Let

$$\hat{x} \doteq T^{-1} x$$

where $T \in \mathbb{R}^{n \times n}$ is a generic non-singular $n \times n$ matrix ($\det(T) \neq 0$).

Then, an equivalent state-space description is given by:

$$\begin{cases} \hat{x}(k+1) = T^{-1} x(k+1) = T^{-1} f(T\hat{x}(k), u(k), k) = \hat{f}(\hat{x}(k), u(k), k) \\ y(k) = g(T\hat{x}(k), u(k), k) = \hat{g}(\hat{x}(k), u(k), k) \end{cases}$$

where $\hat{f}$ and $\hat{g}$ are suitably defined functions.

# Linear systems

# Linear continuous-time dynamic systems

Consider the continuous-time dynamic system state-space representation:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases}$$

The system is said *linear* if the functions $f$ and $g$ are linear with respect to $x(t)$ and $u(t)$.

In that case, the state-space representation can be written as:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ \\ y(t) = C(t)x(t) + D(t)u(t) \end{cases}$$

where $A(t), B(t), C(t), D(t)$, are matrices of suitable dimensions, possibly depending on $t$.

# Linear continuous-time dynamic systems (cont.)

In particular:

$$A(t) = \begin{bmatrix} a_{11}(t) & \cdots & a_{1n}(t) \\ \vdots & \ddots & \vdots \\ a_{n1}(t) & \cdots & a_{nn}(t) \end{bmatrix} \quad B(t) = \begin{bmatrix} b_{11}(t) & \cdots & b_{1m}(t) \\ \vdots & \vdots & \vdots \\ b_{n1}(t) & \cdots & b_{nm}(t) \end{bmatrix}$$

$$C(t) = \begin{bmatrix} c_{11}(t) & \cdots & c_{1n}(t) \\ \vdots & \ddots & \vdots \\ c_{p1}(t) & \cdots & c_{pn}(t) \end{bmatrix} \quad D(t) = \begin{bmatrix} d_{11}(t) & \cdots & d_{1m}(t) \\ \vdots & \vdots & \vdots \\ d_{p1}(t) & \cdots & d_{pm}(t) \end{bmatrix}$$

# Linear time-invariant continuous-time dynamic systems

In the time-invariant case, the matrices $A(t), B(t), C(t), D(t)$ do not depend on $t$, i.e., they are constant matrices $A, B, C, D$:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \vdots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn} \end{bmatrix} \quad D = \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \vdots & \vdots \\ d_{p1} & \cdots & d_{pm} \end{bmatrix}$$

thus, the state-space representation takes the form:

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

# Equilibrium states

Given the time-invariant dynamic system:

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

let $u(t) = \bar{u},\ t \geq 0$ be a constant input.
The equilibrium states (if any) corresponding to $\bar{u}$ are the solutions of the linear equation:

$$0 = Ax + B\bar{u}$$

which is equivalent to

$$Ax = -B\bar{u}.$$

The following two cases have to be considered:

- $\det(A) \neq 0$
- $\det(A) = 0$

# Equilibrium states (cont.)

- $\det(A) \neq 0$. In this case, we get:

$$\bar{x} = -A^{-1}B\bar{u} \quad \Longrightarrow \quad \bar{x} \quad \text{is} \quad \textbf{unique} \; \forall\, \bar{u} \in \mathbb{R}^m$$

  Accordingly, the equilibrium output is given by:

$$\bar{y} = C\bar{x} + D\bar{u} = \left[-CA^{-1}B + D\right]\bar{u}$$

  The matrix $\left[-CA^{-1}B + D\right]$ is defined as the *static gain*.

- $\det(A) = 0$. In this case, depending on the rank of the block matrix $\begin{bmatrix} A & -B\bar{u} \end{bmatrix}$, either there are infinite solutions:

$$\text{rank}\begin{bmatrix} A & -B\bar{u} \end{bmatrix} = \text{rank}\,[A] \quad \Longrightarrow \quad \exists\,\infty\;\bar{x},\;\exists\,\infty\;\bar{y}$$

  or there is no solution:

$$\text{rank}\begin{bmatrix} A & -B\bar{u} \end{bmatrix} \neq \text{rank}\,[A] \quad \Longrightarrow \quad \nexists\;\bar{x},\,\nexists\;\bar{y}$$

# Equivalent state-space representations

Consider the continuous-time linear time-invariant (LTI) dynamic system:

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

Let $\hat{x} \doteq T^{-1}x$, where $T \in \mathbb{R}^{n\times n}$ is a generic non-singular $n \times n$ matrix ($\det(T) \neq 0$). Then, an equivalent state-space description is given by:

$$\begin{cases} \dot{\hat{x}}(t) = T^{-1}\dot{x}(t) = T^{-1}AT\hat{x}(t) + T^{-1}Bu(t) = \hat{A}\hat{x}(t) + \hat{B}u(t) \\ y(t) = CT\hat{x}(t) + Du(t) = \hat{C}\hat{x}(t) + Du(t) \end{cases}$$

Hence, for non-singular $T$, the following tuples are equivalent, meaning that they describe the same linear dynamic system:

$$(A, B, C, D) \qquad \Longleftrightarrow \qquad (T^{-1}AT, T^{-1}B, CT, D)$$

Consider the discrete-time dynamic system state-space representation:

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

The system is said *linear* if the functions $f$ and $g$ are linear with respect to $x(k)$ and $u(k)$.

In that case, the state-space representation can be written as:

$$\begin{cases} x(k+1) = A(k)x(k) + B(k)u(k) \\ \\ y(k) = C(k)x(k) + D(k)u(k) \end{cases}$$

where $A(k), B(k), C(k), D(k)$, are matrices of suitable dimensions, possibly depending on $k$.

In particular:

$$A(k) = \begin{bmatrix} a_{11}(k) & \cdots & a_{1n}(k) \\ \vdots & \ddots & \vdots \\ a_{n1}(k) & \cdots & a_{nn}(k) \end{bmatrix} \quad B(k) = \begin{bmatrix} b_{11}(k) & \cdots & b_{1m}(k) \\ \vdots & \vdots & \vdots \\ b_{n1}(k) & \cdots & b_{nm}(k) \end{bmatrix}$$

$$C(k) = \begin{bmatrix} c_{11}(k) & \cdots & c_{1n}(k) \\ \vdots & \ddots & \vdots \\ c_{p1}(k) & \cdots & c_{pn}(k) \end{bmatrix} \quad D(k) = \begin{bmatrix} d_{11}(k) & \cdots & d_{1m}(k) \\ \vdots & \vdots & \vdots \\ d_{p1}(k) & \cdots & d_{pm}(k) \end{bmatrix}$$

In the time-invariant case, the matrices $A(k), B(k), C(k), D(k)$ do not depend on $k$, i.e., they are constant matrices $A, B, C, D$:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \vdots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn} \end{bmatrix} \quad D = \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \vdots & \vdots \\ d_{p1} & \cdots & d_{pm} \end{bmatrix}$$

thus, the state-space representation takes the form:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ \\ y(k) = Cx(k) + Du(k) \end{cases}$$

# Equilibrium states

Given the linear time-invariant dynamic system:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ \\ y(k) = Cx(k) + Du(k) \end{cases}$$

let $u(k) = \bar{u},\ k \geq 0$ be a constant input. The equilibrium states (if any) corresponding to $\bar{u}$ are the solutions of the following linear equation:

$$x = Ax + B\bar{u}$$

which is equivalent to

$$(I - A)x = B\bar{u}$$

The following two cases have to be considered:

- $\det{(I - A)} \neq 0$
- $\det{(I - A)} = 0$

# Equilibrium states (cont.)

- $\det(I - A) \neq 0$. In this case, we get:

$$\bar{x} = (I - A)^{-1} B\bar{u} \quad \Longrightarrow \quad \bar{x} \text{ is } \textbf{unique } \forall\, \bar{u} \in \mathbb{R}^m$$

  Accordingly, the equilibrium output is given by:

$$\bar{y} = C\bar{x} + D\bar{u} = \left[C(I - A)^{-1}B + D\right]\bar{u}$$

  Matrix $\left[C(I - A)^{-1}B + D\right]$ is defined as *static gain*.

- $\det(I - A) = 0$. In this case, depending on the rank of the block matrix $[\,I - A \quad B\bar{u}\,]$, either there are infinite solutions:

$$\text{rank}\,[\,I - A \quad B\bar{u}\,] = \text{rank}\,[I - A] \quad \Longrightarrow \quad \exists\,\infty\ \bar{x},\ \exists\,\infty\ \bar{y}$$

  or there is no solution:

$$\text{rank}\,[\,I - A \quad B\bar{u}\,] \neq \text{rank}\,[I - A] \quad \Longrightarrow \quad \nexists\ \bar{x},\ \nexists\ \bar{y}$$

# Equivalent state-space representations

Consider the discrete-time linear time-invariant (LTI) dynamic system state-space representation:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

Let $\hat{x} \doteq T^{-1}x$, where $T \in \mathbb{R}^{n \times n}$ is a generic non-singular $n \times n$ matrix $(\det(T) \neq 0)$. Then, the equivalent state-space description is given by:

$$\begin{cases} \hat{x}(k+1) = T^{-1}x(k+1) = T^{-1}AT\hat{x}(k) + T^{-1}Bu(k) = \hat{A}\hat{x}(k) + \hat{B}u(k) \\ y(k) = CT\hat{x}(k) + Du(k) = \hat{C}\hat{x}(k) + Du(k) \end{cases}$$

Hence, for non-singular $T$, the following tuples are equivalent, meaning that they describe the same linear dynamic system:

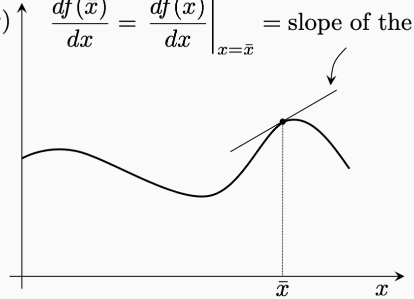
$$(A, B, C, D) \qquad \Longleftrightarrow \qquad (T^{-1}AT, T^{-1}B, CT, D)$$

# Linearization

**Important fact**

- **Linear systems** are provided with **powerful analytical tools** that are not available for nonlinear systems
- **Approximating nonlinear systems by linear ones** in a neighborhood of a state movement may result very useful in practice

$$y = f(x) \qquad \frac{df(\bar{x})}{dx} = \left.\frac{df(x)}{dx}\right|_{x=\bar{x}} = \text{slope of the tangent}$$




$$y(x) \approx y(\bar{x}) + \frac{df(\bar{x})}{dx}(x - \bar{x})$$

# Local linearization around an equilibrium

Consider the nonlinear time-invariant system:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases}$$

Let $f(\bar{x}, \bar{u}) = 0$, meaning that $\bar{x}$ is an equilibrium state associated to the constant input $u(t) = \bar{u}$.

Let us **perturb** the initial state and the input, thus getting a perturbed state movement:

$$\begin{aligned} x(0) &= \bar{x} + \delta\bar{x} \\ u(t) &= \bar{u} + \delta u(t) \end{aligned} \quad \Longrightarrow \quad x(t) = \bar{x} + \delta x(t)$$

where $\delta x(t)$ and $\delta u(t)$ represent the difference between the nominal and the perturbed state and input, respectively.

Hence:

$$\begin{aligned} \dot{\delta x}(t) = \dot{x}(t) = f(x(t), u(t))) &= f(\bar{x} + \delta x(t), \bar{u} + \delta u(t)) \\ &\simeq f(\bar{x}, \bar{u}) + \frac{\partial f(\bar{x}, \bar{u})}{\partial x}\delta x(t) + \frac{\partial f(\bar{x}, \bar{u})}{\partial u}\delta u(t) \end{aligned}$$

# Local linearization around an equilibrium (cont.)

Since $f(\bar{x}, \bar{u}) = 0$, it follows that

$$\begin{aligned}\dot{\delta x}(t) \simeq \quad & \frac{\partial f(\bar{x}, \bar{u})}{\partial x}\delta x(t) + \frac{\partial f(\bar{x}, \bar{u})}{\partial u}\delta u(t) \\ & = A\delta x(t) + B\delta u(t)\end{aligned}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are constant matrices defined as:

$$A = \frac{\partial f(\bar{x}, \bar{u})}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=\bar{x}, u=\bar{u}}$$

$$B = \frac{\partial f(\bar{x}, \bar{u})}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_{x=\bar{x}, u=\bar{u}}$$

# Local linearization around an equilibrium (cont.)

Concerning the perturbed output, we have:

$$\bar{y} = g(\bar{x}, \bar{u}) \quad \text{and} \quad \delta y(t) \doteq y(t) - \bar{y}$$

Hence

$$\begin{aligned} y(t) &= g(x(t), u(t)) = g(\bar{x} + \delta x(t), \bar{u} + \delta u(t)) \\ &\simeq g(\bar{x}, \bar{u}) + \frac{\partial g(\bar{x}, \bar{u})}{\partial x} \delta x(t) + \frac{\partial g(\bar{x}, \bar{u})}{\partial u} \delta u(t) \end{aligned}$$

and then

$$\begin{aligned} \delta y(t) \simeq & \quad \frac{\partial g(\bar{x}, \bar{u})}{\partial x} \delta x(t) + \frac{\partial g(\bar{x}, \bar{u})}{\partial u} \delta u(t) \\ &= C\delta x(t) + D\delta u(t) \end{aligned}$$

where $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are constant matrices defined as follows.

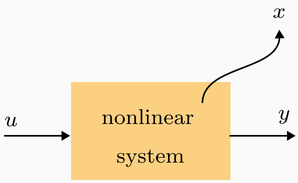
# Local linearization around an equilibrium (cont.)

$$C = \frac{\partial g(\bar{x}, \bar{u})}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_p}{\partial x_1} & \cdots & \frac{\partial g_p}{\partial x_n} \end{bmatrix}_{x=\bar{x}, u=\bar{u}}$$

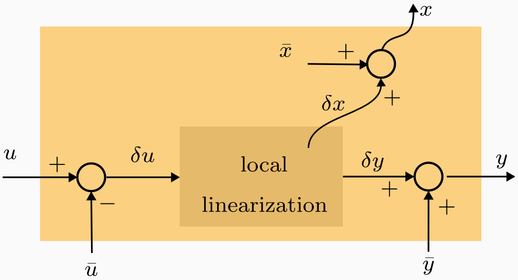
$$D = \frac{\partial g(\bar{x}, \bar{u})}{\partial u} = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial g_p}{\partial u_1} & \cdots & \frac{\partial g_p}{\partial u_m} \end{bmatrix}_{x=\bar{x}, u=\bar{u}}$$

In summary, the linear time-invariant system obtained by linearization in the neighborhood of the equilibrium state $\bar{x}$ corresponding to the constant input $u(t) = \bar{u}$, $t \geq 0$ is

$$\begin{cases} \delta\dot{x}(t) = A\delta x(t) + B\delta u(t) \\ \delta y(t) = C\delta x(t) + D\delta u(t) \end{cases}$$




Nonlinear system.



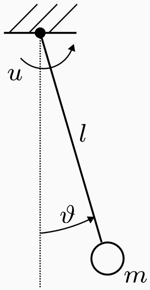

Local approximation obtained from a local linearization.

Note that the linearized system describes the behavior of the nonlinear system in terms of **perturbations**. The input, output and state of the local linearization system are the perturbations with respect to $\bar{u}$, $\bar{y}$, and $\bar{x}$.

# Example

Consider the pendulum: by taking $x_1 = \vartheta$, $x_2 = \dot{\vartheta}$, $y = \vartheta$ and $J = ml^2$, we get:

$$\begin{cases} \dot{x}_1(t) & = & x_2(t) \\ \dot{x}_2(t) & = & -\dfrac{mgl}{J}\sin(x_1(t)) - \dfrac{h}{J}x_2(t) + \dfrac{1}{J}u(t) \\ y(t) & = & x_1(t) \end{cases}$$




To find the equilibrium states corresponding to $u(t) = \bar{u} = 0$ we solve

$$\begin{cases} 0 & = & x_2 \\ 0 & = & -\dfrac{mgl}{J}\sin(x_1) \end{cases}$$

obtaining $\bar{x} = \begin{bmatrix} k\pi \\ 0 \end{bmatrix}$, $k \in \mathbb{Z}$.

By computing the partial derivatives we get

$$\frac{\partial f(\bar{x},\bar{u})}{\partial x} = \left[ \begin{array}{cc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{array} \right]_{\bar{x},\bar{u}} = \left[ \begin{array}{cc} 0 & 1 \\ -\frac{mgl}{J}\cos(x_1) & -\frac{h}{J} \end{array} \right]_{\bar{x},\bar{u}}$$

and by substituting the pair $\bar{x},\bar{u}$ we obtain

$$A^{(\mathrm{e})} = \left[ \begin{array}{cc} 0 & 1 \\ -\frac{mgl}{J} & -\frac{h}{J} \end{array} \right]$$

for $k$ even, and

$$A^{(\mathrm{o})} = \left[ \begin{array}{cc} 0 & 1 \\ \frac{mgl}{J} & -\frac{h}{J} \end{array} \right]$$

for $k$ odd.

# Example (cont.)

The other matrices do not depend on $\bar{x}$:

$$B = \frac{\partial f(\bar{x},\bar{u})}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix}_{\bar{x},\bar{u}} = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix}$$

$$C = \frac{\partial g(\bar{x},\bar{u})}{\partial x} = \begin{bmatrix} \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} \end{bmatrix}_{\bar{x},\bar{u}} = \begin{bmatrix} 1 & 0 \end{bmatrix}_{\bar{x},\bar{u}} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$D = \frac{\partial g(\bar{x},\bar{u})}{\partial u} = \left.\frac{\partial g}{\partial u}\right|_{\bar{x},\bar{u}} = 0$$

# Local linearization around a state movement

Often it is convenient to consider perturbations around arbitrary state movements. Given the nonlinear system:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases}$$

let $\bar{x}(t),\ t \geq 0$ be an **arbitrary state movement**, obtained from the initial state $x(0) = \bar{x}_0$ by applying the input $u(t) = \bar{u}(t),\ t \geq 0$.

Let us perturb the initial state and the input, thus obtaining a perturbed state movement:

$$\begin{matrix} x(0) = \bar{x}_0 + \delta x_0 \\ u(t) = \bar{u}(t) + \delta u(t) \end{matrix} \implies x(t) = \bar{x}(t) + \delta x(t)$$

where $\delta x(t)$ and $\delta u(t)$ represent the difference between the nominal and the perturbed state and input, respectively.

Hence:

$$\dot{\delta x}(t) = \dot{x}(t) - \dot{\bar{x}}(t) = f(x(t), u(t)) - f(\bar{x}(t), \bar{u}(t)).$$

# Local linearization around a state movement (cont.)

On the other hand, by using Taylor's approximation:

$$f(x(t),u(t)) \approx f(\bar{x}(t),\bar{u}(t)) + \frac{\partial f(\bar{x}(t),\bar{u}(t))}{\partial x}\delta x(t) + \frac{\partial f(\bar{x}(t),\bar{u}(t))}{\partial u}\delta u(t)$$

By substituting, we obtain

$$\dot{\delta x}(t) \approx \frac{\partial f(\bar{x}(t),\bar{u}(t))}{\partial x}\delta x(t) + \frac{\partial f(\bar{x}(t),\bar{u}(t))}{\partial u}\delta u(t).$$

The main difference with respect to the linearization around an equilibrium, is that the partial derivatives are now computed along $\bar{x}(t)$ and $\bar{u}(t)$. As a consequence, the linearization leads, in general, to a time **variant** system:

$$\begin{aligned}\dot{\delta x}(t) \quad &\approx \frac{\partial f(\bar{x}(t),\bar{u}(t))}{\partial x}\delta x(t) + \frac{\partial f(\bar{x}(t),\bar{u}(t))}{\partial u}\delta u(t)\\ &= A(t)\delta x(t) + B(t)\delta u(t)\end{aligned}$$

# Local linearization around a state movement (cont.)

where $A(t) \in \mathbb{R}^{n\times n}$, $B(t) \in \mathbb{R}^{n\times m}$ are matrices defined as:

$$A(t) = \frac{\partial f(\bar{x}(t), \bar{u}(t))}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=\bar{x}(t), u=\bar{u}(t)}$$

$$B(t) = \frac{\partial f(\bar{x}(t), \bar{u}(t))}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_{x=\bar{x}(t), u=\bar{u}(t)}$$

# Local linearization around a state movement (cont.)

Concerning the perturbed output, we have:

$$\bar{y}(t) = g(\bar{x}(t), \bar{u}(t)) \quad \text{and} \quad \delta y(t) \doteq y(t) - \bar{y}(t)$$

Hence

$$\begin{aligned} y(t) &= g(x(t), u(t)) = g(\bar{x}(t) + \delta x(t), \bar{u}(t) + \delta u(t)) \\ &\simeq g(\bar{x}(t), \bar{u}(t)) + \frac{\partial g(\bar{x}(t), \bar{u}(t))}{\partial x}\delta x(t) + \frac{\partial g(\bar{x}(t), \bar{u}(t))}{\partial u}\delta u(t) \end{aligned}$$

and then

$$\begin{aligned} \delta y(t) \simeq & \quad \frac{\partial g(\bar{x}(t), \bar{u}(t))}{\partial x}\delta x(t) + \frac{\partial g(\bar{x}(t), \bar{u}(t))}{\partial u}\delta u(t) \\ & = C(t)\delta x(t) + D(t)\delta u(t) \end{aligned}$$

where $C(t) \in \mathbb{R}^{p \times n}$, $D(t) \in \mathbb{R}^{p \times m}$ are matrices defined as follows.

# Local linearization around a state movement (cont.)

$$C(t) = \frac{\partial g(\bar{x}(t), \bar{u}(t))}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_p}{\partial x_1} & \dots & \frac{\partial g_p}{\partial x_n} \end{bmatrix}_{x=\bar{x}(t), u=\bar{u}(t)}$$

$$D(t) = \frac{\partial g(\bar{x}(t), \bar{u}(t))}{\partial u} = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \dots & \frac{\partial g_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial g_p}{\partial u_1} & \dots & \frac{\partial g_p}{\partial u_m} \end{bmatrix}_{x=\bar{x}(t), u=\bar{u}(t)}$$

In summary, the linear system obtained by linearization around a state movement $\bar{x}(t)$ corresponding to the input $u(t) = \bar{u}(t)$, $t \geq 0$ is

$$\begin{cases} \dot{\delta x}(t) = A(t)\delta x(t) + B(t)\delta u(t) \\ \delta y(t) = C(t)\delta x(t) + D(t)\delta u(t) \end{cases}$$

# Local linearization around an equilibrium (discrete-time case)

Consider the nonlinear time-invariant system:

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = g(x(k), u(k)) \end{cases}$$

Let $\bar{x} = f(\bar{x}, \bar{u})$, meaning that $\bar{x}$ is an equilibrium state associated to the constant input $u(k) = \bar{u},\ k \geq 0$.

Let us perturb the initial state and the input, thus getting a perturbed state movement:

$$x(0) = \bar{x} + \delta\bar{x}\,;\ u(k) = \bar{u} + \delta u(k) \implies x(k) = \bar{x} + \delta x(k)$$

Hence:

$$\begin{aligned} x(k+1) = \bar{x} + \delta x(k+1) &= f(\bar{x} + \delta x(k), \bar{u} + \delta u(k)) \\ &\simeq f(\bar{x}, \bar{u}) + \frac{\partial f(\bar{x}, \bar{u})}{\partial x}\delta x(k) + \frac{\partial f(\bar{x}, \bar{u})}{\partial u}\delta u(k) \end{aligned}$$

# Local linearization around an equilibrium (discrete-time case) (cont.)

Since $\bar{x} = f(\bar{x}, \bar{u})$, it follows that

$$\begin{aligned}\delta x(k+1) \simeq & \frac{\partial f(\bar{x}, \bar{u})}{\partial x} \delta x(k) + \frac{\partial f(\bar{x}, \bar{u})}{\partial u} \delta u(k) \\ & = A \delta x(k) + B \delta u(k)\end{aligned}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are constant matrices defined as:

$$A = \frac{\partial f(\bar{x}, \bar{u})}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=\bar{x}, u=\bar{u}}$$

$$B = \frac{\partial f(\bar{x}, \bar{u})}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_{x=\bar{x}, u=\bar{u}}$$

Concerning the perturbed output we have:

$$\bar{y} = g(\bar{x}, \bar{u})\,; \quad y(k) = \bar{y} + \delta y(k)$$

Hence

$$\begin{aligned} y(k) &= g(x(k), u(k)) = g(\bar{x} + \delta x(k), \bar{u} + \delta u(k)) \\ &\simeq g(\bar{x}, \bar{u}) + g_x(\bar{x}, \bar{u})\delta x(k) + g_u(\bar{x}, \bar{u})\delta u(k) \end{aligned}$$

and then

$$\begin{aligned} \delta y(k) \simeq & \quad \frac{\partial g(\bar{x}, \bar{u})}{\partial x}\delta x(k) + \frac{\partial g(\bar{x}, \bar{u})}{\partial u}\delta u(k) \\ & = C\delta x(k) + D\delta u(k) \end{aligned}$$

where $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are constant matrices defined as follows.

# Local linearization around an equilibrium (discrete-time case) (cont.)

$$C = \frac{\partial g(\bar{x},\bar{u})}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_p}{\partial x_1} & \cdots & \frac{\partial g_p}{\partial x_n} \end{bmatrix}_{x=\bar{x},u=\bar{u}}$$

$$D = \frac{\partial g(\bar{x},\bar{u})}{\partial u} = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial g_p}{\partial u_1} & \cdots & \frac{\partial g_p}{\partial u_m} \end{bmatrix}_{x=\bar{x},u=\bar{u}}$$

In summary, the linear time-invariant system obtained by linearization in the neighborhood of a given equilibrium state $\bar{x}$ corresponding to the constant input $u(k) = \bar{u}$ is

$$\begin{cases} \delta x(k+1) = A\delta x(k) + B\delta u(k) \\ \delta y(k) = C\delta x(k) + D\delta u(k) \end{cases}$$

# Example

Consider the nonlinear discrete-time system:

$$\begin{cases} x_1(k+1) = x_1(k) + \alpha(1-\beta x_1(k))x_1(k) - \gamma x_1(k)x_2(k) + u(k) \\ x_2(k+1) = x_2(k) - \zeta x_2(k) + \eta x_1(k)x_2(k) \\ y(k) = x_2(k) \end{cases}$$

By imposing the constant input sequence $\bar{u}(k) = 0$ the following equilibrium states are obtained:

$$\bar{x}^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; ; \; \bar{x}^{(2)} = \begin{bmatrix} \frac{1}{\beta} \\ 0 \end{bmatrix} \; ; \; \bar{x}^{(3)} = \begin{bmatrix} \frac{\zeta}{\eta} \\ \frac{\alpha}{\gamma}\left(1 - \frac{\beta\zeta}{\eta}\right) \end{bmatrix}$$

The general expression for matrix $A$ of the linearized system is:

$$f_x(\bar{x},\bar{u}) = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} \end{bmatrix}_{\bar{x},\bar{u}} =$$

$$= \begin{bmatrix} (1+\alpha-2\alpha\beta x_1-\gamma x_2) & -\gamma x_1 \\ \eta x_2 & 1-\zeta+\eta x_1 \end{bmatrix}_{\bar{x},\bar{u}}$$

Substituting the expressions of the specific equilibrium states we get:

$$\bar{x}^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \implies A^{(1)} = \begin{bmatrix} (1+\alpha) & 0 \\ 0 & 1-\zeta \end{bmatrix}$$

# Example (cont.)

$$\bar{x}^{(2)} = \begin{bmatrix} \frac{1}{\beta} \\ 0 \end{bmatrix} \implies A^{(2)} = \begin{bmatrix} (1-\alpha) & -\frac{\gamma}{\beta} \\ 0 & 1-\zeta+\frac{\eta}{\beta} \end{bmatrix}$$

$$\bar{x}^{(3)} = \begin{bmatrix} \frac{\zeta}{\eta} \\ \frac{\alpha}{\gamma}\left(1-\frac{\beta\zeta}{\eta}\right) \end{bmatrix} \implies A^{(3)} = \begin{bmatrix} \left(1-\frac{\alpha\beta\zeta}{\eta}\right) & -\frac{\gamma\zeta}{\eta} \\ \frac{\alpha\eta}{\gamma}\left(1-\frac{\beta\zeta}{\eta}\right) & 1 \end{bmatrix}$$

# Example (cont.)

Finally, the other matrices $B$, $C$, and $D$ of the linearized systems are given by (their values do not depend on the specific equilibrium states):

$$B = f_u(\bar{x}, \bar{u}) = \begin{bmatrix} \dfrac{\partial f_1}{\partial u} \\ \dfrac{\partial f_2}{\partial u} \end{bmatrix}_{\bar{x}, \bar{u}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = g_x(\bar{x}, \bar{u}) = \begin{bmatrix} \dfrac{\partial g}{\partial x_1} & \dfrac{\partial g}{\partial x_2} \end{bmatrix}_{\bar{x}, \bar{u}} = \begin{bmatrix} 0 & 1 \end{bmatrix}_{\bar{x}, \bar{u}} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$D = g_u(\bar{x}, \bar{u}) = \left.\frac{\partial g}{\partial u}\right|_{\bar{x}, \bar{u}} = 0_{\bar{x}, \bar{u}} = 0$$

# Feedback linearization

Linearizing around an equilibrium or a movement is not the only way to get a linear system out of a nonlinear one.

A remarkable alternative is the *feedback linearization*, based on the idea of **employing the input to cancel out the nonlinearities**.

In the following, we present the approach by restricting our attention to mechanical systems.

The equations of motions of many mechanical systems can be written as follows (for simplicity we drop the dependence of $q$ and $F$ on time):

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + G(q) = F, \tag{5}$$

where:

- $q \in \mathbb{R}^k$ is the *generalized coordinate vector*,
- $M(q) \in \mathbb{R}^{k\times k}$ is the *mass matrix* (symmetric and positive definite),
- $F \in \mathbb{R}^k$ is the *applied forces vector*,
- $G(q) \in \mathbb{R}^k$ is the *conservative forces vector*,
- $B(q, \dot{q}) \in \mathbb{R}^{k\times k}$ is the *centrifugal/Coriolis/friction matrix*

The equation of motion (5) is valid of systems that follow the classical Newton-Euler mechanics or Lagrangian mechanics with a kinetic energy that is quadratic in the derivative of the generalized coordinates and a potential energy that may depend on the generalized coordinates but not on their derivative. Such systems include robot arms, mobile robots, airplanes, helicopters, underwater vehicles and many more.

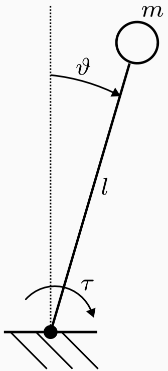
### Example: inverted pendulum

The equation:

$$ml^2\ddot{\vartheta}(t) = mgl\sin\vartheta(t) - b\dot{\vartheta}(t) + \tau(t)$$

(where $b$ is a friction coefficient, $g$ is the gravitational acceleration, and $\tau$ denotes a torque) can be written as (5) provided that we define:

$$q \doteq \vartheta, \qquad F \doteq \tau, \qquad M(q) \doteq ml^2, \qquad B(q) \doteq b, \qquad G(q) \doteq -mgl\sin\vartheta$$



### Example: two-link robot manipulator

The dynamics of the robot arm having two revolute joints shown in figure can be written as (5) provided that we define:

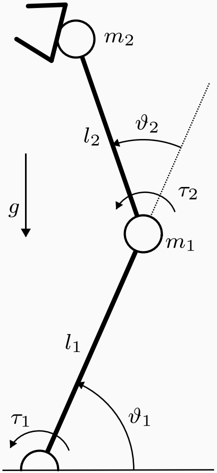
$$q \doteq \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \end{bmatrix}, \qquad F \doteq \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

where $\tau_1$ and $\tau_2$ denote the torques applied at the joints. We have (see Craig (1989), Example 6.3):

$$M(q) = \begin{bmatrix} m_2 l_2^2 + 2 m_2 l_1 l_2 \cos\vartheta_2 + (m_1 + m_2) l_1^2 & m_2 l_2^2 + m_2 l_1 l_2 \cos\vartheta_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos\vartheta_2 & m_2 l_2^2 \end{bmatrix}$$

$$B(q, \dot{q}) = \begin{bmatrix} -2 m_2 l_1 l_2 \dot{\vartheta}_2 \sin\vartheta_2 & -m_2 l_1 l_2 \dot{\vartheta}_2 \sin\vartheta_2 \\ m_2 l_1 l_2 \dot{\vartheta}_1 \sin\vartheta_2 & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} m_2 g l_2 \cos(\vartheta_1 + \vartheta_2) + (m_1 + m_2) g l_1 \cos\vartheta_1 \\ m_2 g l_2 \cos(\vartheta_1 + \vartheta_2) \end{bmatrix}$$



Back to the equation of motion:

$$M(q)\ddot{q} + B(q,\dot{q})\dot{q} + G(q) = F,$$

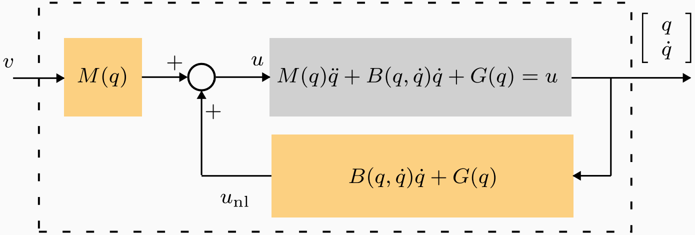
assume that the system is *fully actuated*, meaning that **one has the control over the whole vector of generalized coordinates**. In other words, all the components of $F(t)$ can be manipulated. In that case $F$ can be regarded as the input. Let $F$ be chosen as:

$$F = \underbrace{B(q,\dot{q})\dot{q} + G(q)}_{\doteq u_{\mathsf{nl}}(q,\dot{q})} + M(q)v,$$

where $v$ is a newly introduced auxiliary variable and $u_{\mathsf{nl}}(q,\dot{q})$ is a feedback signal whose purpose is to cancel the nonlinear terms. By substituting, we obtain

$$M(q)\ddot{q} + B(q,\dot{q})\dot{q} + G(q) = B(q,\dot{q})\dot{q} + G(q) + M(q)v \quad \Leftrightarrow \quad M(q)\ddot{q} = M(q)v \quad \Leftrightarrow \quad \ddot{q} = v,$$

where the last equivalence follows from the non-singularity of $M$.

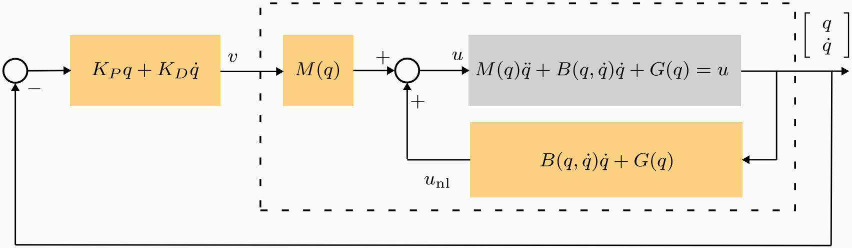



In practice, the nonlinear process has been transformed into a (linear) double integrator:

$$\ddot{q}(t) = v(t).$$

To get a state-space representation we define $x(t) \doteq \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} \in \mathbb{R}^{2k}$, thus obtaining (choosing the generalized coordinates as the output):

$$\dot{x}(t) = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} v, \qquad y(t) = \begin{bmatrix} I & 0 \end{bmatrix} x(t).$$




Now, linear methods can be employed to find a controller for $v$ that results in adequate closed-loop performance for $y = q$. For example, we can employ a proportional-derivative (PD) controller:

$$v(t) = -K_P q(t) - K_D \dot{q}(t) = -\begin{bmatrix} K_P & K_D \end{bmatrix} x(t),$$

leading to the closed-loop dynamics:

$$\dot{x}(t) = \left( \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} K_P & K_D \end{bmatrix} \right) x(t) = \begin{bmatrix} 0 & I \\ -K_P & -K_D \end{bmatrix} x(t), \qquad y(t) = \begin{bmatrix} I & 0 \end{bmatrix} x(t).$$

# References

# References

Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control, 2nd edition*. Addison Wesley.

Ljung, L. and Glad, T. (1994). *Modeling of dynamic systems*. Prentice-Hall, Inc.

322MI –Spring 2023

Lecture 1
Generalities: systems, models, and control

END