System Representation and Charactization

Course Feedback Control and Real-time Systems

Master of Science in Informatics at Grenoble
Univ. Grenoble Alpes, Laboratory Verimag
`thao.dang@univ-grenoble-alpes.fr`

Modelling

A model is a **description** of a (physical, biological, economical, *etc*...) phenomenon, in a given language (for example, mathematical language).

A model is defined by a collection of **variables** et describes their evolution over time:

- **Predict** the values of the variables
- **Explain** complex phenomena from simpler or more general phenomena/principles

Modelling steps

- **Formalisation**: define the input and output variables, and the equations describing their relations. The equations may contain parameters.
- **Identification**: determine the parameter values in a given context
- **Validation**: verify if the model is coherent with the observations

Simulation: solving the equations to find the relations between the input and output variables. The resolution can be analytical, numerical, *etc*.

Other usages of models: design of controllers, formal verification of properties, code generation

Signal and System

A signal is an application from time to a domain $X : T \to D_X$

- T can be either continuous in R, or logical/discrete N, Z
- D_X specifies the signal type, R, N, $Bool$

A system is a **signal transformer**: $S : (T \to D_X) \to (T \to D_Y)$

Example 1: A modem transforms a binary signal into a continuous electrical signal (system in **open loop**, *i.e.* the output is determined directly from the input)

Example 2: A thermostat is a system in **closed loop** (with a feedback loop from the output to the input)

System - Input/Output

A system: establishes a cause-effect link between the **input signals** (excitations) and the **output signals** (responses).

Among the inputs, we distinghuishes:

- the **controls**
- the **disturbances**

Defining a dynamical system (1)

A system is a transformer of signals

$S : (T \to D_X) \to (T \to D_Y)$

To define a system

- **Identify the inputs and outputs**: a plant (inputs: raw material, outputs: products), a computer (input/output: information coming from the input/output interfaces)
- **Choose the types of the input/output signals**: D_X et D_Y
- **Choose the domain of time** T: Z, N (discrete time), or R, R_+ (continuous time), or collection of moments at which some events occur

Defining a dynamical system (2)

Define directly the function S is **difficult**!!

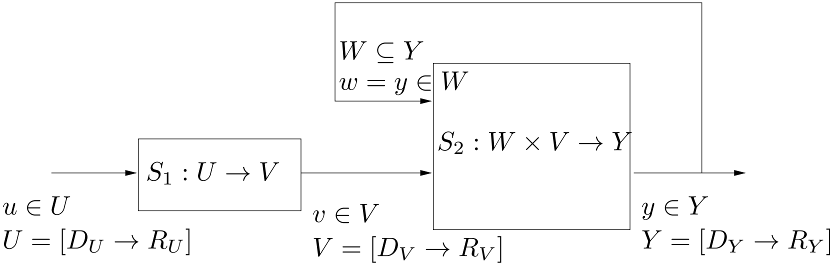
We need to use associated analysis tools

- For continuous-time systems: differential and integral calculus
- For discrete-time systems: algebra

Composition using block diagrams (1)

- Block diagrams: graphical description of connections between the components. Each component is associated with a function of signal transformation
- Connection $\Rightarrow$ composition of functions
- Hierarchical, easy to understand

Composition using block diagrams (2)

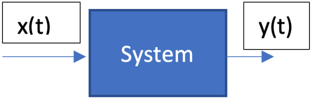


The global systeml $S_3 : U \to Y$ t.q.

$\forall u \in U : S_3(u) = S_2(S_3(u), S_1(u))$

The connection between y et w is called **'feedback'**. We need to solve the equation $z = S_2(z, S_1(z))$

Properties, characteristics of a system (1)



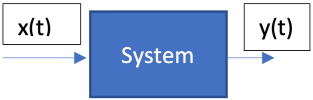
Linear systems vs non-linear systems

A system is **linear** iff it satisfies the following properties:

- Properties of **additivity**: If the input is $x_1(t)$, the output is $y_1(t)$. If the input is $x_2(t)$, the output is $y_2(t)$. Thus, if the input is $x(t) = x_1(t) + x_2(t)$, the output is $y(t) = y_1(t) + y_2(t)$.
- Proprerties of **homogeneity** :
  If the input is $x_1(t)$, the output is $y_1(t)$. Thus, for $\forall \alpha \neq 0$, if the input is $x(t) = \alpha x_1(t)$, the output is $y(t) = \alpha y_1(t)$.

Properties, characteristics of a system (2)

Stationary systems (time-invariant)



More formally, we say that the system commutes with a delay:

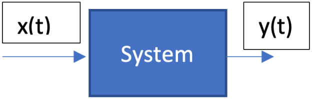
$$S(x(t-\delta)) = (Sx)(t-\delta)$$

The linear stationary systems form a class important historically and practically

Properties, characteristics of a system (3)

Causal system

Principle of causality: the effects should not precede the causes.

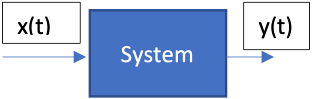


If the input $x(t)$ is nul for $t < 0$, then the output $y(t)$ is also nul for $t < 0$.

Properties, characteristics of a system (4)

Instantenous system vs dynamical system

Instantenous system (without memory or static): at a given instant, the output depends only on the input at that instant



For example, $y(t) = a(t)x(t)$ defines a static system.

Dynamical system: non-static, with memory

Properties, characteristics of a system (5)

Dynamical system In continuous system, memory is formalised by an integrator. The input/output relation is described by differential equations involving $y(t)$ and their derivatives

$$y'(t) = lim_{h \to 0} \frac{(y(t) - y(t-h))}{h}$$

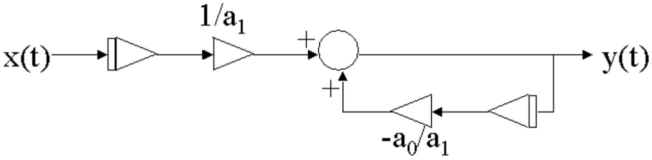
- $y'(t)$: information about the growth
- $y(t)$: present instant
- $y(t-h)$: past instant

Dynamical system - example

Consider a continuous dynamical system described by the following first-order linear differential equation:

$$a_1 y'(t) + a_0 y(t) = x(t)$$

$$y'(t) = -\frac{a_0}{a_1} y(t) + \frac{1}{a_1} x(t)$$



This schema is not optimised in the sense that one single integrator would suffice

Representation mode

For a continuous system:

- Differential equations
- Functional representation
- State space representation
- Representation in a transformed space, for example via the Laplace transform

Functional representation

The **functional schema** is deduced from differential equations and allows a more direct way to numerical simulation

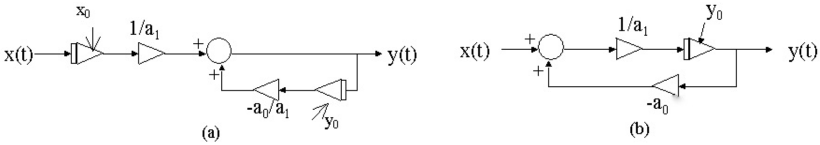
- It is a **program** in a **graphical language** connecting the **functional blocks**
- A **compiler** translates this schema into a computer program for the **numerical resolution of differential equations**
- To describe a linear continuous system, we need the following functional blocks: **Gain**, **Sum/Substraction**, **integrators** (memory blocks).
- This representation is not unique

Functional representation - example

Consider a continuous dynamical system described by:

$$a_1 y'(t) + a_0 y(t) = x(t)$$

$$y'(t) = -\frac{a_0}{a_1} y(t) + \frac{1}{a_1} x(t)$$



Integration uses initial conditions:

$$y(t) = \int_0^t y'(t)dt + y_0$$

How to obtain a functional representation (1)

Obtain systematically a functional representation associated with a differential equation ($x(t)$ is input and $y(t)$ is output)

$$
\begin{aligned}
\frac{d^n y(t)}{dt^n} \quad = \quad & -a_{n-1} \frac{d^{n-1} y(t^{n-1})}{dt} - \ldots - a_1 \frac{dy(t)}{dt} - a_0 y(t) \\
& + b_m \frac{d^m x(t)}{dt^m} + \ldots + b_1 \frac{dx(t)}{dt} + b_0 x(t)
\end{aligned}
$$

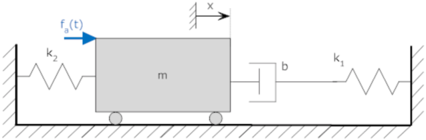
Example 1:

$$
\frac{d^2 y(t)}{dt^2} = -a_1 \frac{dy(t)}{dt} - a_0 y(t) + b_0 x(t)
$$

How to obtain a functional representation (2)

Example 2:

$f_a(t)$ as input and $x(t)$ as output.



The system is represented by the differential equation:

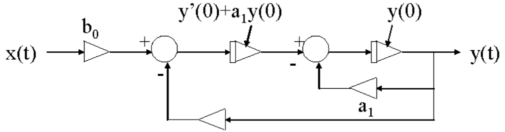
$$mb\frac{d^3x(t)}{dt^3} + mk_1\frac{d^2x(t)}{dt^2} + b(k_1+k_2)\frac{dx(t)}{dt} + k_1k_2x(t) = b\frac{df_a(t)}{dt} + k_1f_a(t)$$

How to obtain a functional representation (3)

$$\frac{d^2y(t)}{dt^2} = -a_1\frac{dy(t)}{dt} - a_0y(t) + b_0x(t)$$

We write $u_2(t) = y(t)$ and $u_1'(t) = -a_0y(t) + b_0x(t)$. Hence,
$u_2'' = -a_1u_2' + u_1'$.
After the first integration $u_2' = -a_1u_2 + u_1$.
Now in integrating $u_2'(t)$ we obtain $y(t)$

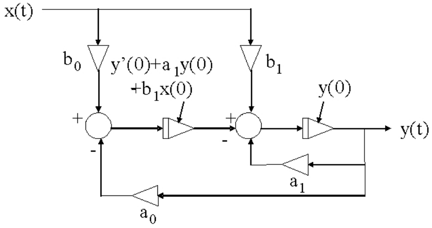
$$\begin{aligned} u_1'(t) &= -a_0y(t) + b_0x(t) \\ u_2'(t) &= -a_1u_2(t) + u_1(t) \\ y(t) &= u_2(t) \end{aligned}$$



How to obtain a functional representation (4)

$$\frac{d^2y(t)}{dt^2} = -a_1\frac{dy(t)}{dt} - a_0y(t) + b_1\frac{dx(t)}{dt} + b_0x(t)$$

We write $u_1'(t) = -a_0y(t) + b_0x(t)$ et $u_2(t) = y(t)$. Hence, $u_2'' = -a_1u_2' + b_1x' + u_1'$.



State space representation (1)

Notion of state

- To specify the function S, we often need a **collection** $\mathcal{X}$ of **internal states**.
- ***More formally***, the state is a vector containing a minimal number of variables such that:
  The initial output value $y(t_0)$ is known $\Rightarrow$ for all $t > t_0$,
  $y(t)$ can be determined uniquely if the input $x(t)$ is known for the interval $[t_0, t]$

State space representation (2)

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

$$\begin{aligned} v(t) &= \frac{1}{C}\int_{-\infty}^{t} i(\tau)d\tau = \frac{1}{C}\int_{-\infty}^{t_0} i(\tau)d\tau + \frac{1}{C}\int_{t_0}^{t} i(\tau)d\tau \\ &= v(t_0) + \frac{1}{C}\int_{t_0}^{t} i(\tau)d\tau \end{aligned}$$

- Specifying $v(t_0)$ is more "economical" than specifying all the evolution $i(t)$ from $t = -\infty$ to $t = t_0$
- The state at the instant t_0 of the system must form the memory of the system
- The state can be a representation *more compact* than the complete history of the system.

State space representation (3)

- The state at the instant t_0 of the system must form the minimal memory of the past, necessary to determine the future
- The state represented by the internal variables provides a complete description of the evolution of the system
- This formalism allows **transforming** all the linear differential equations of order n into a system of differential equations of order 1.
- **The choice of state representation is not unique**

How to obtain a state space representation

Consider the precedent example

$$\frac{d^2y(t)}{dt^2} = -a_1\frac{dy(t)}{dt} - a_0y(t) + b_1\frac{dx(t)}{dt} + b_0x(t)$$

We have set

$$\begin{aligned} u_1'(t) &= -a_0y(t) + b_0x(t) = -a_0u_2(t) + b_0x(t) \\ u_2'(t) &= -a_1u_2(t) + b_1x(t) + u_1(t) \\ y(t) &= u_2(t) \end{aligned}$$

In a matrix form

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} 0 & -a_0 \\ 1 & -a_1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} x$$

and

$$y = (0 \;\; 1) \begin{pmatrix} u_1' \\ u_2' \end{pmatrix}$$

State space representation

$$
\begin{aligned}
u' &= Au + Bx \\
y &= Cu + Dx
\end{aligned}
$$

- The matrix A: **state matrix**, of dimension $n \times n$
- The matrix B: **input matrix**, of dimension $n \times p$
- The matrix C: **output matrix**, of dimension $q \times n$
- The matrix D: **coupling matrix**, of dimension $q \times p$

From a structural viewpoint

System of first order (second ordre, …):

differential			recurrent, automaton, object program		
$X(0)$			$X(0)$		
X'	$=$	$F(X,U)$	X_{n+1}	$=$	$F(X_n, U_{n+1})$
Y	$=$	$G(X,U)$	Y_n	$=$	$G(X_n, U_n)$

Order of the system: **dimension of X**

Remark: not intrinsic

Finite-state system: Automaton, finit-state machine