Faculty of Automotive and Construction Machinery Engineering

WARSAW UNIVERSITY OF TECHNOLOGY

Theory of Machines and Automatic Control

Winter 2019/2020

Lecturer: Sebastian Korczak, PhD Eng.

Lecture 15

Modern control theory overview.

Control theory

Classical control theory	modern control theory (1950-now)
single input, single output (SISO)	multiple input, multiple output (MIMO)
usually linear systems	often nonlinear systems
time independent systems	time dependent systems
description by a transfer functions	description by a state equations
time and frequency domain analysis	time domain analysis
system response is the most important	system state is the most important

State-space representation

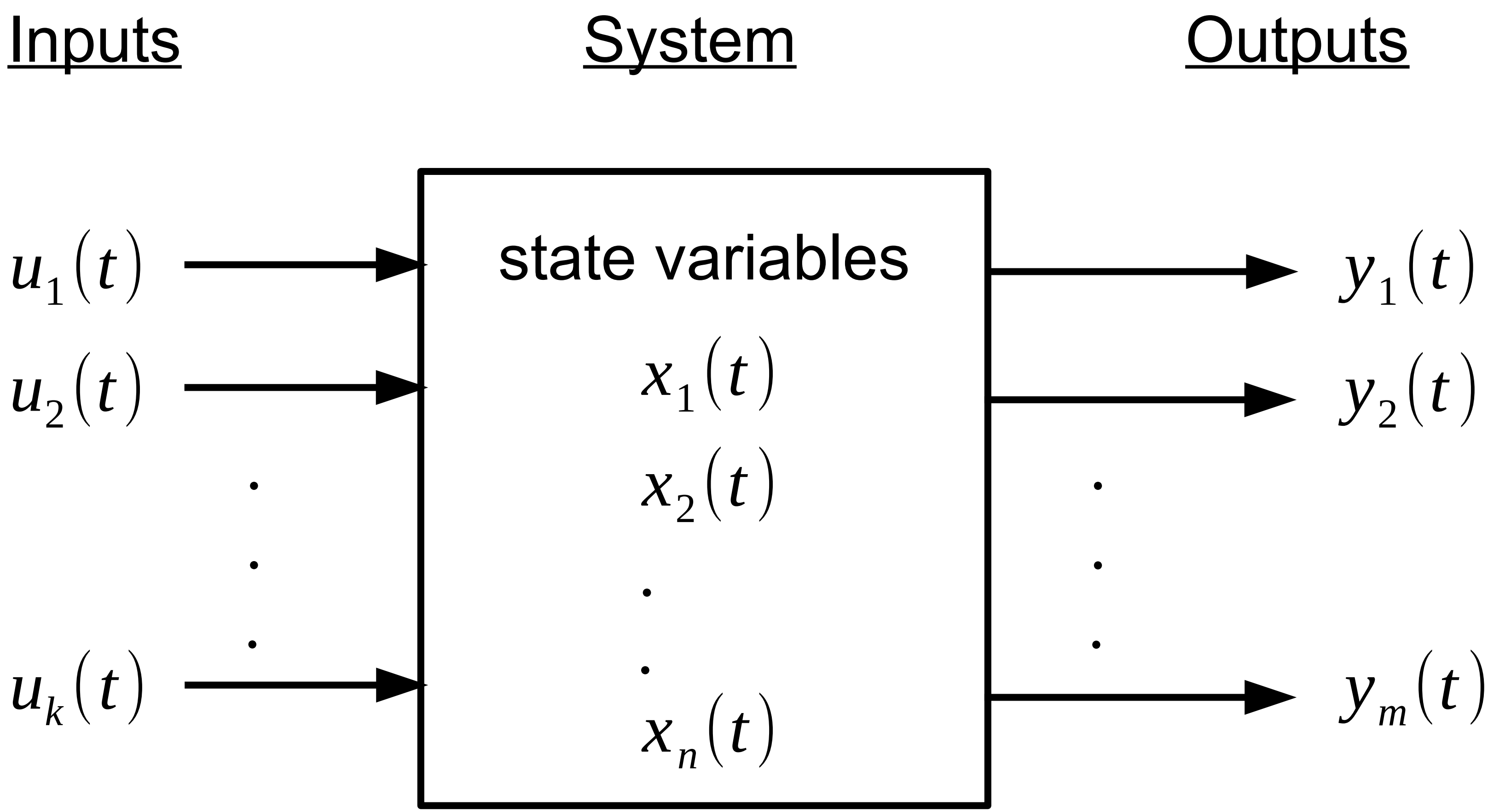
State-space representation is an alternative to transfer function form for writing system models.

State variable or set of state variables – representation of status of a system at any time.

Typical state variables: position, velocity, temperature, pressure, volume flow, current, voltage.

There are many different state variable representations for the same system, but input-output relation does not depend on its' selection.

State-space representation



State-space representation

For continouse, linear and time-invariant system

State-space equation: $\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)$

$\boldsymbol{x}_{n x 1}(t)$ - state variables vector

$\boldsymbol{A}_{n x n}$ - state matrix (system matrix)

$\boldsymbol{B}_{n x k}$ - input matrix

$\boldsymbol{u}_{k x 1}(t)$ - control inputs

External outputs equation: $\boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)$

$\boldsymbol{y}_{m x 1}(t)$ - outputs vector

$\boldsymbol{C}_{m x n}$ - output matrix

$\boldsymbol{D}_{m x k}$ - transmition matrix (direct feedthrough matrix)

$\boldsymbol{u}_{k x 1}(t)$ - control inputs

State-space representation

Example for n = 2, k = 4, m = 3

State-space equation: $\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\,\boldsymbol{x}(t)+\boldsymbol{B}\,\boldsymbol{u}(t)$

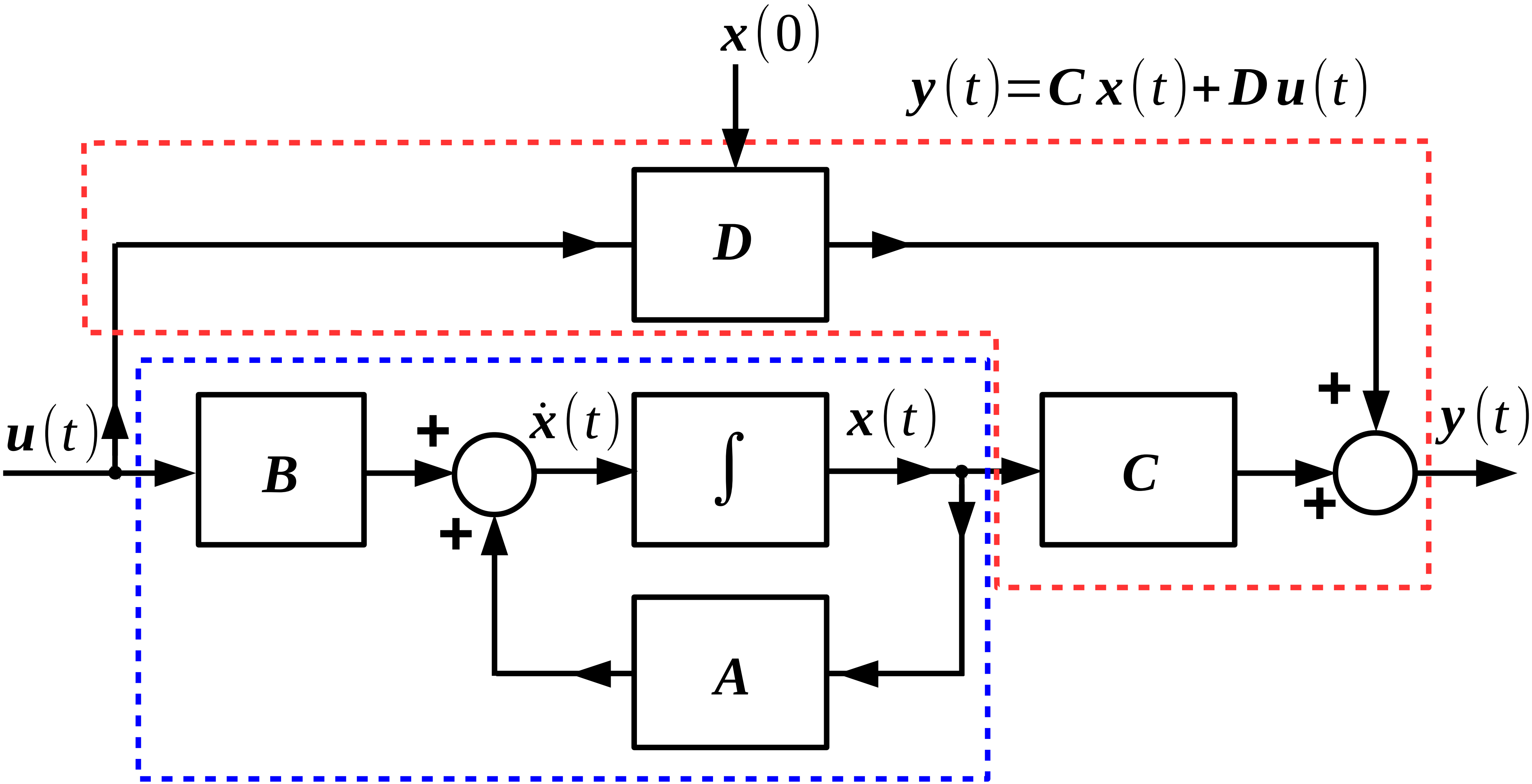
$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix}=\begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}+\begin{bmatrix}B_{11} & B_{12} & B_{13} & B_{14}\\ B_{21} & B_{22} & B_{23} & B_{24}\end{bmatrix}\begin{bmatrix}u_1(t)\\ u_2(t)\\ u_3(t)\\ u_4(t)\end{bmatrix}$$

Outputs equation: $\boldsymbol{y}(t)=\boldsymbol{C}\,\boldsymbol{x}(t)+\boldsymbol{D}\,\boldsymbol{u}(t)$

$$\begin{bmatrix}y_1(t)\\ y_2(t)\\ y_3(t)\end{bmatrix}=\begin{bmatrix}C_{11} & C_{12}\\ C_{21} & C_{22}\\ C_{31} & C_{32}\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}+\begin{bmatrix}D_{11} & D_{12} & D_{13} & D_{14}\\ D_{21} & D_{22} & D_{23} & D_{24}\\ D_{31} & D_{32} & D_{33} & D_{34}\end{bmatrix}\begin{bmatrix}u_1(t)\\ u_2(t)\\ u_3(t)\\ u_4(t)\end{bmatrix}$$

State-space representation

Time-domain block diagram representation

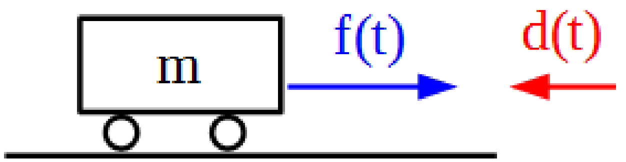


$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\,\boldsymbol{x}(t)+\boldsymbol{B}\,\boldsymbol{u}(t)$$

$$\boldsymbol{y}(t)=\boldsymbol{C}\,\boldsymbol{x}(t)+\boldsymbol{D}\,\boldsymbol{u}(t)$$

State-space representation – example 1

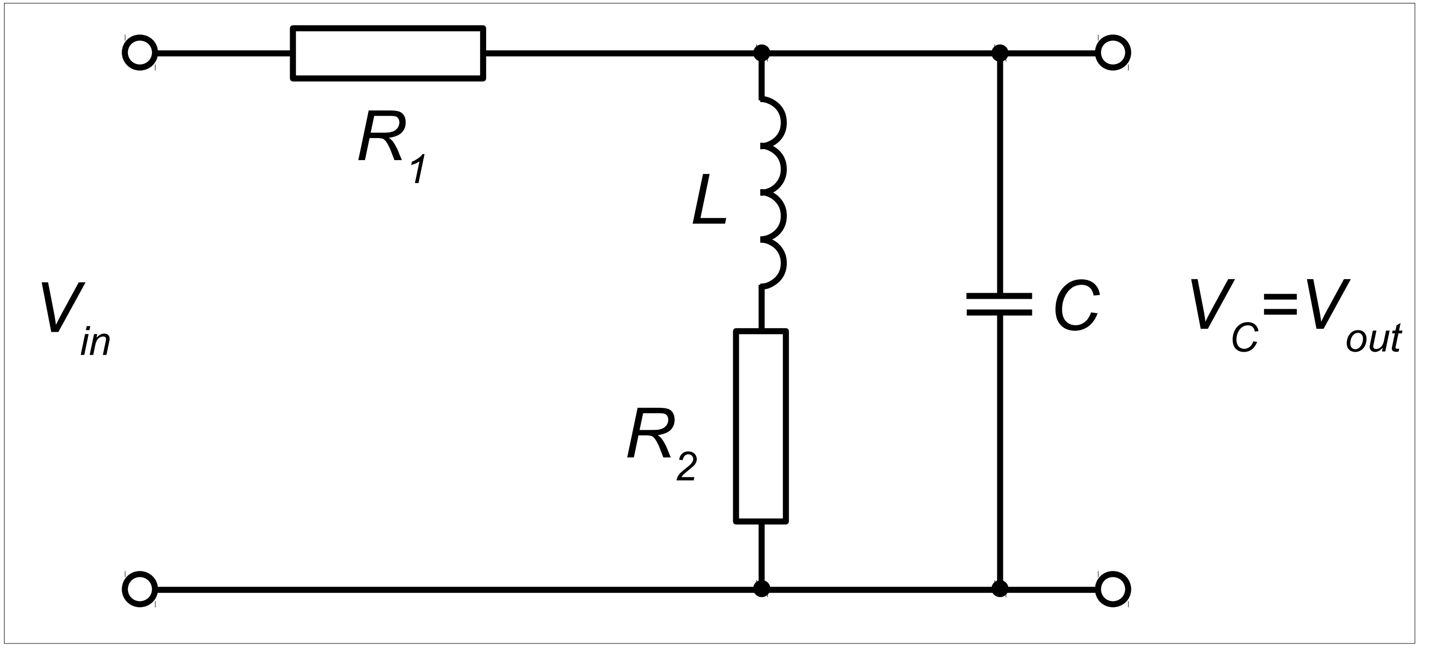
Car on a flat surface
m – mass,
f(t) – driving force,
d(t)=c*v(t) – air ressistance,
x(t) – displacement



$$m\frac{d^2x(t)}{dt^2}=f(t)-d(t)$$

input: force f(t)
output: velocity v(t)

State-space representation – example 2



$$V_{\text{in}}(t)=i_1(t)R_1+L\frac{d\,i_2(t)}{dt}+i_2(t)R_2$$

$$V_{\text{in}}(t)=i_1(t)R_1+V_C(t)$$

$$C\frac{dV_C(t)}{dt}=i_3(t)$$

$$i_1(t)=i_2(t)+i_3(t)$$

State-space representation – solution

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)$$

$$\boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)$$

Solution:

$$\boldsymbol{y}(t)=\boldsymbol{C}e^{\boldsymbol{A}t}\boldsymbol{x}_0+\boldsymbol{C}\int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)\,d\tau+\boldsymbol{D}\boldsymbol{u}(t)$$

satisfying initial conditions (free response)

convolution of an input with system impulse responses (forced response)

State-space representation – solution

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)$$

$$\boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)$$

Solution:

$$\boldsymbol{y}(t)=\boldsymbol{C}e^{\boldsymbol{A}t}\boldsymbol{x}_0+\boldsymbol{C}\int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)\,d\tau+\boldsymbol{D}\boldsymbol{u}(t)$$

satisfying initial conditions (free response)

convolution of an input with system impulse responses (forced response)

Problem: calculation of exp(At)

Calculation of exp(At)

Transformation of A into Jordan normal form

$$e^{\boldsymbol{A}t} = \boldsymbol{S}\,\mathrm{diag}\left(e^{\lambda_1 t}, \ldots, e^{\lambda_n t}\right)\boldsymbol{S}^{-1}, \text{ where:}$$

$\boldsymbol{S}$ - matrix of eigenvectors of A

$\lambda_1,\ \lambda_2,\ \lambda_n$ - eigenvalues of A

Partial fraction expansion of *(Is - A)*$^{-1}$

$$\text{if } \left(\boldsymbol{I}s - \boldsymbol{A}\right)^{-1} = \sum_{i=1}^{N}\sum_{j=i}^{n_i} T_{ij}\frac{1}{\left(s-\lambda_i\right)^j} \text{ then } e^{\boldsymbol{A}t} = \sum_{i=1}^{N}\sum_{j=i}^{n_i} T_{ij}\frac{t^{j-1}}{(j-1)!}e^{\lambda t}$$

State-space representation to transfer function conversion

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\,\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t) \qquad \boldsymbol{y}(t)=\boldsymbol{C}\,\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)$$

$\mathcal{L}$ + zero IC (applied to both equations)

$$s\,\boldsymbol{X}(s)=\boldsymbol{A}\,\boldsymbol{X}(s)+\boldsymbol{B}\boldsymbol{U}(s)$$

$$s\,\boldsymbol{X}(s)-\boldsymbol{A}\,\boldsymbol{X}(s)=\boldsymbol{B}\boldsymbol{U}(s)$$

$$(s\boldsymbol{I}-\boldsymbol{A})\boldsymbol{X}(s)=\boldsymbol{B}\boldsymbol{U}(s)$$

for $\det(s\boldsymbol{I}-\boldsymbol{A})\neq 0$

$$\boldsymbol{X}(s)=(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(s)$$

$$\boldsymbol{Y}(s)=\boldsymbol{C}\,\boldsymbol{X}(s)+\boldsymbol{D}\boldsymbol{U}(s)$$

$$\boldsymbol{Y}(s)=\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(s)+\boldsymbol{D}\boldsymbol{U}(s)$$

$$\boldsymbol{Y}(s)=\left(\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}\right)\boldsymbol{U}(s)$$

$$\boldsymbol{G}(s)=\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}$$

State-space representation to transfer function conversion

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\,\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)$$

$$\boldsymbol{y}(t)=\boldsymbol{C}\,\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)$$

Rosenbrock's system matrix

$$\boldsymbol{P}(s)=\begin{bmatrix} s\boldsymbol{I}-\boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}$$

Transfer function elements for i-th input and j-th output:

$$g_{ij}=\frac{\begin{vmatrix} s\boldsymbol{I}-\boldsymbol{A} & -b_i \\ c_j & d_{ij} \end{vmatrix}}{\left|s\boldsymbol{I}-\boldsymbol{A}\right|}$$

State-space representation to transfer function conversion

Example 1

$$\boldsymbol{A}=\begin{bmatrix} 2 & 4 \\ -4 & 2 \end{bmatrix} \quad \boldsymbol{B}=\begin{bmatrix} 1 & 1 & 2 \\ 0 & -1 & 3 \end{bmatrix} \quad \boldsymbol{C}=\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \boldsymbol{D}=\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 1 \\ 3 & -1 & 2 \end{bmatrix}$$

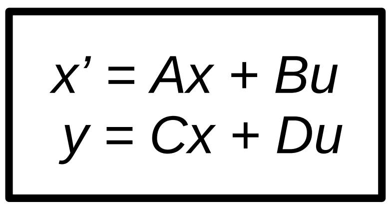
State-space representation to transfer function conversion

Example 1

$$\boldsymbol{A}=\begin{bmatrix} 2 & 4 \\ -4 & 2 \end{bmatrix} \quad \boldsymbol{B}=\begin{bmatrix} 1 & 1 & 2 \\ 0 & -1 & 3 \end{bmatrix} \quad \boldsymbol{C}=\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \boldsymbol{D}=\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 1 \\ 3 & -1 & 2 \end{bmatrix}$$

State-space representation – software

MATLAB & Simulink

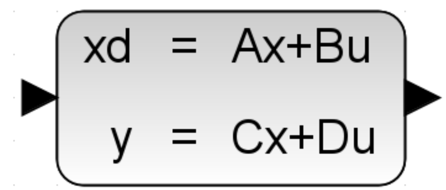


State space

State-space representation to transfer function conversion:

[num,den] = ss2tf(A,B,C,D,iu)

Scilab & Xcos

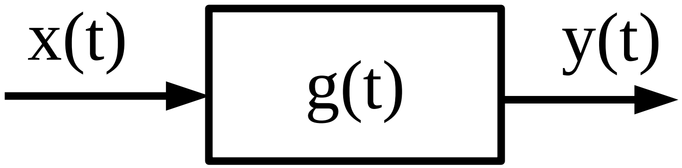


State-space representation to transfer function conversion:

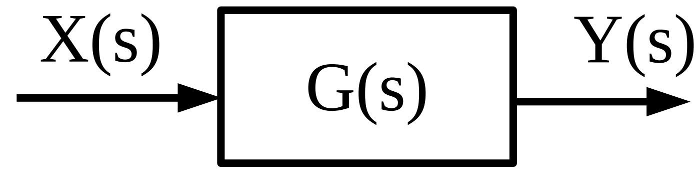
[h] = ss2tf(sl)

Block diagrams

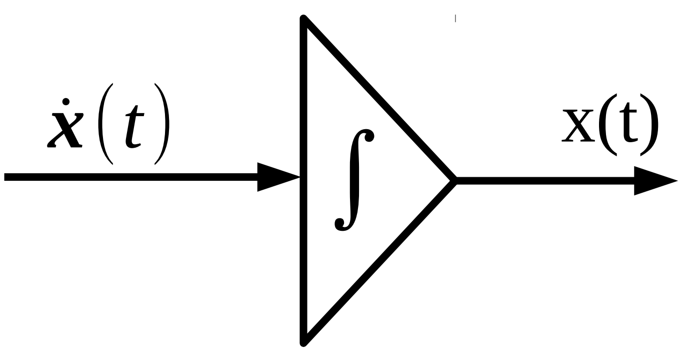
transfer function in time domain



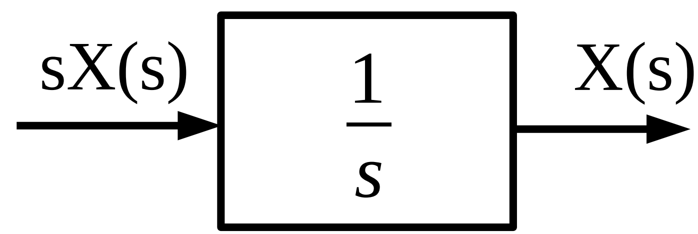
transfer function in complex domain



inegration in time domain



integration in complex domain



Transfer function to state-space representation conversion

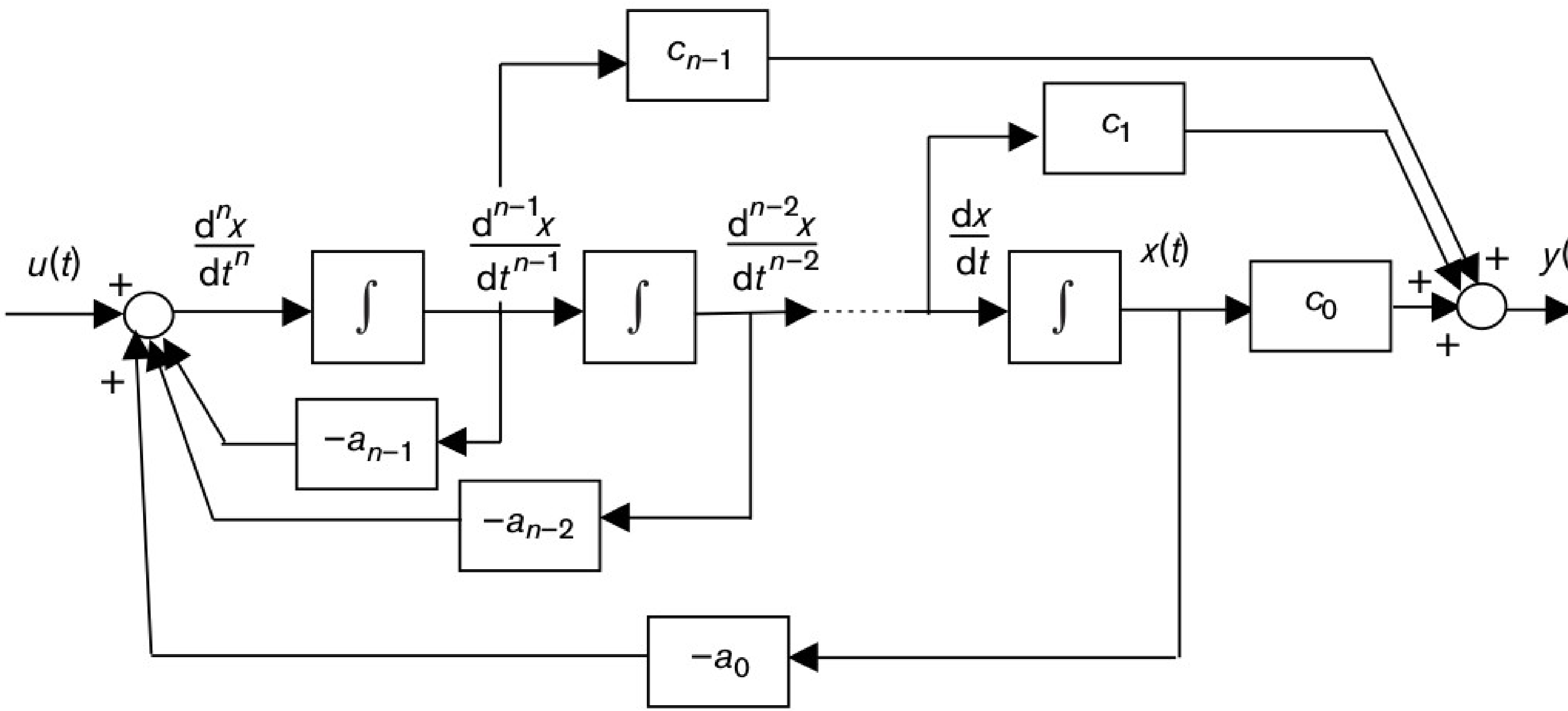
Direct method (controllable canonical form) – direct construction of the block diagram with coefficients from transfer function. We can read state-space representation from the diagram.

Parallel method (diagonal canonical form) – partial fraction decomposition of a transfer function is needed, then we can create block diagram. State matrix A will be then diagonal.

Iterative method – factored form of a transfer function is needed (poles and zeroes visible).

Direct method (controllable canonical form) for SISO system

$$H(s)=\frac{Y(s)}{U(s)}=\frac{c_m s^m+c_{m-1}s^{m-1}+\ldots+c_1 s+c_0}{s^n+a_{n-1}s^{n-1}+\ldots+a_1 s+a_0}$$



picture source: Jacqueline Wilkie, Michael Johnson, Reza Katebi, Control engineering - An introductory course, 2002

Direct method (controllable canonical form) for SISO system

$$H(s)=\frac{Y(s)}{U(s)}=\frac{c_m s^m + c_{m-1} s^{m-1} + \ldots + c_1 s + c_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} \quad \text{for } n > m$$

$$\boldsymbol{A}=\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}_{\mathrm{nxn}} \qquad \boldsymbol{B}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}_{\mathrm{nxk}}$$

$$\boldsymbol{C}=\begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{m-1} & c_m & 0 & \cdots & 0 \end{bmatrix}_{\mathrm{mxn}}$$

$$\boldsymbol{D}=[0]_{\mathrm{mxk}}$$

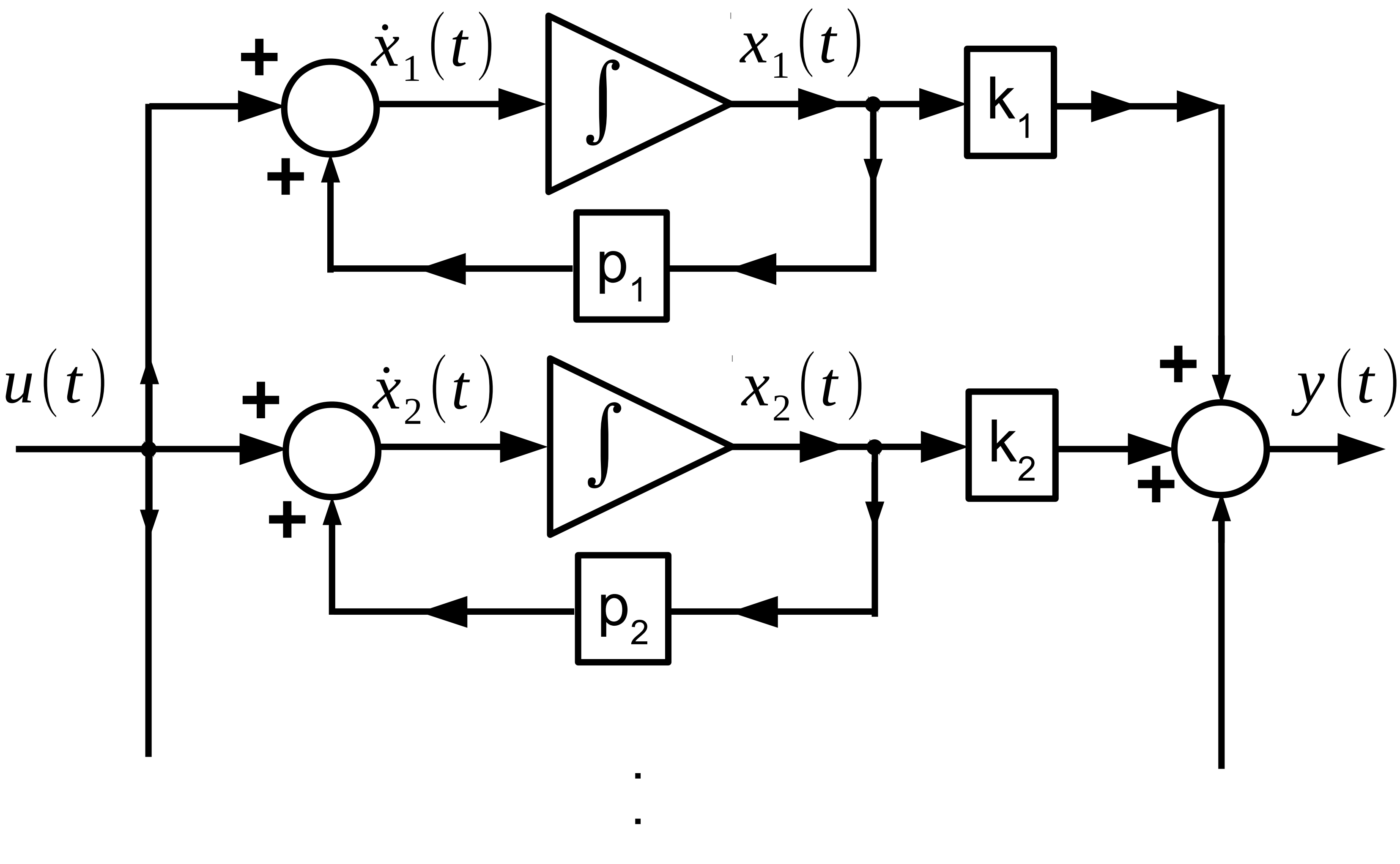
Direct method (canonical control form) for SISO system

Example

$$H(s)=\frac{Y(s)}{U(s)}=\frac{2s^3+s+5}{s^4+3s^3+2s^2+s+12}$$

Parallel method (diagonal canonical form) for SISO system

$$H(s)=\frac{Y(s)}{U(s)}=\frac{k_1}{s-p_1}+\frac{k_2}{s-p_2}+\ldots+\frac{k_n}{s-p_n} \quad \text{for } n>m$$



Parallel method (diagonal canonical form) for SISO system

$$H(s)=\frac{Y(s)}{U(s)}=\frac{k_1}{s-p_1}+\frac{k_2}{s-p_2}+\ldots+\frac{k_n}{s-p_n} \quad \text{for } n>m$$

$$\boldsymbol{A}=\begin{bmatrix} p_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & p_n \end{bmatrix} \qquad \boldsymbol{B}=\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}_{\mathrm{nx1}}$$

$$\boldsymbol{C}=\begin{bmatrix} k_1 & k_2 & k_3 & \cdots & k_{n-1} & k_n \end{bmatrix}_{\mathrm{nx1}}$$

$$\boldsymbol{D}=0$$

<u>For repeated poles use Jordan form</u>

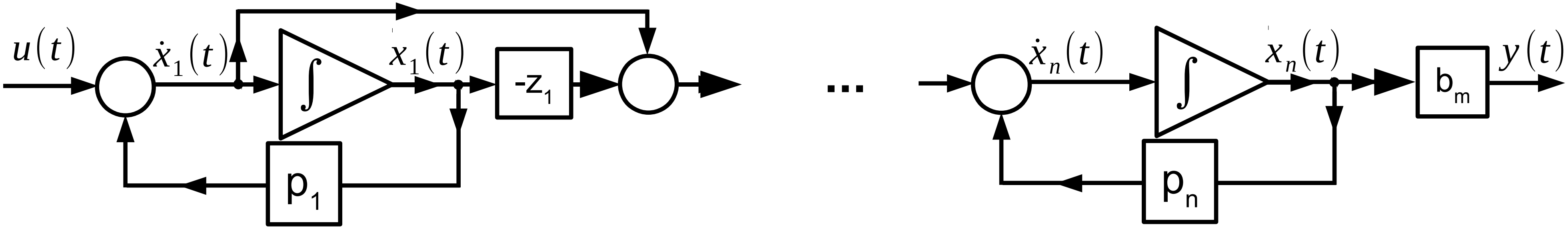
Parallel method (diagonal canonical form) for SISO system

Example

$$H(s)=\frac{Y(s)}{U(s)}=\frac{s^2+s}{10s^3+17s^2+10s+2}$$

Iterative method for SISO system

$$H(s)=\frac{Y(s)}{U(s)}=b_m\frac{s-z_1}{s-p_1}\frac{s-z_2}{s-p_2}\frac{s-z_3}{s-p_3}\cdots\frac{s-z_m}{s-p_m}\frac{1}{s-p_{m+1}}\cdots\frac{1}{s-p_n}$$



Iterative method for SISO system

$$H(s)=\frac{Y(s)}{U(s)}=b_m\frac{s-z_1}{s-p_1}\frac{s-z_2}{s-p_2}\frac{s-z_3}{s-p_3}\cdots\frac{s-z_m}{s-p_m}\frac{1}{s-p_{m+1}}\cdots\frac{1}{s-p_n}$$

$$\boldsymbol{A}=\begin{bmatrix} p_1 & 0 & 0 & \cdots & 0 \\ p_1-z_1 & p_2 & 0 & \cdots & 0 \\ p_1-z_1 & p_2-z_2 & p_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_1-z_1 & p_2-z_2 & p_3-z_3 & \cdots & 0 \\ p_1-z_1 & p_2-z_2 & p_3-z_3 & \cdots & p_n \end{bmatrix} \qquad \boldsymbol{B}=\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}_{\mathrm{nx1}}$$

$$\boldsymbol{C}=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & b_m \end{bmatrix}_{\mathrm{nx1}}$$

$$\boldsymbol{D}=0 \text{ , if } n>m$$

Iterative method for SISO system

Example

$$H(s)=\frac{Y(s)}{U(s)}=\frac{s^2+3s+2}{s^4+11s^3+41s^2+61s+30}$$

Controllabilty

<u>Controllability</u> – ability to change the state of a system from any initial state to any other final state in finite time interval with permissible inputs.

Linear time-invariant system is controllable if the Kalmana matrix [B AB A^2B …] has full rank.

Nonlinear system is controllable if the accessibility distribution created with Lie bracket operation has full rank.

Linear time-invariant system in form:

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

is controllable wrt output if:

rank [B AB A^2B … A^{n-1}B] = n,

where:

n – number of state variables

Controllabilty

Example

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_1(t)$$

Controllabilty wrt output

Controllability with respect to output – ability to change the output of a system from any initial value to any other desired output in finite time interval with permissible inputs.

Linear time-invariant system in form:

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

is controllable wrt output if:

rank $[CB \;\; CAB \;\; CA^2B \;\; \ldots \;\; CA^{n-1}B] = k$,

where: k – number of inputs

Controllabilty wrt output

<u>Example</u>

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_1(t) \qquad \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u_1(t)$$

Observability

<u>Observability</u> – for any possible sequence of past state and control inputs, the current state of the system can be determined.

Linear time-invariant system in form:

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

is observable if:

rank $[C\ \ CA\ \ CA^2 \ldots CA^{n-1}]^T = n$

where:
n – number of state variables

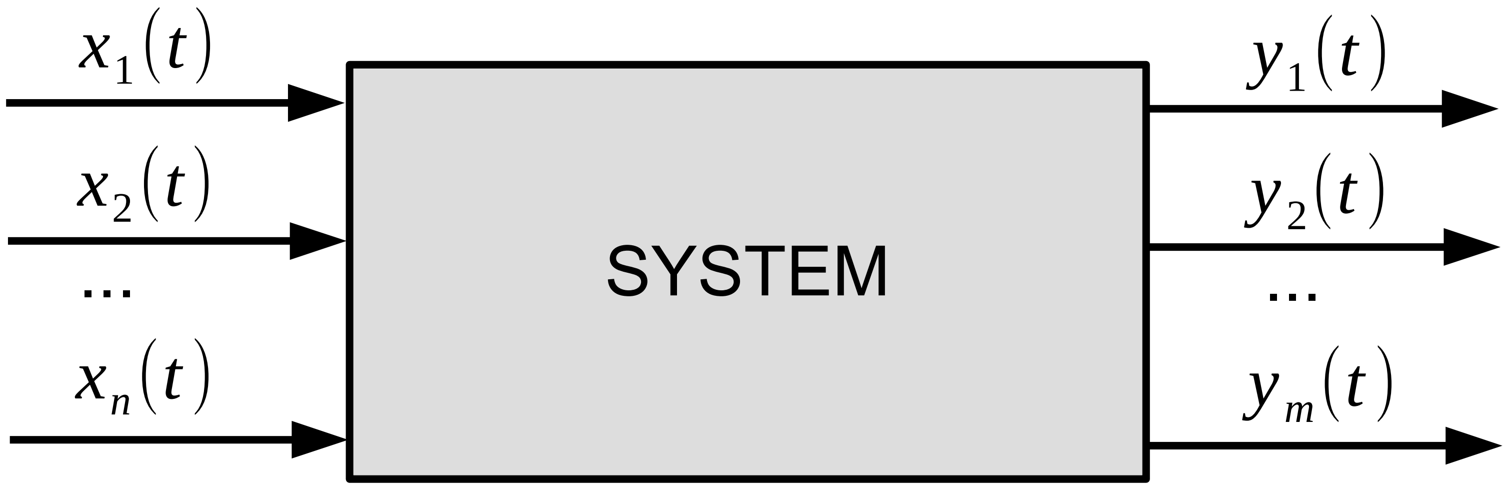
Observability

Example

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -1 & -2 & -3\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u_1(t)$$

$$\begin{bmatrix}y_1(t)\\ y_2(t)\\ y_3(t)\end{bmatrix}=\begin{bmatrix}1 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}u_1(t)$$

Multiple Input Multiple Output (MIMO) system



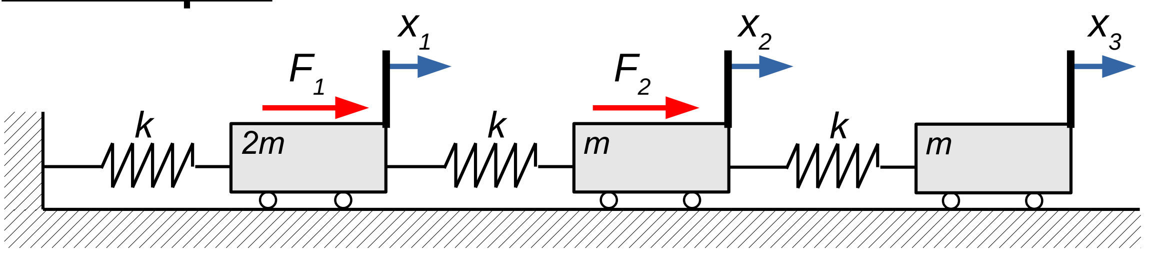
Multi-dimensional matrix of transfer functions

$$\boldsymbol{G}(s)=\begin{bmatrix} G_{11} & G_{12} & G_{13} & \cdots & G_{1k} \\ G_{21} & G_{22} & G_{23} & \cdots & G_{2k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ G_{l1} & G_{l2} & G_{l3} & \cdots & G_{lk} \end{bmatrix}$$

$$G_{ij}(s)=\frac{y_j(s)}{u_i(s)}, \text{ for zero initial conditions and } u_p=0 \text{ if } p\neq i$$

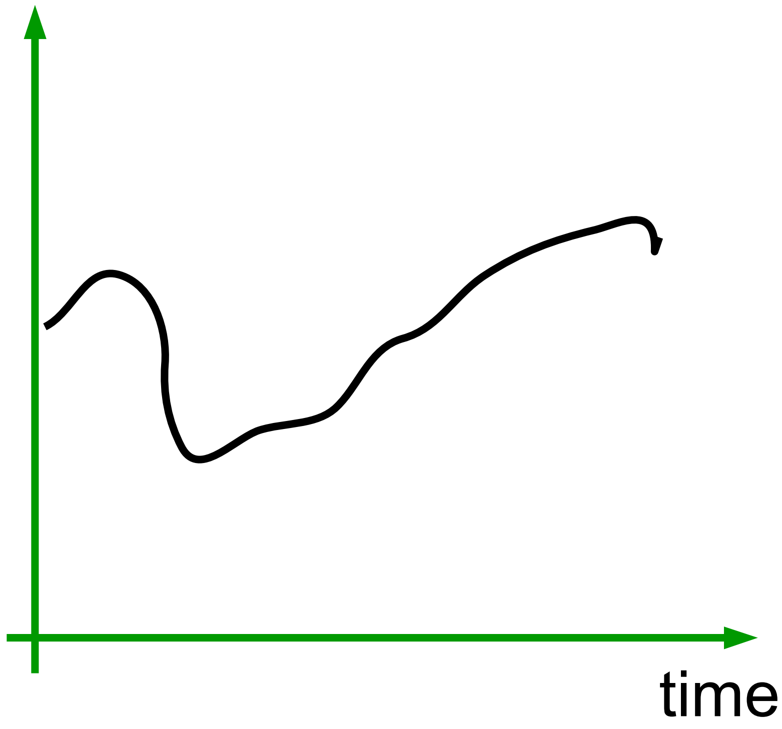
Multi-dimensional matrix of transfer functions

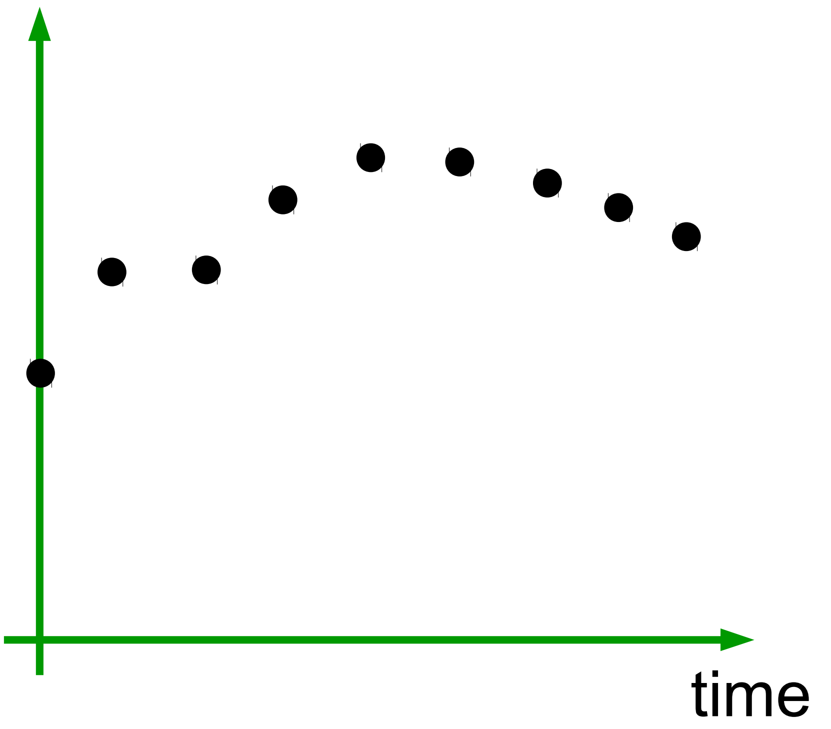
Example



input forces: F_1 and F_2
mass: m
stiffness: k
outputs: $x_2, v_3 = \dot{x}_3$

Continuous/discrete signals

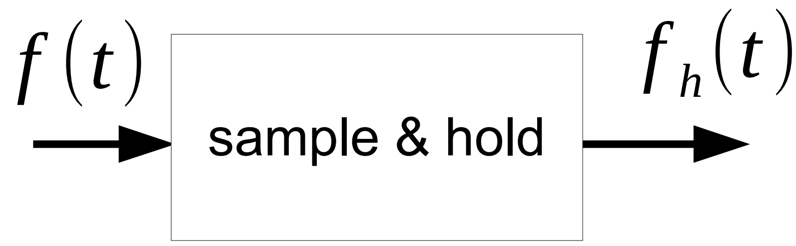


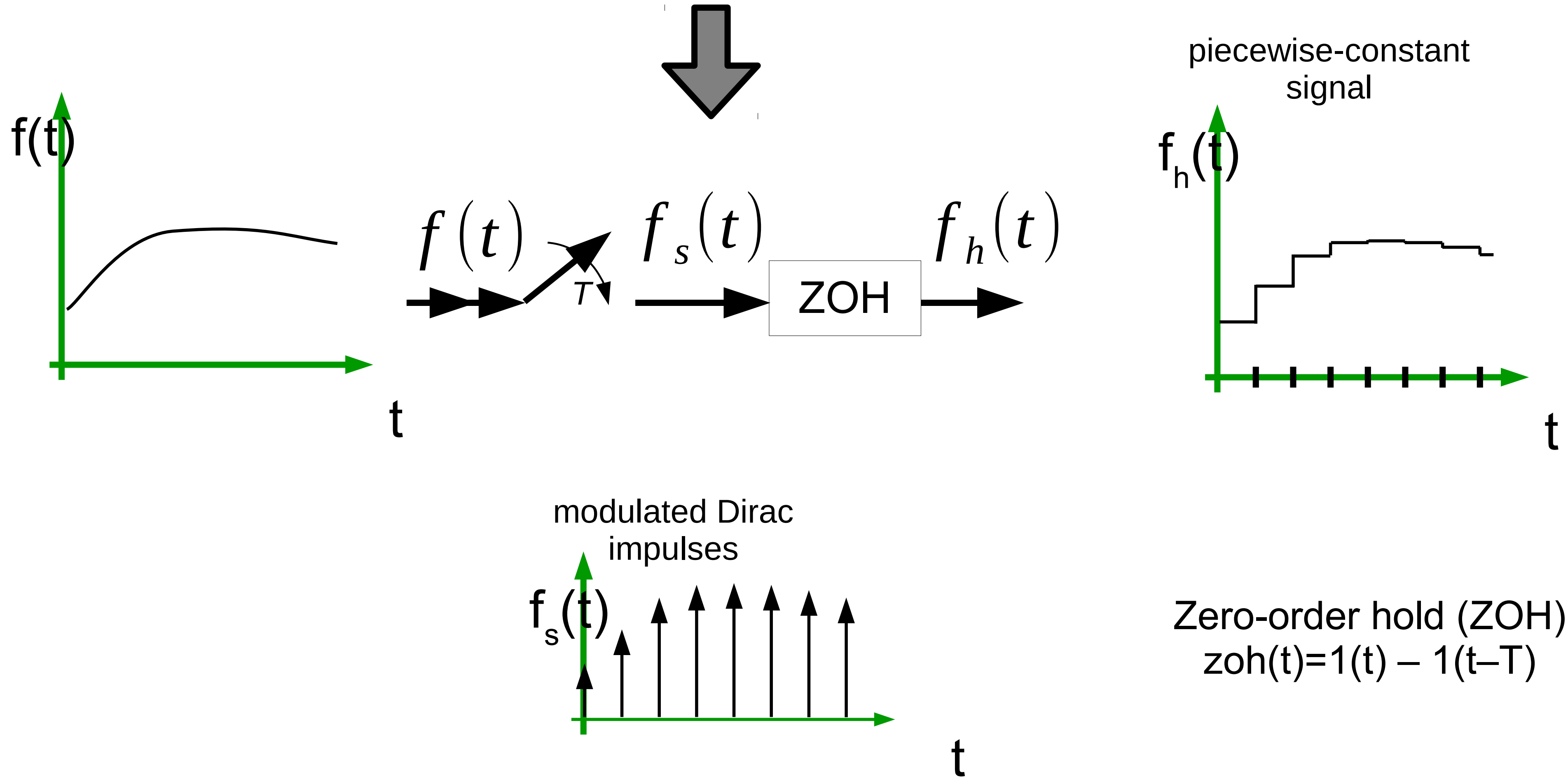


Continuous/discrete systems

Continuous-time signal x(t) for t ≥ 0	Discrete-time signal x[n] for n ≥ 0
Laplace transform	Unilateral Z-transform
$$X(s)=L\{x(t)\}=\int_0^{\infty} x(t)e^{-st}dt$$	$$X(z)=Z\{x[n]\}=\sum_{n=0}^{\infty} x[n]z^{-n}$$ (defined by W. Hurewicz)
$$L[\delta(t)]=1$$	$$Z[\delta(n)]=1$$
$$L[1(t)]=\frac{1}{s}$$	$$Z[1(n)]=\frac{z}{z-1}$$
$$L[1-e^{bt}]=\frac{s}{s-b}$$	$$Z[a^n 1(n)]=\frac{z}{z-a}$$

Sample and hold (ADC)





Zero-order hold (ZOH)
zoh(t)=1(t) – 1(t–T)

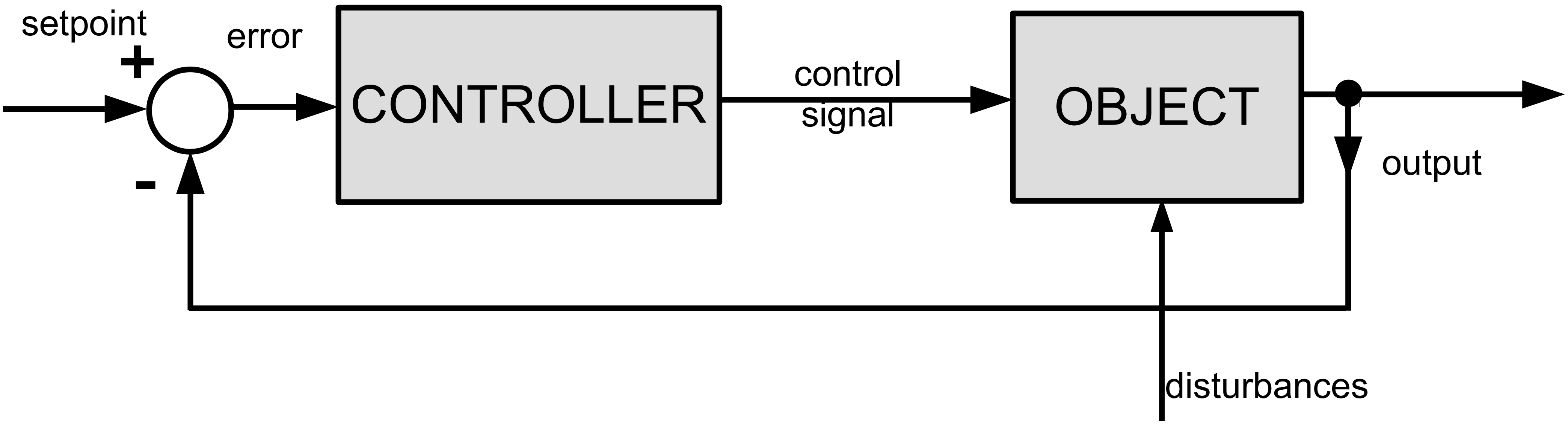
IMPORTANT DEFINITIONS

Robust control – controller is designed to work assuming that certain system parameters will be not constant but bounded. Control law is not changing.

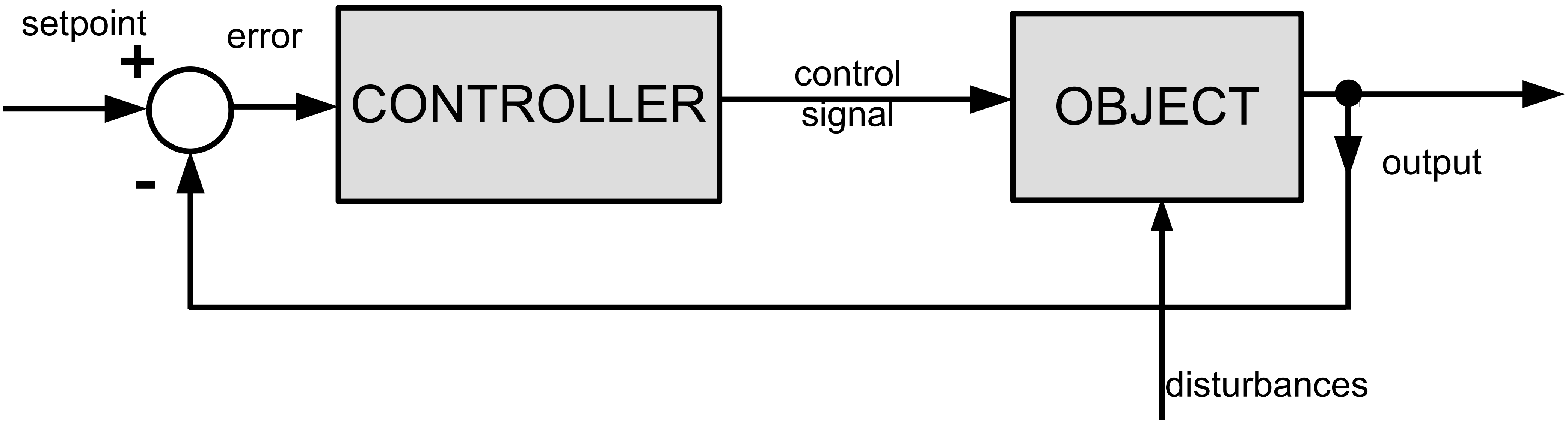
Adaptive control – controller adapts its parameters or changes its control law for varying parameters of the system. Parametric estimation of the system is used.

Intelligent control – control techniques that uses e.g. neural networks, fuzzy logic, machine learning or genetic algorithms.

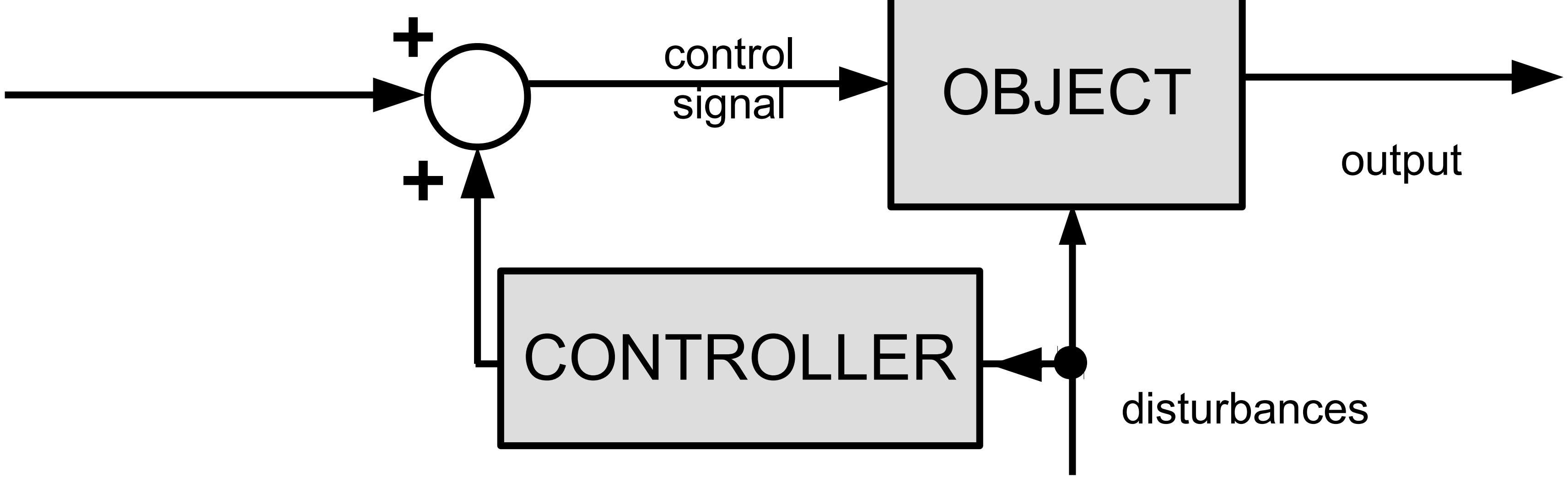
Feedback control



Feedback control

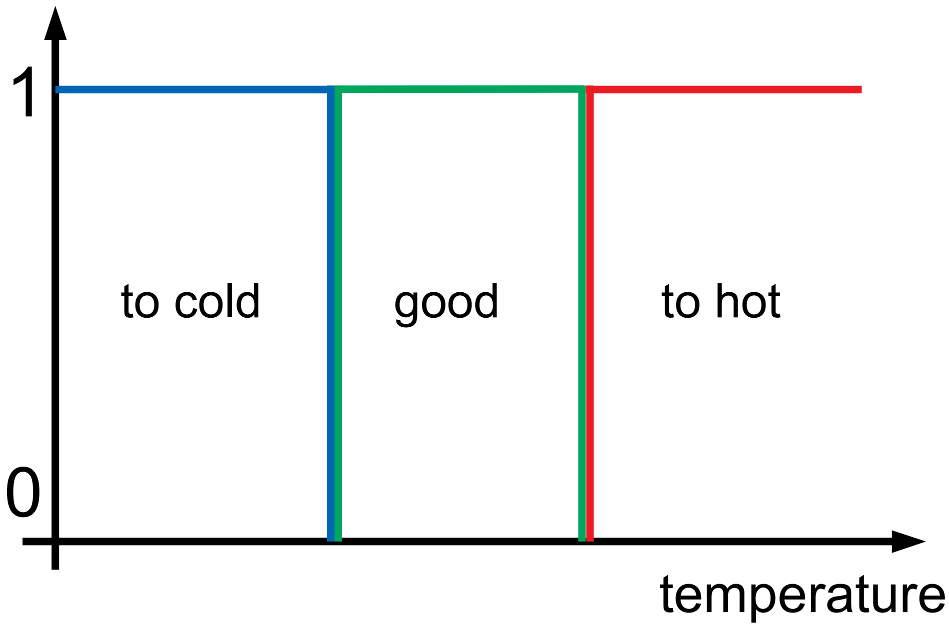


Feedforward control



Fuzzy logic by examples

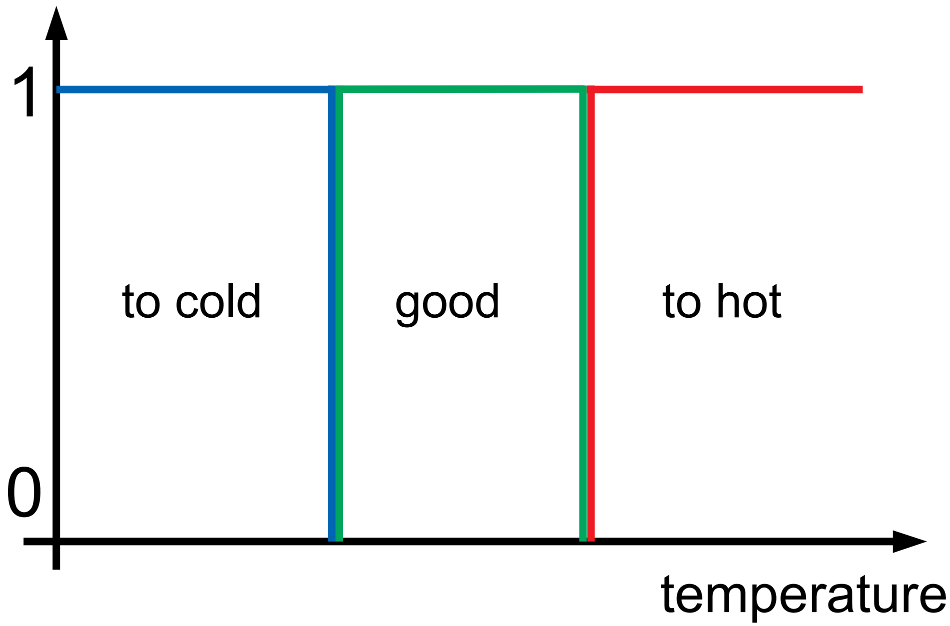
Classical logic



to cold:	
good:	
to hot:	

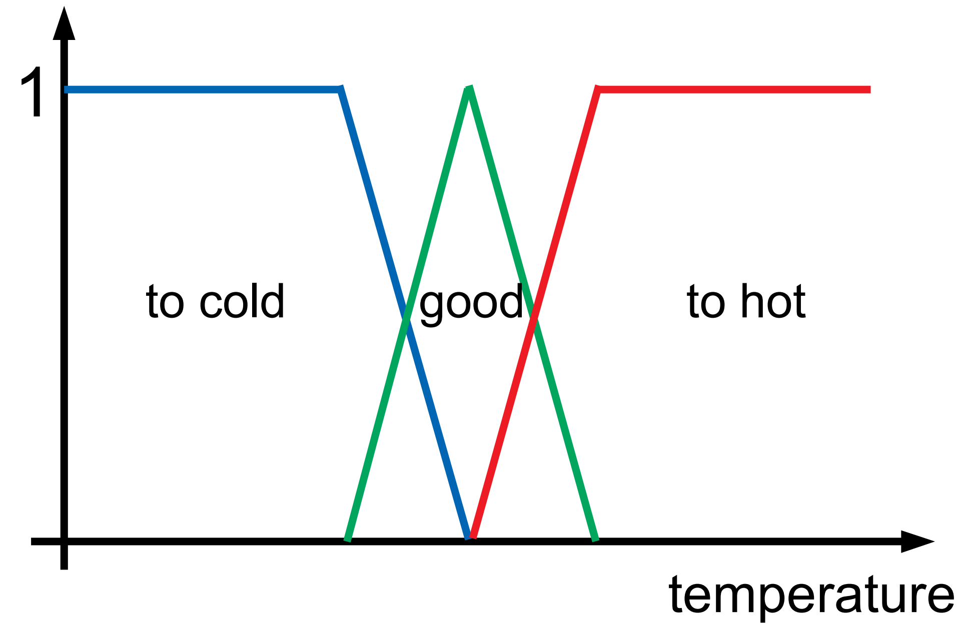
Fuzzy logic by examples

Classical logic



to cold:	
good:	
to hot:	

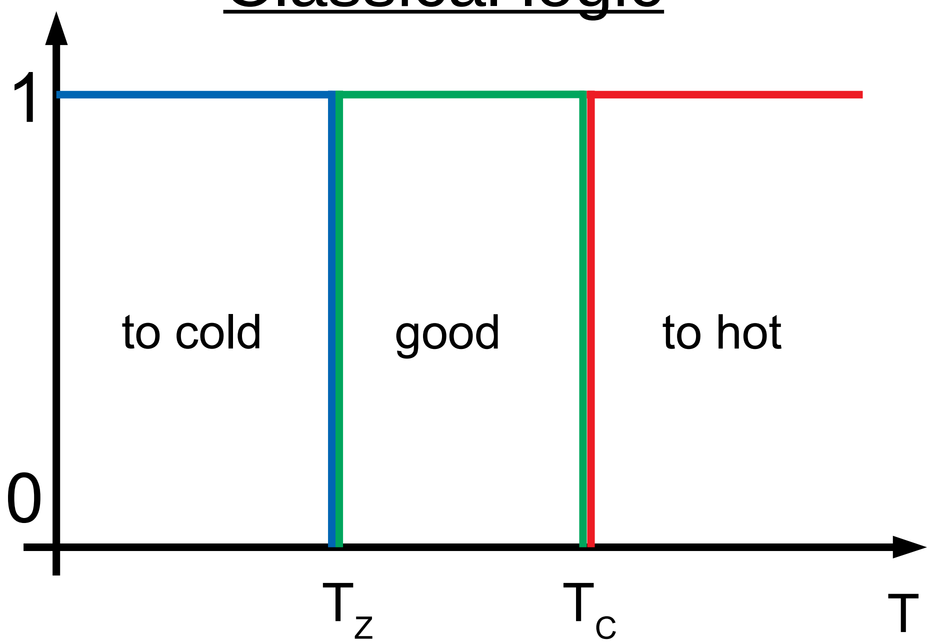
Fuzzy logic



to cold:	
good:	
to hot:	

Fuzzy logic by examples

Classical logic



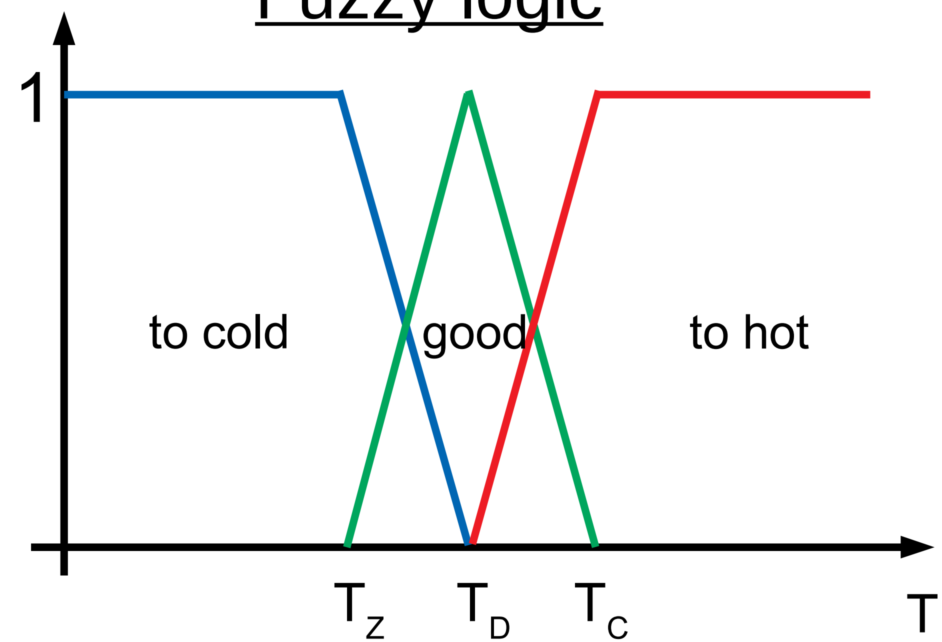
Membership functions:

$$\text{cold: } \begin{cases} 1\text{, if } T < T_Z \\ 0\text{, in other cases} \end{cases}$$

$$\text{good: } \begin{cases} 1\text{, if } T_Z < T < T_C \\ 0\text{, in other cases} \end{cases}$$

$$\text{hot: } \begin{cases} 1\text{, if } T > T_C \\ 0\text{, in other cases} \end{cases}$$

Fuzzy logic



Membership functions:

$$\text{cold: } \begin{cases} 1\text{, if } T < T_Z \\ \frac{(T_D - T)}{(T_D - T_Z)}\text{, if } T_Z < T < T_D \\ 0\text{, in other cases} \end{cases}$$

...

Fuzzy logic

Łukasiewicz-Tarski logic

Jan Łukasiewicz (1878-1956)

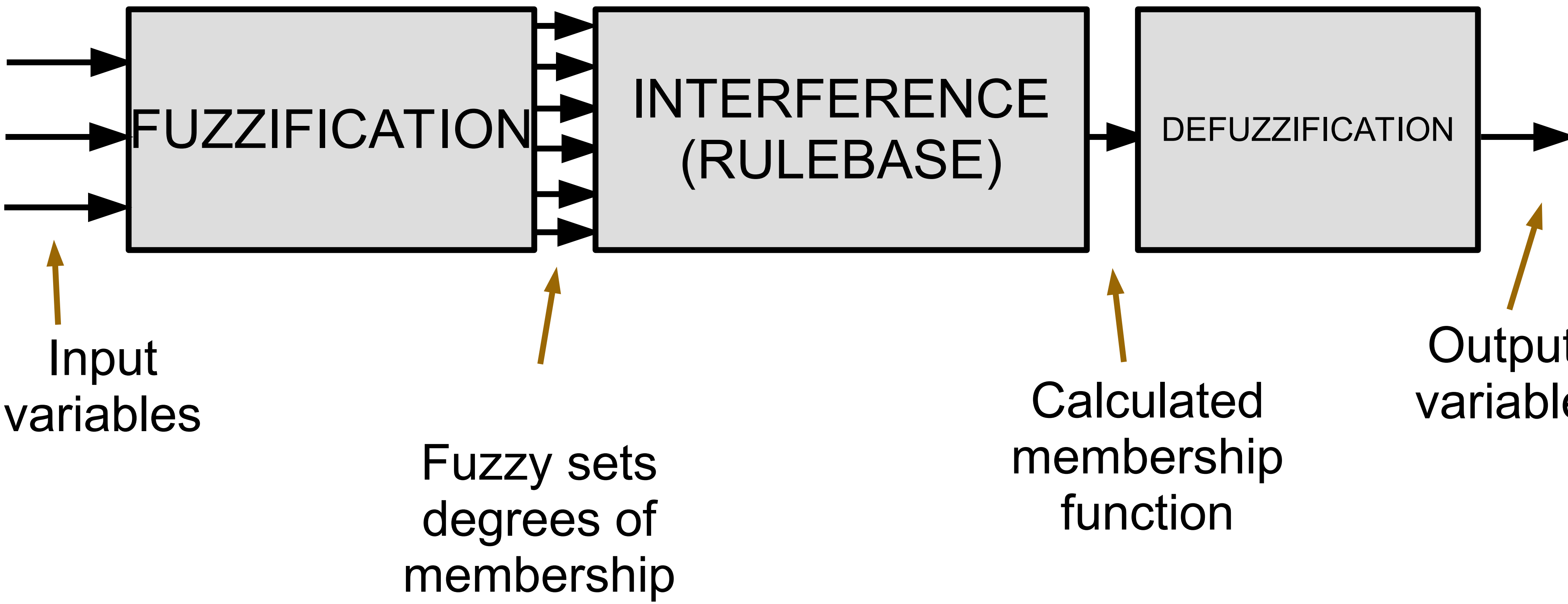
Alfred Tarski (1901-1983)

Basic operators:

- sum (OR) -----> MAX(x,y)
- multiplication (AND) -----> MIN(x,y)
- negation (NOT) -----> NOT(x)=1-x

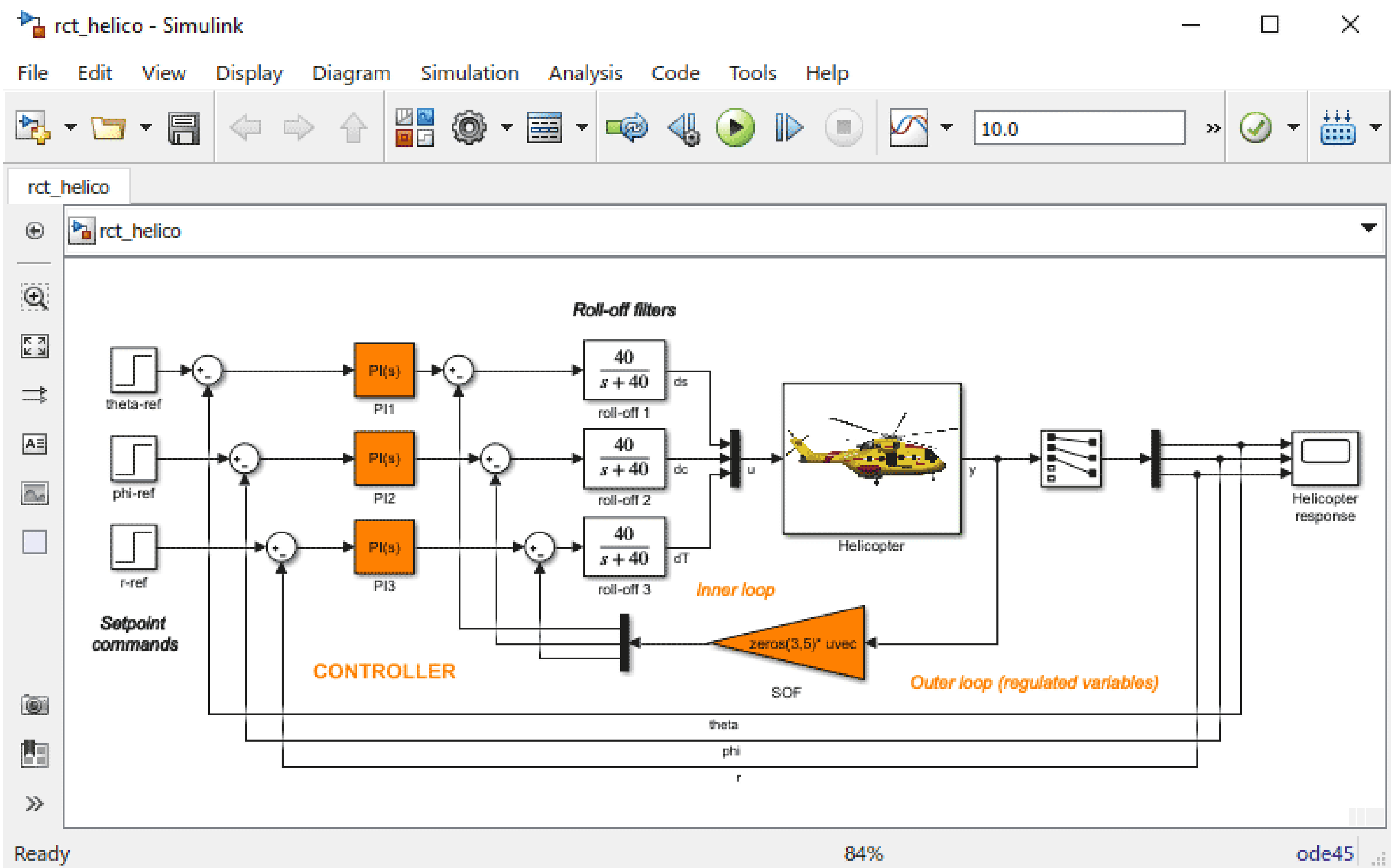
Fuzzy logic by examples

Fuzzy controller



Numerical simulations - software

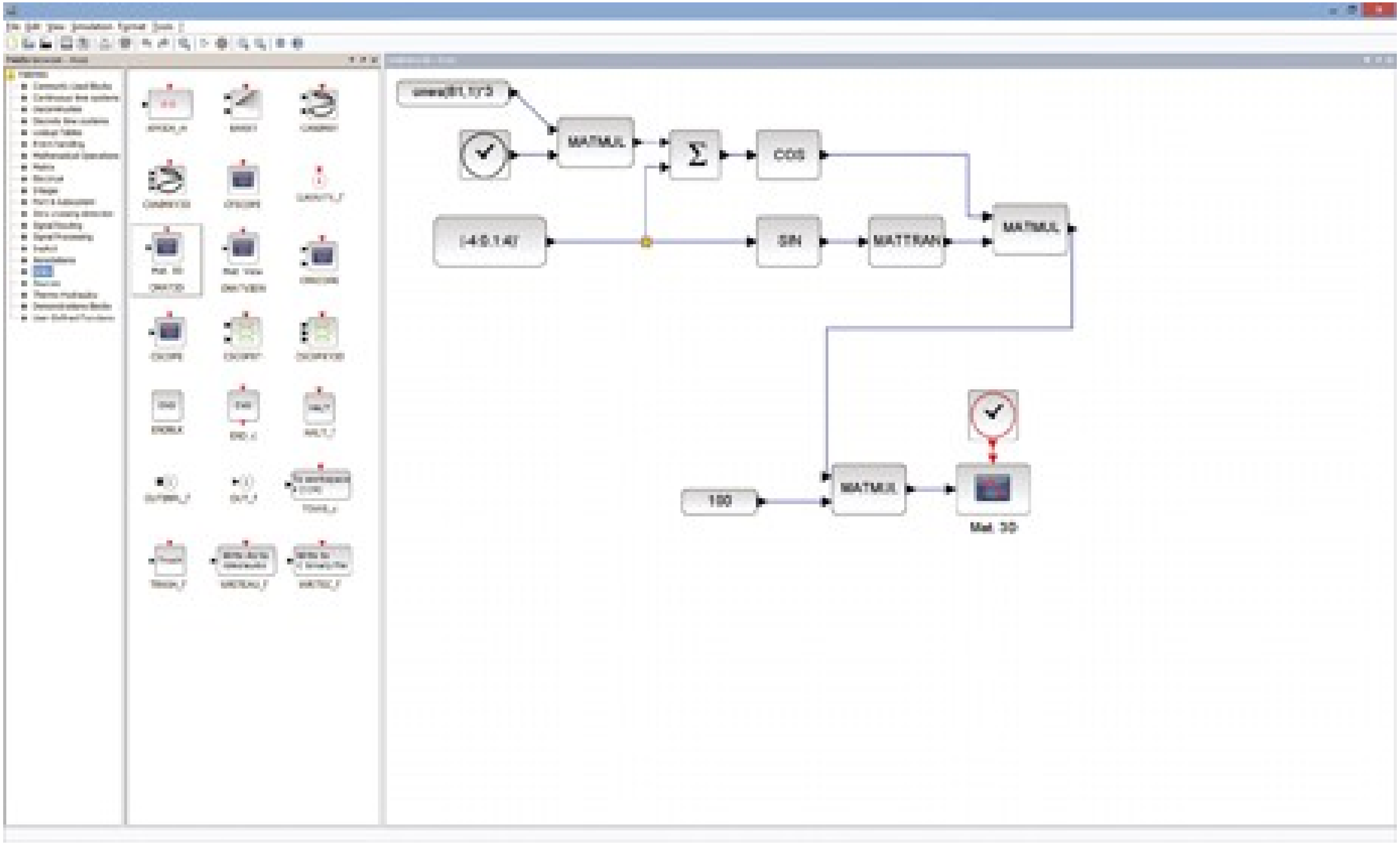
Matlab® / Simulink®



source: https://www.mathworks.com

Numerical simulations - software

Scilab / Xcos



source: https://scilab.org/scilab/features/xcos

OTHER THINGS

Control tasks:

* stabilization
* trajectory tracking
* path following

Bbackstepping method

Sliding mode control

Optimal control

Control with differential flatness property

Model-based control

Computed torque technique

Linear-quadratic regulator (LQR)

Examples

https://www.youtube.com/watch?v=URmxzxYlmtg

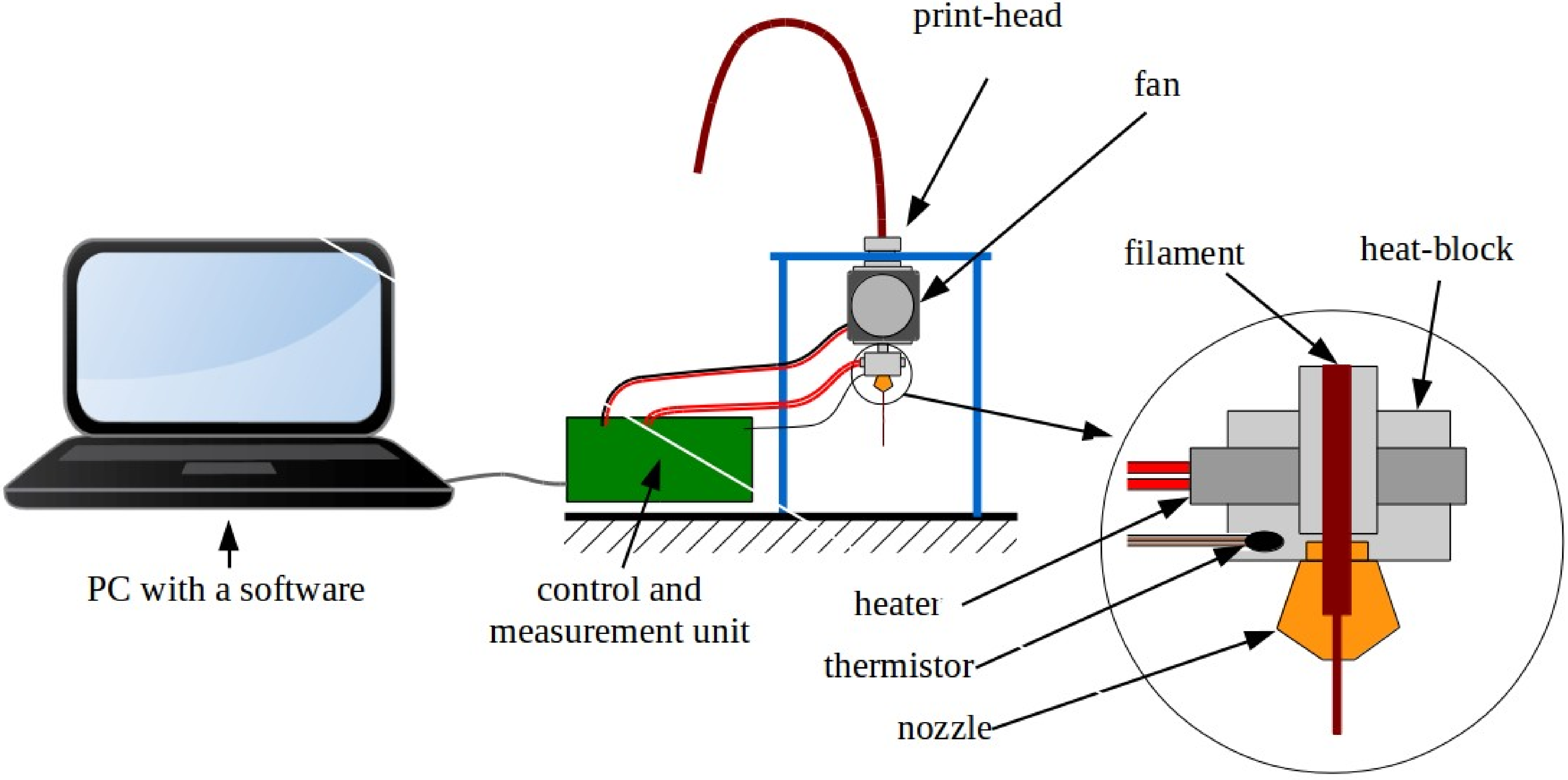
https://vimeo.com/192179726

https://www.youtube.com/watch?v=geqip_0Vjec

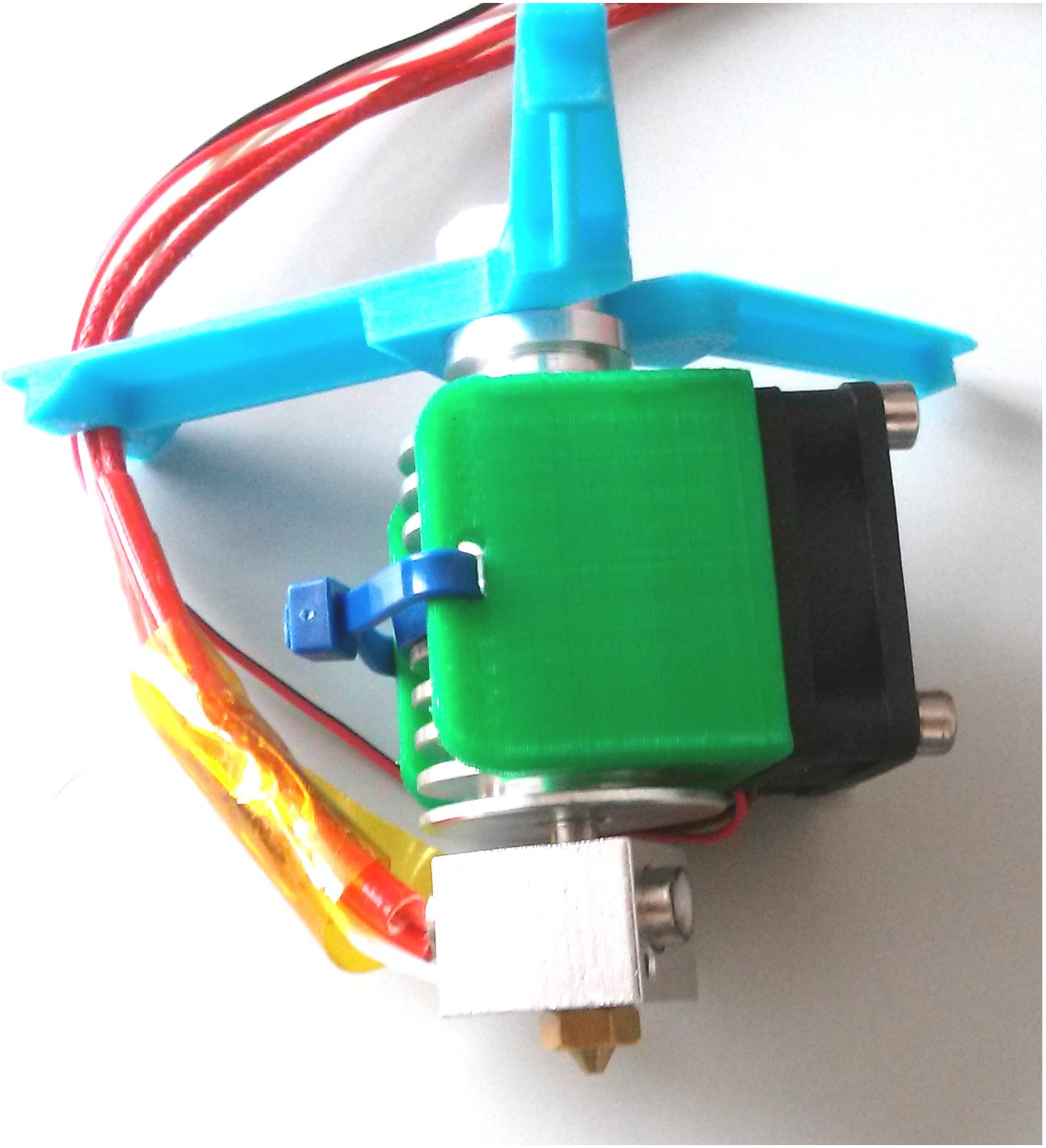
https://www.youtube.com/watch?v=w2itwFJCgFQ

https://www.youtube.com/watch?v=g11LN0UIynY

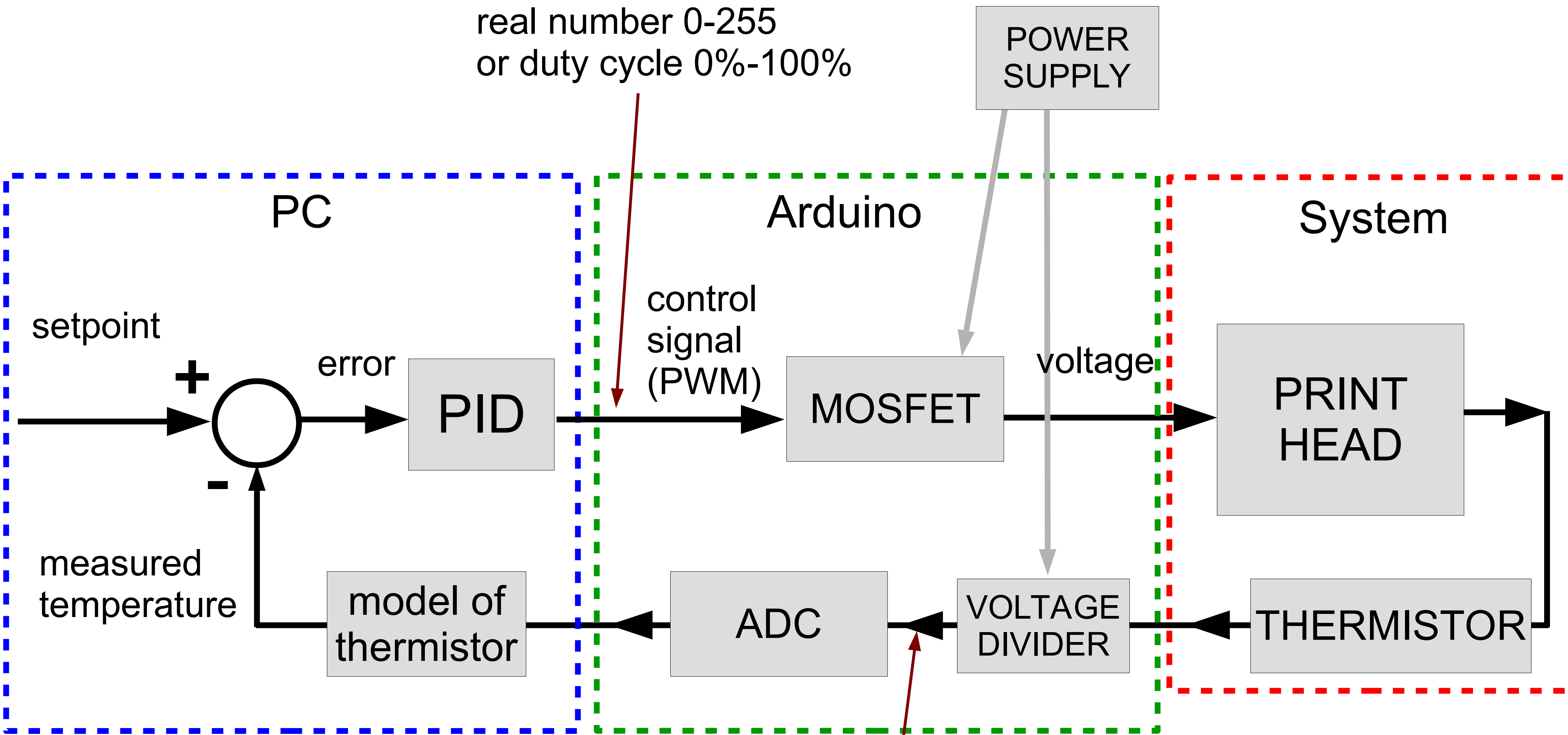
Experiment – control of the 3D printer's print-head temperature.



Experiment – control of the 3D printer's print-head temperature.



Experiment – control of the 3D printer's print-head temperature.



$$T(u_2)=\frac{1}{\frac{1}{T_0}-\frac{1}{\beta}\ln\left(\frac{R_0}{R_2}\left(\frac{u_1}{u_2}-1\right)\right)}$$

thermistor's
voltage
dropdown

$$R(T)=R_0\exp\left(\beta\left(\frac{1}{T}-\frac{1}{T_0}\right)\right)$$