MM2: Modeling, model description and simulation

- ✓ Models and modeling: concepts
  - definitions
  - simulation
  - mathematical models
- ✓ Model descriptions
  - Transfer functions
  - state space
  - block diagrams
- ✓ Discretization methods
- ✓ Simulation of linear and nonlinear dynamic systems with Matlab

Models and modeling: concepts

Definitions:

Model: a representation - in a usable form - of the essential aspects of a system

System identification: building mathematical models of dynamic systems based on observed data from the system:

- many measurement data are obtained as sampled values of the inputs and outputs
- a computer is used for processing the data
- model parameters are estimated by minimization of an error criterion.

Characterization of models and modeling

Models:

mathematical – other

parametric – nonparametric

continuous-time - discrete-time

input/output - state-space

linear – nonlinear

dynamic – static

time-invariant - time-varying

SISO - MIMO

Modelling / system identification:

theoretical (physical) – experimental

white-box - **grey-box** - black-box

structure determination - parameter estimation

time-domain - frequency-domain

direct - indirect

Physical parameters

Physical parameters are model parameters with an apparent physical meaning or significance

Typically they are the coefficients in basic physical laws, e.g. Newtons, Hooks, Ohms, and Kirchoffs laws

Some corresponding physical parameters are

- mechanical parameters: mass, friction coefficients, stiffness
- electrical parameters: resistance, inductance, capacitance
- thermal parameters: thermal resistance, specific heat
- besides: static gain, time constant, natural frequency and damping ratio

Non-physical parameters: coefficients in z-transform of state space description

Simulation

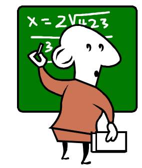
Purpose

- To obtain an understanding of the system
- Prediction of future system output
- Design and test of control systems
- Optimization of constructions
- Training of system operators on real time simulators
- Model parameter estimation from experiments

A computer simulation requires:

1. a discrete-time model
2. means for performing experiments on the model, e.g. giving specific inputs signals and parameter values
3. graphical tools for presenting the results.

Mathematical models



Model descriptions

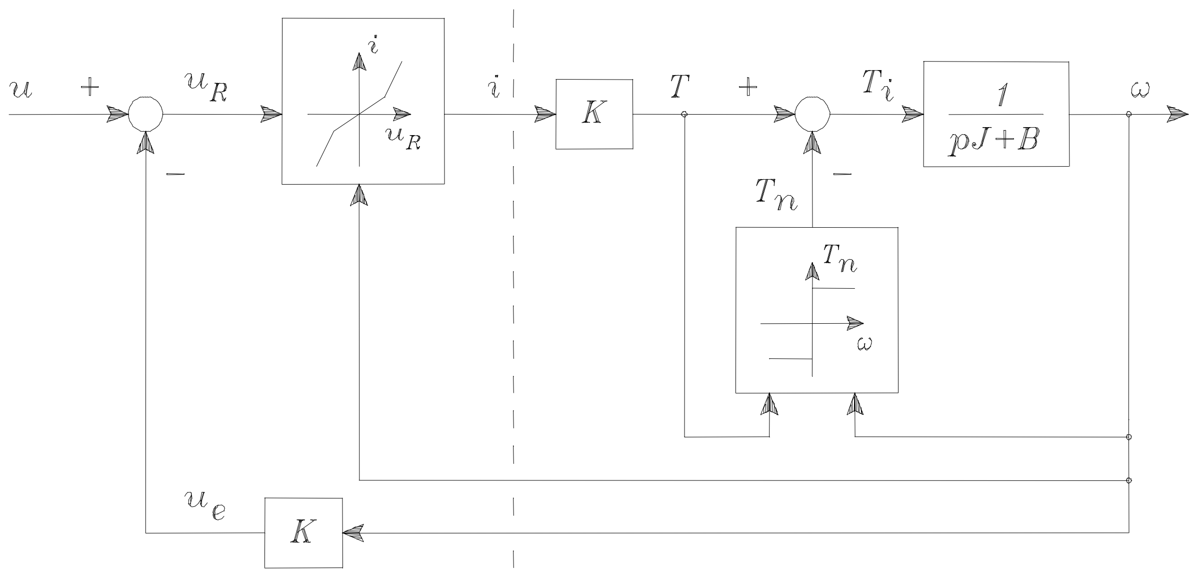
- Transfer functions
- state space
- block diagrams

Notation for continuous-time and discrete-time models:

Complex Laplace variable s   differential operator p: $x(t)=\partial x(t)/\partial t=px(t)$

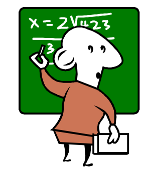
Complex Z-transform variable z   and shift operator q: $x(k+1)=qx(k)$

Block diagram of nonlinear system (DC-motor):



Discretization methods

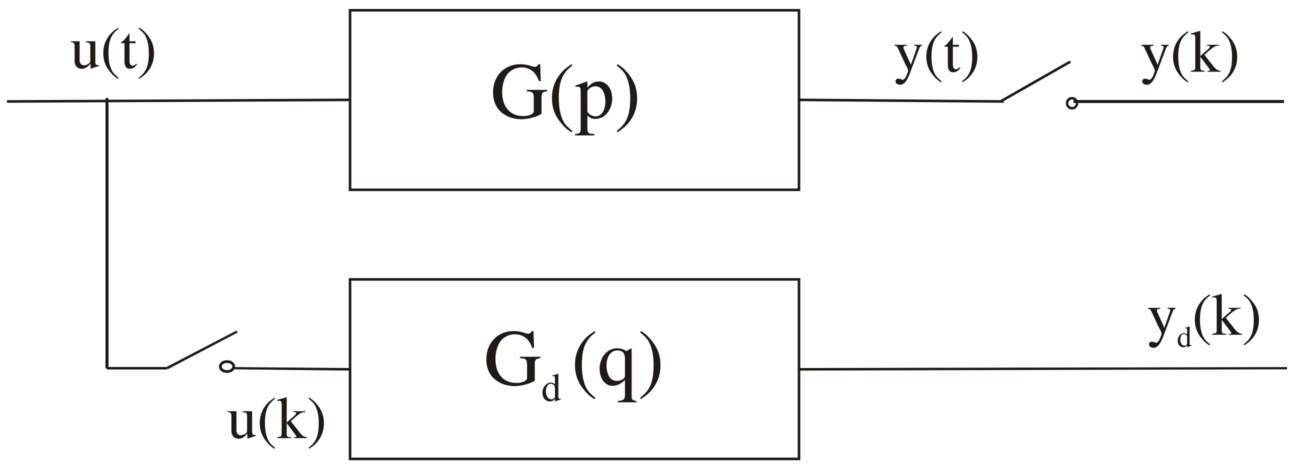
Name	Algorithm	Characteristics
Eulers forward rule	$s \rightarrow \frac{z - 1}{T}$	x'(t) constant over the period
Tustin (Bilinear Transformation)	$s \rightarrow \frac{2}{T}\frac{z - 1}{z + 1}$	x'(t) varies linearly over the period
Step Invariant (ZOH equivalent)	$G_d(z) = (1 - z^{-1})\, Z\{\frac{1}{s} G(s)\}$	u(t) constant over the period
Ramp Invariant (Tr H equivalent)	$G_d(z) = \frac{(1 - z^{-1})^2}{z^{-1} T} Z\{\frac{1}{s^2} G(s)\}$	u(t) varies linearly over the period
Pole-zero mapping	$z_0 = e^{s_o T}$	

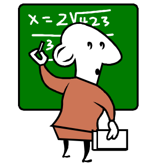


Invariance transformations

Given an analog system G(p). Determine the transfer function $G_d(q)$ of a digital system (the model) so the outputs are equal at the sampling times t=kT:

$$y_d(k)=y(k)$$





Simulating linear and nonlinear systems with Matlab

Senstools require a Matlab function: y=simproces(u,t,par)

Example: linear system K/(1+ sτ)

a) Using state space and for-loop

b) Using 'filter'

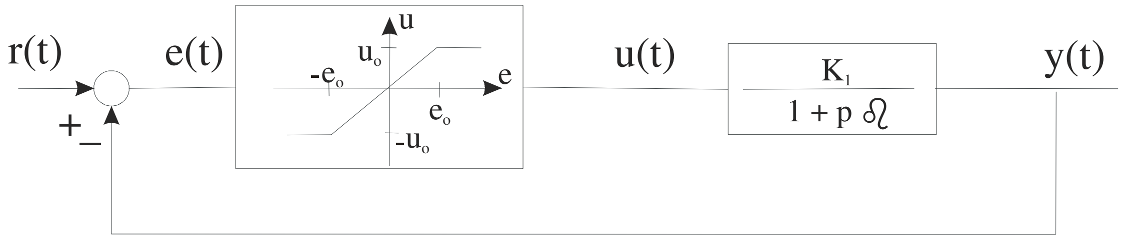
c) Using 'lsim'

Evaluation of the three methods:

a) is slow (loops in Matlab shall be avoided if possible), and should only be used only for nonlinear systems.

b) and c) are similar in performance

Simulating nonlinear system with Matlab



Saturation:

$e < -e_o$:	$u = -u_o$
$-e_o < e < e_o$:	$u = k\,e$
$e_o < e$:	$u = u_o$

This can be implemented in Matlab as

```
u=ke;
if e>e0,  u=u0;  end
if e<e0,  u=-u0; end
```

Using logical operations (a false statement has the value 0) more efficient as:

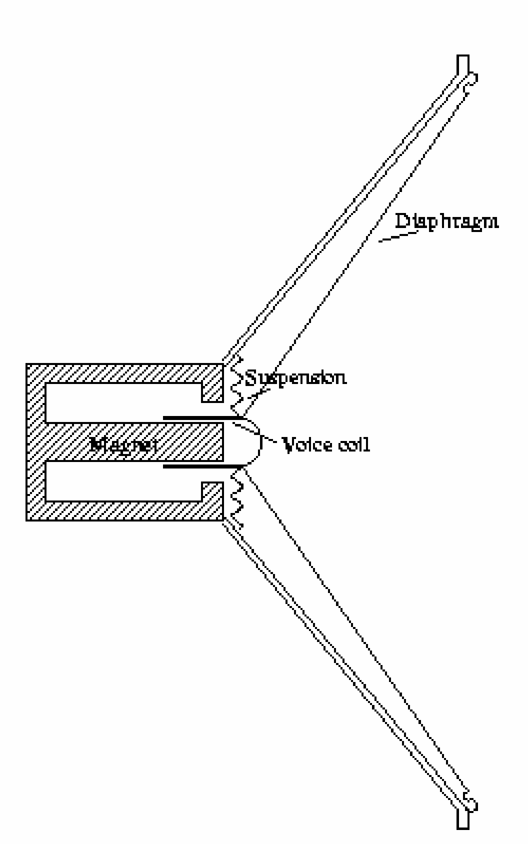
```
u=k*e-(abs(e)>e0)*k*(e-sign(e)*e0);
```

as $u_o=ke_o$ can be written as $u_o=ke-k(e-e_o)$ and $-u_o=ke-k(e+e_o)$

Modelling and simulation of loudspeaker

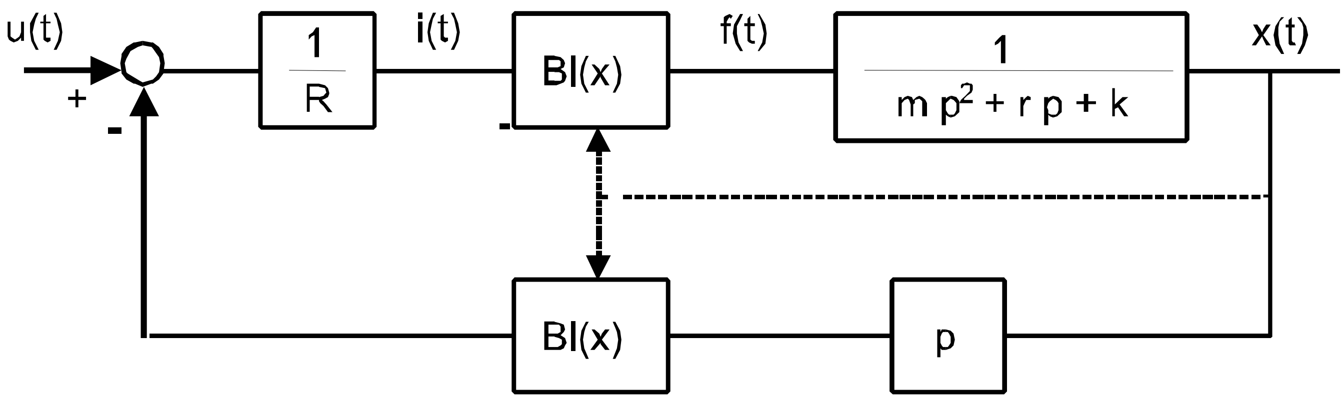
Symbols used:

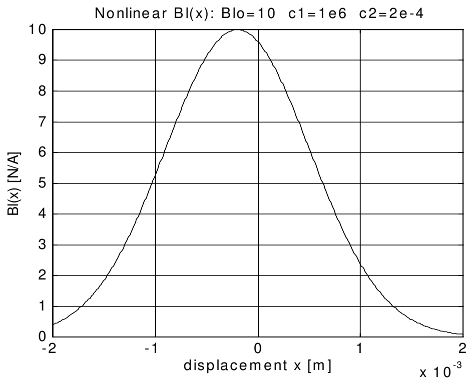
u   applied voltage   [V]
i   current in voice coil   [A]
x   displacement of voice coil   [m]
R   resistance of voice coil   [Ohm]
Bl   force factor   [N/A]
m   mass of moving system   [kg]
r   viscous friction coefficient   [Ns/m]
k   suspension stiffness   [N/m]



Nonlinear loudspeaker model

Displacement dependent force factor:





Blx=Bl*exp(-c1*(x+c2)^2);