UNIVERSITÄT
DES
SAARLANDES

Aktorik und Sensorik mit intelligenten Materialsystemen 4

Computer Lecture: Analytical Linearization of a DE-Mass-Spring Actuator in Matlab

Gianluca Rizzello

gianluca.rizzello@mmsl.uni-saarland.de

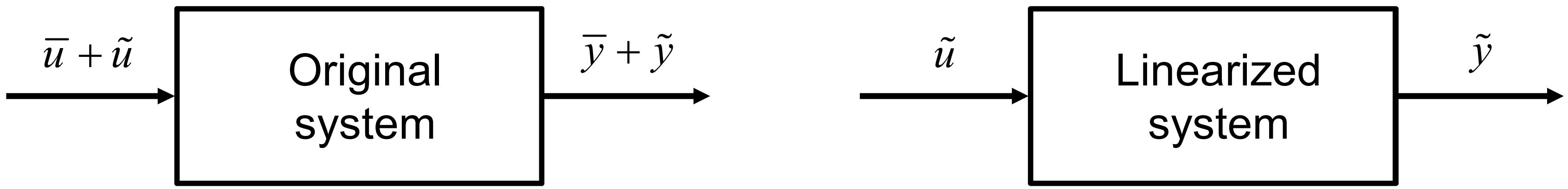
Saarland University

Computer Lecture Overview

- GOALS:
  - Obtain a linearized model describing the response of a DE-mass-spring actuator around an equilibrium state via the Taylor expansion method
  - Comparison between voltage-displacement response of nonlinear and linearized systems
  - Comparison between load force-displacement response of nonlinear and linearized systems

Analytical linearization

- Analyrical linearization: obtain a linearized model describing relationship between input deviations an output deviations from an equilibrium point, based on a Taylor expansion of model equations



- The analytical linearization requires:
  - Identification of an equilibrium state
  - Computation of partial derivatives of system functions

- These operations can be performed systematically in Matlab via Symbolic Toolbox

State-space modeling: DE mass spring

- State-space representation of the overall DE mass spring model, by considering the special case of a second order viscoelastic model

$$
\begin{cases}
\dot{x}_1 = x_2 \\
\dot{x}_2 = -\dfrac{k}{m}\left(x_1 - y_0\right) - g - \dfrac{V}{m x_1}\left\{\displaystyle\sum_{i=1}^{3}\mu_i\left[\left(\dfrac{x_1}{L_1}\right)^{\alpha_i} - \left(\dfrac{x_1}{L_1}\right)^{-\alpha_i}\right] - \varepsilon_0\varepsilon_r\left(\dfrac{x_1}{L_1 L_3}u\right)^2 + \displaystyle\sum_{j=1}^{2}k_{v,j}\left(\dfrac{x_1}{L_1} - 1 - x_{j+2}\right) + b_{v,0}\dfrac{x_2}{L_1}\right\} + \dfrac{1}{m}d \\
\dot{x}_3 = -\dfrac{k_{v,1}}{b_{v,1}}x_3 + \dfrac{k_{v,1}}{b_{v,1}}\left(\dfrac{x_1}{L_1} - 1\right) \\
\dot{x}_4 = -\dfrac{k_{v,2}}{b_{v,2}}x_4 + \dfrac{k_{v,2}}{b_{v,2}}\left(\dfrac{x_1}{L_1} - 1\right) \\
y = x_1
\end{cases}
$$

- Control input $u = v$
- Disturbance input $d = F_L$
- Output $y = l_1$
- State $x = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^T$

Symbolic Toolbox

- The symbolic toolbox offers a powerful framework to solve complex problems in an analytical way, such as computation of gradients of functions or the solution of nonlinear systems of equations

- The following steps need to be implemented:
  1) Definition of equilibrium control input u_{eq} and disturbance input d_{eq};
  2) Definition of symbolic variables for system states $x = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^T$, control input u, and disturbance input d;
  3) Definition of system state functions $f = [f_1 \;\; f_2 \;\; f_3 \;\; f_4]^T$ as well as output function $h = h_1$;
  4) Find an equilibrium state x_{eq} by solving $f(x_{eq}, u_{eq}, d_{eq}) = 0$, for the given values of u_{eq} and d_{eq};
  5) Compute the linearized model matrices by differentiating f and h and compute the corresponding gradient matrices on the obtained values for x_{eq}, u_{eq}, and d_{eq}.

- All these steps have been implemented in the Matlab script `Model_linearization.m`

Symbolic Toolbox

- Useful commands of the Symbolic Toolbox:
  - `x = sym('x')` – define symbolic variable `x`
  - `x = sym('x','real')` – define symbolic real variable `x`
  - `f = 2*x^2 + 3` – define `f` as a symbolic function of the symbolic variable `x`
  - `g = subs(f,x,2)` – given a symbolic function `f` of argument `x`, obtain a new symbolic function `g` by substituting `x = 2` in `f`
  - `g = subs(f,{x,y},{2,3})` – given a symbolic function `f` of arguments `x` and `y`, obtain a new symbolic function `g` by substituting `x = 2` and `y = 3` in `f`
  - `xnum = double(f)` – given a symbolic function `f` which, after a substitution, does not depend on any symbolic variable, convert it into the corresponding numerical value `xnum` which can be manipulated as a double
  - `sol = vpasolve(f == 0,x)` – given a symbolic function `f` of argument `x`, obtain `sol` as the `x` which solves the equation `f == 0`
  - `sol = vpasolve(f == 0,x,[-1 2])` – given a symbolic function `f` of argument `x`, obtain `sol` as the `x` which solves the equation `f == 0`, with `x` constrained in `[-1, 2]`
  - `sol = vpasolve(f == 0,[x y],[-1 2;0 inf])` – given a symbolic function `f` of arguments `x` and `y`, obtain `sol` as the `[x; y]` which solves `f == 0`, with `x` constrained in `[-1, 2]` and `y` constrained in `[0, inf]`
  - `g = diff(f,x)` – given a symbolic function `f` having `x` among its argument, obtain a new symbolic function `g` given by the partial derivative of `f` over `x`

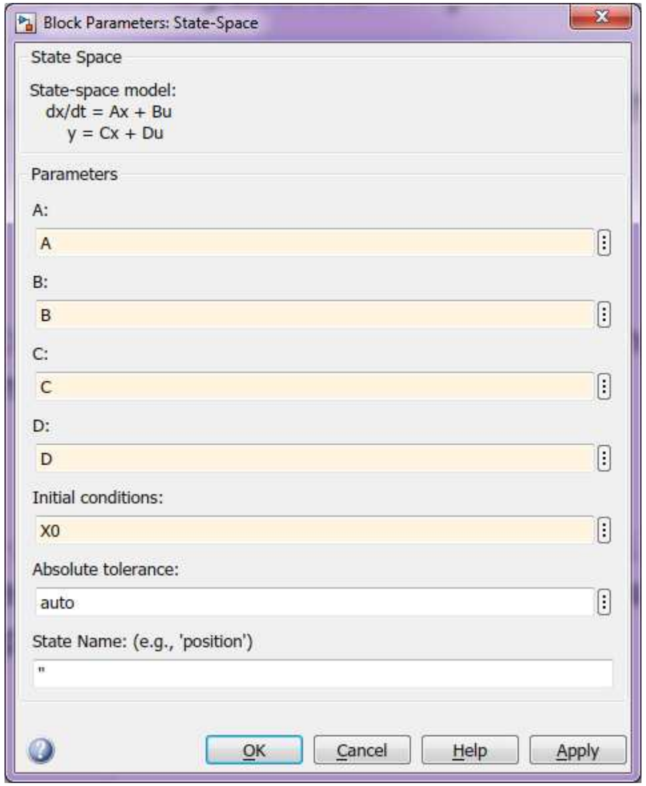
Linear Systems Implementation

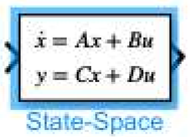
- Representations of linear state-space models in the form

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

- In Matlab:
  `S = ss(A,B,C,D)` – creates object `S` representing the linear state-space model defined by matrix variables `A`, `B`, `C`, `D`

- In Simulink:
  The block `state-space` located in `Simulink/Continuous` can be used to implement a linear state-space model, once matrices `A`, `B`, `C`, `D` and initial condition `X0` are specified





Assignment

- Given the DE-mass-spring model, perform the corresponding studies in Simulink:
  1) Consider the equilibrium corresponding to $u = 2500\ \mathrm{V}$ and $d = 0\ \mathrm{N}$, compare the output displacement of both nonlinear and linearized models when applying a voltage square wave perturbation from the equilibrium given by a square wave with period of 10 s, duty cycle of 50%, and amplitude equal to -10 V, -100 V, and -1000 V
  2) Consider the equilibrium corresponding to $u = 1250\ \mathrm{V}$ and $d = 0\ \mathrm{N}$, repeat the study of case 1 by considering voltage square waves with amplitude 10 V, 100 V, and 1000, -10 V, -100 V, and -1000 V
  3) Consider the equilibrium corresponding to $u = 0\ \mathrm{V}$ and $d = 0\ \mathrm{N}$, repeat the study of case 1 by considering voltage square waves with amplitude 10 V, 100 V, and 1000 V
  4) Consider the equilibrium corresponding to $u = 0\ \mathrm{V}$ and $d = 0\ \mathrm{N}$, repeat the study of case 1 by considering force square waves with amplitude 0.01 N, 0.1 N, and 1 N
  5) Repeat one of the above simulations by combining arbitrary equilibrium perturbations in both voltage and force inputs