Object Oriented CRONE Toolbox for system identification and control

Rachid MALTI, Stéphane VICTOR, Patrick LANUSSE, Pierre MELCHIOR, and Alain OUSTALOUP

Journées de l'Automatique du GDR MACS

5 et 6 Octobre 2015 – Grenoble

Bordeaux, FRANCE

Outline

Outline

Fractional derivatives and integrals

- Grünwald-Letnikov fractional derivatives
  - $\mathbf{D}^{\nu}x(t) = \lim_{h\to 0} \frac{1}{h^{\nu}} \sum_{k=0}^{\infty} (-1)^k \binom{k}{\nu} x(t-kh).$
  - $\nu = 1 \Rightarrow \lim_{h\to 0} \frac{x(t-kh)-x(t)}{h}.$
- Fractional differential equations
  - $y(t)+a_1\mathbf{D}^{\alpha_1}y(t)+\cdots+a_N\mathbf{D}^{\alpha_N}y(t) = b_0\mathbf{D}^{\beta_0}u(t)+\cdots+b_M\mathbf{D}^{\beta_M}u(t),$
- Laplace transform
  - $\mathscr{L}\{\mathbf{D}^{\nu}x(t)\} = s^{\nu}X(s).$
- History of the CRONE Toolbox
  - Development started in the late 1990's as a standard non-OO toolbox,
  - Development of the OO-CRONE toolbox starting from 2004,
  - Free downloads from 2011.

Fractional polynomials

- Fractional explicit polynomials (Cole-Cole transfer functions):

$$p(s) = \sum_{i=0}^{L} c_i s^{\gamma_i}$$

Characterized by two linked sequences

$[c_0, c_1, \ldots, c_L]$ and $[\gamma_0, \gamma_1, \ldots, \gamma_L]$.

- Fractional implicit polynomials (Havriliak-Negami transfer functions):

$$\tilde{p}(s) = p(s)^{\beta} = \left(\sum_{i=0}^{L} c_i s^{\gamma_i} \right)^{\beta},$$

Characterized by

An explicit polynomial and a diff order β.

Fractional polynomials

- Fractional explicit polynomials (Cole-Cole transfer functions):

$$p(s) = \sum_{i=0}^{L} c_i s^{\gamma_i}$$

Characterized by two linked sequences

$[c_0, c_1, \ldots, c_L]$ and $[\gamma_0, \gamma_1, \ldots, \gamma_L]$.

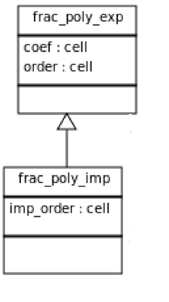
- Fractional implicit polynomials (Havriliak-Negami transfer functions):

$$\tilde{p}(s) = p(s)^{\beta} = \left(\sum_{i=0}^{L} c_i s^{\gamma_i}\right)^{\beta},$$

Characterized by

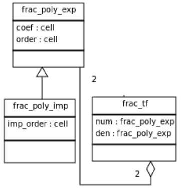
An explicit polynomial and a diff order β.

UML diagram



Fractional system representation

Frac_tf



- Fractional transfer function in a developed form:

$$H(s) = \frac{\sum_{i=0}^{M} b_i s^{\beta_i}}{1+\sum_{j=1}^{N} a_j s^{\alpha_j}}$$

- Fractional transfer function in a factorized form:

$$H(s) = K\frac{\prod_{i=0}^{m} (s^{\nu}+z_i)}{\prod_{j=0}^{n} (s^{\nu}+p_j)}$$

- Fractional (or pseudo-) state space representation:

$$\mathbf{D}^{\nu}\mathbf{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

Fractional system representation

- Fractional transfer function in a developed form:

$$H(s) = \frac{\sum_{i=0}^{M} b_i s^{\beta_i}}{1+\sum_{j=1}^{N} a_j s^{\alpha_j}}$$

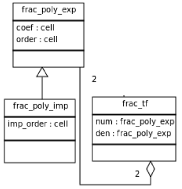
- Fractional transfer function in a factorized form:

$$H(s) = K\frac{\prod_{i=0}^{m} (s^{\nu}+z_i)}{\prod_{j=0}^{n} \left(s^{\nu}+p_j\right)}$$

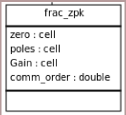
- Fractional (or pseudo-) state space representation:

$$\mathbf{D}^{\nu}\mathbf{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

Frac_tf



Frac_zpk



Fractional system representation

- Fractional transfer function in a developed form:

$$H(s) = \frac{\sum_{i=0}^{M} b_i s^{\beta_i}}{1+\sum_{j=1}^{N} a_j s^{\alpha_j}}$$

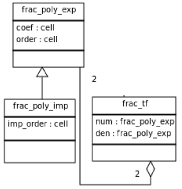
- Fractional transfer function in a factorized form:

$$H(s) = K \frac{\prod_{i=0}^{m} (s^{\nu}+z_i)}{\prod_{j=0}^{n} (s^{\nu}+p_j)}$$

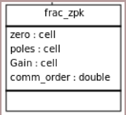
- Fractional (or pseudo-) state space representation:

$$\mathbf{D}^{\nu}\mathbf{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

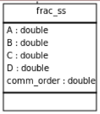
Frac_tf



Frac_zpk



Frac_ss



Outline

Methods based on discrete-time models

- Approximation of a fractional differentiator by its discrete-time equivalent: $s^\nu \to \psi(z^{-1})$
- As a result a discrete-time transfer function is obtained:

$$\mathcal{H}(z^{-1}) = H(\psi(z^{-1})) = \frac{\sum_{i=0}^{M} b_i \psi(z^{-1})^{\beta_i}}{1 + \sum_{j=1}^{N} a_j \psi(z^{-1})^{\alpha_j}}$$

- The discretization operator $\psi(z^{-1})$ of analogue circuits can be any of the usual operators. Euler's operator (Grünwald definition) is implemented in the CRONE toolbox:

$$\psi(z^{-1}) = \left(\frac{1-z^{-1}}{T_s}\right)^\nu = \left(\frac{1}{T_s}\right)^\nu \sum_{k=0}^{\infty} (-1)^k \binom{\nu}{k} z^{-k},$$

Characterized by

a sampling period T_s.

Methods based on discrete-time models

- Approximation of a fractional differentiator by its discrete-time equivalent: $s^\nu \to \psi(z^{-1})$
- As a result a discrete-time transfer function is obtained:

$$\mathcal{H}(z^{-1}) = H(\psi(z^{-1})) = \frac{\sum_{i=0}^{M} b_i \psi(z^{-1})^{\beta_i}}{1 + \sum_{j=1}^{N} a_j \psi(z^{-1})^{\alpha_j}}$$

- The discretization operator $\psi(z^{-1})$ of analogue circuits can be any of the usual operators. Euler's operator (Grünwald definition) is implemented in the CRONE toolbox:

$$\psi(z^{-1}) = \left(\frac{1-z^{-1}}{T_s}\right)^\nu = \left(\frac{1}{T_s}\right)^\nu \sum_{k=0}^{\infty} (-1)^k \binom{\nu}{k} z^{-k},$$

Characterized by

a sampling period T_s.

Methods based on continuous-time models

Based on the approximation of a fractional model by a rational continuous-time one in a given frequency band.

- Let $s^\gamma = s^\gamma_{[\omega_A,\omega_B]} \quad \forall\omega \in [\omega_A, \omega_B]$ with $0 < \gamma < 1$
- Oustaloup's approximation: $s^\gamma_{[\omega_A,\omega_B]} \approx \mathscr{A}^{(\gamma)}_{\mathrm{Oust}} = C_{(\gamma)} \left(\frac{1+\frac{s}{\omega_h}}{1+\frac{s}{\omega_b}}\right)^\gamma$
  - $\omega_b = \sigma^{-1}\omega_A$ and $\omega_h = \sigma\omega_B$ (σ is usually set to 10)
  - $C_{(\gamma)}$ is chosen to get a unit gain at $\omega = 1 \text{ rad s}^{-1}$:
  - $C_{(\gamma)} = \left|\frac{1+\mathrm{j}\frac{1}{\omega_h}}{1+\mathrm{j}\frac{1}{\omega_b}}\right|^{-\gamma} = \left(\frac{\omega_h}{\omega_b}\right)^\gamma \left(\frac{1+\omega_b^2}{1+\omega_h^2}\right)^{\frac{\gamma}{2}}$
- Trigeassou's variant: $s^\gamma_{[\omega_A,\omega_B]} \approx \mathscr{A}^{(\gamma)}_{\mathrm{Trig}} = C_{(\gamma-1)}s\left(\frac{1+\frac{s}{\omega_h}}{1+\frac{s}{\omega_b}}\right)^{\gamma-1}$

Oustaloup's approximation	Trigeassou's variant
ω_A, ω_B, N	ω_A, ω_B, N

Methods based on continuous-time models

- Oustaloup and Trigeassou approximate the irrational part by a recursive distribution of poles and zeros:
  - $\left(\frac{1+\frac{s}{\omega_h}}{1+\frac{s}{\omega_b}}\right)^{\gamma} \approx \prod_{k=1}^{N}\left(\frac{1+\frac{s}{\omega_k}}{1+\frac{s}{\omega'_k}}\right)$
  - $\frac{\omega_{k+1}}{\omega_k} = \frac{\omega'_{k+1}}{\omega'_k} = \alpha\eta, \quad \frac{\omega_k}{\omega'_k} = \alpha, \quad \frac{\omega'_{k+1}}{\omega_k} = \eta, \quad \gamma = \frac{\log(\alpha)}{\log(\alpha)+\log(\eta)}$
  - $\alpha = \left(\frac{\omega_h}{\omega_b}\right)^{\frac{\gamma}{N}}$ and $\eta = \left(\frac{\omega_h}{\omega_b}\right)^{\frac{1-\gamma}{N}}$
- Rational TF, equivalent a fractional TF:

$$H(s) = \frac{\sum_{i=0}^{M} b_i s^{\beta_i}}{1+\sum_{j=1}^{N} a_j s^{\alpha_j}} \approx \mathscr{H}(s) = \frac{\sum_{i=0}^{M} b_i s^{\lfloor\beta_i\rfloor}\mathscr{A}^{(\beta_i-\lfloor\beta_i\rfloor)}}{1+\sum_{j=1}^{N} a_j s^{\lfloor\alpha_j\rfloor}\mathscr{A}^{(\alpha_j-\lfloor\alpha_j\rfloor)}},$$

where $\mathscr{A}^{(\gamma)}$ is either of Oustaloup's or Trigeassou's approximation of the fractional operators s^{γ}, with $0 < \gamma < 1$.

Methods based on continuous-time models

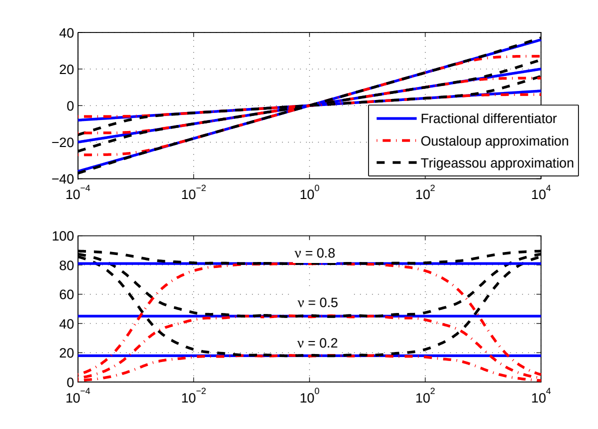


Figure : Approximation of ideal differentiators using Oustaloup's and Trigeassou's methods

Methods based on continuous-time models

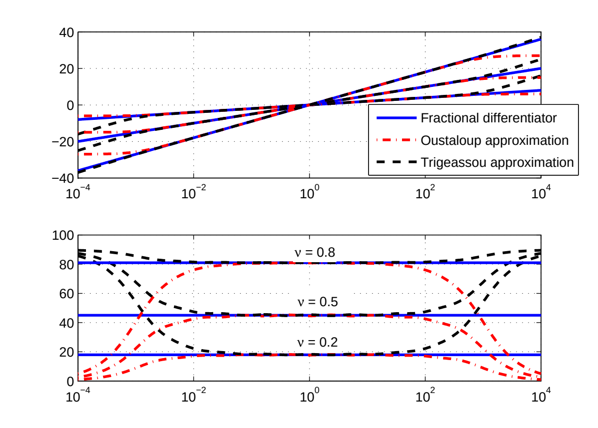
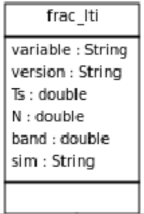


Figure : Approximation of ideal differentiators using Oustaloup's and Trigeassou's methods

Oustaloup's and Trigeassou's approximations

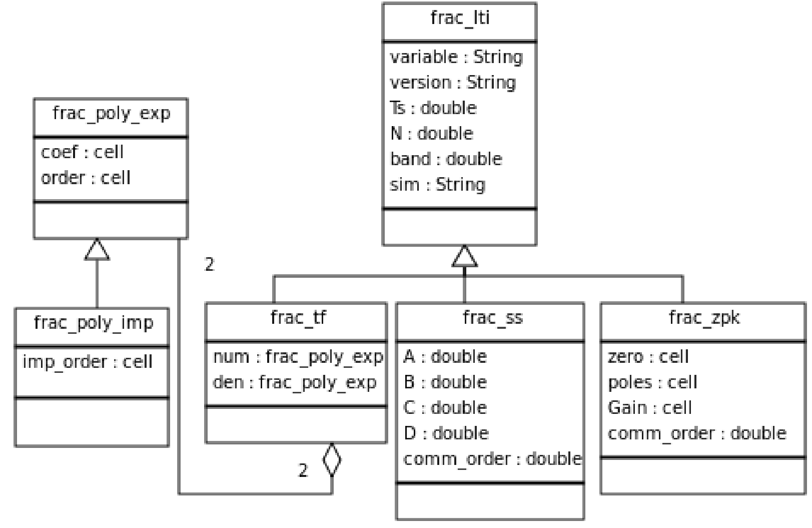
ω_A, ω_B, N

frac_lti class



Outline

Class diagram of the OO-CRONE toolbox – Attributes



Methods associated to the OO-CRONE toolbox

- General purpose methods
  - get, set used to access data (data encapsulation),
  - isnan, isempty, size, length, iscomplex for general purpose op,
  - horzcat, vertcat, subsref, subsasgn for MIMO system handling,
- Methods associated to operator overloading
  - The main operators $(+,-,\times,.\times,/,\backslash,',=,\ =,...)$ are overloaded by rewritting plus, minus, uminus, mtimes, times, ldivide, rdivide, transpose, eq, ne, display scripts,
- Methods associated automatic control (many methods developed for fractional TF, some methods implemented for MIMO TF)
  - Frequency-domain simulation bode, nichols, nyquist,
  - Time-domain simulation lsim (with various options),
  - System identification oe, lssvf, ivsvf, srivcf, oosrivcf,
  - Stability isstable.

A focus on isstable method in the frac_tf-class

Based on Matignon's stability theorem :

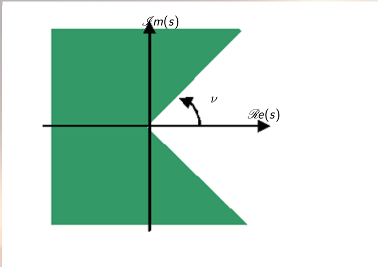


Figure : Matignon's stability theorem

Hence,

$$F(s) = \frac{1}{s + s^{0.5} + 1}$$

must have all its $s^{0.5}$-poles in the sector defined by:

$$|\arg(s^{0.5})| > \pi/4$$

An open problem

However, due to floating point arithmetics, the following TF might be coded in computers instead of $F(s)$:

$$F_\epsilon(s) = \frac{1}{s + s^{0.5+\epsilon} + 1}$$

$F_\epsilon(s)$ is comm. of ordre ϵ. Hence all the roots of the (with $p = s^\epsilon$):

$$p^{\frac{1}{\epsilon}} + p^{\frac{0.5}{\epsilon}+1} + 1$$

needs to be evaluated, which is impossible if ϵ is the machine-ϵ.

Problem formulation

- If $F(s) = \frac{1}{s+s^{0.5}+1}$ is stable with a certain margin, is it possible to conclude on the stability of $F_\epsilon(s) = \frac{1}{s+s^{0.5+\epsilon}+1}$?
- How to find that margin?

Outline

System Identification 1/2

- The toolbox supports transfer function models of the following forms:

$$y(t_k) = \frac{B(\mathcal{D})}{A(\mathcal{D})} u(t_k) + \frac{1}{F(\mathcal{D})} e(t_k), \tag{1}$$

- In the output error context, the following criterion is minimized

$$J = \sum_{k=1}^{k=K} (y(t_k) - y_m(t_k))^2 \tag{2}$$

$$H(s) = \frac{\sum\limits_{i=0}^{M} b_i s^{\beta_i}}{1 + \sum\limits_{j=1}^{N} a_j s^{\alpha_j}}$$

$$H(s) = \frac{\sum\limits_{i=0}^{m} \tilde{b}_i s^{i\nu}}{1 + \sum\limits_{j=1}^{n} \tilde{a}_j s^{j\nu}}$$

$$\nu \in (0, 2)$$

System Identification 2/2

Sys. Id.	OE	ARX
oe	✓	
lssvf		✓
ivsvf		✓
srivcf	✓	
oosrivcf	✓	

Table : Methods for system identification using fractional models

Sys. Id. methods	Coefficients estimation	Commensurate order estimation	All order estimation
oe	✓	✓	✓
lssvf	✓		
ivsvf	✓		
srivcf	✓		
oosrivcf	✓	✓	✓

Table : Coefficient and/or order estimation.

Crone Control System Design tools 1/2

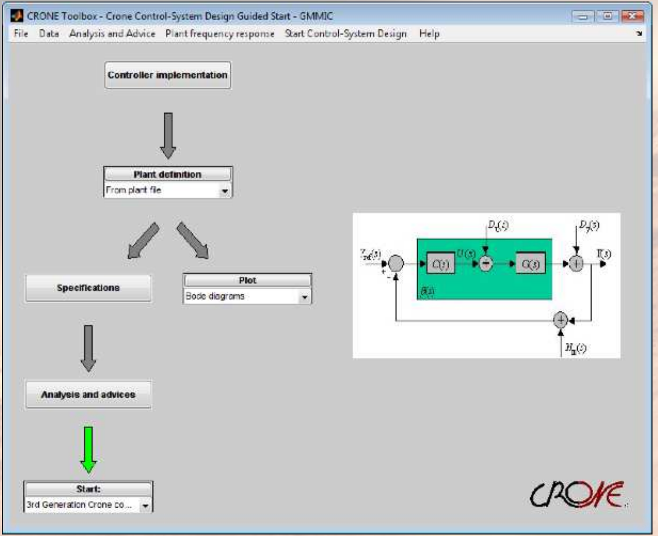


Figure : Crone CSD – user interface.

Crone Control System Design tools 2/2

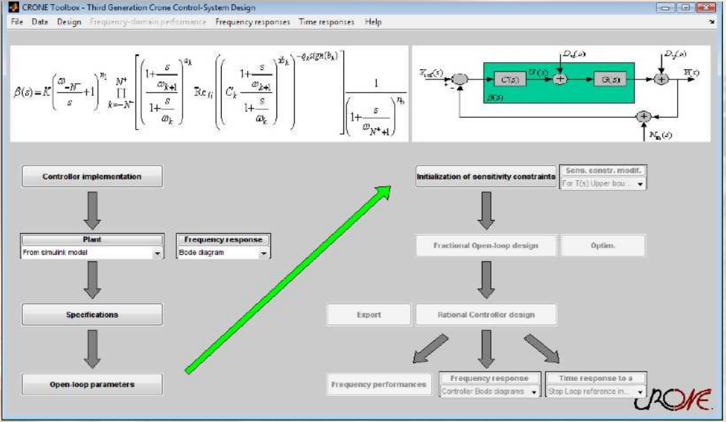


Figure : Crone CSD – user interface.

Outline

Example 1 – Mass-spring-fractor

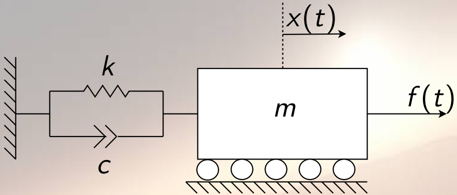


Figure : Mechanical system including a fractor $\nu \in [0\ 1]$.

Constitutive equation in time domain ($x(t) = 0, f(t) = 0 \forall t < 0$):

$$m\mathcal{D}^2 x(t) + c\mathcal{D}^\nu x(t) + kx(t) = f(t)$$

In the $\mathscr{L}$aplace domain (with $\omega_0 = \sqrt{\frac{k}{m}}, \zeta = \frac{ck^{\frac{\nu}{2}-1}}{2m^{\frac{\nu}{2}}}$):

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2 + cs^\nu + k} = \frac{1/k}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\left(\frac{s}{\omega_0}\right)^\nu + 1}.$$

Example 1 – Matlab code – Frequency domaine behaviour

$$\frac{X(s)}{F(s)} = \frac{1/k}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\left(\frac{s}{\omega_0}\right)^\nu + 1}.$$

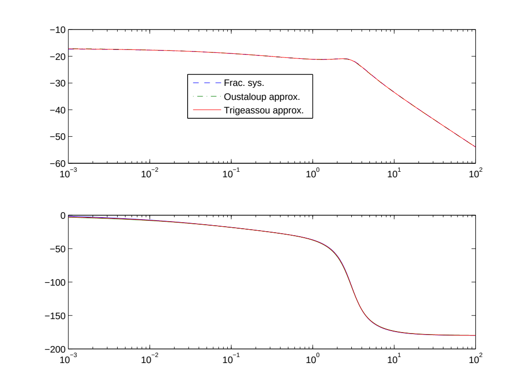
$m = 25\text{Kg},\ k = 50\text{N/m},\ \nu = 0.5,\ c = 110\text{Kg/s}^\nu.$

```
m = 25; k = 50; c = 110; nu = 0.5; %Kg, N/m, kg/s^\nu
zeta = c * k^(nu/2 - 1)/(2*m^(nu/2)); w0 = sqrt(k/m)
M = frac_tf(1/k, ...
     frac_poly_exp([1/w0^2  2*zeta/w0^nu  1], ...
     [2, nu, 0]), 10, [1e-3 1e2])
M_Oust = frac2int(M);
set(M, 'sim', 'Trig'); M_Trig = frac2int(M);

[G, Ph, w] = bode(M, [1e-3 1e2]);
[GO, PhO, w] = bode(M_Oust, w);
[GT, PhT, w] = bode(M_Trig, w);
```

Example 1 – Frequency domain behaviour

$$\frac{X(s)}{F(s)} = \frac{1/k}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\left(\frac{s}{\omega_0}\right)^\nu + 1}.$$

- $\omega_0 = 1.41, \zeta = 1.31$
- Oustaloup's approx and Trigeassou's variant
  - $[\omega_a, \omega_b] = [10^{-2}, 10^3]$
  - $N = 10$



Example 1 – Matlab code – Time-domain simulation

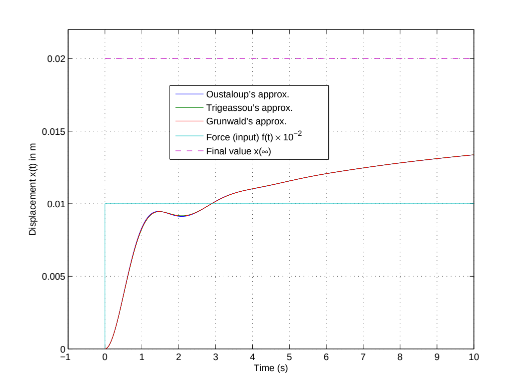
$$\frac{X(s)}{F(s)} = \frac{1/k}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\left(\frac{s}{\omega_0}\right)^\nu + 1}.$$

```
Ts = 0.001; u = [zeros(0, 1) ; ones(10000, 1)];
t = ((0:length(u)-1)*Ts)';
set(M, 'sim', 'Oust'); yO = lsim(M, u, t);
set(M, 'sim', 'Trig'); yT = lsim(M, u, t);
set(M, 'sim', 'grun'); yG = lsim(M, u, t);
figure, plot(t, yO, t, yT, t, yG, [0; t], ...
     [0; u]*1e-2, [0, 10], [1/k  1/k], '--')
```

Example 1 – Time domain simulation

$$\frac{X(s)}{F(s)} = \frac{1/k}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\left(\frac{s}{\omega_0}\right)^{\nu} + 1}.$$

- $\omega_0 = 1.41, \zeta = 1.31$
- Oustaloup's approx and Trigeassou's variant
  - $[\omega_a, \omega_b] = [10^{-2}, 10^3]$
  - $N = 10$



Example 2 – Thermal diffusion

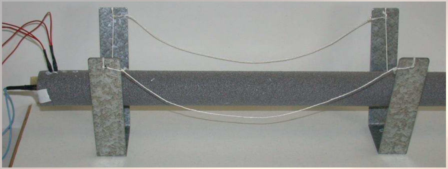


Figure : Thermal aluminium rod heated at one end.

Assumptions

1. The rod is perfectly isolated,
2. the rod is considered as a semi-infinite homogeneous plane medium with conductivity λ and diffusivity α,
3. at rest, the rod is at ambient temperature,
4. losses on the surface where the thermal flux is applied are neglected.

Example 2 – Thermal diffusion – Physical modeling

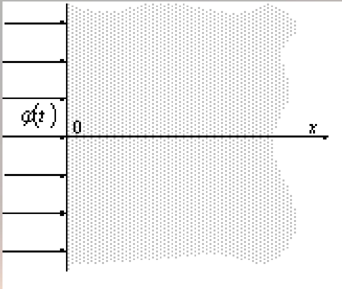


Figure : Semi-infinite plannar medium

[Battaglia et al, 2001]

$$\begin{cases} \frac{\partial T(x,t)}{\partial t} = \alpha \frac{\partial^2 T(x,t)}{\partial x^2}, 0 < x < \infty, t > 0 \\ -\lambda \frac{\partial T(x,t)}{\partial x} = \varphi(t), \quad x = 0, \quad t > 0 \\ T(x,t) = 0, \quad 0 \le x < \infty, \quad t = 0 \end{cases}$$

- Evaluating the Laplace transform:

$$\frac{\partial^2 \bar{T}(x,s)}{\partial x^2} - \frac{s}{\alpha} \bar{T}(x,s) = 0,$$
$$\text{where } \bar{T}(x,s) = \mathscr{L}\{T(x,t)\}.$$

- Solving with respect to x yields:

$$\bar{T}(x,s) = K_1(s)\, e^{-x\sqrt{\frac{s}{\alpha}}} + K_2(s)\, e^{x\sqrt{\frac{s}{\alpha}}}.$$

- Taking into account limit conditions, the following transfer function is obtained:

$$H(x,s) = \frac{\bar{T}(x,s)}{\bar{\varphi}(s)} = \frac{\sqrt{\alpha}}{\lambda\sqrt{s}}\, \mathrm{e}^{-x\sqrt{\frac{s}{\alpha}}}.$$

Example 2 – Thermal diffusion – Physical modeling

P^{th}-order Padé approximation of $H(x, s)$, at $x = x^*$:

$$H(x^*, s) \approx H_P(s) = \frac{\sqrt{\alpha}}{\lambda\sqrt{s}} \frac{\sum_{k=0}^{P} \frac{(2P-k)!}{k!(P-k)!}\left(-x^*\sqrt{\frac{s}{\alpha}}\right)^k}{\sum_{k=0}^{P} \frac{(2P-k)!}{k!(P-k)!}\left(x^*\sqrt{\frac{s}{\alpha}}\right)^k}.$$

The integrator $H_0(s)$ and a first order order Padé approximation $H_1(s)$:

$$H_0(s) = \frac{4.21 \times 10^{-5}}{s^{0.5}},$$

$$H_1(s) = \frac{10^{-5}}{s^{0.5}} \left(\frac{-2.11 s^{0.5} + 8.43}{0.50 s^{0.5} + 2.00}\right).$$

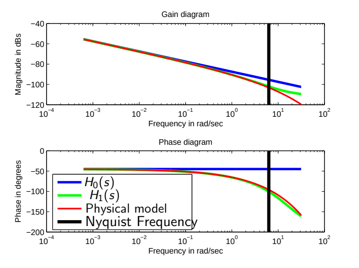


Figure : Physical model $H(x, s)$ and its Padé approximations $H_0(s)$ and $H_1(s)$

Example 2 – System Identification from experimental data

- The aluminium rod is driven to a steady state by injecting a constant heat flux.
- a prbs signal is generated around the constant flux.

```
load('ThermalRodData.mat')
data = iddata(y,u,Ts);
sys_init = frac_tf(1, frac_poly_exp([1 1 1], ...
    [1.2 0.6 0]), 24, [1e-5 5e1]);
sys_oe_coef = oe(sys_init, data,[],'coef');
sys_oe_comm = oe(sys_oe_coef, data,[],'comm');
sys_oe_all = oe(sys_oe_comm, data,[],'all');
figure, subplot(211),plot(t,y,'b',...
    t,lsim(sys_oe_coef,u,t),'r--',...
    t,lsim(sys_oe_comm,u,t),'g-.',...
    t,lsim(sys_oe_all,u,t),'m')
```

Figure : Matlab script showing the use of oe routine (system identification) of the CRONE toolbox

Example 2 – System Identification from experimental data

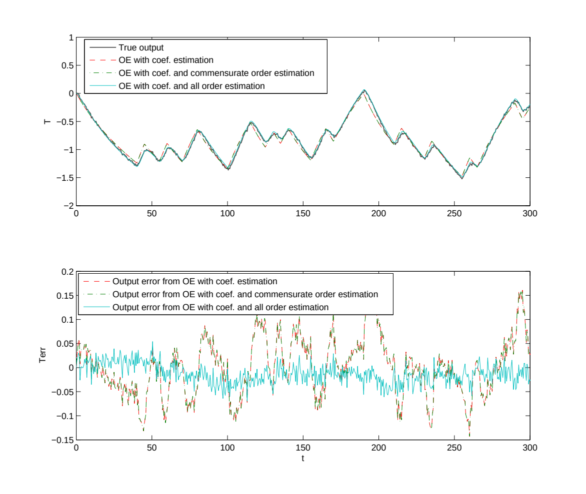


Figure : Comparison among the three outputs of the oe routine. Similar results are obtained with oosrivcf routine

Example 3 – MIMO systems

$sys = \frac{1}{5s^{1.5}+1}, \quad [\omega_b, \omega_h] = [10^{-2}, 10^{2}] \Rightarrow [\omega_A, \omega_B] = [10^{-1}, 10^{1}],\ N = 4$

Are the approximations satisfactory in the frequency-domain?

Example 3 – MIMO systems

$$sys = \frac{1}{5s^{1.5}+1}, \quad [\omega_b, \omega_h] = [10^{-2}, 10^{2}] \Rightarrow [\omega_A, \omega_B] = [10^{-1}, 10^{1}],\ N = 4$$

Are the approximations satisfactory in the frequency-domain?

```
% Frequency plot  -- bode --
set(sys,'sim','Oust'); ApproxOust=frac2int(sys);
set(sys,'sim','Trig'); ApproxTrig=frac2int(sys);

[G, ph, w] = bode(sys, [1e-2  1e1]);
[GOust, phOust] = bode(ApproxOust, w);
[GTrig, phTrig] = bode(ApproxTrig, w);
figure(2), subplot(211)
semilogx(w, 10*log10(squeeze(GOust)), '--',...
    w, 10*log10(squeeze(GTrig)), '-.', ...
    w,10*log10(squeeze(G))),grid,
subplot(212), semilogx(w,squeeze(phOust),w,...
    squeeze(phTrig), w, squeeze(ph)), grid,
```

Example 3 – MIMO systems

$$sys = \frac{1}{5s^{1.5}+1}, \quad [\omega_b, \omega_h] = [10^{-2}, 10^2] \Rightarrow [\omega_A, \omega_B] = [10^{-1}, 10^1],\ N = 4$$

Are the approximations satisfactory in the frequency-domain?

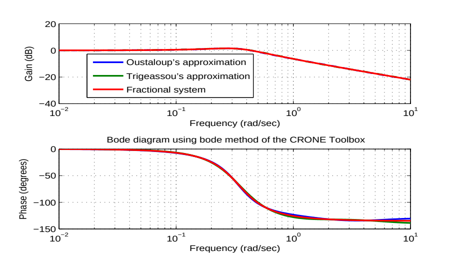


Figure : Approximations using Oustaloup's and Trigeassou's methods in the frequency band $[10^{-1} 10^{1}]$ with $N = 4$

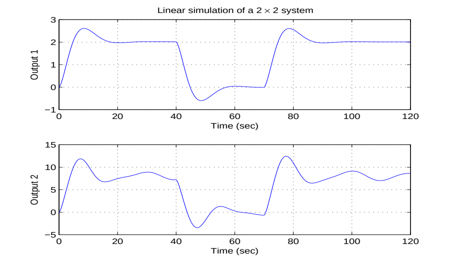
Overloading operators and MIMO time-response

```
% Example of operator overloading
sys2=(-sys^3+5*sys)

 Frac_tf transfer function :
        ( 625 s^4.5 + 375 s^3 + 70 s^1.5 + 4 )
-------------------------------------------------
( 625 s^6 + 500 s^4.5 + 150 s^3 + 20 s^1.5 + 1 )

% MIMO example
sysMIMO = [sys, sys ; sys2, sys2]
figure(3), lsim(sysMIMO, [u ; u], t);

Frac tf from input 1 to output:
#1 :  Frac_tf transfer function :
        ( 1 )
----------------
( 5 s^1.5 + 1 )
#2 :  Frac_tf transfer function :
```

Example 3

$$sys = \left(\frac{\frac{1}{5s^{1.5}+1}}{-\left(\frac{1}{5s^{1.5}+1}\right)^3 + 5 \times \frac{1}{5s^{1.5}+1}} \quad \frac{\frac{1}{5s^{1.5}+1}}{-\left(\frac{1}{5s^{1.5}+1}\right)^3 + 5 \times \frac{1}{5s^{1.5}+1}} \right)$$

Time-domain simulation of a fractional MIMO system



Outline

Prospectives – General information

- Provide technical manuals
- Develop a GUI

Download

http://cronetoolbox.ims-bordeaux.fr

Any question, bug report, etc.

A forum is provided.

Test functions

Every developed function is tested with multiple cases. All test functions are provided for the users.

Enjoy!

Prospectives – General information

- Provide technical manuals
- Develop a GUI

Download

http://cronetoolbox.ims-bordeaux.fr

Any question, bug report, etc.

A forum is provided.

Test functions

Every developed function is tested with multiple cases. All test functions are provided for the users.

Enjoy!

Website – Free downloads

Figure : http://cronetoolbox.ims-bordeaux.fr