ISS0023 Intelligent Control Systems
Fractional-order Calculus based Modeling and Control of Dynamic Systems

Aleksei Tepljakov, Ph.D.

December 1, 2017
Lecture overview

- Mathematical basis of fractional-order calculus;
- Fractional-order calculus in modeling and control:
  - Analysis of fractional models;
  - Implementations of fractional-order systems;
  - $PI^\lambda D^\mu$ controllers and their design.
- Overview of CACSD tools and examples of practical applications:
  - Introduction to FOMCON toolbox for MATLAB;
  - Control design and implementation examples.
Part I: Mathematical Basis of Fractional-order Calculus
The concept of the differentiation operator $\mathcal{D} = \frac{d}{dx}$ is a well-known fundamental tool of modern calculus. For a suitable function $f$ the $n$-th derivative is well defined as

$$\mathcal{D}^n f(x) = \frac{d^n f(x)}{dx^n},$$

where $n$ is a positive integer.

What happens if we extend this concept to a situation, when the order of differentiation is arbitrary, for example, fractional?

That was the very same question L’Hôpital addressed to Leibniz in a letter in 1695. Since then the concept of fractional calculus has drawn the attention of many famous mathematicians, including Euler, Laplace, Fourier, Liouville, Riemann, Abel.
For the power function $f(x) = x^k$ the fractional derivative can be shown to be

$$\frac{d^\alpha f(x)}{dx^\alpha} = \frac{\Gamma(k + 1)}{\Gamma(k - \alpha + 1)}x^{k-\alpha}.$$  \hspace{1cm} (2)

The function $\Gamma(\cdot)$ above is the Gamma function—the generalization of the factorial function:

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt, \quad x > 0.$$ \hspace{1cm} (3)

Example:

$$\frac{d^{1/2}(x^2)}{dx^{1/2}} = \frac{\Gamma(3)}{\Gamma(5/2)}x^{3/2} = \frac{8x^{3/2}}{3\sqrt{\pi}}.$$
The Gamma function
Example: fractional-order derivative of a function $f(x) = x$
We observe, what happens when we repeatedly differentiate the function \( f(x) = \sin x \):

\[
\frac{d}{dx} \sin x = \cos x, \quad \frac{d^2}{dx^2} \sin x = -\sin x, \quad \frac{d^3}{dx^3} \sin x = -\cos x, \ldots
\]

The pattern can be deduced: for the \( n \)th derivative, the function \( \sin x \) is shifted by \( n\pi/2 \) radians. This can be observed from studying the graph of the function. Thus, if we replace \( n \) by \( \alpha \in \mathbb{R}_+ \), we have

\[
\frac{d^\alpha}{dx^\alpha} \sin x = \sin \left( x + \frac{\alpha\pi}{2} \right). \quad (4)
\]

Obviously, a similar equation holds for the cosine function as well.
Half derivative of a sine function

\[ \frac{d^{0.5}}{dt^{0.5}} \sin(t) \]

\[ \frac{\pi}{4} \]

Amplitude

Time
Repeated differentiation: Backward difference equation

Recall the backward difference definition of $f'(x)$ given by

$$f'(x) = \lim_{h \to 0} \frac{f(x) - f(x - h)}{h}. \quad (5)$$

It follows, that

$$f''(x) = \lim_{h \to 0} \frac{f'(x) - f'(x - h)}{h} = \lim_{h \to 0} \frac{f(x) - 2f(x - h) + f(x - 2h)}{h^2}.$$  

Furthermore,

$$f'''(x) = \lim_{h \to 0} \frac{f(x) - 3f(x - h) + 3f(x - 2h) - f(x - 3h)}{h^3}.$$  

And in general

$$f^{(n)}(x) = \lim_{h \to 0} \frac{1}{h^n} \sum_{k=0}^{n} (-1)^k \binom{n}{k} f(x - kh). \quad (6)$$
Can we generalize this to the case $n \in \mathbb{R}_+$?

Of course! All we need to do is to consider the factorial formula for the binomial coefficient and use the ever so kind Gamma function to lend a helping hand in case we have $\alpha \in \mathbb{R}_+$:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad \rightarrow \quad \binom{\alpha}{k} = \frac{\Gamma(\alpha + 1)}{\Gamma(k + 1)\Gamma(\alpha - k + 1)}.$$  \hfill (7)

We find that this approach is the very basis for Grünwald-Letnikov’s definition of the fractional-order derivative. In fact, here it is:

**Definition 1.** (Grünwald-Letnikov)

$$^GLD^\alpha f(t)\big|_{t=nh} = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{k=0}^{n} (-1)^k \binom{\alpha}{k} f(nh - kh).$$  \hfill (8)
Fractional-order derivative: Important alternative definitions

Definition 2. (Riemann-Liouville)

\[ R_a^\alpha D_t^\alpha f(t) = \frac{1}{\Gamma(m - \alpha)} \left( \frac{d}{dt} \right)^m \left[ \int_a^t \frac{f(\tau)}{(t - \tau)^{\alpha-m+1}} d\tau \right], \quad (9) \]

where \( m - 1 < \alpha < m \), \( m \in \mathbb{N}, \alpha \in \mathbb{R}_+ \).

Definition 3. (Caputo)

\[ C_0^\alpha D_t^\alpha f(t) = \frac{1}{\Gamma(m - \alpha)} \int_0^t \frac{f^{(m)}(\tau)}{(t - \tau)^{\alpha-m+1}} d\tau, \quad (10) \]

where \( m - 1 < \alpha < m \), \( m \in \mathbb{N} \).
The generalized operator

Fractional calculus is a generalization of integration and differentiation to non-integer order operator $aD_t^\alpha$, where $a$ and $t$ denote the limits of the operation and $\alpha$ denotes the fractional order such that

$$aD_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \Re(\alpha) > 0, \\ 1 & \Re(\alpha) = 0, \\ \int_a^t (d\tau)^{-\alpha} & \Re(\alpha) < 0, \end{cases}$$

where generally it is assumed that $\alpha \in \mathbb{R}$, but it may also be a complex number. We restrict our attention to the former case.
Properties of fractional-order differentiation

Fractional-order differentiation has the following properties:

1. If $\alpha = n$ and $n \in \mathbb{Z}_+$, then the operator $0D_t^\alpha$ can be understood as the usual operator $d^n/dt^n$.

2. Operator of order $\alpha = 0$ is the identity operator:
   
   $$0D_t^0 f(t) = f(t).$$

3. Fractional-order differentiation is linear; if $a, b$ are constants, then
   
   $$0D_t^\alpha [af(t) + bg(t)] = a 0D_t^\alpha f(t) + b 0D_t^\alpha g(t). \quad (12)$$

4. If $f(t)$ is an analytic function, then the fractional-order differentiation $0D_t^\alpha f(t)$ is also analytic with respect to $t$. 
5. For the fractional-order operators with $\Re(\alpha) > 0$, $\Re(\beta) > 0$, and under reasonable constraints on the function $f(t)$ it holds the additive law of exponents:

\[ 0D_t^\alpha [0D_t^\beta f(t)] = 0D_t^\beta [0D_t^\alpha f(t)] = 0D_t^{\alpha+\beta} f(t) \quad (13) \]

6. The fractional-order derivative commutes with integer-order derivative

\[ \frac{d^n}{dt^n} (aD_t^\alpha f(t)) = aD_t^\alpha \left( \frac{d^n f(t)}{dt^n} \right) = aD_t^{\alpha+n} f(t), \quad (14) \]

and if $t = a$ we have $f^{(k)}(a) = 0$, $(k = 0, 1, 2, \ldots, n - 1)$. 
On the meaning of the fractional-order derivative

We shall call $\mathcal{F}(f_t(\cdot), t)$ a hereditary operator acting on a cause process $f_t(\cdot)$ to produce a time-shifted effect $g(t)$ which depends on the history of the process $\{f_t(\tau); \tau < t\}$:

$$g(t) = \mathcal{F}[f_t(\cdot); t].$$  \hfill (15)

We can replace $g(t)$ by the function $f(t)$ or its derivatives, i.e.

$$\frac{df(t)}{dt} = \mathcal{F}[f_t(\cdot); t]$$  \hfill (16)

and so on. (Again we see repeated differentiation/integration.)

Some hereditary process examples from physics: Brownian motion; Viscoelasticity; Heat transfer; Long transmission.
Exercise: Integration

Compute a fractional-order derivative of order $1/2$ for the function $f(t) = t^2$ using the Caputo definition. Hint: $\Gamma(1/2) = \sqrt{\pi}$.

$$C_0^\mathcal{D}_t^{1/2}t^2 = \frac{1}{\Gamma(1-1/2)} \int_0^t \frac{(\tau^2)'}{(t-\tau)^{1/2-1+1}}d\tau = ?$$

Solution: Compute the indefinite integral

$$\int \frac{(\tau^2)'}{(t-\tau)^{1/2-1+1}}d\tau = \int \frac{2\tau}{\sqrt{t-\tau}}d\tau =$$

$$u = t-\tau \quad 2 \int \frac{u-t}{\sqrt{u}}du = 2 \int \sqrt{u}du - 2t \int \frac{1}{\sqrt{u}}du =$$

$$\frac{4}{3}u^{3/2} - 4t\sqrt{u} + C = \frac{4}{3}(t-\tau)^{3/2} - 4t\sqrt{t-\tau} + C.$$  

The answer is

$$\left.\frac{1}{\sqrt{\pi}} \cdot \left(\frac{4}{3}(t-\tau)^{3/2} - 4t\sqrt{t-\tau} + C\right)\right|_0^t = \frac{1}{\sqrt{\pi}} \cdot \left(-\frac{4}{3}t^{3/2} + 4t^{3/2}\right) = \frac{8t^{3/2}}{3\sqrt{\pi}}.$$
Part II: Factional-order Modeling of Dynamic Systems
A function $F(s)$ of the complex variable $s$ is called the Laplace transform of the original function $f(t)$ and defined as

$$F(s) = \mathcal{L}[f(t)] = \int_0^\infty e^{-st}f(t)\,dt \tag{17}$$

The original function $f(t)$ can be recovered from the Laplace transform $F(s)$ by applying the inverse Laplace transform

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{j2\pi} \int_{c-j\infty}^{c+j\infty} e^{st}F(s)\,ds, \tag{18}$$

where $c$ is greater than the real part of all the poles of $F(s)$. 
Fractional-order derivative definitions: Laplace transform

Definition 4. (Riemann-Liouville)

\[
\mathcal{L} \left[ R \mathcal{D}^\alpha f(t) \right] = s^\alpha F(s) - \sum_{k=0}^{m-1} s^k \left[ \mathcal{D}^{\alpha-k-1} f(t) \right]_{t=0}.
\] (19)

Definition 5. (Caputo)

\[
\mathcal{L} \left[ C \mathcal{D}^\alpha f(t) \right] = s^\alpha F(s) - \sum_{k=0}^{m-1} s^{\alpha-k-1} f^{(k)}(0).
\] (20)

Definition 6. (Grünwald-Letnikov)

\[
\mathcal{L} \left[ L \mathcal{D}^\alpha f(t) \right] = s^\alpha F(s).
\] (21)

For the first two definitions we have \((m - 1 \leq \alpha < m)\).
A linear, fractional-order continuous-time dynamic system can be expressed by a fractional differential equation of the following form

\[ a_n \mathcal{D}^{\alpha_n} y(t) + a_{n-1} \mathcal{D}^{\alpha_{n-1}} y(t) + \cdots + a_0 \mathcal{D}^{\alpha_0} y(t) = b_m \mathcal{D}^{\beta_m} u(t) + b_{m-1} \mathcal{D}^{\beta_{m-1}} u(t) + \cdots + b_0 \mathcal{D}^{\beta_0} u(t), \]  

where \( a_k, b_k \in \mathbb{R} \). The system is said to be of *commensurate-order* if in (22) all the orders of derivation are integer multiples of a base order \( \gamma \) such that \( \alpha_k = k \gamma, \beta_k = k \gamma, \gamma \in \mathbb{R}_+ \). The system can then be expressed as

\[ \sum_{k=0}^{n} a_k \mathcal{D}^{k\gamma} y(t) = \sum_{k=0}^{m} b_k \mathcal{D}^{k\gamma} u(t). \]  

(23)
If in (23) the order is $\gamma = 1/q$, $q \in \mathbb{Z}_+$, the system will be of rational order. The diagram with linear time-invariant (LTI) system classification is given in the following diagram.
Fractional-order transfer functions

Applying the Laplace transform to (22) with zero initial conditions the input-output representation of the fractional-order system can be obtained in the form of a transfer function:

\[
G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}}. \tag{24}
\]

In the case of a system with commensurate order \( \gamma \) we have

\[
G(s) = \frac{\sum_{k=0}^{m} b_k (s^{\gamma})^k}{\sum_{k=0}^{n} a_k (s^{\gamma})^k}. \tag{25}
\]
Taking $\lambda = s^\gamma$ the function (25) can be viewed as a pseudo-rational function $H(\lambda)$:

$$H(\lambda) = \frac{\sum_{k=0}^{m} b_k \lambda^k}{\sum_{k=0}^{n} a_k \lambda^k}.$$  

(26)

Based on the concept of the pseudo-rational function, a state-space representation can be established in the form:

$$\mathcal{D}^\gamma x(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t).$$  

(27)
Example: From a FO transfer function to the FO state-space form

Suppose that we are given a fractional-order transfer function

\[ G(s) = \frac{s^{0.25} + 2.5}{3s^{1.75} + 2s^{0.5} + 1}. \]

We find, that the commensurate order for this system is \( \gamma = 0.25 \). Then we use \( H(s) = C(sI - A)^{-1}B + D \) and arrive at the following state-space matrices

\[
A = \begin{bmatrix}
0 & 0 & 0 & 0 & -0.66 & 0 & -0.33 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}, \quad
B = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix},
\]

\[
C = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0.33 & 0.83
\end{bmatrix}, \quad
D = 0.
\]
Example: Fractional system composition

Let us assume that a fractional system is given by a block diagram

Here

\[ G_1(s) = \frac{1}{s^{0.5} + 1}, \quad G_2(s) = \frac{s^{0.3} + 1}{s^{2.5} + s + 1}, \]
\[ G_3(s) = \frac{2}{s^{0.1} + 1}, \quad G_4(s) = \frac{1}{15s + 1}. \]

Compute the transfer function resulting from the interconnection above.
The fractional-order systems we consider are linear. Therefore, the usual operations for computing system interconnections hold. In this case the complete system is given by

$$G(s) = \frac{-30 s^{3.5} - 2 s^{2.5} - 30 s^2 + 15 s^{1.4} + 15 s^{1.3}}{15 s^{4.1} + 15 s^4 + 15 s^{3.6} + 15 s^{3.5} + s^{3.1} + s^3 + 16 s^{2.6} + 14 s^{2.5} + 15 s^{2.1} + 15 s^2 + 16 s^{1.6} + 16 s^{1.5} + 16 s^{1.1} + 14 s + s^{0.6} + s^{0.5} + s^{0.4} + s^{0.3} + 2 s^{0.1}}.$$  

It can be seen from this example that from relatively simple initial systems a fairly complicated fractional-order transfer function was obtained. In this case we find, that the commensurate order of the system is $\gamma = 0.1$. 

Theorem 1. (Matignon’s stability theorem) The fractional transfer function \( G(s) = Z(s)/P(s) \) is stable if and only if the following condition is satisfied in \( \sigma \)-plane:

\[
|\arg(\sigma)| > q \frac{\pi}{2}, \forall \sigma \in \mathbb{C}, \ P(\sigma) = 0,
\]  

(28)

where \( \sigma := s^q \). When \( \sigma = 0 \) is a single root of \( P(s) \), the system cannot be stable. For \( q = 1 \), this is the classical theorem of pole location in the complex plane: no pole is in the closed right plane of the first Riemann sheet.

Algorithm summary: Find the commensurate order \( q \) of \( P(s) \), find \( a_1, a_2, \ldots a_n \) in (25) and solve for \( \sigma \) the equation \( \sum_{k=0}^{n} a_k \sigma^k = 0 \). If all obtained roots satisfy the condition (28), the system is stable.
Stability regions

Stable region

Unstable region

\[ q \frac{\pi}{2} \]
Exercise: Stability

Determine the commensurate order $\gamma$ of the fractional-order system given below. Then, write out and solve the characteristic equation $P(\lambda) = 0$. Hint: $\lambda = s^\gamma$.

$$G(s) = \frac{s + 1}{s - 2s^{0.5} + 5}.$$

Solution: The commensurate order is $\gamma = 0.5$, so we have $\lambda = s^{0.5}$. Therefore, the characteristic equation is

$$P(\lambda) = \lambda^2 - 2\lambda + 5.$$

Solving $P(\lambda) = 0$ yields complex roots $\lambda_{1,2} = 1 \pm j2$. Notice, that in case of a classical integer-order system this result would immediately imply instability. However, in case of this system we have

$$|\arg(1 \pm j2)| \approx 1.1071 > 0.7854 \approx \frac{0.5\pi}{2},$$

hence the system under analysis is stable.
Example: Stability evaluation of a relatively complex system

The transfer function is

\[ G(s) = \frac{-2s^{0.63} + 4}{2s^{3.501} + 3.8s^{2.42} + 2.6s^{1.798} + 2.5s^{1.31} + 1.5} \]

and the commensurate order \( q = 0.01 \). It is found to be stable.
Time-domain analysis

Consider a revised Grünwald-Letnikov definition rewritten as

\[
a \mathcal{D}_t^\alpha f(t) = \lim_{h \to 0} \frac{1}{h^\alpha} \left[ \frac{t-a}{h} \right] \sum_{j=0}^{[\frac{t-a}{h}]} w_j^{(\alpha)} f(t - jh),
\]

(29)

where \( h \) is the computation step-size and \( w_j^{(\alpha)} = (-1)^j \binom{\alpha}{j} \) can be evaluated recursively from

\[
w_0^{(\alpha)} = 1, \quad w_j^{(\alpha)} = \left( 1 - \frac{\alpha + 1}{j} \right) w_{j-1}^{(\alpha)}, \quad j = 1, 2, \cdots.
\]

(30)

Further manipulations provide an algorithm for fixed-step numerical time-domain evaluation of fractional-order transfer functions. Please see [3] for details.
Frequency-domain analysis

Frequency-domain response may be obtained by substituting \( s = j \omega \) in (24). The complex response for a frequency \( \omega \in (0; \infty) \) can then be computed as follows:

\[
G(\omega) = \frac{b_m(j\omega)^{\beta_m} + b_{m-1}(j\omega)^{\beta_{m-1}} + \cdots + b_0(j\omega)^{\beta_0}}{a_n(j\omega)^{\alpha_n} + a_{n-1}(j\omega)^{\alpha_{n-1}} + \cdots + a_0(j\omega)^{\alpha_0}},
\]

(31)

where \( j \) is the imaginary unit.

It should be noted, that frequency-domain analysis is a very important tool where fractional-order modeling and control design are concerned.
Approximation of fractional operators

The Oustaloup recursive filter gives a very good approximation of fractional operators in a specified frequency range and is widely used in fractional calculus. For a frequency range $(\omega_b, \omega_h)$ and of order $N$ the filter for an operator $s^\gamma$, $0 < \gamma < 1$, is given by

$$s^\gamma \approx K \prod_{k=-N}^{N} \frac{s + \omega'_k}{s + \omega_k}, \quad K = \omega_h^\gamma, \quad \omega_r = \frac{\omega_h}{\omega_b},$$

(32)

$$\omega'_k = \omega_b(\omega_r)\frac{k+N+\frac{1}{2}(1-\gamma)}{2N+1}, \quad \omega_k = \omega_b(\omega_r)\frac{k+N+\frac{1}{2}(1+\gamma)}{2N+1}.$$

The resulting model order is $2N + 1$.

A modified Oustaloup filter has been proposed in literature [3].
A general method for approximating a fractional-order model by an integer-order one may be proposed. Recall the property in (14):

- The fractional-order derivative commutes with integer-order derivative

\[ \frac{d^n}{dt^n} \left( aD_t^\alpha f(t) \right) = aD_t^\alpha \left( \frac{d^n f(t)}{dt^n} \right) = aD_t^{\alpha+n} f(t). \]

Thus, for fractional orders \( \alpha \geq 1 \) it holds

\[ s^\alpha = s^n s^\gamma, \]

(33)

where \( n = \alpha - \gamma \) denotes the integer part of \( \alpha \) and \( s^\gamma \) is obtained by the Oustaloup approximation in (32).
The fractional-order transfer function is

\[ G(s) = \frac{1}{14994s^{1.31} + 6009.5s^{0.97} + 1.69}, \]

and approximation parameters \( \omega = [10^{-4}; 10^4] \), \( N = 5 \).
Continuous zeros and poles, obtained using the Oustaloup recursive filter, are directly mapped to their discrete-time counterparts by means of the relation

\[ z = e^{sT_s}, \tag{34} \]

where \( T_s \) is the desired sampling interval. The gain of the resulting discrete-time system \( H(z) \) must be corrected by a proper factor.

For the synthesis of continuous zeros and poles using the Oustaloup method with the intent to obtain a discrete-time approximation the transitional frequency \( \omega_h \) may be chosen such that

\[ \omega_h \leq \frac{2}{T_s}. \tag{35} \]
We now address the issue of implementing the fractional-order integrator component. A continuous-time integrator of order $\lambda$ has to be implemented as

$$G_I(s) = \frac{1}{s^\lambda} = \frac{s^{1-\lambda}}{s}$$

to ensure a nice control effect at lower frequencies. Its discrete-time equivalent is given by

$$H_I(z^{-1}) = H^{1-\lambda}(z^{-1}) \cdot H_I(z^{-1}), \quad (36)$$

where $H^{1-\lambda}(z)$ is computed using the method presented above, and

$$H_I(z^{-1}) = \frac{T_s}{(1 - z^{-1})} \quad (37)$$

is a simple discrete-time integrator.
Given the transfer function model in (24)

\[ G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}} \]

we search for a parameter set \( \theta = [ a_p \; \alpha_p \; b_z \; \beta_z ] \), such that

\[
\begin{align*}
    a_p &= [ a_n \; a_{n-1} \; \cdots \; a_0 ], \\
    \alpha_p &= [ \alpha_n \; \alpha_{n-1} \; \cdots \; \alpha_0 ], \\
    b_z &= [ b_m \; b_{m-1} \; \cdots \; b_0 ], \\
    \beta_z &= [ \beta_n \; \beta_{n-1} \; \cdots \; \beta_0 ],
\end{align*}
\]

by employing numerical optimization with an objective function given by an output error norm \( \| e(t) \|_2^2 \), where \( e(t) = y(t) - \tilde{y}(t) \) is obtained by taking the difference of the original model output \( y(t) \) and simulated model output \( \tilde{y}(t) \).
Consider the following generalizations of conventional process models used in industrial control design.

<table>
<thead>
<tr>
<th>Type</th>
<th>Transfer Function</th>
<th>Delay Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>(FO)FOPDT</td>
<td>$G(s) = \frac{K}{1+Ts}e^{-Ls}$</td>
<td>$G(s) = \frac{K}{1+Ts^{\alpha}}e^{-Ls}$</td>
</tr>
<tr>
<td>(FO)IPDT</td>
<td>$G(s) = \frac{K}{s}e^{-Ls}$</td>
<td>$G(s) = \frac{K}{s^{\alpha}}e^{-Ls}$</td>
</tr>
<tr>
<td>(FO)FOIPDT</td>
<td>$G(s) = \frac{K}{s(1+Ts)}e^{-Ls}$</td>
<td>$G(s) = \frac{K}{s(1+Ts^{\alpha})}e^{-Ls}$</td>
</tr>
</tbody>
</table>

Therefore, due to additional parameters $K$ (gain) and $L$ (delay) we may update the identified parameter set discussed previously to

$$\theta = [K \quad L \quad a_p \quad \alpha_p \quad b_z \quad \beta_z \quad].$$
Denote by $y_r$ the experimental plant output, and by $y_m$ the identified model output. We consider the SISO case, so both $y_r$ and $y_m$ should be vectors of size $N \times 1$. In the following, we address the problem of statistical analysis of modeling residuals. Residuals are given by a vector containing the model output error

$$\varepsilon = y_r - y_m.$$  \hspace{1cm} (38)

The percentage fit may be expressed as

$$Fit = \left( 1 - \frac{\|\varepsilon\|}{\|y_r - \bar{y}_r\|} \right) \cdot 100\%,$$  \hspace{1cm} (39)

where $\|\cdot\|$ is the Euclidean norm, and $\bar{y}_r$ is the mean value of $y_r$. 
• Maximum absolute error

\[
\varepsilon_{max} = \max_k |\varepsilon(k)|, \quad (40)
\]

shows the maximum deviation from the expected behavior of the model over the examined time interval; however, it may be misleading in case of disturbances or strong noise;

• The mean squared error

\[
\varepsilon_{MSE} = \frac{1}{N} \sum_{k=1}^{N} \varepsilon_k^2 = \frac{\|\varepsilon\|_2^2}{N}, \quad (41)
\]

may serve as a general measure of model quality. The lower it is, the more likely the model represents an adequate description of the studied process.
Residual Analysis: Autocorrelation of Residuals

Additional useful information is given by an estimate for autocorrelation of residuals for lag \( \tau = 1, 2, \ldots, \tau_{\text{max}} < N \), which may be computed by means of

\[
R_\varepsilon(\tau) = \frac{1}{(N - \tau)} \sum_{k=1}^{N-\tau} \varepsilon(k)\varepsilon(k + \tau).
\]  

(42)

The vector \( r^\varepsilon = \left[ R_\varepsilon(1) \quad R_\varepsilon(2) \quad \cdots \quad R_\varepsilon(\tau_{\text{max}}) \right] \) is constructed and is normalized such that \( r^{\varepsilon,\text{norm}} = r^\varepsilon / R_\varepsilon(1) \). Assuming normal distribution of residuals the confidence band \( \hat{\eta} \) is then approximated for a confidence percentage \( p_{\text{conf}} \in (0, 1] \) around zero mean as an interval

\[
\hat{\eta} = \left[ \left( 0 - \Phi^{-1}(c_p) \right) / \sqrt{N}, \left( 0 + \Phi^{-1}(c_p) \right) / \sqrt{N} \right],
\]  

(43)

where \( c_p = 1 - 0.5(1 - p_{\text{conf}}) \) and \( \Phi^{-1}(x) = \sqrt{2} \text{erf}^{-1}(2x - 1) \) is the quantile function. If the residual samples represent uncorrelated white noise, then ideally:

\[
r^{\varepsilon,\text{norm}}_i \in \hat{\eta} \quad \forall i = 1, 2, \ldots, \tau_{\text{max}}.
\]  

(44)
Identification data is collected from a system

\[ \Psi = \Psi_G + \mathcal{N}, \quad (45) \]

where \( \Psi_G \) is given by a continuous-time fractional-order transfer function of the form

\[ \Psi_G(s) = \frac{1.5}{0.11s^{1.93} + 0.79s^{0.31} + 1}, \quad (46) \]

and the noise term has an amplitude of \( \mathcal{N} = \pm 0.05 \). A pseudo-random binary sequence is used as the excitation signal for obtaining the transient response with a sample time of 0.01s.

In this example, the initial model structure is chosen such that its pseudo-order is \( n = 2 \) and commensurate order \( \gamma = 1 \).
Time domain identification: Excitation signal
Time domain identification: Trust-Region-Reflective identification

![Graph showing output error and autocorrelation of residuals.]

Mean squared error: 0.051619; Max abs error: 0.87922

Autocorrelation of residuals (with P=0.95 confidence)
Time domain identification: Levenberg-Marquardt algorithm

Mean squared error: 0.0099879; Max abs error: 0.40139

Autocorrelation of residuals (with P=0.95 confidence)
Part III: Factional-order PID Controllers
The control law of the PI$^\lambda$D$^\mu$ controller can be expressed as follows:

\[ u(t) = K_p e(t) + K_i \mathcal{D}^{-\lambda} e(t) + K_d \mathcal{D}^{\mu} e(t), \]  

(47)

where \( e(t) = y_{sp}(t) - y(t) \) is the error signal. After applying the Laplace transform to (47) assuming zero initial conditions, the following equation is obtained:

\[ C(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^{\mu} \]  

(48)

Obviously, when taking \( \lambda = \mu = 1 \) the result is the classical integer-order PID controller.
Fractional-order Control: $\text{PI}^\lambda \text{D}^\mu$ control loop

\[ + \_ \quad \text{PI}^\lambda \text{D}^\mu \quad \text{Plant} \]
Basics of fractional control: Fractional control actions

Let a basic fractional control action be defined as $C(s) = K \cdot s^\gamma$. The control actions in the time domain for $\gamma \in [-1, 1]$ with $K = 1$ under different input signals are given below.
PID controller vs. $PI^{0.5}D^{0.5}$ controller: frequency-domain characteristics
We would like to establish tuning methods for the FOPID controller similar to conventional ones (e.g. Ziegler-Nichols tuning formulae). Several methods have been proposed in literature so far. Consider the F-MIGO method suitable for tuning PI$^\lambda$ controllers [3]. Suppose we are given a FOPDT process model

$$G(s) = \frac{K}{Ts + 1} e^{-Ls}, \quad \tau = \frac{L}{L + T},$$

(49)

where $\tau$ is the relative dead-time of the system. Then

$$\lambda = \begin{cases} 
1.1, & \text{if } \tau \geq 0.6 \\
1.0, & \text{if } 0.4 \leq \tau < 0.6 \\
0.9, & \text{if } 0.1 \leq \tau < 0.4 \\
0.7, & \text{if } \tau < 0.1.
\end{cases}$$

and

$$K_p = \frac{1}{K} \left( \frac{0.2978}{\tau + 0.000307} \right), \quad K_i = \frac{K_p (\tau^2 - 3.402\tau + 2.405)}{0.8578T}. $$
Optimization provides general means of tuning a fractional-order PID controller given a cost function and suitable optimization constraints. There are several aspects to the problem of designing a proper controller using constrained optimization:

- The type of plant to be controlled (integer or noninteger order, nonlinear);
- Optimization criterion (cost function);
- Fractional controller design specifications;
- Specific parameters to optimize in the set \( \{ K_p, K_i, K_d, \lambda, \mu \} \);
- Selection of initial controller parameters.
Optimization based PI$^\lambda$D$^\mu$ tuning: Cost function

In case of a linear model we use time-domain simulation of a typical negative unity feedback loop

$$G_{cs}(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}.$$  \hspace{1cm} (50)

For the cost function we consider performance indicies:

- integral square error $ISE = \int_0^\tau e^2(t)\,dt$,
- integral absolute error $IAE = \int_0^\tau |e(t)|\,dt$,
- integral time-square error $ITSE = \int_0^\tau te^2(t)\,dt$,
- integral time-absolute error $ITAE = \int_0^\tau t|e(t)|\,dt$. 
Optimization based $PI^\lambda D^\mu$ tuning: Constraints

The design specifications include:

- Gain margin $G_m$ and phase margin $\varphi_m$ specifications;
- Complementary sensitivity function $T(j\omega)$ constraint, providing $A$ dB of noise attenuation for frequencies $\omega > \omega_t$ rad/s;
- Sensitivity function $S(j\omega)$ constraint for output disturbance rejection, providing a sensitivity function of $B$ dB for frequencies $\omega < \omega_s$ rad/s;
- Robustness to plant gain variations: a flat phase of the system is desired within a region of the system critical frequency $\omega_{cg}$;
- For practical reasons, a constraint on the control effort $u(t)$ may also be set.
Gain and phase margin specifications

(See http://a-lab.ee/edu/ajs/freq/ for details.)
Consider the original integer-order PID controller of the form

\[ C_{PID}(s) = K_P + K_I s^{-1} + K_D s. \] (51)

Let \( C_R(s) \) be a controller of the form

\[ C_R(s) = \frac{K_2 s^\beta + K_1 s^\alpha - K_D s^2 + (K_0 - K_P) s - K_I}{K_D s^2 + K_P s + K_I}, \] (52)

where the orders \( \alpha \) and \( \beta \) are such, that \(-1 < \alpha < 1\) and \(1 < \beta < 2\). The PI\(^\lambda\)D\(^\mu\) controller resulting from a classical PID controller will have the following coefficients

\[ K_P^* = K_0, \quad K_I^* = K_1, \quad K_D^* = K_2, \] (53)

and the orders will be

\[ \lambda = 1 - \alpha, \quad \mu = \beta - 1. \] (54)
It can be shown, that this structure may be replaced by a negative unity feedback where the controller is

\[
C'(s) = (C_R(s) + 1) \cdot C_{PID}(s).
\]  

(55)
After acquiring a set of discrete-time zeros and poles by means of (34), the fractional-order controller may be implemented in form of a IIR filter represented by a discrete-time transfer function $H(z^{-1})$. In general, one has two choices:

1. Implement each fractional-order component approximation of the controller in (48) separately as $H^\lambda(z^{-1})$ and $H^\mu(z^{-1})$; this method offers greater flexibility, since the components may be reused in the digital signal processing chain, but requires more memory and is generally more computationally expensive;

2. Compute a single LTI object approximating the whole controller; this method is suitable when there is a need for a static description of a fractional-order controller.
In this particular work we choose the second option, that is we seek a description of the controller in the form

$$H(z^{-1}) = K \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}.$$  \hspace{1cm} (56)

For practical reasons, the equivalent IIR filter should be comprised of second-order sections. This allows to improve computational stability when the target signal digital processing hardware has limited DSP capabilities. Thus, the discrete-time controller must be transformed to yield

$$H(z^{-1}) = K_c \prod_{k=1}^{N} \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}.$$  \hspace{1cm} (57)
Biquad IIR filter: Transposed form II
FOPID Controller Hardware Prototype
Part IV: CACSD Tools: FOMCON
FOMCON project: Fractional-order Modeling and Control

- Official website: http://fomcon.net/
- Toolbox for MATLAB available, development via GitHub: https://github.com/AlekseiTepljakov/fomcon-matlab
- Recently: Added initial support for studying FO MIMO systems.
FOMCON toolbox: Structure

Fractional-order systems analysis (time domain, frequency domain)

Identification
- Time domain
- Frequency domain

Control design
- Integer-order
- Fractional-order

Implementation
- Continuous approximations
- Discrete approximations
- Analog filters
- Digital filters
FOMCON toolbox: FOTF Viewer

- FOTF systems in MATLAB workspace
- Stability test
- Add, Edit or Delete FOTF systems
- Time-domain analysis
- Export FOTF systems to other formats
- Frequency-domain analysis

Control Lab
www.a-lab.ee
Aleksei Tepljakov
FOMCON toolbox: Time-domain identification

- Time-domain simulation parameters
- Optimization algorithm selection
- Identified model; Structure selection
- Other available identification options
- Experimental data
- Initial guess model generation
- Identified model export to workspace
FOMCON toolbox: Optimization based PI$^\lambda$D$^\mu$ tuning

- **Linear plant model**
- **Fractional PID parameters**
- **Simulation parameters**
- **Optimization options**
- **Frequency-domain specifications**
- **Control signal constraints**
Part V: Applications of Fractional-order Control
The system is modeled in continuous time in the following way:

\[ \dot{x}_1 = \frac{1}{A} u_1 - d_{12} - w_1 c_1 \sqrt{x_1}, \quad (58) \]

\[ \dot{x}_2 = \frac{1}{A} u_2 + d_{12} - w_2 c_2 \sqrt{x_2}, \]

where \( x_1 \) and \( x_2 \) are levels of fluid, \( A \) is the cross section of both tanks; \( c_1, c_2, \) and \( c_{12} \) are flow coefficients, \( u_1 \) and \( u_2 \) are pump powers; valves are denoted by \( w_i : w_i \in \{0, 1\} \) and

\[ d_{12} = w_{12} \cdot c_{12} \cdot \text{sign}(x_1 - x_2) \sqrt{|x_1 - x_2|}. \]
Our task is to control the level in the first tank. We identify the real plant from a step experiment with \( w_1 = w_{12} = 1, w_2 = 0 \) in (58). The resulting fractional-order model is described by a transfer function

\[
G_2 = \frac{2.442}{18.0674s^{0.9455} + 1}e^{-0.1s}.
\] 

(59)

We notice, that this model does not tend to exhibit integer-order dynamics. Due to the value of the delay term the basic tuning formulae for integer-order PID tuning do not provide feasible results. It is possible to select some starting point manually and run optimization several times. However, it is important to choose the correct frequency domain specifications to ensure control system stability.
Case study (1): Experiments with controller implementation: Hardware platform

Aleksei Tepljakov

Control Lab
www.a-lab.ee
In our case the goal is to minimize the impact of disturbance, so constraints on the sensitivity functions could be imposed. Our choice is such that \(|T(j\omega)| \leq -20 \, dB, \forall \omega \geq 10 \, \text{rad/s} \) and \(|S(j\omega)| \leq -20 \, dB, \forall \omega \leq 0.1 \, \text{rad/s} \). The gain and phase margins are set to \(G_m = 10 \, \text{dB} \) and \(\varphi = 60^\circ \), respectively. Additionally, in order to limit the overshoot, the upper bound of control signal saturation was lowered from 100% to 60%. Using the IAE performance metric we finally arrive at the following PI\(^\lambda\)D\(^\mu\) controller parameters by optimizing the response of the nonlinear system in Simulink:

\[
K_p = 6.9514, \quad K_i = 0.13522, \quad K_d = -0.99874, \\
\lambda = 0.93187, \quad \mu = 0.29915. \quad (60)
\]
Case study (1): Fractional-order control of the coupled tank system (continued)

Real plant, controller in Simulink
- Discrete–time simulation
- Real controller, model in Simulink

Real plant

Level [m]

Control law u(t) [%]

w2
This system can be described by the following differential equations:

\[
\dot{x}_1 = \frac{1}{\eta_1(x_1)} \left( u_p(v) - C_1 x_1^{\alpha_1} - \zeta_1(v_1) x_1^{\alpha_1 v_1} \right),
\]

\[
\dot{x}_2 = \frac{1}{\eta_2(x_2)} \left( q + r - C_2 x_2^{\alpha_2} - \zeta_2(v_2) x_2^{\alpha_2 v_2} \right),
\]

where \( x_1 \) and \( x_2 \) are levels in the upper tank and middle tank, respectively, \( \eta_1(x_1) = A = aw \) and \( \eta_2(x_2) = cw + x_2bw/x_{2max} \) are cross-sectional areas of the upper and middle tank, respectively, \( u_p(v) \) is the pump capacity, such that depends on the normalized input \( v(t) \in [0,1] \); \( \zeta_1(v_1) \) and \( \zeta_2(v_2) \) are variable flow coefficients of the automatic valves controlled by normalized inputs \( v_1(t), v_2(t) \in [0,1] \), \( q = C_1 x_1^{\alpha_1} \) and \( r = \zeta_1(v_1) x_1^{\alpha_1 v_1} \).
Case study (2): Statement of the control problem

- The task is to design a controller for the upper tank such that would keep the level of fluid within reasonable bounds at the desired set point in the presence of disturbances caused by the controlled output valve.

- It is required to design a controller for the middle tank, such that would keep the level of fluid at the desired set point using controlled valves of the upper tank and also its own valve.

- The tanks are, in fact, coupled, so only a limited range of fluid level values is achievable in the middle tank and it is related to the level in the upper tank.

- The outflow of liquid from the upper tank through the automatic valve forms part of the control for the middle tank and is considered a disturbance from the perspective of level control in the upper tank.
Case study (2): The real-life Multi-Tank system
First, linear approximations are obtained from the nonlinear model by means of time-domain identification at system working points 
\((0.7029, 0.1)\) and \((0.7879, 0.2)\). The following models are found:

\[
G_1(s) = \frac{0.14464}{18.728s^{0.91746} + 1}
\]

and

\[
G_2(s) = \frac{0.25881}{27.859s^{0.9115} + 1}.
\]

Next, controllers are designed for level control in the upper tank using the FOPID optimization tool of FOMCON toolbox. For this a nonlinear model of the system is used for simulations in the time domain, the set value corresponds to the particular operating point. Linear approximations, corresponding to the working points, are used to constrain the optimization by means of frequency-domain specifications.
Case study (2): Tuning the FOPID controller for the upper tank

We use a two-point GOS scheme, therefore we have two controllers. The specifications are as follows:

- In case of the first controller, a phase margin is set to \( \varphi_m \geq 60^\circ \), sensitivity and complementary sensitivity function constraints are set such that \( \omega_t = 0.02 \) and \( \omega_s = 0.1 \) with \( A = B = -20 \text{ dB} \). Robustness to gain variations specification is also used with the critical frequency \( \omega_c = 0.1 \).

- For the second controller, the phase margin specification is changed to \( \varphi_m = 85^\circ \) and the bandwidth limitation specified by \( \omega_c \) is removed.

Due to the flexibility of the tuning tool, it is possible to retune the controllers by considering the composite control law during the controller optimization process.
As a result, two FOPID controllers are obtained:

\[ C_1(s) = 6.1467 + \frac{1.0712}{s^{0.9528}} + 0.8497s^{0.8936} \]

and

\[ C_2(s) = 5.1524 + \frac{0.3227}{s^{1.0554}} + 2.4827s^{0.010722} \].

The composite control law

\[ C(s) = \frac{(1 - \gamma(x_1))C_1(s) + \gamma(x_1)C_2(s)}{2} \]

is then verified with both models \( G_1(s) \) and \( G_2(s) \) using the stability test with step size of \( \Delta\gamma = 0.01 \) and minimum commensurate order \( q_{min} = 0.01 \). The result of the test is that the closed-loop systems are stable in case of both fractional models.
Once the gain and order scheduled composite controller is designed, it is plugged into the simulated control system, and a FOPID controller is designed for the second tank using the same optimization tool. In addition, we consider the following:

- Frequency-domain specifications are not applicable, since we do not have a linear model of this process.
- The application of the $D^\mu$ component is not very desirable in this case due to higher levels of noise.

Therefore we design a FOPI controller based only on optimization of the transient response of the control system in the time domain. The following controller is obtained:

$$C_3(s) = 5.0000 + \frac{0.06081}{s^{0.1029}}$$

which is essentially a proportional controller with a weak fractional-order integrator.
Case study (2): Control system performance
Case study (2): GOS FOPID control of level in the first tank via visual feedback
Case study (2): GOS FOPID control of level in the first tank via visual feedback: Results

![Graph showing water level and control law over time.](image)

- Reference
- Water level (visual detection)
- Water level (sensor data)

Control law $u(t)$ vs. Time [s]
We use the following model of the MLS:

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= -\frac{c(x_1)}{m} x_3^2 + \frac{x_2}{x_1} + g, \\
\dot{x}_3 &= \frac{f_{ip2} \cdot i(u) - x_3}{f_{ip1} \cdot e^{-x_1/f_{ip2}}},
\end{align*}
\]  

(61)

where \(x_1\) is the position of the sphere, \(x_2\) is the velocity of the sphere, and \(x_3\) is the coil current, \(f_{ip1}\) and \(f_{ip2}\) are constants, \(c(x_1)\) is a 4th order polynomial and \(i(u)\) is a 2nd order polynomial.
A real-life MLS is used in this experiment. The MATLAB/Simulink environment acts as an interface between the two devices.
The following FOPID controller was implemented in the retuning configuration:

\[ C_1^*(s) = -45.839 - 18.504s^{-1.06} - 3.0559s^{0.94}, \]

The parameters of the retuning controller in (52) were computed, and an implementation of the form (57) was obtained using the Oustaloup method with \( N = 4 \) and \( \omega = [0.001, 2/T_s] \), where \( T_s = 0.001s \) is the desired sample rate.
Case study (3): Experimental Results

Video: https://youtu.be/NXbqjK6oIcw
Recall the example, where our goal was to obtain an analog implementation a fractional controller for a model of a position servo

\[ G(s) = \frac{1.4}{s(0.7s + 1)}e^{-0.05s}. \]

We now provide the results of approximating the controller

\[ C(s) = \left( \frac{2.0161s + 1}{0.0015s + 1} \right)^{0.7020} \]

by an electrical network by using a deterministic method, implemented as part of the unified network generation framework in FOMCON, for obtaining the parameters of the network.
In order to implement it, the following steps are carried out:

- We choose $R_1 = 200 \, k\Omega$ and $C_1 = 1 \, \mu F$ due to the time constant $\tau$.

- The basic structure is the Foster II form $RC$ network and the implementation is done by means of the mentioned algorithm.

- To obtain the differentiator, we use the property $Z_d(s) = 1/Z_i(s)$, where $Z_d(s)$ and $Z_i(s)$ correspond to impedances of a differentiator and an integrator, respectively.

- This is done by setting the impedances in the active filter circuit such that $Z_1(s) = Z_i(s)$ and $Z_2(s) = R_k$, where $R_k$ serves as a gain correction resistor.
\[ b = 2.0161; \quad wz = 1/b; \]
\[ \alpha = 0.702; \]
\[ Gc = \text{fotf}('s')^\alpha / wz^\alpha; \]

\[
\text{params} = \text{struct}; \quad \text{params.R1} = 200e3; \\
\text{params.C1} = 1e-6; \quad \text{params.N} = 4; \\
\text{params.}\varphi = 0.01; \\
\]

\[
\text{imp2} = \text{frac_rcl}(1/Gc, ... \\
\quad '\text{frac_struct_rc_foster2}', ... \\
\quad '\text{frac_imp_rc_foster2_abgen}', ... \\
\quad \text{logspace}(-2,3,1000), ... \\
\quad \text{params}); 
\]
The controller is obtained from the object using
\[ C = \frac{1}{\text{zpk}(\text{imp2});} \]
Now we set the resistor values to the preferred series with 5\% tolerance, and the capacitor values are substituted for closest components out of the 10\%-series:
\[ \text{imp2} = \text{imp2}.\text{prefnum}('5\%','10\%',[],'5\%'); \]
Finally, the bill of materials can be generated using \text{engnum}():
\[ [\text{vals}, \text{str}] = \text{engnum}(	ext{imp2}.\text{R}); \]
The variable \text{str} will contain the following:
\[ '360 \text{ k}' '200 \text{ k}' '75 \text{ k}' '27 \text{ k}' '9.1 \text{ k}' \]
The gain setting resistor \( R_k \) has the preferred value of 390\( k\Omega \).
Case study (4): Electrical network approximations (continued)

```
.ac list {w/(2*pi)}
.step dec param w 0.001 1000 150
```
Case study (4): Electrical network approximations (continued)
Case study (4): Electrical network approximations (continued)

Aleksei Tepljakov 96 / 101

Bode Diagram

System: Compensated
Gain Margin (dB): 24.9
At frequency (rad/s): 23.3
Closed loop stable? Yes

System: Uncompensated
Gain Margin (dB): 23.2
At frequency (rad/s): 5.28
Closed loop stable? Yes

System: Compensated
Phase Margin (deg): 85.5
Delay Margin (sec): 0.71
At frequency (rad/s): 2.1
Closed loop stable? Yes

System: Uncompensated
Phase Margin (deg): 49.1
Delay Margin (sec): 0.774
At frequency (rad/s): 1.11
Closed loop stable? Yes
Case study (4): Electrical network approximations (continued)
Case study (4): Frequency response around $\omega_{cg} = 2.2 \text{ rad/s}$
Case study (4): Frequency response around $\omega_{cg} = 2.2 \, \text{rad/s}$

Bode Diagram

- Fractional lead compensator
- Electrical network approximation (real)
Case study (4): Electrical network approximations: Results

![Graph showing the response with simulated model and real controller, compared to simulated response and set point.](image)

**Graph Description:**
- **Response with simulated model and real controller**
- **Simulated response**
- **Set point**

**Control Law u(t):**
- **Analog controller**
- **Simulated controller**
References


