Fractional Order Control - A Tutorial

YangQuan Chen, Ivo Petráš and Dingyü Xue

http://fractionalcalculus.googlepages.com

Abstract— Many real dynamic systems are better characterized using a non-integer order dynamic model based on fractional calculus or, differentiation or integration of non-integer order. Traditional calculus is based on integer order differentiation and integration. The concept of fractional calculus has tremendous potential to change the way we see, model, and control the nature around us. Denying fractional derivatives is like saying that zero, fractional, or irrational numbers do not exist. In this paper, we offer a tutorial on fractional calculus in controls. Basic definitions of fractional calculus, fractional order dynamic systems and controls are presented first. Then, fractional order PID controllers are introduced which may make fractional order controllers ubiquitous in industry. Additionally, several typical known fractional order controllers are introduced and commented. Numerical methods for simulating fractional order systems are given in detail so that a beginner can get started quickly. Discretization techniques for fractional order operators are introduced in some details too. Both digital and analog realization methods of fractional order operators are introduced. Finally, remarks on future research efforts in fractional order control are given.

I. INTRODUCTION

Fractional calculus is a more than 300 years old topic. The number of applications where fractional calculus has been used rapidly grows. These mathematical phenomena allow to describe a real object more accurately than the classical "integer-order" methods. The real objects are generally fractional [61], [77], [56], [99], however, for many of them the fractionality is very low. A typical example of a non-integer (fractional) order system is the voltage-current relation of a semi-infinite lossy transmission line [98] or diffusion of the heat through a semi-infinite solid, where heat flow is equal to the half-derivative of the temperature [76].

The main reason for using the integer-order models was the absence of solution methods for fractional differential equations. At present time there are lots of methods for approximation of fractional derivative and integral and fractional calculus can be easily used in wide areas of applications (e.g.: control theory - new fractional controllers and system models, electrical circuits theory - fractances, capacitor theory, etc.).

As pointed in [13], clearly, for closed-loop control systems, there are four situations. They are 1) IO (integer order) plant with IO controller; 2) IO plant with FO (fractional order) controller; 3) FO plant with IO controller and 4) FO plant with FO controller. From control engineering point of view, doing something better is the major concern. Existing evidences have confirmed that the best fractional order controller can outperform the best integer order controller. It has also been answered in the literature why to consider fractional order control even when integer (high) order control works comparatively well [49], [52]. Fractional order PID controller tuning has reached to a matured state of practical use. Since (integer order) PID control dominates the industry, we believe FO-PID will gain increasing impact and wide acceptance. Furthermore, we also believe that based on some real world examples, fractional order control is ubiquitous when the dynamic system is of distributed parameter nature [13].

A comprehensive review of fractional order control and its applications can be found in the coming monograph [51]. For computational methods in fractional calculus, we refer to the book [101]. Note that, the textbook [102] is the first control textbook containing a dedicated chapter on fractional order control.

In this paper, we offer a tutorial on fractional calculus in controls. Basic definitions of fractional calculus, fractional order dynamic systems and controls are presented first in Sec. II. Then, fractional order PID controllers are introduced in Sec. III which may make fractional order controllers ubiquitous in industry. Additionally, several typical known fractional order controllers are introduced and commented in Sec. IV. Numerical methods for simulating fractional order systems are given in detail in Sec. V so that a beginner can get started quickly. Discretization techniques for fractional order operators are introduced in some details in Sec. VI and implementation in Sec. VII . Finally, in Sec. VIII remarks on future research effort in fractional order control are given.

II. FRACTIONAL CALCULUS AND FRACTIONAL ORDER DYNAMIC SYSTEMS

The idea of fractional calculus has been known since the development of the regular (integer-order) calculus, with the first reference probably being associated with Leibniz

YangQuan Chen is with the Center for Self-Organizing and Intelligent Systems (CSOIS), Department of Electrical and Computer Engineering, Utah State University, 4120 Old Main Hill, Logan, UT 84322-4120, USA. Corresponding author and Tutorial Session Organizer. Email: yqchen@ieee.org, Tel./Fax: 1(435)797-0148/3054; Web: http://www.csois.usu.edu or http://yangquan.chen.googlepages.com

Ivo Petráš is with the Institute of Control and Informatization of Production Processes, BERG Faculty, Technical University of Košice, B. Němcovej 3, 042 00 Košice, Slovak Republic, ivo.petras@tuke.sk, Tel./Fax: +421-55-602-5194.

Dingyü Xue is with the Faculty of Information Sciences and Engineering, Northeastern University, Shenyang 110004, P R China, xuedingyu@mail.neu.edu.cn, Tel. +86-24-83689762, Fax. +86-24-23899982.

and L'Hôpital in 1695 where half-order derivative was mentioned.

Fractional calculus is a generalization of integration and differentiation to non-integer order fundamental operator $_aD_t^r$, where a and t are the limits of the operation. The continuous integro-differential operator is defined as

$$_aD_t^r = \begin{cases} \mathrm{d}^r/\mathrm{d}t^r & \Re(r) > 0, \\ 1 & \Re(r) = 0, \\ \int_a^t (\mathrm{d}\tau)^{-r} & \Re(r) < 0, \end{cases}$$

where r is the order of the operation, generally $r \in R$ but r could also be a complex number [62].

A. Definition of Fractional Differintegral

Two definitions used for the general fractional differintegral are the Grunwald-Letnikov (GL) definition and the Riemann-Liouville (RL) definition [58], [76]. The GL is given here

$$_aD_t^r f(t) = \lim_{h \to 0} h^{-r} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^j \binom{r}{j} f(t - jh), \quad (1)$$

where [.] means the integer part. The RL definition is given as

$$_aD_t^r f(t) = \frac{1}{\Gamma(n-r)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{r-n+1}} \mathrm{d}\tau, \quad (2)$$

for $(n - 1 < r < n)$ and $\Gamma(.)$ is the *Gamma* function.

The Laplace transform method is routinely used for solving engineering problems. The formula for the Laplace transform of the RL fractional derivative (2) has the following form [76]:

$$\int_0^\infty e^{-st} {_0D_t^r} f(t) \, \mathrm{d}t = s^r F(s) - \sum_{k=0}^{n-1} s^k \, {_0D_t^{r-k-1}} f(t) \Big|_{t=0}, \quad (3)$$

for $(n - 1 < r \leq n)$, where $s \equiv j\omega$ denotes the Laplace transform variable.

A geometric and physical interpretation of fractional integration and fractional differentiation can be found in Podlubny's work [79].

B. Properties of Fractional Calculus

The main properties of fractional derivatives and integrals are the following:

1) If $f(t)$ is an analytical function of t, its fractional derivative $_0D_t^\alpha f(t)$ is an analytical function of z and α.

2) For $\alpha = n$, where n is an integer, the operation $_0D_t^\alpha f(t)$ gives the same result as classical differentiation of integer order n.

3) For $\alpha = 0$ the operation $_0D_t^\alpha f(t)$ is the identity operator:

$$_0D_t^0 f(t) = f(t).$$

4) Fractional differentiation and fractional integration are linear operations:

$$_0D_t^\alpha af(t) + bg(t) = a {_0D_t^\alpha} f(t) + b {_0D_t^\alpha} g(t).$$

5) The additive index law (semigroup property)

$$_0D_t^\alpha {_0D_t^\beta} f(t) = {_0D_t^\beta} {_0D_t^\alpha} f(t) = {_0D_t^{\alpha+\beta}} f(t)$$

holds under some reasonable constraints on the function $f(t)$.

The fractional-order derivative commutes with integer-order derivative

$$\frac{\mathrm{d}^n}{\mathrm{d}t^n}({_aD_t^r} f(t)) = {_aD_t^r} \left(\frac{\mathrm{d}^n f(t)}{\mathrm{d}t^n} \right) = {_aD_t^{r+n}} f(t),$$

under the condition $t = a$ we have $f^{(k)}(a) = 0$, $(k = 0, 1, 2, \ldots, n-1)$. The relationship above says the operators $\frac{d^n}{dt^n}$ and $_aD_t^r$ commute. (see [76, Chapter 2] for other commute properties).

C. Fractional Order Dynamic Systems

A fractional-order dynamic system can be described by a fractional differential equation of the following form [74], [76], [91]:

$$a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots + a_0 D^{\alpha_0} y(t)$$
$$= b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots + b_0 D^{\beta_0} u(t), \quad (4)$$

where $D^\gamma \equiv {_0D_t^\gamma}$; a_k $(k = 0, \cdots n)$, b_k $(k = 0, \cdots m)$ are constants; and α_k $(k = 0, \cdots n)$, β_k $(k = 0, \cdots m)$ are arbitrary real numbers.

Without loss of generality we can assume that $\alpha_n > \alpha_{n-1} > \cdots > \alpha_0$, and $\beta_m > \beta_{m-1} > \cdots > \beta_0$.

For obtaining a discrete model of the fractional-order system (4), we have to use discrete approximations of the fractional-order integro-differential operators and then we obtain a general expression for the discrete transfer function of the controlled system [91]

$$G(z) = \frac{b_m \left(w\left(z^{-1}\right) \right)^{\beta_m} + \ldots + b_0 \left(w\left(z^{-1}\right) \right)^{\beta_0}}{a_n \left(w\left(z^{-1}\right) \right)^{\alpha_n} + \ldots + a_0 \left(w\left(z^{-1}\right) \right)^{\alpha_0}}, \quad (5)$$

where $(\omega(z^{-1}))$ denotes the discrete equivalent of the Laplace operator s, expressed as a function of the complex variable z or the shift operator z^{-1}.

The fractional-order linear time-invariant system can also be represented by the following state-space model

$$\begin{aligned} {_0D_t^q} x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \quad (6)$$

where $x \in R^n$, $u \in R^r$ and $y \in R^p$ are the state, input and output vectors of the system and $A \in R^{n \times n}$, $B \in R^{n \times r}$, $C \in R^{p \times n}$, q is the fractional commensurate order.

D. Stability of LTI Fractional Order Systems

It is known from the theory of stability that an LTI (linear, time-invariant) system is stable if the roots of the characteristic polynomial are negative or have negative real parts if they are complex conjugate. It means that they are located on the left half of the complex plane. In the fractional-order LTI case, the stability is different from the integer one. Interesting notion is that a stable fractional system may have roots in the right half of complex plane

(see Fig. 1). It has been shown that system (6) is stable if the following condition is satisfied [45]

$$|\arg(\mathrm{eig}(\boldsymbol{A}))| > q\frac{\pi}{2}, \qquad (7)$$

where $0 < q < 1$ and $\mathrm{eig}(\boldsymbol{A})$ represents the eigenvalues of matrix $\boldsymbol{A}$.

Matignon's stability theorem says [45]: *The fractional transfer function $G(s) = Z(s)/P(s)$ is stable if and only if the following condition is satisfied in σ-plane:*

$$|arg(\sigma)| > q\frac{\pi}{2}, \ \forall \sigma \in C, \ P(\sigma) = 0, \qquad (8)$$

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 half plane of the first Riemann sheet.
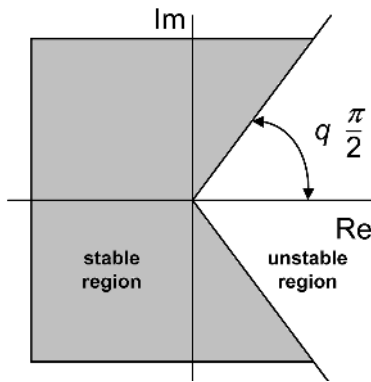

Fig. 1. Stability region of LTI fractional order systems with order $0 < q \le 1$

Generally, consider the following commensurate fractional order system in the form:

$$D^q \boldsymbol{w} = \boldsymbol{f}(\boldsymbol{w}), \qquad (9)$$

where $0 < q < 1$ and $\boldsymbol{w} \in R^n$. The equilibrium points of system (9) are calculated via solving the following equation

$$\boldsymbol{f}(\boldsymbol{w}) = 0. \qquad (10)$$

The equilibrium points are asymptotically stable if all the eigenvalues $\lambda_j, (j = 1, 2, \ldots, n)$ of the Jacobian matrix $\boldsymbol{J} = \partial \boldsymbol{f}/\partial \boldsymbol{w}$, evaluated at the equilibrium, satisfy the condition:

$$|\arg(\mathrm{eig}(\boldsymbol{J}))| = |\arg(\lambda_j)| > q\frac{\pi}{2}, \ j = 1, 2, \ldots, n. \qquad (11)$$

Figure 1 shows the stable and unstable regions of the complex plane for such case.

III. FRACTIONAL-ORDER PID CONTROLLERS

According to a survey [103] of the state of process control systems in 1989 conducted by the Japan Electric Measuring Instrument Manufacturer's Association, more than 90 percent of the control loops were of the PID type. It was also indicated in [6] that a typical paper mill in Canada has more than 2,000 control loops and that 97 percent use PI control. Therefore, the industrialist had concentrated on PI/PID controllers and had already developed *one-button type*

relay auto-tuning techniques for fast, reliable PI/PID control yet with satisfactory performance [36], [3], [104], [88], [18].

Intuitively, with noninteger order controllers for integer order plants, there are more flexibilities in adjusting the gain and phase characteristics than using IO controllers. These flexibilities make FO control a powerful tool in designing robust control system with less controller parameters to tune. The *key point* is that using few tuning knobs, FO controller achieves similar robustness achievable by using *very* high-order IO controllers.

$\mathrm{PI}^\lambda \mathrm{D}^\mu$ controller, also known as $\mathrm{PI}^\lambda \mathrm{D}^\delta$ controller, was studied in time domain in [77] and in frequency domain in [67]. In general form, the transfer function of $\mathrm{PI}^\lambda \mathrm{D}^\delta$ is given by

$$C(s) = \frac{U(s)}{E(s)} = K_p + T_i s^{-\lambda} + T_d s^\delta, \qquad (12)$$

where λ and δ are positive real numbers; K_p is the proportional gain, T_i the integration constant and T_d the differentiation constant. Clearly, taking $\lambda = 1$ and $\delta = 1$, we obtain a classical PID controller. If $\lambda = 0$ $(T_i = 0)$ we obtain a PD^δ controller, etc. All these types of controllers are particular cases of the $\mathrm{PI}^\lambda \mathrm{D}^\delta$ controller. The time domain formula is that

$$u(t) = K_p e(t) + T_i \mathrm{D}_t^{-\lambda} e(t) + T_d \mathrm{D}_t^\delta e(t). \ \ (\mathrm{D}_t^{(*)} \equiv {}_0 \mathrm{D}_t^{(*)}). \qquad (13)$$

It can be expected that $\mathrm{PI}^\lambda \mathrm{D}^\delta$ controller (13) may enhance the systems control performance due to more tuning knobs introduced. Actually, in theory, $\mathrm{PI}^\lambda \mathrm{D}^\delta$ itself is an infinite dimensional linear filter due to the fractional order in differentiator or integrator. For controller tuning techniques, refer to [50], [15].

Similar to the fact that, every year, numerous PI/PID papers have been published, we can foresee that, more and more FO PI/D papers will be published in the future. In general, the following issues should be addressed:

- How to tell there is a need to use FO PI/D controller while integer order PI/D control works well in the existing controlled systems?
- How to predict the net performance gain by using FO PI/D controller?
- How to best tune the FO PI/D controller by taking minimum experimental efforts?
- How to best design the experiments to tune FO PI/D controller?
- For a given class of plants to be controlled, how to best design FO PI/D controller?

In the interest of space, we conclude this section by referring to the two recent Ph.D. dissertations [105], [49] and the references therein.

Remark 3.1: We comment that since PID control is ubiquitous in industry process control, FO PID control will be also ubiquitous when tuning and implementation techniques are well developed.

IV. SOME TYPICAL FRACTIONAL-ORDER CONTROLLERS

As already widely known, the early attempts to apply fractional-order derivative to systems control can be found

in [42], [4], [63]. In this section, three other representative fractional-order controllers in the literature will be briefly introduced, namely, TID (tilted integral derivative) controller, CRONE controller and fractional lead-lag compensator [100], [102]. For detailed introduction and comparison, refer to [100]. For the latest developments, we refer to [7], [47], [60], [51], [52].

- *TID Controller.* In [38], a feedback control system compensator of the PID type is provided, wherein the proportional component of the compensator is replaced with a **t**ilted component having a transfer function $s^{-\frac{1}{n}}$. The resulting transfer function of the entire compensator more closely approximates an optimal loop transfer function, thereby achieving improved feedback control performance. Further, as compared to conventional PID compensators, the TID compensator allows for simpler tuning, better disturbance rejection, and smaller effects of plant parameter variations on the closed-loop response.

  The objective of TID is to provide an improved feedback loop compensator having the advantages of the conventional PID compensator, but providing a response which is closer to the theoretically optimal response. In the TID patent [38], an analog circuit using op-amps plus capacitors and resistors is introduced with a detailed component list which is useful in some cases where the computing power to implementing $T_3(s)$ digitally is not possible. An example is given in [38] to illustrate the benefits from using TID over applying conventional PID in both time and frequency domains.

- *CRONE Controller.* The CRONE control was proposed by Oustaloup in pursuing *fractal robustness* [65], [66]. CRONE is a French abbreviation for "*Contrôle Robuste d'Ordre Non Entier*" (which means non-integer order robust control). In this subsection, we shall follow the basic concept of *fractal robustness*, which motivated the CRONE control, and then mainly focus on the second generation CRONE control scheme and its synthesis based on the desired frequency template which leads to fractional transmittance.

  In [61], "fractal robustness" is used to describe the following two characteristics: the iso-damping and the vertical sliding form of frequency template in the Nichols chart. This desired robustness motivated the use of fractional-order controller in classical control systems to enhance their performance.

  With a unit negative feedback, for the characteristic equation

  $$1 + (\tau s)^{\alpha} = 0,$$

  the forward path transfer function, or the open-loop transmittance, is that

  $$\beta(s) = \left(\frac{1}{\tau s}\right)^{\alpha} = \left(\frac{\omega_u}{s}\right)^{\alpha}, \qquad (14)$$

  which is the transmittance of a *non integer integrator* in which $\omega_u = 1/\tau$ denotes the unit gain (or transitional) frequency.

  In controller design, the objective is to achieve such a similar frequency behavior, in a medium frequency range around ω_u, knowing that the closed-loop dynamic behavior is exclusively linked to the open-loop behavior around ω_u. Synthesizing such a template defines the non-integer order approach that the second generation CRONE control uses.

  There are a number of real life applications of CRONE controller such as the car suspension control [66], flexible transmission [65], hydraulic actuator [34] etc. CRONE control has been evolved to a powerful non-conventional control design tool with a dedicate MATLAB toolbox for it [64]. For an extensive overview, refer to [62] and the references therein.

- *Fractional Lead-Lag Compensator.* In the above, fractional controllers are directly related to the use of fractional-order differentiator or integrator. It is possible to extend the classical lead-lag compensator to the fractional-order case which was studied in [83], [53]. The fractional lead-lag compensator is given by

  $$C_r(s) = C_0 \left(\frac{1 + s/\omega_b}{1 + s/\omega_h}\right)^r \qquad (15)$$

  where $0 < \omega_b < \omega_h$, $C_0 > 0$ and $r \in (0, 1)$. The autotuning technique has been presented in [53].

We conclude this section by offering the following remark.

Remark 4.1: Just like the non-integers are ubiquitous between integers, noninteger order control schemes will be ubiquitous by extending the existing integer order control schemes into their noninteger counterparts. For example, fractional sliding mode control with fractional order sliding surface dynamics; model reference adaptive control using fractional-order parameter updating law etc. The opportunities for extensions of existing integer-order controls are almost endless. However, the question remains: we need a good reason for such extensions. Performance enhancement as demonstrated in previous sections is only part of the reason.

V. HOW TO SIMULATE? INTRODUCING FOTF MATLAB TOOLBOX

In order to carry out numerical computation of fractional order operators, the revised version of (1) is rewritten as

$$_aD_t^{\alpha} f(t) = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{[(t-a)/h]} w_j^{(\alpha)} f(t - jh) \qquad (16)$$

where h is the step-size in computation, and $w_j^{(\alpha)}$ can be evaluated recursively from

$$w_0^{(\alpha)} = 1, \quad w_j^{(\alpha)} = \left(1 - \frac{\alpha + 1}{j}\right) w_{j-1}^{(\alpha)}, \; j = 1, 2, \cdots. \quad (17)$$

Laplace transform (3) can be applied to the fractional-order derivatives of a given signal. In particular, if the function $f(t)$ and its derivatives at $t = a$ are all equal to 0, one has

$$\mathscr{L}[_aD_t^{\alpha} f(t)] = s^{\alpha} \mathscr{L}[f(t)]. \qquad (18)$$

Fractional-order control systems are often modeled by fractional-order differential equations and a standard form of a linear time-invariant fractional-order differential equation is given in (4), from which the fractional-order transfer function (FOTF) model can be established

$$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}}. \qquad (19)$$

It can be seen that essential parameters of the FOTF model are, the order vectors of numerator and denominator and the coefficient vectors of them, which are summarized below

$$\begin{aligned} \boldsymbol{n}_b &= [\beta_m, \beta_{m-1}, \cdots, \beta_0], \quad \boldsymbol{n}_a = [\alpha_n, \alpha_{n-1}, \cdots, \alpha_0] \\ \boldsymbol{b} &= [b_m, b_{m-1}, \cdots, b_0], \quad \boldsymbol{a} = [a_n, a_{n-1}, \cdots, a_0]. \end{aligned} \qquad (20)$$

A MATLAB class FOTF is designed, and based on it, a series of overload functions are provided which are useful in the evaluation of block diagram modeling. A mini-toolbox based on the FOTF models is designed. The stability test, time- and frequency-domain analysis of FOTF models are presented with the use of the toolbox.

A. FOTF-Object Programming

In this section, MATLAB object-oriented programming technique is illustrated, with an application of the establishment of the FOTF-class, with which the fractional-order transfer function can be expressed. Based on the class, a series of overload functions can be written, aiming at describing ways for FOTF block interconnections and simplification. An illustrative example is given to show the fractional order feedback control system modeling.

1) FOTF-Class Creation: To define a class in MATLAB, folder with the name started with `@` is required. For instance for FOTF class, a folder named `@fotf` should be created first. Then in the folder, two files are essential: A `foft.m` file is used in defining the class, and a `display.m` file is used to define the way in which the class is displayed.

The lists of the two files are given below, respectively

```
function G=fotf(a,na,b,nb) %fotf.m
if nargin==0,
   G.a=[]; G.na=[]; G.b=[];
   G.nb=[]; G=class(G,'fotf');
elseif isa(a,'fotf'), G=a;
elseif nargin==1 & isa(a,'double'),
   G=fotf(1,0,a,0);
else,
   ii=find(abs(a)<eps); a(ii)=[];
   na(ii)=[]; ii=find(abs(b)<eps);
   b(ii)=[]; nb(ii)=[]; G.a=a;
   G.na=na; G.b=b; G.nb=nb;
   G=class(G,'fotf');
end

function display(G) %display.m
strN=polydisp(G.b,G.nb);
strD=polydisp(G.a,G.na);
nn=length(strN); nd=length(strD);
nm=max([nn,nd]); n1=(nm-nn)/2;
disp([char(' '*ones(1,floor(-n1))) strN])
disp(char('-'*ones(1,nm)))
disp([char(' '*ones(1,floor(n1)) strD])
function strP=polydisp(p,np)
P=''; [np,ii]=sort(np,'descend'); p=p(ii);
for i=1:length(p),
```

```
   P=[P,'+',num2str(p(i)),...
      's^{',num2str(np(i)),'}'];
end
P=P(2:end); P=strrep(P,'s^{0}','');
P=strrep(P,'+-','-');
P=strrep(P,'^{1}','');
P=strrep(P,'+1s','+s');
strP=strrep(P,'-1s','-s');
if length(strP)>=2 & strP(1:2)=='1s',
   strP=strP(2:end);
end
```

If one wants to specify a FOTF object

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

the model can be entered into MATLAB

```
>> b=[-2,4]; na=[3.501,2.42,1.798,1.31,0];
   nb=[0.63,0]; a=[2 3.8 2.6 2.5 1.5];
   G=fotf(a,na,b,nb)
```

and the FOTF object is displayed as

```
                  -2s^{0.63}+4
-------------------------------------------------
2s^{3.501}+3.8s^{2.42}+2.6s^{1.798}+2.5s^{1.31}+1.5
```

2) Overload Functions Programming: In mathematical representation, if two blocks, $G_1(s)$ and $G_2(s)$, are connected in series, the overall model can be evaluated from $G_2(s)G_1(s)$, and if they are in parallel, the overall model can be $G_2(s) + G_1(s)$.

The multiplication and plus operation for FOTF objects can be implemented with the overload facilities. Specifically, the `mtimes.m` and `mplus.m` functions can be designed in the `@fotf` folder.

```
function G=mtimes(G1,G2) % mtimes.m
G2=fotf(G2); a=kron(G1.a,G2.a);
b=kron(G1.b,G2.b); na=[]; nb=[];
for i=1:length(G1.na),
   na=[na,G1.na(i)+G2.na];
end
for i=1:length(G1.nb),
   nb=[nb,G1.nb(i)+G2.nb];
end
G=unique(fotf(a,na,b,nb));

function G=plus(G1,G2) % plus.m
a=kron(G1.a,G2.a); na=[]; nb=[];
b=[kron(G1.a,G2.b),kron(G1.b,G2.a)];
for i=1:length(G1.a),
   na=[na G1.na(i)+G2.na];
   nb=[nb, G1.na(i)+G2.nb];
end
for i=1:length(G1.b),
   nb=[nb G1.nb(i)+G2.na];
end
G=unique(fotf(a,na,b,nb));
```

A common function, `unique.m` is designed to collect polynomial terms and to simplify the FOTF model.

```
function G=unique(G)  % common unique.m
[a,n]=polyuniq(G.a,G.na); G.a=a; G.na=n;
[a,n]=polyuniq(G.b,G.nb); G.b=a; G.nb=n;
function [a,an]=polyuniq(a,an)
[an,ii]=sort(an,'descend'); a=a(ii);
ax=diff(an); key=1;
for i=1:length(ax)
   if ax(i)==0, a(key)=a(key)+a(key+1);
      a(key+1)=[]; an(key+1)=[];
```

```
    else, key=key+1; end
  end
```

Thus, whenever the ⋆ and + commands are used on FOTF objects, the corresponding overload function will be called automatically to perform the right task. This is the beauty of object-oriented programming under MATLAB.

Other functions should also be designed, such as the `feedback()`, `minus()`, `uminus()`, `inv()`, and the files should be placed in the `@fotf` directory to overload the existing ones.

```
function G=feedback(F,H)    % feedback.m
H=fotf(H); b=kron(F.b,H.a); na=[]; nb=[];
a=[kron(F.b,H.b), kron(F.a,H.a)];
for i=1:length(F.b),
   nb=[nb F.nb(i)+H.nb];
   na=[na,F.nb(i)+H.nb];
end
for i=1:length(F.a),
   na=[na F.na(i)+H.na];
end
G=unique(fotf(a,na,b,nb));

function G=uminus(G1)      % uminus.m
G=fotf(G1.a,G1.na,-G1.b,G1.nb);

function G=minus(G1,G2)    % minus.m
G=G1+(-G2);

function G=inv(G1)         % inv.m
G=fotf(G1.b,G1.nb,G1.a,G1.na);
```

With the above overloaded functions, the series, parallel and feedback connections of FOTF blocks, as well as the inverse and other manipulation operations can easily be achieved in a similar manner with the existing MATLAB Control Systems Toolbox.

Also if one designs a `fotf.m` file below in the `@tf` folder, the integer-order transfer function can be converted directly into a FOTF object.

```
function G1=fotf(G)
[n,d]=tfdata(G,'v');
i1=find(abs(n)<eps); i2=find(abs(d)<eps);
if length(i1)>0 & i1(1)==1, n=n(i1(1)+1:end); end
if length(i2)>0 & i2(1)==1, d=d(i2(1)+1:end); end
G1=fotf(d,length(d)-1:-1:0,n,length(n)-1:-1:0);
```

3) Illustrations of FOTF Modeling: With the use of the overload functions, interconnected systems can easily be simplified.

Suppose in the unity negative feedback system, the models are given by

$$G(s) = \frac{0.8s^{1.2} + 2}{1.1s^{1.8} + 0.8s^{1.3} + 1.9s^{0.5} + 0.4}$$

$$G_c(s) = \frac{1.2s^{0.72} + 1.5s^{0.33}}{3s^{0.8}}.$$

To find the overall model, the two fractional-order transfer function blocks should be entered first

```
>> G=fotf([1.1,0.8 1.9 0.4],...
        [1.8 1.3 0.5 0],[0.8 2],[1.2 0]);
   Gc=fotf([3],[0.8], [1.2 1.5],[0.72 0.33]);
   H=fotf(1,0,1,0); GG=feedback(G*Gc,H)
```

and the closed-loop FOTF is given by

$$G(s) = \frac{0.96s^{1.92} + 1.2s^{1.53} + 2.4s^{0.72} + 3s^{0.33}}{3.3s^{2.6} + 2.4s^{2.1} + 0.96s^{1.92} + 1.2s^{1.53} + 5.7s^{1.3}}.$$
$$+ 1.2s^{0.8} + 2.4s^{0.72} + 3s^{0.33}$$

It can be seen from the above illustrations that, although both the plant and controller models are relatively simple, extremely complicated closed-loop models may be obtained. This makes the analysis and design of the fractional-order system a difficult task.

B. Analysis of FOTF-Objects

In this section, three important system analysis methods are explored to FOTF blocks, with MATLAB implementations.

1) Stability: The stability assessment problems for a class of FOTFs can easily be carried out with the method in Section II-D. It should be noted that, only the denominator is meaningful in stability assessment and the numerator does not affect the stability of a FOTF.

The following MATLAB function can be written to test approximately the stability of a given FOTF model. In order to avoid the case that the order of the polynomial is too high, the resolution of commensurate-order is restricted to $q = 0.01$. The returned argument K is the stability of the system, with 1 for stable and 0 for unstable.

```
function [K,q,err,apol]=isstable(G)
a=G.na; a1=fix(100*a); n=gcd(a1(1),a1(2));
for i=3:length(a1), n=gcd(n,a1(i)); end
q=n/100, a=fix(a1/n), b=G.a;
c(a+1)=b; c=c(end:-1:1);
p=roots(c); p=p(abs(p)>eps);
err=norm(polyval(c,p));
plot(real(p),imag(p),'x',0,0,'o')
apol=min(abs(angle(p))); K=apol>q*pi/2;
xm=xlim; xm(1)=0; line(xm,q*pi/2*xm)
```

Consider the system model given below

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

The following statements can be given

```
>> b=[-2,4]; na=[3.501,2.42,1.798,1.31,0];
   nb=[0.63,0]; a=[2 3.8 2.6 2.5 1.5];
   [K,q,err,apol]=isstable(G)
```

Using the `isstable` function, the denominator is truncated automatically by $2s^{3.50} + 3.8s^{2.42} + 2.6s^{1.80} + 2.5s^{1.31} + 1.5$, with a least common divisor of 0.01. Thus it is found that $K = 1$, indicating the system is stable, with $q = 0.01$. The pole position plot is also obtained as shown in Fig. 2, and from the zoomed plot, it is immediately found that all the poles of the $s^{0.01}$ polynomial are located in the stable area, which means that the system is stable.

2) Time-Domain Analysis: Time domain responses of fractional-order systems can be evaluated with different methods. For instance, the impulse and step responses of commensurate-order systems can be obtained by the use of the well-established Mittag-Leffler functions[76]. However the solution method is time consuming and tedious.
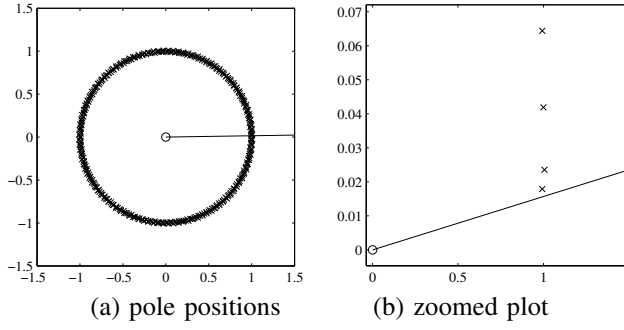
(a) pole positions   (b) zoomed plot

Fig. 2.   Pole positions with zoomed area

A closed-form solutions presented in [106] is useful in evaluating time-domain responses of linear fractional-order systems. Let us first consider a simple case. Suppose that the right hand side of (4) is $u(t)$ such that

$$a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots + a_0 D^{\alpha_0} y(t) = u(t). \quad (21)$$

Recall the Grünwald-Letnikov definition in (16). Substituting it to (21), the closed-form numerical solution to the fractional-order differential equation can be obtained as

$$y_t = \frac{1}{\displaystyle\sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}}} \left[u_t - \sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \sum_{j=1}^{[(t-a)/h]} w_j^{(\alpha_i)} y_{t-jh} \right]. \quad (22)$$

Now let us consider the full equation in (4), where the right-hand-side is not $u(t)$ but $\hat{u}(t)$ where

$$\hat{u}(t) = b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots + b_0 D^{\beta_0} u(t). \quad (23)$$

Thus $\hat{u}(t)$ should be evaluated first using the algorithm in (16), then from the closed-form solution in (22) the time response under input signal $u(t)$ can be obtained. A MATLAB function `lsim()` is implemented below

```
function y=lsim(G,u,t)
a=G.a; eta=G.na; b=G.b; gamma=G.nb;
nA=length(a); h=t(2)-t(1);
D=sum(a./[h.^eta]); W=[]; nT=length(t);
vec=[eta gamma]; D1=b(:)./h.^gamma(:);
y1=zeros(nT,1); W=ones(nT,length(vec));
for j=2:nT,
    W(j,:)=W(j-1,:).*(1-(vec+1)/(j-1));
end
for i=2:nT, A=[y1(i-1:-1:1)]'*W(2:i,1:nA);
    y1(i)=(u(i)-sum(A.*a./[h.^eta]))/D;
end
for i=2:nT,
    y(i)=(W(1:i,nA+1:end)*D1)'*[y1(i:-1:1)];
end
```

In the function call, the vector u of input samples at time vector t must be given, and the time response vector y can be obtained.

Step response of a fractional-order system is also very important, and since the step signal equal to one at all time t, an overload function `step()` can be implemented as

```
function y=step(G,t)
u=ones(size(t)); y=lsim(G,u,t);
if nargout==0, plot(t,y); end
```

If no returned argument is specified in the function call, the unit step response curves can be drawn automatically.

Still consider the stable system model

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

Selecting a step-size $h = 0.01$ and an interested time interval of $t \in [0, 30]$, the step response of the system can be obtained as shown in Fig. 3 (a), with the following statements

```
>> b=[-2,4]; na=[3.501,2.42,1.798,1.31,0];
   nb=[0.63,0]; a=[2 3.8 2.6 2.5 1.5];
   G=fotf(a,na,b,nb);
   t=0:0.01:30; step(G,t);
```


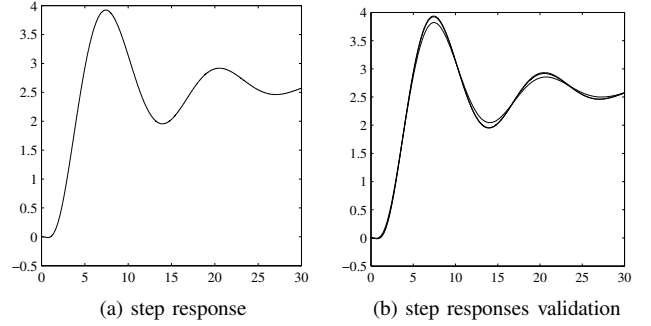(a) step response   (b) step responses validation

Fig. 3.   Step response and its validation

It should be noted that, since fixed-step computation is involved in the time response evaluation, the accuracy of the result may depend upon the step-size used. Thus a crucial procedure in the computation should not be neglected — the validation of the results. Due to the lack of analytical solution method, the only plausible way to validate the results is that, select different step-sizes and see whether they yield the same results. To validate the results, the step-sizes of 0.1 and 0.001 are selected, and with a new set of commands, the results are obtained as shown in Fig. 3 (b).

```
>> hold on
   t=0:0.1:30; step(G,t);
   t=0:0.001:30; step(G,t);
```

It can be observed from the results that, the step-size 0.1 yields the least computation effort, and there are slight differences between the solutions with other step-sizes. The step responses under the other two step-sizes are almost identical, and cannot be distinguished from each other, which means that the step response obtained under $h = 0.01$ is accurate.

3) Frequency-Domain Analysis: For FOTF models, the variable s can be replaced by $j\omega$ such that the frequency-domain response data can be obtained directly. Based on this fact, an overload function `bode()` can be written as

```
function H=bode(G,w)
a=G.a; na=G.na; b=G.b; nb=G.nb; j=sqrt(-1);
if nargin==1, w=logspace(-4,4); end
for i=1:length(w)
   P=b*((j*w(i)).^nb.');
   Q=a*((j*w(i)).^na.'); H1(i)=P/Q;
end
```

```
H1=frd(H1,w);
if nargout==0, bode(H1); else, H=H1; end
```

Also, overload Nyquist and Nichols plots functions can also be written

```
function nyquist(G,w)
if nargin==1, w=logspace(-4,4); end;
H=bode(G,w); nyquist(H);

function nichols(G,w)
if nargin==1, w=logspace(-4,4); end;
H=bode(G,w); nichols(H);
```

With the above overload functions, the Bode plot and Nyquist plot of the previous FOTF object can easily be drawn as shown in Fig. 4.

```
>> b=[-2,4]; na=[3.501,2.42,1.798,1.31,0];
   nb=[0.63,0]; a=[2 3.8 2.6 2.5 1.5];
   G=fotf(a,na,b,nb); w=logspace(-1,2);
   bode(G,w), figure; nyquist(G,w)
```
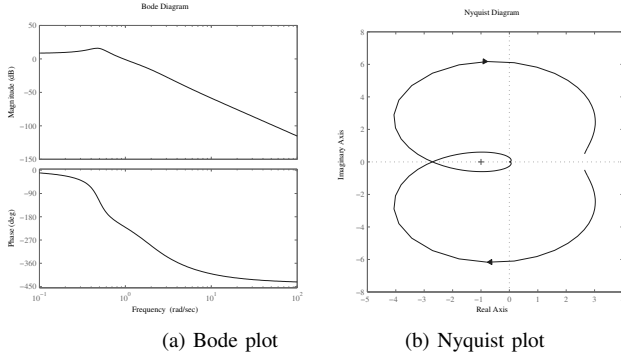

(a) Bode plot (b) Nyquist plot

Fig. 4. Bode and Nyquist plots

VI. APPROXIMATE REALIZATION TECHNIQUES

In general, if a function $f(t)$ is approximated by a grid function, $f(nh)$, where h the grid size, the approximation for its fractional derivative of order α can be expressed as [25]:

$$y_h(nh) = h^{\mp\alpha}\left(\omega\left(\zeta^{-1}\right)\right)^{\pm\alpha} f_h(nh) \qquad (24)$$

where ζ^{-1} is the shift operator, and $\omega\left(\zeta^{-1}\right)$ is a generating function, where $s \approx \omega\left(\zeta^{-1}\right)$. This generating function and its expansion determine both the form of the approximation and its coefficients [37]. List of some generating fucntions is presented in Table VI-D.

It is worth mentioning that, in general, the case of controller realization is not equivalent to the cases of simulation or numerical evaluation of the fractional integral and differential operators. In the case of controller realization it is necessary to take into account some important considerations. First of all, the value of h, the step when dealing with numerical evaluation, is the value of the sample period T, and it is limited by the characteristics of the microprocessor-based system, used for the controller implementation, in two ways: (i) each microprocessor-based system has its own minimum value for the sample period, and (ii) it is necessary to perform all the computations required by the control law

between two samples. Due to this last reason, it is very important to obtain good approximations with a minimal set of parameters. On the other hand, when the number of parameters in the approximation increases, it increases the amount of the required memory and speed too.

It is also important to have discrete equivalents or approximations with poles and zeros, that is, in a rational form.

In the following, the notation normally used in control theory is adopted, that is: T, the sample period, is used instead of h, and z, the complex variable resulting from the application of the Z transform to the functions $y(nT), f(nT)$ considered as sequences, is used instead of ζ.

A. Discrete Approximations Using PSE

The simplest and most straightforward method is the direct discretization using finite memory length expansion from GL definition (1). In general, the discretization of fractional-order differentiator/integrator $s^{\pm r}, (r \in R)$ can be expressed by the generating function $s = \omega(z^{-1})$.

Using the generating function corresponding to the backward fractional difference rule, $\omega\left(z^{-1}\right) = \left(1 - z^{-1}\right)$, and performing the power series expansion (PSE) of $\left(1 - z^{-1}\right)^{\alpha}$, the Grünwald–Letnikov formula (1) for the fractional derivative of order α is obtained.

In any case, the resulting transfer function, approximating the fractional-order operators, can be obtained by applying the following relationship:

$$Y(z) = T^{\mp\alpha}\text{PSE}\left\{\left(1 - z^{-1}\right)^{\pm\alpha}\right\}F(z) \qquad (25)$$

where T is the sample period, $Y(z)$ is the Z transform of the output sequence $y(nT)$, $F(z)$ is the Z transform of the input sequence $f(nT)$, and PSE$\{u\}$ denotes the expression, which results from the power series expansion of the function u.

Doing so gives:

$$D^{\pm\alpha}(z) = \frac{Y(z)}{F(z)} = T^{\mp\alpha}\ \text{PSE}\left\{\left(1 - z^{-1}\right)^{\pm\alpha}\right\} \simeq T^{\mp\alpha}P_p(z^{-1}) \qquad (26)$$

where $D^{\pm\alpha}(z)$ denotes the discrete equivalent of the fractional-order operator, considered as processes.

By using the short memory principle [76], the discrete equivalent of the fractional-order integro-differential operator, $(\omega(z^{-1}))^{\pm\alpha}$, is given by

$$D^{\pm\alpha}(z) = (\omega(z^{-1}))^{\pm\alpha} = T^{\mp\alpha}z^{-[L/T]}\sum_{j=0}^{[L/T]}(-1)^j\binom{\pm\alpha}{j}z^{[L/T]-j} \qquad (27)$$

where T is the sampling period, L is the memory length and $(-1)^j\binom{\pm\alpha}{j}$ are the binomial coefficients $w_j^{(\alpha)}, (j = 0, 1, \ldots)$ computed according to relation (17).

For practical numerical calculation or simulation of the fractional derivative and integral we can derive from the GL definition (1) and (27) the following formula

$$\begin{aligned}{}_{(k-L/T)}D_{kT}^{\pm\alpha}f(t) &\approx T^{\mp\alpha}\sum_{j=v}^{k}(-1)^j\binom{\pm\alpha}{j}f_{k-j} \\ &= T^{\mp\alpha}\sum_{j=v}^{k}c_j^{(\pm\alpha)}f_{k-j},\end{aligned} \qquad (28)$$

where $v = 0$ for $k < (L/T)$ or $v = k - (L/T)$ for $k > (L/T)$ in the relation (28).

Obviously, for this simplification we pay a penalty in the form of some inaccuracy. If $f(t) \leq M$, we can easily establish the following estimate for determining the memory length L, providing the required accuracy ε:

$$L \geq \left(\frac{M}{\varepsilon |\Gamma(1-\alpha)|} \right)^{1/\alpha}. \quad (29)$$

An evaluation of the short memory effect and convergence relation of the error between short and long memory were clearly described and also proved in [76].

Performing the PSE of the function $\left(1 - z^{-1}\right)^{-\alpha}$ leads to the formula given by Lubich for the fractional integral/derivative of order α [37]:

$$\nabla_T^{\pm\alpha} f(nT) = T^{\pm\alpha} \sum_{k=0}^{\infty} (-1)^k \binom{\mp\alpha}{k} f((n-k)T). \quad (30)$$

Another possibility for the approximation is the use of the trapezoidal rule, that is, the use of the generating function and then the PSE

$$\omega(z^{-1}) = 2 \frac{1 - z^{-1}}{1 + z^{-1}} \quad (31)$$

It is known that the forward difference rule is not suitable for applications to causal problems [25], [37].

It should be mentioned that, at least for control purposes, it is not very important to have a closed-form formula for the coefficients, because they are usually pre-computed and stored in the memory of the microprocessor. In such a case, the most important is to have a limited number of coefficients because of the limited available memory of the microprocessor system.

It is very important to note that the PSE scheme leads to approximations in the form of polynomials, that is, the discretized fractional order derivative is in the form of FIR filters. Taking into account that our aim is to obtain discrete equivalents to the fractional integrodifferential operators in the Laplace domain, $s^{\pm\alpha}$, the following considerations have to be made:

1) s^α, $(0 < \alpha < 1)$, viewed as an operator, has a branch cut along the negative real axis for arguments of s on $(-\pi, \pi)$ but is free of poles and zeros.
2) A dense interlacing of simple poles and zeros along a line in the s plane is, in some way, equivalent to a branch cut.
3) It is well known that, for interpolation or evaluation purposes, rational functions are sometimes superior to polynomials, roughly speaking, because of their ability to model functions with zeros and poles. In other words, for evaluation purposes, rational approximations frequently converge much more rapidly than PSE and have a wider domain of convergence in the complex plane.
4) The trapezoidal rule maps adequately the stability regions of the s plane on the z plane, and maps the

points $s = 0, s = -\infty$ to the points $z = 1$ and $z = -1$ respectively.

It should be pointed out that, for control applications, the obtained approximate discrete-time rational transfer function should be stable and minimum phase. Furthermore, for a better fit to the continuous frequency response, it would be of high interest to obtain discrete approximations with poles and zeros interlaced along the line $z \in (-1, 1)$ in the z plane. As it will be shown later, the direct discretization approximations enjoy the above desirable properties.

B. Discrete Approximations Using CFE

The approximations, considered in the previous section, lead to discrete transfer functions in the form of polynomials, and this is not convenient, at least from the control point of view. On the other hand, it can be recalled that the CFE (continuous fractional expansion) leads to approximations in rational form, and often converges much more rapidly than PSE and has a wider domain of convergence in the complex plane, and, consequently, a smaller set of coefficients will be necessary for obtaining a good approximation.

In view of these reasons, a method for obtaining discrete equivalents of the fractional-order operators, which combines the well known advantages of the trapezoidal rule in the control theory and the advantages of the CFE, is introduced here. The method involves the following:

- Use of the generating function

$$\omega(z^{-1}) = 2 \frac{1 - z^{-1}}{1 + z^{-1}},$$

where z is a complex variable, and z^{-1} is the shifting operator,
- The continued fraction expansion (CFE) of

$$\left(\omega(z^{-1})\right)^{\pm\alpha} = \left(2 \frac{1 - z^{-1}}{1 + z^{-1}}\right)^{\pm\alpha}$$

for obtaining the coefficients and the form of the approximation.

It is well known that the Continued Fraction Expansion (CFE) is a method for evaluation of functions that frequently converges much more rapidly than power series expansions, and in a much larger domain in the complex plane [82]. The result of such approximation for an irrational function, $G(z)$, can be expressed in the form:

$$\begin{aligned} G(z) \quad &\simeq \quad a_0(z) + \cfrac{b_1(z)}{a_1(z) + \cfrac{b_2(z)}{a_2(z) + \cfrac{b_3(z)}{a_3(z) + \cdots}}} \\ &= \quad a_0(z) + \frac{b_1(z)}{a_1(z)+} \frac{b_2(z)}{a_2(z)+} \frac{b_3(z)}{a_3(z)+} \cdots \quad (32) \end{aligned}$$

where a_i and b_i are either rational functions of the variable z or constants. The application of the method yields a rational function, $\widehat{G}(z)$, which is an approximation of the irrational function $G(z)$.

The resulting discrete transfer function, approximating fractional-order operators, can be expressed as:

$$D^{\pm\alpha}(z) = \frac{Y(z)}{F(z)} = \left(\frac{2}{T}\right)^{\pm\alpha} \text{CFE}\left\{ \left(\frac{1-z^{-1}}{1+z^{-1}}\right)^{\pm\alpha} \right\}_{p,q}$$

$$= \left(\frac{2}{T}\right)^{\pm\alpha} \frac{P_p(z^{-1})}{Q_q(z^{-1})}, \qquad (33)$$

where T is the sample period, CFE$\{u\}$ denotes the function resulting from applying the continued fraction expansion to the function u, $Y(z)$ is the Z transform of the output sequence $y(nT)$, $F(z)$ is the Z transform of the input sequence $f(nT)$, p and q are the orders of the approximation, and P and Q are polynomials of degrees p and q, correspondingly, in the variable z^{-1}.

As for generating function for CFE, we in fact can use any of function listed in Table VI-D. For example, Chen and Moore in [14] used the so-called Al-Alaoui operator, which is the weighted sum of rectangular rule or Euler operator (0.25) and the trapezoidal rule (0.75), and CFE in the form:

$$D^{\pm\alpha}(z) \approx \left(\frac{8}{7T}\right)^{\pm\alpha} \text{CFE}\left\{ \left(\frac{1-z^{-1}}{1+z^{-1}/7}\right)^{\pm\alpha} \right\}_{p,q}$$

$$= \left(\frac{8}{7T}\right)^{\pm\alpha} \frac{P_p(z^{-1})}{Q_q(z^{-1})}, \qquad (34)$$

where CFE$\{u\}$ denotes the continued fraction expansion of u; p and q are the orders of the approximation and P and Q are polynomials of degrees p and q. Normally, we can also set $p = q = n$.

We can use the well-known backward rule difference as a generating function and apply a CFE. The better way is use a method proposed by Vinagre [94] and obtain an IIR form of approximation by using a PSE of backward rule difference in the following form:

$$D^{\mp\alpha}(z) = \frac{1}{(1-z^{-1})^{\pm\alpha}} = \frac{T^{\mp\alpha}}{\sum\limits_{j=0}^{\infty}(-1)^j \binom{\pm\alpha}{j} z^{-j}} \simeq \frac{T^{\mp\alpha}}{Q_q(z^{-1})}, \qquad (35)$$

where $Q_q(z^{-1})$ is polynomial of order q in variable z^{-1}.

C. Discretization by Muir's Recursion

One of the key points of Tustin discretization of fractional-order differentiator is how to get a recursive formula similar to (17) in the preceding subsection. Here we introduce the so-called Muir-recursion originally used in geophysical data processing with applications to petroleum prospecting [16]. The Muir-recursion motivated in computing the vertical plane wave reflection response via the impedance of a stack of n-layered earth can be used in recursive discretization of fractional-order differentiator of Tustin generating function. In the following, without loss of generality, assume that $\alpha \in [-1, 1]$. Moreover, in order to simplify the presentation, we only give the recursive formula for positive α (see e.g.

[14], [96]).

$$(\omega(z^{-1}))^\alpha = \left(\frac{2}{T}\right)^\alpha \left(\frac{1-z^{-1}}{1+z^{-1}}\right)^\alpha = \left(\frac{2}{T}\right)^\alpha \lim_{n\to\infty} \frac{A_n(z^{-1}, \alpha)}{A_n(z^{-1}, -\alpha)} \qquad (36)$$

where

$$A_0(z^{-1}, \alpha) = 1, \quad A_n(z^{-1}, \alpha) = A_{n-1}(z^{-1}, \alpha) - c_n z^n A_{n-1}(z, \alpha) \qquad (37)$$

and

$$c_n = \begin{cases} \alpha/n & ; \quad n \text{ is odd}; \\ 0 & ; \quad n \text{ is even}. \end{cases} \qquad (38)$$

For any given order of approximation n, we can use MATLAB Symbolic Toolbox to generate an expression for $A_n(z^{-1}, \alpha)$. Therefore,

$$s^\alpha \approx \left(\frac{2}{T}\right)^\alpha \frac{A_n(z^{-1}, \alpha)}{A_n(z^{-1}, -\alpha)}.$$

For a ready reference, we listed $A_n(z^{-1}, \alpha)$ in Table I up to $n = 9$, which should be sufficient in many applications [14].

TABLE I
TABLE OF FORMULAE $A_n(z^{-1}, \alpha)$ FOR $n = 1, \cdots, 9$

n	$A_n(z^{-1}, \alpha)$
0	1
1	$-\alpha z^{-1} + 1$
3	$-\frac{1}{3}\alpha z^{-3} + \frac{1}{3}\alpha^2 z^{-2} - \alpha z^{-1} + 1$
5	$-\frac{1}{5}\alpha z^{-5} + \frac{1}{5}\alpha^2 z^{-4} - \left(\frac{1}{3}\alpha + \frac{1}{15}\alpha^3\right)z^{-3} + \frac{2}{5}\alpha^2 z^{-2} - \alpha z^{-1} + 1$
7	$-\frac{1}{7}\alpha z^{-7} + \frac{1}{7}\alpha^2 z^{-6} - \left(\frac{1}{5}\alpha + \frac{2}{35}\alpha^3\right)z^{-5} + \left(\frac{26}{105}\alpha^2 + \frac{1}{105}\alpha^4\right)z^{-4} - \left(\frac{1}{3}\alpha + \frac{2}{21}\alpha^3\right)z^{-3} + \frac{3}{7}\alpha^2 z^{-2} - \alpha z^{-1} + 1$
9	$-\frac{1}{9}\alpha z^{-9} + \frac{1}{9}\alpha^2 z^{-8} - \left(\frac{1}{7}\alpha + \frac{1}{21}\alpha^3\right)z^{-7} + \left(\frac{34}{189}\alpha^2 + \frac{2}{189}\alpha^4\right)z^{-6} - \left(\frac{1}{5}\alpha + \frac{16}{189}\alpha^3 + \frac{1}{945}\alpha^5\right)z^{-5} + \left(\frac{17}{63}\alpha^2 + \frac{1}{63}\alpha^4\right)z^{-4} - \left(\frac{1}{3}\alpha + \frac{1}{9}\alpha^3\right)z^{-3} + \frac{4}{9}\alpha^2 z^{-2} - \alpha z^{-1} + 1$

Remark 6.1: To examine the correctness of the Muir-recursion used for the recursive discretization of the fractional-order derivative operator, one can compare the symbolic Taylor expansion of (VI-B). It has been verified that the proposed recursive formula is as correct as Taylor series expansion till the order of approximation.

D. Other Direct Discretization Methods

Besides the above methods described we can also use an approach proposed by Machado [40]. Instead binomial series for expansion of generating functions may be used a Taylor series. The fractional order conversion scheme leads to non-rational formula, where final algorithm corresponds to a n-term truncated series.

Approximations, listed in Table 5.5, and their properties were analyzed and implemented to control system in Machado's papers (see e.g. [40], [41]).

Similar approach was described by Duarte in [24], where instead of a Tylor series, a MacLaurin series truncated after

Methods	$s \to z$ conversion
Euler	$s^{\alpha} \approx \left\lceil \dfrac{1 - z^{-1}}{T} \right\rceil^{\alpha}$
Al-Alaoui	$s^{\alpha} \approx \left\lceil \dfrac{8}{7T} \dfrac{1 - z^{-1}}{1 + z^{-1}/7} \right\rceil^{\alpha}$
Tustin	$s^{\alpha} \approx \left\lceil \dfrac{2}{T} \dfrac{1 - z^{-1}}{1 + z^{-1}} \right\rceil^{\alpha}$
Simpson	$s^{\alpha} \approx \left\lceil \dfrac{3}{T} \dfrac{(1 + z^{-1})(1 - z^{-1})}{1 + 4z^{-1} + z^{-2}} \right\rceil^{\alpha}$

Fig. 5. Block diagram of the canonical representation of IIR filter form

n-th order term was used. The general expressions were described as well.

Both mentioned approaches lead to non-rational approximation in the form of a FIR filter.

There are many other suggestions to discretize fractional calculus and numerical solution of fractional differential equations. We should mention Diethelm's work [20], where discretization algorithm was based on the quadrature formula approach. Another method is proposed by Leszczynski. In his proposal the algorithm for numerical solution is obtained by using a decomposition of the fractional differential equation into a system of ordinary differential equation of integer order and inverse form of Abel's integral equation [35]. Last but not least we should mention the approach proposed by Hwang, which is based on B-splines function [29] and Podlubny's matrix approach [78], [81].

VII. HOW TO IMPLEMENT? PROPOSED REALIZATIONS

Basically, there are two methods for realization of the FOC. One is a digital realization based on microprocessor devices and appropriate control algorithm and the second one is an analogue realization based on analogue circuits so-called *fractance*. This section describe both of the realizations.

A. Digital realization: Control algorithm

This realization can be based on implementation of the control algorithm in the microprocessor devices, e.g.: PLC controller [71], processor C51 or PIC [69], PCL I/O card [95], etc. Some experimental measurements with processor and PCL card were already done in [69], [95].

Generally, the control algorithm may be based on canonical form of IIR filter, which can be expressed as follow

$$F(z^{-1}) = \frac{U(z^{-1})}{E(z^{-1})} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-N}}, \quad (39)$$

where $a_0 = 1$ for compatible with the definitions used in MATLAB. Normally, we choose $M = N$.

The FOC in form of IIR filter can be directly implemented to any microprocessor based devices as for instance PLC or PIC. A direct form of such implementation using canonical form shown in Fig. 5, with input $e(k)$ and output $u(k)$ range mapping to the interval $0 - U_{FOC}$ [V] divided into two sections: initialization code and loop code. The pseudo-code has the following form:
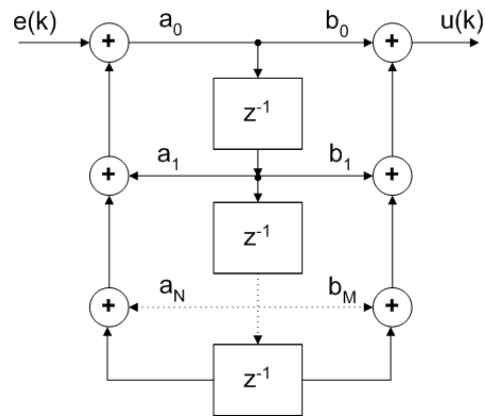
```
(* initialization code *)
scale := 32752; % input and output
order := 5; % order of approximation
U_FOC := 10; % input and output voltage range:
             % 5[V], 10[V], ...

a[0] := ...; a[1] := ...; a[2] := ...;
a[3] := ...; a[4] := ...; a[5] := ...;
b[0] := ...; b[1] := ...; b[2] := ...;
b[3] := ...; b[4] := ...; b[5] := ...;

loop i := 0 to order do
  s[i] := 0;
endloop

(* loop code *)
in := (REAL(input)/scale) * U_FOC; feedback := 0;
feedforward := 0; loop i:=1 to order do
  feedback := feedback - a[i] * s[i];
  feedforward := feedforward + b[i] * s[i];
endloop
s[0] := in + a[0] * feedback;
out := b[0] * s[1] + feedforward;
loop i := order downto 1 do
  s[i] := s[i-1];
endloop
output := INT(out*scale)/U_FOC;
```

The disadvantage with this solution is that the complete controller is calculated using floating point arithmetic.

All discrete techniques described in Sec.VI allow us use a fractional operator in discrete form but we have to realize the capacity and speed limitation of real devices as for example PIC microprocessor.

B. Analogue Realization: Fractance Circuits

A circuit exhibiting fractional-order behavior is called a *fractance* [76]. The fractance devices have the following characteristics [56], [59], [30]. First, the phase angle is constant independent of the frequency within a wide frequency band. Second, it is possible to construct a filter which has moderated characteristics which can not be realized by using the conventional devices.

Generally speaking, there are three basic fractance devices. The most popular is a domino ladder circuit network. Very often used is a tree structure of electrical elements and finally, we can find out also some transmission line circuit. Here

we must mention that all basic electrical elements (resistor, capacitor and coil) are not ideal [11], [99].

Design of fractances can be done easily using any of the rational approximations [72] or a truncated CFE, which also gives a rational approximation.

Truncated CFE does not require any further transformation; a rational approximation based on other methods must be transformed to the form of a continued fraction. The values of the electric elements, which are necessary for building a fractance, are then determined from the obtained finite continued fraction. If all coefficients of the obtained finite continued fraction are positive, then the fractance can be made of classical passive elements (resistors and capacitors). If some of the coefficients are negative, then the fractance can be made with the help of negative impedance converters [72].

Domino ladder lattice networks can approximate fractional operator more effectively than the lumped networks [23].
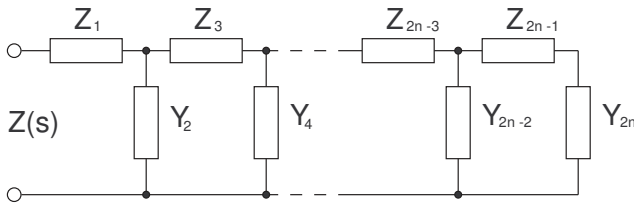

Fig. 6.   Finite ladder circuit

Let us consider the circuit depicted in Fig. 6, where $Z_{2k-1}(s)$ and $Y_{2k}(s)$, $k = 1,\ldots,n$, are given impedances of the circuit elements. The resulting impedance $Z(s)$ of the entire circuit can be found easily, if we consider it in the right-to-left direction:

$$Z(s) = Z_1(s) + \cfrac{1}{Y_2(s) + \cfrac{1}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots \cfrac{1}{Y_{2n-2}(s) + \cfrac{1}{Z_{2n-1}(s) + \cfrac{1}{Y_{2n}(s)}}}}} \quad (40)$$

The relationship between the finite domino ladder network, shown in Fig. 6, and the continued fraction (40) provides an easy method for designing a circuit with a given impedance $Z(s)$. For this one has to obtain a continued fraction expansion for $Z(s)$. Then the obtained particular expressions for $Z_{2k-1}(s)$ and $Y_{2k}(s)$, $k = 1,\ldots,n$, will give the types of necessary components of the circuit and their nominal values.

Rational approximation of the fractional integrator/differentiator can be formally expressed as:

$$s^{\pm\alpha} \approx \left\{ \frac{P_p(s)}{Q_q(s)} \right\}_{p,q} = Z(s), \quad (41)$$

where p and q are the orders of the rational approximation, P and Q are polynomials of degree p and q, respectively.

For direct calculation of circuit elements was proposed method by Wang [98]. This method was designed for constructing resistive-capacitive ladder network and transmission lines that have a generalized Warburg impedance $As^{-\alpha}$, where A is independent of the angular frequency and $0 < \alpha < 1$. This impedance may appear at an electrode/electrolyte interface, etc. The impedance of the ladder network (or transmission line) can be evaluated and rewritten as a continued fraction expansion:

$$Z(s) = R_0 + \frac{1}{C_0 s+} \frac{1}{R_1+} \frac{1}{C_1 s+} \frac{1}{R_2+} \frac{1}{C_2 s+} \cdots \quad (42)$$

If we consider that $Z_{2k-1} \equiv R_{k-1}$ and $Y_{2k} \equiv C_{k-1}$ for $k = 1,\ldots,n$ in Fig. 6, then the values of the resistors and capacitors of the network are specified by

$$R_k = 2h^\alpha P(\alpha)\frac{\Gamma(k+\alpha)}{\Gamma(k+1-\alpha)} - h^\alpha \delta_{ko}$$

$$C_k = h^{1-\alpha}(2k+1)\frac{\Gamma(k+1-\alpha)}{P(\alpha)\Gamma(k+1+\alpha)},$$

$$P(\alpha) = \frac{\Gamma(1-\alpha)}{\Gamma(\alpha)}, \quad (43)$$

where $0 < \alpha < 1$, h is an arbitrary small number, δ_{ko} is the Kronecker delta, and k is an integer, $k \in [0, \infty)$.
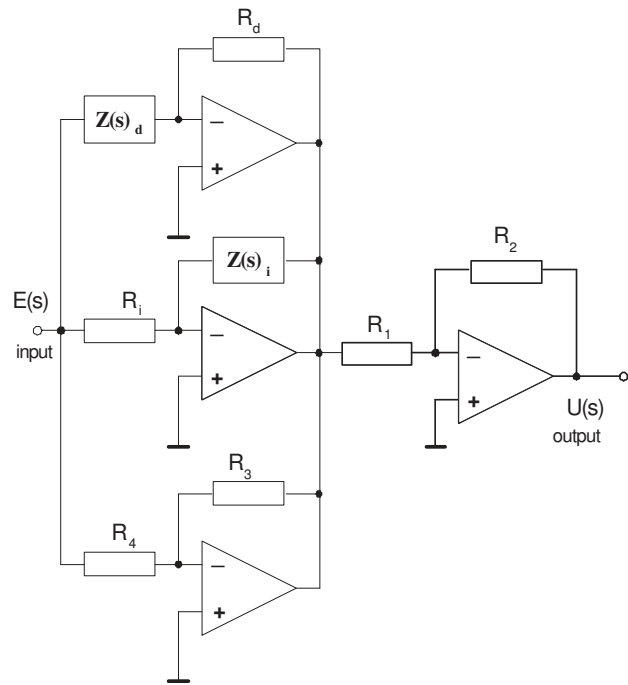

Fig. 7.   Analogue fractional-order $PI^\lambda D^\delta$ controller

Figure 7 depicts an analogue implementation of fractional-order $PI^\lambda D^\delta$ controller. Fractional order differentiator is approximated by general Warburg impedance $Z(s)_d$ and fractional order integrator is approximated by impedance $Z(s)_i$, where orders of both approximations are $0 < \alpha < 1$. For orders greater than 1, the Warburg impedance can be combined with classical integer order circuit block. Usually we assume $R_2 = R_1$ in Fig. 7.

For proportional gain K_p we can write the formula:

$$K_p = \frac{R_3}{R_4}.$$

The integration constant T_i can be computed from relationship

$$T_i = \frac{Z(s)_i}{R_i}.$$

For derivative constant T_d we can write formula:

$$T_d = \frac{R_d}{Z(s)_d}.$$

In the case, if we use identical resistors (R-series) and identical capacitors (C-shunt) in the fractances, then the behavior of the circuit will be as a half-order integrator/differentiator. Realization and measurements of such kind controllers were done in [72].

Instead of fractance circuit the new electrical element introduced by G. Bohannan called *Fractor* can be used as well [9]. This element - Fractor made from a material with the properties of $LiN_2H_5SO_4$ has been already used for temperature control [5].

Last but not least we should mention the implementation technique based on the memristive devices recently suggested in [17]. This new implementation involves using memristors and other memristive systems for realization of the fractional-order controllers.

VIII. CONCLUSIONS AND FUTURE PERSPECTIVES

In this tutorial article, we offer a tutorial on fractional calculus in controls. Basic definitions of fractional calculus, fractional order dynamic systems and controls are presented first. Then, fractional order PID controllers are introduced which may make fractional order controllers ubiquitous in industry. Additionally, several typical known fractional order controllers are introduced and commented. Numerical methods for simulating fractional order systems are given in detail so that a beginner can get started quickly. Discretization techniques for fractional order operators are introduced in some details. Both analog and digital realization of fractional operators are introduced.

As for the future perspectives, we briefly offer the following remarks for future investigation

- Power law Lyapunov stability theory should replace the exponential law based Lyapunov stability theory?
- Power law phenomena are due to time-/spatial-fractional order dynamics?
- Time frequency analysis, multi-resolution analysis (wavelet), fractional Fourier transformation, and fractional order calculus are inter-related?
- Long range dependence of stochastic process is due to fractional order dynamics?
- $\cdots$

Some of our recent investigations have already shown that, some of the above speculations are true.

As final remarks, the readers are reminded that whenever the following words appears:

- power law
- long range dependence
- porous media
- particulate

- granular
- lossy
- anomaly
- disorder
- scale-free, scale-invariant
- complex dynamic system
- $\cdots$

please think about fractional order dynamics and controls. In general, fractional order dynamics and controls are ubiquitous.

IX. ACKNOWLEDGMENTS

REFERENCES

[1] M. A. Al-Alaoui: Novel digital integrator and differentiator, *Electron. Lett.*, vol. 29, no. 4, 1993, pp. 376–378.

[2] M. A. Al-Alaoui: Filling the gap between the bilinear and the backward difference transforms: An interactive design approach, *Int. J. Elect. Eng. Edu.*, vol. 34, no. 4, 1997, pp. 331–337.

[3] Tore Hagglund Karl J. Astrom. *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems, and Automation Society (2nd edition), 1995.

[4] M. Axtell and E. M. Bise: Fractional calculus applications in control systems. *Proc. of the IEEE 1990 Nat. Aerospace and Electronics Conf.*, New York, 1990, pp. 563–566.

[5] T. Bashkaran, Y. Q. Chen, and G. Bohannan: Practical tuning of fractional order proportional and integral controller (II): Experiments, *Proc. of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, September 4-7, 2007, Las Vegas, Nevada, USA.

[6] W. L. Bialkowski. Dreams versus reality: A view from both sides of the gap. *Pulp and Paper Canada*, 11:19–27, 1994.

[7] Blas M. Vinagre and YangQuan Chen. Lecture notes on fractional calculus applications in automatic control and robotics. In Blas M. Vinagre and YangQuan Chen, editors, *The 41st IEEE CDC2002 Tutorial Workshop # 2*, pages 1–310. [Online] http://mechatronics.ece.usu.edu/foc/cdc02_tw2_ln.pdf, Las Vegas, Nevada, USA, 2002.

[8] H. W. Bode. *Network Analysis and Feedback Amplifier Design*, Tung Hwa Book Company, 1949.

[9] G. Bohannan: Analog Realization of a Fractional Control Element - Revisited, *Proc. of the 41st IEEE Int. Conf. on Decision and Control, Tutorial Workshop 2: Fractional Calculus Applications in Automatic Control and Robotics*, December 9, 2002, Las Vegas, USA.

[10] G. E. Carlson and C. A. Halijak: Simulation of the fractional derivative operator and the fractional integral operator, *Kansas State University Bulletin*, vol. 45, no. 7, 1961, pp. 1–22.

[11] G. E. Carlson and C. A. Halijak: Approximation of fractional capacitors $(1/s)^{1/n}$ by a regular Newton process, *IEEE Trans. on Circuit Theory*, vol. 11, no. 2, 1964, pp. 210–213.

[12] A. Charef, H. H. Sun, Y. Y. Tsao, and B. Onaral: Fractal system as represented by singularity function, *IEEE Trans. on Automatic Control*, vol. 37, no. 9, September 1992, pp. 1465–1470.

[13] YangQuan Chen. "Ubiquitous Fractional Order Controls?", (12 pages plenary talk paper) The Second IFAC Symposium on Fractional Derivatives and Applications (IFAC FDA06) 19 - 21 July, 2006. Porto, Portugal.

[14] Y. Q. Chen and K. L. Moore: Discretization schemes for fractional - order differentiators and integrators. *IEEE Trans. On Circuits and Systems - I: Fundamental Theory and Applications*, vol. 49, no. 3, 2002, pp. 363–367.

[15] YangQuan Chen, Kevin L. Moore, Blas M. Vinagre, and Igor Podlubny. Robust pid controller autotuning with a phase shaper. In *Proceedings of The First IFAC Symposium on Fractional Differentiation and its Applications (FDA04)*, Bordeaux, France, 2004a.

[16] J. F. Claerbout: *Fundamentals of Geophysical Data Processing with applications to petroleum prospecting*, Blackwell Scientific Publications, 1976, http://sepwww.stanford.edu/oldreports/fgdp2/.

[17] C. Coopmans, I. Petras and Y. Q. Chen: Analogue fractional-order generalized memristive devices. Submitted to *Proc. of the ASME 2009*, IDETC/CIE 2009, San Diego, USA.

[18] Shankar P. Bhattacharyya Aniruddha Datta, Ming-Tzu Ho. *Structure and Synthesis of PID Controllers*. Springer-Verlag, London, 2000.

[19] J. J. D'Azzo and C. H. Houpis: *Linear Control System Analysis and Design: Conventional and Modern*, McGraw-Hill, New York, 1995.

[20] K. Diethelm: An Algorithm for the Numerical Solution of Differential Equations of Fractional Order, *Elec. Trans. On Num. Analysis*, vol. 5, 1997, pp. 1–6.

[21] Ľ. Dorčák: *Numerical models for simulation the fractional-order control systems*, UEF-04-94, Slovak Academy of Science, Kosice, 1994. http://xxx.lanl.gov/abs/math.OC/0204108/

[22] R. C. Dorf and R. H. Bishop: *Modern Control Systems*. Addison-Wesley, New York, 1990.

[23] S. C. Dutta Roy: On the realization of a constant-argument immitance of fractional operator, *IEEE Trans. on Circuit Theory*, vol. 14, no. 3, 1967, pp. 264–374.

[24] V. Duarte and J.S.Costa: Time-domain implementations of non-integer order controllers, *Proceedings of Controlo 2002*, Sept. 5–7, Portugal, pp. 353 – 358.

[25] R. Gorenflo: Fractional Calculus: Some Numerical Methods, *CISM Lecture Notes*, Udine, Italy, 1996.

[26] T. C. Haba, M. Martos, G. Ablart, and P. Bidan: Composants électroniques á impédance fractionnaire, *Proceeding of Fractional Differential Systems: Models, Methods and Applications*, vol. 5, 1998, pp. 99–109.

[27] D. Heleschewitz and D. Matignon: Diffusive realisations of fractional integrodifferential operators: structural analysis under approximation. *IFAC Conference on System, Structure and Control*, Vol. 2, Nantes, France, 1998, pp. 243-248.

[28] N. Heymans and J.-C. Bauwens: Fractal rheological models and fractional differential equations for viscoelastic behavior, *Rheologica Acta*, vol. 33, 1994, pp. 210–219.

[29] C. Hwang, J. F. Leu and S.Y. Tsay: A note on time-domain simulation of feedback fractional order systems, *IEEE Trans. On Automatic Control*, vol. 47, no. 4, 2002, pp. 625 – 631.

[30] M. Ichise, Y. Nagayanagi, and T. Kojima: An analog simulation of non-integer order transfer functions for analysis of electrode processes, *J. Electroanal. Chem.*, vol. 33, 1971, pp. 253–265.

[31] H. E. Jones and B. A. Shenoi: Maximally flat lumped-element approximation to fractional operator immitance function, *IEEE Trans. on Circuit and Systems*, vol. 17, no. 1, 1970, pp. 125–128.

[32] W. B. Jones and W. J. Thron: *Continued Fractions: Analytic Theory and Applications*, Addison-Wesley, Reading, 1980. (Russian translation: W. B. Jones and W. J. Thron: *Nepreryvnye drobi: analiticheskaya teoriya i prilozhenia*, Mir, Moscow, 1985).

[33] A. N. Khovanskii: *Prilozhenie tsepnykh drobei i ikh obobshchenii k voprosam priblizhennogo analiza*, Gostekhizdat, Moscow, 1956 (in Russian). (English translation: A. N. Khovanskii, *The Application of Continued Fractions and Their Generalizations to Problems in Approximation Theory*, Noordhoff, Groningen, 1963).

[34] P. Lanusse, V. Pommier, and A. Oustaloup. Fractional control system design for a hydraulic actuator. In *Proc. of the 1 First IFAC conference on Mechatronics systems, Mechatronic 2000*, Darmstadt, Germany, September 2000.

[35] J. Leszczynski and M. Ciesielski: A numerical method for solution of ordinary differential equation of fractional order, *Parallel Processing and Applied Mathematics Conference*, Springer-Verlag, 2001.

[36] A. Leva. PID autotuning algorithm based on relay feedback. *IEEE Proc. Part-D*, 140(5):328–338, 1993.

[37] CH. Lubich: Discretized fractional calculus. *SIAM J. Math. Anal.*, vol. 17, no. 3, 1986, pp. 704–719.

[38] B. J. Lurie: Three-Parameter Tunable Tilt-Integral-Derivative (TID) Controller, *United States Patent, 5 371 670*, USA, 1994.

[39] P. A. Lynn and W. Fuerst: *Digital Signal Processing with Computer Applications*, John Wiley & Sons, New York, 1994.

[40] J. A. T. Machado: Analysis and design of fractional-order digital control systems, *J. Syst. Anal. Modeling-Simulation*,vol. 27, 1997, pp. 107–122.

[41] J. A. T. Machado: Discrete-time fractional-order controllers, *Fractional Calculus & Applied Analysis*, vol. 4, no. 1, 2001, pp. 47–66.

[42] S. Manabe: The Non-Integer Integral and its Application to Control Systems. *ETJ of Japan*, vol. 6, no. 3-4, 1961, pp. 83–87.

[43] F. Mainardi: Fractional Calculus: Some Basic Problems in Continuum and Statistical Mechanics. *CISM Lecture Notes*, Udine, Italy, 1996.

[44] B. Mbodje and G. Montseny: Boundary Fractional Derivative Control of the Wave Equation. *IEEE Transactions on Automatic Control*, vol. 40, no. 2, 1995. pp. 378–382.

[45] D. Matignon. *Generalized Fractional Differential and Difference Equations: Stability Properties and Modelling Issues*. Proc. of Math. Theory of Networks and Systems Symposium, Padova, Italy, 1998.

[46] K. Matsuda and H. Fujii: H_∞–optimized wave-absorbing control: analytical and experimental results, *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 6, 1993, pp. 1146–1153.

[47] A. Le Mehaut, J. A. Tenreiro Machado, J. C. Trigeassou, and J. Sabatier, editors. *Proceedings of The First IFAC Symposium on Fractional Differentiation and its Applications (FDA04)*, Bordeaux, France, July 19-21 2004. IFAC, Elsevier Science Ltd., Oxford, UK.

[48] K. S. Miller and B. Ross. *An Introduction to the Fractional Calculus and Fractional Differential Equations*. John Wiley & Sons. Inc., New York, 1993.

[49] Concepción Alicia Monje Micharet. *Design Methods of Fractional Order Controllers for Industrial Applications*. PhD thesis, University of Extremadura, Spain, 2006.

[50] C. A. Monje, B. M. Vinagre, Y. Q. Chen, V. Feliu, P. Lanusse, and J. Sabatier. Proposals for fractional $PI^\lambda D^\mu$ tuning. In *Proceedings of The First IFAC Symposium on Fractional Differentiation and its Applications (FDA04)*, Bordeaux, France, 2004.

[51] Concepción A. Monje, YangQuan Chen, Blas Vinagre, Dingyü Xue and Vincente Fileu. "Fractional Order Controls - Fundamentals and Applications". Springer-Verlag London, Advances in Industrial Control series, (Invited book project, to be published in 2009)

[52] Concepción A. Monje, Blas M. Vinagre, Vicente Feliu, and YangQuan Chen. "Tuning and Auto-Tuning of Fractional Order Controllers for Industry Applications". IFAC Journal of Control Engineering Practice, Volume: 16 Issue: 7 Pages: 798-812 Year: July 2008.

[53] C. A. Monje, B. M. Vinagre, A. J. Calderón, V. Feliu and Y. Q. Chen. *Self-tuning of Fractional Lead-Lag Compensators*, Prague, Czech, July 4-8 2005. IFAC World Congress.

[54] G. Montseny, J. Audounet and B. Mbodje: Optimal models of fractional integrators and applications to systems with fading memory. *Proceedings of the Conference IEEE Systems, Man and Cybernetics*, Le Touquet, France, 1993, pp. 65–70.

[55] M. Moshrefi–Torbati and J. K. Hammond: Physical and geometrical interpretation of fractional operators, *Journal of Franklin Institute*, vol. 335B, no. 6, 1998, pp. 1077–1086.

[56] M. Nakagava and K. Sorimachi: Basic characteristics of a fractance device, *IEICE Trans. fundamentals*, vol. E75 - A, no. 12, 1992, pp. 1814–1818.

[57] S. Nimmo, A. K. Evans: The Effects of Continuously Varying the Fractional Differential Order of Chaotic Nonlinear Systems, *Chaos, Solitons & Fractals*, vol. 10, no. 7, 1999, pp. 1111–1118.

[58] K. B. Oldham and J. Spanier: *The Fractional Calculus*. Academic Press, New York, 1974.

[59] K. B. Oldham and C. G. Zoski: Analogue instrumentation for processing polarographic data, *Journal of Electroanal. Chem.*, vol. 157, 1983, pp. 27–51.

[60] A. Oustaloup, editor. *Proceedings of The Second IFAC Symposium on Fractional Differentiation and its Applications (FDA06)*, Porto, Portugal, July 19-21 2006. IFAC, Elsevier Science Ltd., Oxford, UK.

[61] A. Oustaloup: *La Dérivation non Entière*. Hermes, Paris, 1995.

[62] A. Oustaloup, F. Levron, B. Mathieu, and F. M. Nanot: Frequency-band complex noninteger differentiator: characterization and synthesis,

IEEE Trans. on Circuit and Systems - I: Fundamental Theory and Application, vol. 47, no. 1, 2000, pp. 25–39.

[63] A. Oustaloup: *Systèmes Asservis Linéaires d'Ordre Fractionnaire: Théorie et Pratique*. Editions Masson, Paris (1983).

[64] A. Oustaloup, P. Melchoir, P. Lanusse, C. Cois, and F. Dancla. The CRONE toolbox for Matlab. In *Proceedings of the 11th IEEE International Symposium on Computer Aided Control System Design - CACSD*, Anchorage, USA, September 2000b.

[65] A. Oustaloup, B. Mathieu, and P. Lanusse. The CRONE control of resonant plants: application to a flexible transmission. *European Journal of Control*, 1(2), 1995.

[66] A. Oustaloup, X. Moreau, and M. Nouillant. The CRONE suspension. *Control Engineering Practice*, 4(8):1101–1108, 1996.

[67] I. Petráš: The fractional-order controllers: methods for their synthesis and application, *J. of Electrical Engineering*, vol. 50, 1999, no. 9-10, pp. 284–288.

[68] I. Petráš and Ľ. Dorčák: The Frequency Method for Stability Investigation of Fractional Control Systems, *Journal of SACTA*, vol. 2, no. 1-2, 1999, pp. 75–85.

[69] I. Petráš and Š. Grega: Digital fractional order controllers realized by PIC microprocessor: Experimental results, *Proceedings of the ICCC2003*, High Tatras, Slovak Republic, May 26-29, pp. 873–876.

[70] I. Petráš, B. M. Vinagre, Ľ. Dorčák and V. Feliu: Fractional digital control of a heat solid: Experimental results, *Proc. of the ICCC'02*, Malenovice, Czech Republic, May 27 - 30, 2002, pp. 365 – 370.

[71] I. Petráš, Ľ. Dorčák, I. Podlubny, J. Terpák, and P. O'Leary: Implementation of fractional-order controllers on PLC B&R 2005, *Proceedings of the ICCC2005*, Miskolc-Lillafured, Hungary, May 24-27, pp. 141–144.

[72] I. Petráš, I. Podlubny, P. O'Leary, Ľ. Dorčák, B.M. Vinagre: *Analogue Realizations of Fractional Order Controllers*, Faculty BERG, TU Kosice, 2002, p. 84, ISBN 80-7099-627-7.

[73] I. Petráš, Ľ. Dorčák: Fractional-Order Control Systems: Modelling and Simulation, *Fractional Calculus and Applied Analysis, An International Journal for Theory and Applications*, vol. 6, no. 2, 2003, pp. 205 – 232.

[74] I. Podlubny: *The Laplace Transform Method for Linear Differential Equations of the Fractional Order*, UEF-02-94, Slovak Acad. Sci., Kosice, 1994. http://xxx.lanl.gov/abs/funct-an/9710005/

[75] I. Podlubny: *Fractional-Order Systems and Fractional-Order Controllers*, UEF-03-94, Slovak Acad. Sci., Kosice, 1994.

[76] I. Podlubny: *Fractional Differential Equations*. Academic Press, San Diego, 1999.

[77] I. Podlubny. Fractional-order systems and $PI^\lambda D^\mu$-controllers. *IEEE Transactions on Automatic Control*, vol. 44, no. 1, 1999, pp. 208–214.

[78] I. Podlubny. Matrix approach to discrete fractional calculus, *Fractional Calculus & Applied Analysis*, vol. 3, no. 4, 2000, pp. 359–386.

[79] I. Podlubny. Geometric and Physical Interpretation of Fractional Integration and Fractional Differentiation. *Fractional Calculus and Applied Analysis*, vol. 5, no. 4, pp. 367-386, 2002.

[80] I. Podlubny, I. Petráš, B. M. Vinagre, P. O'Leary, Ľ. Dorčák: Analogue Realization of Fractional-Order Controllers. *Nonlinear Dynamics*, Kluwer Academic Publishers, vol. 29, no. 1 - 4, July - September, 2002, pp. 281–296.

[81] I. Podlubny, A. V. Chechkin, T. Skovranek, Y. Q. Chen, B. Vinagre: Matrix approach to discrete fractional calculus II: partial fractional differential equations. *Journal of Computational Physics*, vol. 228, no. 8, 1 May 2009, pp. 3137–3153.

[82] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery: *Numerical Recipes in C. The Art of Scientific Computing*, 2nd edition, Cambridge University Press, 1992.

[83] H. F. Raynaud and A. Zergaïnoh. State-space representation for fractional order controllers. *Automatica*, 36:1017–1021, 2000.

[84] S. G. Samko, A. A. Kilbas and O. I. Maritchev. *Integrals and Derivatives of the Fractional Order and Some of Their Applications*. Nauka i Tekhnika, Minsk, 1987 (in Russian).

[85] M. Sugi, Y. Hirano Y., Y. F. Miura, K. Saito: Simulation of fractal immittance by analog circuits: an approach to the optimized circuits, *IEICE Trans. Fundamentals*, vol. E82–A, no. 8, 1999, pp. 1627–1635.

[86] M. Sugi, Y. Hirano Y., Y. F. Miura, K. Saito: $\sqrt{\omega}$–Variation of AC admittance in the inhomogeneously distributed RC lines, *Jpn. J. Appl. Phys.*, vol. 39, no. 9A, 2000, pp. 5367–5368.

[87] H. H. Sun, A. A. Abdelwahab, and B. Onaral: Linear approximation of transfer function with a pole of fractional power, *IEEE Trans. on Automatic Control*, vol. AC-29, no. 5, May 1984, pp. 441–444.

[88] Kok Kiong Tan, Wang Qing-Guo, Hang Chang Chieh, and Tore Hagglund. *Advances in PID Controllers*. Advances in Industrial Control. Springer-Verlag, London, 2000.

[89] P. J. Torvik and R. L. Bagley. On the Appearance of the Fractional Derivative in the Behavior of Real Materials. *Transactions of the ASME*, vol. 51, 1984, pp. 294–298.

[90] A. Tustin, J. T. Allanson, J. M. Layton and R. J.Jakeways: The Design of Systems for Automatic Control of the Position of Massive Objects, *The Proceedings of the Institution of Electrical Engineers*, 105C (1), 1958.

[91] B. M. Vinagre, I. Podlubny, A. Hernandez, and V. Feliu: On realization of fractional-order controllers. *Conference Internationale Francophone d'Automatique*, Lille, Jule 5-8, 2000, pp. 945–950.

[92] B. M. Vinagre, I. Podlubny, A. Hernández, and V. Feliu: Some approximations of fractional order operators used in control theory and applications, *Fractional Calculus & Applied Analysis*, vol. 3, no. 3, 2000, pp. 231–248.

[93] B. M. Vinagre, I. Podlubny, Ľ. Dorčák and V. Feliu: On Fractional PID Controllers: A Frequency Domain Approach, *IFAC Workshop on Digital Control - PID'00*, Terrassa, Spain, 2000.

[94] B. M. Vinagre: Modelado y Control de Sistemas Dinamicos Caracterizados por Ecuaciones Integro-Diferenciales de Orden Fraccional, PhD Thesis, 2001.

[95] B. M. Vinagre, I. Petráš, P. Merchan and Ľ. Dorčák: Two digital realizations of fractional controllers: Application to temperature control of a solid, *Proc. of the ECC'01*, Porto, Portugal, September 4 - 7, 2001, pp. 1764 – 1767.

[96] B. M. Vinagre, Y. Q. Chen, I. Petráš: Two Direct Tustin Discretization Methods for Fractional-Order Differentiator/Integrator, *Journal of The Franklin Institute*, vol. 340, no.5, 2003, pp. 349–362.

[97] B. M. Vinagre, C. A. .Monje, A. J. Calderon, J. I. Suarez: Fractional PID Controllers for Industry Application. A Brief Introduction, *Journal of Vibration and Control*, vol. 13, no.910, 2007, pp. 14191429.

[98] J. C. Wang: Realizations of generalized warburg impedance with RC ladder networks and transmission lines, *J. of Electrochem. Soc.*, vol. 134, no. 8, August 1987, pp. 1915–1920.

[99] S. Westerlund and L. Ekstam: Capacitor theory, *IEEE Trans. on Dielectrics and Electrical Insulation*, vol. 1, no. 5, Oct 1994, pp. 826–839.

[100] D. Xue and Y. Q. Chen: A Comparative Introduction of Four Fractional Order Controllers, *Proceedings of the 4th World Congress on Intelligent Control and Automation*, June 10 - 14, Shanghai, China, 2002.

[101] Dingyu Xue and YangQuan Chen. "Solving Advanced Applied Mathematical Problems Using Matlab". Taylor and Francis CRC Press. 2008 (448 pages in English, ISBN-13: 978-1420082500.)

[102] Dingyu Xue, YangQuan Chen and Derek Atherton. "Linear Feedback Control - Analysis and Design with Matlab". SIAM Press, 2007, ISBN: 978-0-898716-38-2. (348 pages) Chapter-8: Fractional-order Controller - An Introduction.

[103] S. Yamamoto and I. Hashimoto. Recent status and future needs: The view from Japanese industry. In Arkun and Ray, editors, *Proceedings of the fourth International Conference on Chemical Process Control*, Texas, 1991. Chemical Process Control – CPCIV.

[104] Cheng-Ching Yu. *Autotuning of PID Controllers: Relay Feedback Approach*. Advances in Industrial Control. Springer-Verlag, London, 1999.

[105] Chunna Zhao. *Research on Analyse and Design Methods of Fractional Order System*. PhD thesis, Northeastern University, China, 2006.

[106] C. N. Zhao and D. Xue. Closed-form solutions to fractional-order linear differential equations. *Frantiers of Electrical and Electronic Engineering in China*, 3(2):214–217, 2008.