Fitting of experimental data using Mittag-Leffler function

Igor Podlubny, Ivo Petráš, Tomáš Škovránek
Institute of Control and Informatization of Production Processes
BERG Faculty, Technical University of Kosice
B. Nemcovej 3, 04200 Kosice, Slovakia
Email: igor.podlubny@tuke.sk

***Abstract*—For fitting the measured data, a new approach is suggested, which is based on using the Mittag-Leffler function and which means that the data are fitted by a solution of an initial-value problem for a two-term fractional differential equation.**

I. Introduction

Fitting experimental data is, undoubtedly, the most important step on the way to good models of considered processes for modeling and control purposes.

Formulation of basic laws of physics, chemistry, electrotechnics, and other fields of science, normally starts from experimental observations. An experiment is set up, data are collected from the experiment, and a researcher postulates a hypothesis based on his assessment of those experimental data. Such assessment can be intuitive or exact; nowadays, researchers usually pre-process the collected data in order to remove artifacts, outliers, noise, or other disturbances, and then use some simple function $y(t) = f(t, params)$, where t is the independent variable and $params$ are model parameters, in order to describe the results analytically. The main problem is to find such a set of parameters $params$, which gives satisfactory agreement between the experimental data and the fitting function $y(t)$.

A good choice of the fitting function $y(t)$ with appropriate number of parameters (having less parameters is better) can serve as a general model for a wide class of objects or processes.

Let us recall a historical example of the discovery of Hooke's law in the theory of elasticity. Experimenting with elongation of various kinds of elastic materials led Robert Hooke to the observation that for small deformations the stress σ in the material due to its deformation is approximately proportional to the deformation ε:

$$\sigma \sim \varepsilon$$

It is well known that he formulated this observation in the form of an anagram, which after decoding sounds as "Ut tensio, sic vis" ("As the extension, so the force"). In other words, he suggested the model which we now call the Hooke's law:

$$\sigma = E\,\varepsilon,$$

where E (which later got the name of the modulus of elasticity or the Young modulus) is a constant depending on a particular material; taking different materials, we obtain different values of the parameter E.

II. Basic Notions of the Fractional-Order Calculus

Fractional calculus is more than 300 years old topic, which during recent decades became a powerful and widely used tool for better modeling and control of processes in many fields of science and engineering [1], [2], [3], [4], [5]. The term "fractional calculus" has some historical background and is used for denoting the theory of integration and differentiation of arbitrary real (not necessarily integer) order.

The standard notation for denoting the left-sided fractional-order differentiation of a function $f(t)$ defined in the interval $[a, b]$ is ${}_aD_t^\alpha f(t)$, with $\alpha \in$ R. Sometimes a simplified notation $f^{(\alpha)}(t)$ or $d^\alpha f(t)/dt^\alpha$ is used. In some applications also right-sided fractional derivatives ${}_tD_b^\alpha f(t)$ are used, but in the present article we will use only left-sided fractional derivatives. Even from the notation one can see that evaluation of the left-sided fractional-order operators require the values of the function $f(t)$ in the interval $[a, t]$. When α becomes an integer number, this interval shrinks to the vicinity of the point t, and we obtain the classical integer-order derivatives as particular cases.

There are several definitions of the fractional derivatives and integrals, of which we need only the following two.

The Caputo definition of fractional differentiation can be written as [1]:

$$ {}_a^C D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \qquad (1)$$
$$(n-1 \le \alpha < n)$$

where $\Gamma(z)$ is Euler's gamma function.

Above Caputo definition is extremely useful in the time domain studies, because the initial conditions for the fractional-order differential equations with the Caputo derivatives can be given in the same form as for the integer-order differential equations. This is an advantage in applied problems, which require the use of initial conditions containing starting values of the function and its integer-order derivatives $f(a)$, $f^{'}(a)$, $f^{''}(a)$, …, $f^{(n-1)}(a)$.

978-1-4577-1868-7/12/$26.00 ©2012 IEEE

The formula for the Laplace transform of the Caputo fractional derivative (1) has the form [1]:

$$\int_0^{\infty} e^{-st}\, {}_0^C D_t^{\alpha} f(t)\, dt = s^{\alpha} F(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} f^{(k)}(0), \quad (2)$$

$$(n-1 \le \alpha < n).$$

If the process $f(t)$ is considered from the state of absolute rest, so $f(t)$ and its integer-order derivatives are equal to zero at the starting time $t = 0$, then the Laplace transform of the α-th derivative of $f(t)$ is simply $s^{\alpha} F(s)$.

III. The Mittag-Leffler function

As it is obvious from its name, the Mittag-Leffler function was introduced by G. M. Mittag-Leffler. This function is a generalization of exponential function, and it plays in the fractional-order calculus the same fundamental role as the exponential function plays in the classical integer-order calculus and integer-order differential equations. Many known functions, which we used to consider as different, are, in fact, just particular cases of the Mittag-Leffler function.

IV. Data fitting using the Mittag-Leffler function

In order to provide a tool for quick and easy creation of models of arbitrary real (integer and non-integer) order, we have developed a new approach to data fitting, which is based on using the Mittag-Leffler function.

The idea of our method is based on the following. When it comes to obtaining a mathematical models from measurements or observations, it is a common practice in many fields of science and engineering to choose the type of the fitting curve and identify its parameters using some criterion (usually a least squares method). We would like to point out that choosing a particular type of a curve means that, in fact, the process is modeled by a differential equation, for which that curve is a solution.

For example, fitting data using the equation $y(t) = at + b$ (known as linear regression model) means that the process is modeled by the solution of a simple second-order differential equation under two initial conditions:

$$y'' = 0, \quad y(0) = b, \quad y'(0) = a. \quad (3)$$

Similarly, the fitting function in the form $y = a\sin(\omega t) + b\cos(\omega t)$ means that the process is modeled by the solution of the initial value problem of the form

$$y'' + \omega^2 y = 0, \quad y(0) = b, \quad y'(0) = a\omega. \quad (4)$$

Choosing the fitting function in another frequently used form, $y = ae^{bt}$, means that the process is modeled by the solution of the initial value problem

$$y' - b\,y = 0, \quad y(0) = a. \quad (5)$$

Thinking in this way, we conclude that instead of postulating the shape of the fitting curve it is possible to postulate the form of the initial-value problem and identify the parameters appearing in the differential equation and in the initial conditions. For the first time this method was suggested and used in [1, Chapter 10]. We would like to emphasize that obtaining a fitting function $y(t)$ for measurements of a dynamic process immediately means that that process is described by an initial-value problem of which $y(t)$ is the solution.

Suppose that the measured data are fitted by

$$y = y_0\, E_{\alpha,1}(a\, t^{\alpha}) \quad (6)$$

where $E_{\alpha,\beta}(z)$ is the Mittag-Leffler function defined as [1]

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}. \quad (7)$$

The parameters to be identified are α, a, and y_0.

For example, If the data are fitted by the function (6), then this means that they are modeled by the solution of the following initial-value problem for a two-term fractional-order differential equation containing the Caputo fractional derivative of order α:

$${}_0^C D_t^{\alpha} y(t) - a\, y(t) = 0, \quad y(0) = y_0. \quad (8)$$

V. Examples

The proposed method of fitting is illustrated below on several examples, which includes "restoration" of the Mittag-Leffler function from its noised values, fitting a complimentary error function, fitting a sine wave, and fitting damped oscillations.

A. Fitting back the noised Mittag-Leffler function:

The first example is the "restoration" of the function $y(x) = 0.8E_{1.5}(-0.2x^{1.5})$. Three series of "measured" data are created by adding noise to the values of the Mittag-Leffler function at the same set of nodes x. Such noisy data are fitted using MLFFIT1.M function.

```
% Define the set of nodes (x)
% and the parameter $\alpha$ of the
% Mittag-Leffler function:
x = 0:0.35:20;
alfa = 1.5;

% Since for computing the one-parameter
% Mittag-Leffler function
% we call the Matlab function for computing
% the two-parameter Mittag-Leffler function,
% the second parameter is equal to 1:
beta = 1;

% Now let us simulate measurements
% by adding noise to the exact values
% of the original function
y1 = 0.8*mlf(alfa, beta, -0.2*x.^alfa, 6) ...
+ (-.05 + .1*rand(size(x)));
y2 = 0.8*mlf(alfa, beta, -0.2*x.^alfa, 6) ...
+ (-.05 + .1*rand(size(x)));
y3 = 0.8*mlf(alfa, beta, -0.2*x.^alfa, 6) ...
+ (-.05 + .1*rand(size(x)));

% and fit these "measurements"
% by calling MLFFIT1:
[c, R2] = mlffit1([x x x], [y1 y2 y3], ...
```

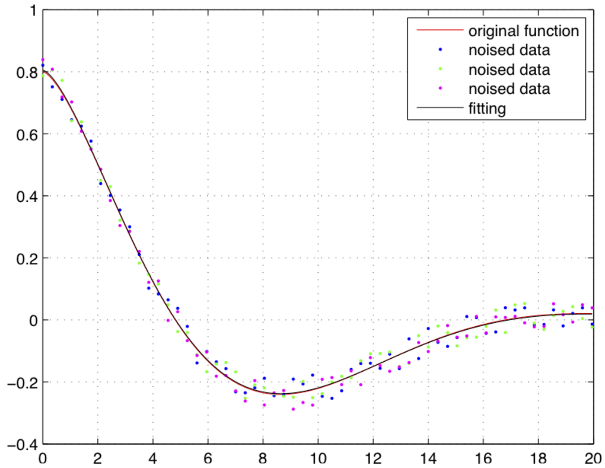

Fig. 1. Restoring the function $y(x) = 0.8E_{1.5}(-0.2x^{1.5})$

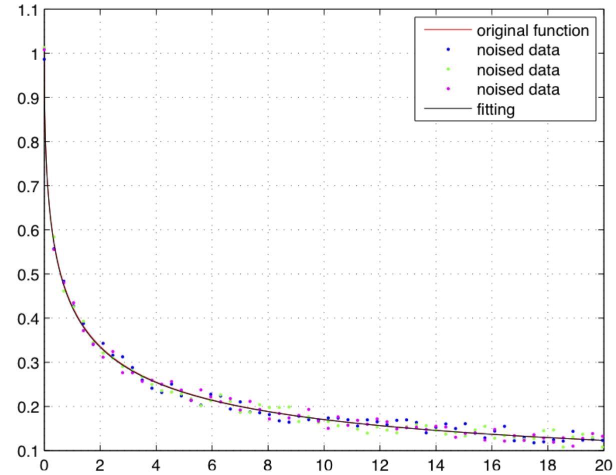

Fig. 2. Restoring the function $y(x) = e^{x}erfc(\sqrt{x})$

```
[0.5; 0.5; 0.5; -0.5], 6);

% Let us check if the coefficients
% of the fitting Mittag-Leffler function
% are close to the original coefficients:
alpha = c(1)
C = c(3)
a = c(4)

% Finally, we can plot the "measurements",
% the original function and the function
% that fits the "measurement":

xfine = x(1):0.01:x(end);
yfit = c(3)*mlf(c(1),c(2),c(4)*xfine.^c(1),6);
yorig = 0.8*mlf(alfa,beta,-0.2*xfine.^alfa,6);

figure(1)
plot(xfine, yorig, 'r', x,y1, '.b', x, y2, '.g', ...
x, y3, '.m', xfine, yfit, 'k')
grid on
legend('original function', 'noised data', ...
'noised data', 'noised data', 'fitting')
```

The output of the above code was

$$\alpha = 1.4949, \quad C = 0.8053, \quad a = -0.2008,$$

which is very close to the values of these parameters for the original function $y(x) = 0.8E_{1.5}(-0.2x^{1.5})$. The noised data, and the fitting curve are shown in Fig. 1.

B. Fitting the classics (complementary error function):

Let us "restore" the following function: $y(x) = e^{x}erfc(\sqrt{x})$. it should be mentioned that this function can be written as $y(x) = E_{1/2,1}(-x^{1/2})$.

```
% Define the set of nodes (x)
x = 0:0.35:20;

% Now let us simulate measurements
% by adding noise to  the exact values
% of the original function
y1 = exp(x).*erfc(sqrt(x))  ...
+ (-.02 + .04*rand(size(x)));
y2 = exp(x).*erfc(sqrt(x)) ...
 + (-.02 + .04*rand(size(x)));
y3 = exp(x).*erfc(sqrt(x)) ...
 + (-.02 + .04*rand(size(x)));

% and fit these "measurements"
% by calling MLFFIT1:
[c, R2] = mlffit1([x x x], [y1 y2 y3], ...
[0.5; 0.5; 0.5; -0.5], 6);

% Let us output the coefficients
% of the fitting Mittag-Leffler function:
alpha = c(1), C = c(3), a = c(4)

% Finally, we can plot the "measurements",
% the original function and the function
% that fits the "measurement":
xfine = x(1):0.01:x(end);
yfit = c(3)*mlf(c(1),c(2),c(4)*xfine.^c(1),6);
yorig = exp(xfine).*erfc(sqrt(xfine));

plot(xfine, yorig,'r',x,y1,'.b',...
x,y2,'.g',x,y3,'.m',xfine,yfit,'k')
grid on
legend('original function','noised data',...
'noised data','noised data','fitting')
```

The output of the above code was

$$\alpha = 0.4966, \quad C = 1.0028, \quad a = -1.0175,$$

which is very close to the values of these parameters for the original function $y(x) = e^{x}erfc(\sqrt{x}) = E_{1/2,1}(-x^{1/2})$. The noised data, and the fitting curve are shown in Fig. 2.

C. Dumped oscillation fitting:

Let us test if the Mittag-Leffler function is abe to fit dumped oscillations: $y(x) = e^{(} - \alpha x)\cos(x)$.

```
% Define the dumping coefficient:
alfa = 0.2;

% Define the set of nodes (x)
x = 0:0.35:20;

% Now let us simulate measurements by adding noise to
% the exact values of the original function
y1 = exp(-alfa*x).*cos(x)+(-.05+.1*rand(size(x)));
y2 = exp(-alfa*x).*cos(x)+(-.05+.1*rand(size(x)));
y3 = exp(-alfa*x).*cos(x)+(-.05+.1*rand(size(x)));

% and fit these "measurements" by calling MLFFIT1:
[c, R2] = mlffit1([x x x], [y1 y2 y3], ...
[0.5; 0.5; 0.5; -0.5], 6);

% Let us output if the coefficients
% of the fitting Mittag-Leffler function:
```

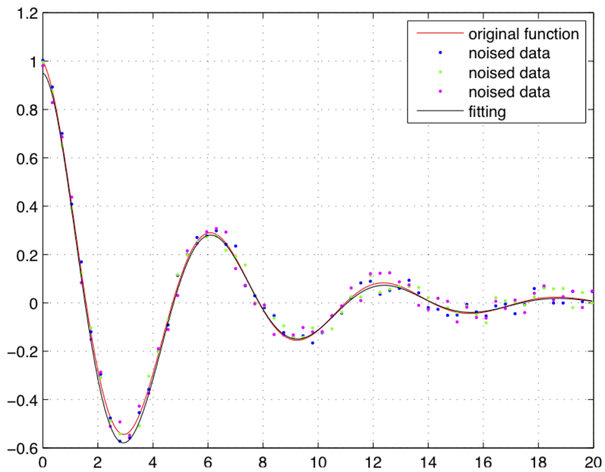

Fig. 3. Fitting damped oscillations $y(x) = e^(-0.2\,x)\cos(x)$

```
alpha = c(1), C = c(3), a = c(4)

% Finally, we can plot the "measurements",
% the original function and the function
% that fits the "measurement":
xfine = x(1):0.01:x(end);
yfit = c(3)*mlf(c(1),c(2),c(4)*xfine.^c(1),6);
yorig = exp(-alfa*xfine).*cos(xfine);

plot(xfine,yorig,'r',x,y1,'.b',x,y2,'.g', ...
x,y3,'.m',xfine,yfit,'k')
legend('original function','noised data',...
'noised data','noised data','fitting')
```

The output of the above code was

$$\alpha = 1.7631, \quad C = 0.9495, \quad a = -1.0340.$$

The noised data, and the fitting curve are shown in Fig. 3.

VI. Conclusion

The Mittag-Leffler function can be used as a universal fitting function, which is capable of capturing the behavior of various types of processes, including such practically important cases as monotonic processes, oscillatory behavior, and damped oscillations. There is no need in postulating a narrow type of the fitting function anymore; the MIttag-Leffler function flexibly uncovers the nature of the fitted data. The broad field of potential applicaitons the proposed approach was applied recently for providing the first example of identification of variable-order systems [8].

In addition, as soon as the data are fitted with the help of the Mittag-Leffler function, the process can be described by a two-term fractional-order differential equation. This is an important advantage of the proposed approach to fitting experimental data – it opens a way to creating standard models of elementary processes.

Acknowledgment

The authors would like to acknowledge partial support from grants VEGA 1/0497/11, SK-PT-0025-10, SK-AT-0024-10, and SK-FR-0037-09.

References

[1] I. Podlubny, *Fractional Differential Equations*, San Diego: Academic Press, 1999.
[2] K. B. Oldham and J. Spanier, *The Fractional Calculus*, New York: Academic Press, 1974.
[3] R. L. Magin, *Fractional Calculus in Bioengineering*, Connecticut: Begell House Publishers, 2006.
[4] C. A. Monje, Y. Q. Chen, B. M. Vinagre, D. Xue and V. Feliu, *Fractional-order Systems and Controls*, Series: Advances in Industrial Control, Springer, 2010.
[5] R. Caponetto, G. Dongola, L. Fortuna, and I. Petráš, *Fractional Order Systems: Modeling and Control Applications*, Singapore: World Scientific, 2010.
[6] I. Podlubny. *Fitting data using the Mittag-Leffler function*. Matlab Central File Exchange, submission #32170, 11 Jul 2011 (Updated 19 Jul 2011)
http://www.mathworks.com/matlabcentral/fileexchange/32170
[7] I. Podlubny. *Mittag-Leffler function*. Matlab Central File Exchange, submission #8738, 17 Oct 2005 (Updated 25 Mar 2009)
http://www.mathworks.com/matlabcentral/fileexchange/8738
[8] D. Sierociuk, I. Podlubny, and I. Petras; , Experimental evidence of variable-order behavior of ladders and nested ladders, EEE Transactions on Control Systems Technology I, 2012, doi: 10.1109/TCST.2012.2185932 (early access).