OptimFOPID: A MATLAB Interface for Optimum Fractional-order PID Controller Design for Linear Fractional-order Plants ⋆

Dingyü Xue * **YangQuan Chen** **

* *School of Information Science and Engineering Northeastern University, Shenyang 110004, P R China (e-mail: xuedingyu@mail.neu.edu.cn).*
** *Department of Electrical & Computer Engineering Center for Self-Organizing & Intelligent Systems (CSOIS) Utah State University, Logan, UT 84322-4120, USA (e-mail: yqchen@ieee.org).*

Abstract: In the paper, a MATLAB based graphical user interface OptimFOPID is presented for designing optimum fractional-order PID-type controllers of different types, under different criteria, for linear fractional-order plants. Numerical optimization algorithms, including global optimization algorithms are embedded in the interface in designing the controllers. Illustrative examples of optimum fractional-order PID controller design are given to demonstrate the merit of optimum fractional-order controllers over the traditional ones.

Keywords: fractional-order control, PI$^\lambda$D$^\mu$ controller, optimum PID, optimal control, optimization, graphical user interface, global optimization, ITAE criterion, integral performance indices, MATLAB

1. INTRODUCTION

PID-type controllers are the most widely used controllers in process industry, there are a great amount of published algorithms and applications of PID controllers, see O'Dwyer (2003), Åström and Hägglund (1995), Johnson and Moradi (2005), Silva et al. (2005), and some of the tuning algorithms collected in the books are already adopted in real applications. To further improve the behavior of the conventional PID controllers, fractional-order PID controllers, proposed in Podlubny (1999b), can also be adopted.

The standard form of the fractional-order PID controller, also known as PI$^\lambda$D$^\mu$ controller, is expressed as

$$G_\mathrm{c} = K_\mathrm{p} + \frac{K_\mathrm{i}}{s^\lambda} + K_\mathrm{d} s^\mu \qquad (1)$$

where λ and μ are arbitrary real numbers. The fractional-order derivatives and integrals are defined with

$${}_a\mathscr{D}_t^\alpha f(t) == \lim_{h\to 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(t-a)/h]} (-1)^j \begin{pmatrix} \alpha \\ j \end{pmatrix} f(t-jh) \qquad (2)$$

where $w_j^{(\alpha)} = (-1)^j \begin{pmatrix} \alpha \\ j \end{pmatrix}$ is the jth coeffcient of the polynomial $(1-z)^\alpha$. The above definition is referred to as the Grünwald-Letnikov definition, and there are several other definitions in Podlubny (1999a); Monje et al. (2010).

There are many tuning algorithms for the fractional-order PID controllers such as Podlubny (1999b), however most of the algorithms are for particular types of plant models. For instance, the tuning algorithms for integer-order plants, see Li and Chen (2008); Chen et al. (2008); Chang and Lee (2008), and the tuning formula for some special fractional-order plants. However there is no to-date optimum parameter tuning algorithms for arbitrary linear fractional-order plants exist. In this paper, a MATLAB based graphical user interface, named OptimFOPID, is developed and presented for optimum fractional-order PID-type controller design for ordinary linear fractional-order plants. Integral performance indices are used, and the most meaningful criterion is recommended in servo control systems in Section 3. In Section 4, a brief tutorial and descriptions to OptimFOPID is given, based on the FOTF objects in Monje et al. (2010) for the plants described by linear fractional-order transfer functions. In Section 5, some illustrative examples are given to show the benefit of the proposed interface. Comparisons are made on the examples for fractional-order as well as conventional integer-order PID controllers. Global optimization tools are embedded in the interface and they can be invoked directly with extra toolboxes such as Global Optimization Toolbox in MathWorks Inc (2011), GAOT in Houck et al. (1995) and PSOt in Birge (2003), to ensure the best controllers to be found.

⋆ Supported financially by the National Natural Science Foundation China, under Grant 61174145.

2. VARIATIONS OF FRACTIONAL-ORDER PID CONTROLLERS

A typical fractional-order PID control framework for process system is shown in Fig. 1, where the plant model $G(s)$ is a fractional-order transfer function given by

$$G(s) = \frac{b_1 s^{\gamma_1} + b_2 s^{\gamma_2} + \cdots + b_m s^{\gamma_m}}{a_1 s^{\eta_1} + a_2 s^{\eta_2} + \cdots + a_{n-1} s^{\eta_{n-1}} + a_n s^{\eta_n}} \tag{3}$$

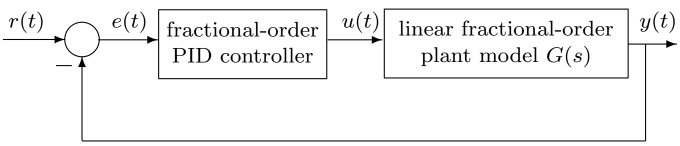
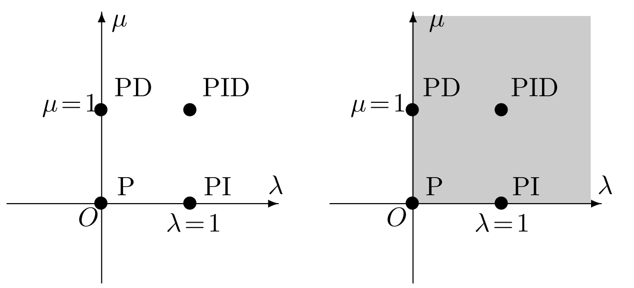


Fig. 1. Fractional-order PID control structure

The standard form of the fractional-order PID controller, denoted as $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$, is presented again

$$G_{\mathrm{c}}(s) = K_{\mathrm{p}} + \frac{K_{\mathrm{i}}}{s^{\lambda}} + K_{\mathrm{d}} s^{\mu} \tag{4}$$

where λ and μ are real, and typically $\lambda, \mu \in (0, 2)$. There are five tunable parameters in the controller. Compared with conventional-order PID controller, the extra two parameters λ and μ are demonstrated in Fig. 2, and it can be seen that the fractional-order PID controller has two more degrees-of-freedom than the conventional PID controller, which makes it more flexible. Conventional-order PID-type controllers are special cases of fractional-order PID-type controllers.



(a) conventional (b) fractional-order

Fig. 2. Comparison of fractional-order and conventional PID controllers

Some special cases of fractional-order PID-type controllers are also considered in the paper, as shown in Table 1.

Table 1. Variations of fractional-order PID-type controllers

notation	description	parameters
PI^{λ}	fractional-order PI	$K_{\mathrm{d}} = 0$
PD^{μ}	fractional-order PD	$K_{\mathrm{i}} = 0$
PID^{μ}	integer-order integral	$\lambda = 1$
$\mathrm{PI}^{\lambda}\mathrm{D}$	integer-order derivative	$\mu = 1$
PID	conventional PID controller	$\lambda = \mu = 1$

3. INTEGRAL PERFORMANCE INDICES

The target of fractional-order PID control is to tune the parameters of $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ such that the dynamic tracking error signal $e(t)$ as small as possible. Integral performance indices to the error signal $e(t)$ are often good choices to assess the behaviors of the tracking error signal $e(t)$. In particular, the following integral performance indices are used in the interface

$$I_1 = \int_0^{\infty} e^2(t)\mathrm{d}t, \quad I_2 = \int_0^{\infty} |e(t)|\mathrm{d}t, \quad I_3 = \int_0^{\infty} t|e(t)|\mathrm{d}t \tag{5}$$

$$I_4 = \int_0^{\infty} te^2(t)\mathrm{d}t, \quad I_5 = \int_0^{\infty} t^2 e^2(t)\mathrm{d}t, \quad I_6 = \int_0^{\infty} t^2|e(t)|\mathrm{d}t \tag{6}$$

The above criteria are abbreviated respectively as ISE, IAE, ITAE, ISTE, $\mathrm{IT}^2\mathrm{SE}$, and $\mathrm{IT}^2\mathrm{AE}$ criteria, and some of those are usually adopted in literatures.

It can be seen that in the ISE and IAE criteria, the values of the error signals $e(t)$ at any time instances are treated equally, whereas in the ITAE criterion, the value of error signal is penalized when the time t gets larger. That is to say that the ITAE criterion is more suitable for servo control problems, since the error signal is forced to settle down at zero as soon as possible. Later, an example will be given to demonstrate the advantages of the ITAE criterion.

Since ITAE criterion can only be evaluated with simulation approach, infinite integrals cannot be evaluated. Finite-time ITAE integral defined as

$$I_{\mathrm{FT-ITAE}} = \int_0^{t_{\mathrm{f}}} t|e(t)|\mathrm{d}t \tag{7}$$

can be used instead to approximate the ITAE criterion, if the finite time t_{f} is selected properly, since the error signal $|e(t)|$ may settle down at zero for larger t, such that the integral after t_{f} may have zero contribution to the total value of the ITAE integral.

4. OPTIMFOPID: GRAPHICAL USER INTERFACE

A MATLAB based graphical user interface, OptimFOPID, is developed and it can be used directly in optimum PID controllers design. The package can be downloaded from MathWorks' File Exchange web-site (See Xue (2012)). It can be unzipped to a folder and this folder should be added to the MATLAB search path. Once all these are done, the command `optimfopid` can be issued and the main interface shown in Fig. 3 can be displayed.

If the plant model to be controlled is a linear time invariant fractional-order transfer function in (3), and the class `fotf` in Monje et al. (2010) can be used to represent it. In the `fotf` package, the plant model can be expressed by four compatible vectors, representing respectively the coefficients and orders of the numerator and denominator pseudo polynomials. For instance, the fractional-order transfer function

$$G(s) = \frac{s^{0.4} + 5s^{0.8} + 2}{s^{1.5} + 1.2s^{0.7} + 1.8s^{0.4} + 5}$$

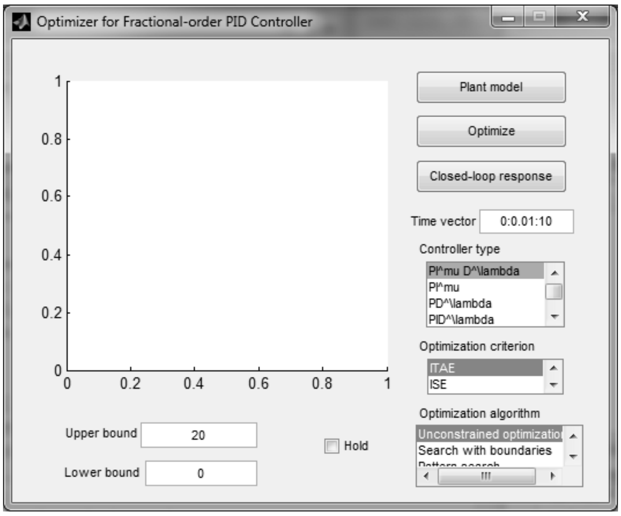


Fig. 3. The main interface of OptimFOPID

can be entered into MATLAB workspace by

```
>> G=fotf([1 1.2 1.8,5],[1.5 0.7 0.4 0],...
          [1 5 2],[0.4 0.8 0])
```

For the complicated fractional-order transfer functions

$$G(s) = \frac{1}{s^{0.2}(s^{0.5} + 0.8s^{0.2} + 1)(s^{0.2} + 3)}$$

the command `s=fotf('s')` can be used to declare the Laplace operator s first, Thus the following MATLAB command can be used to enter the fractional-order transfer function model

```
>> s=fotf('s');
   G=1/s^(0.2)/(s^0.5+0.8*s^0.2+1)/(s^0.2+3)
```

The above factorized model can be converted automatically to a standard FOTF object

$$G(s) = \frac{1}{s^{0.9} + 3s^{0.7} + 0.8s^{0.6} + 3.4s^{0.4} + 3s^{0.2}}$$

The essential procedures of optimum fractional-order PID controller design with OptimFOPID are:

(1) Using MATLAB command to specify the plant model as a FOTF object, in the variable G in MATLAB workspace, and then click the Plant model button to load the plant model into the interface.
(2) The finite time t_f or an evenly spaced time vector $\boldsymbol{t}$ should also be entered into the Time vector edit box.
(3) Click Optimize button to start the controller design process. The intermediate information will be displayed in the command window. After the search process, the controller can be returned on the MATLAB variable Gc, as a FOTF object.

Below are optional settings in controller design:

(1) The controller type such as $\mathrm{PI}^\lambda\mathrm{D}^\mu$, PI^λ, PD^μ, PID^μ, $\mathrm{PI}^\lambda\mathrm{D}$, and conventional PID controller can be selected from the Controller type listbox.
(2) The optimization algorithms can be selected from the listbox of Optimization algorithm. Apart from conventional optimization algorithms, global optimization algorithms such as genetic algorithm, particle swarm optimization and pattern search algorithm are also supported.
(3) Upper and lower bounds of the parameters K_p, K_i and K_d, where the orders λ and μ are assumed $\lambda, \mu \in (0, 2)$.
(4) Different criteria ITAE, ISE, IAE, and so on as defined in (4) and (5), can be adopted from the Optimization criterion listbox, with ITAE the recommended one.
(5) Unit step response of the closed-loop system under the designed controller can be obtained by clicking the Closed-loop response. If Hold checkbox is checked, the closed-loop of other controllers designed can be displayed in the same axis.

Besides, in order to design satisfactory controllers, the following key points should also be considered:

(1) The selection of the terminate time t_f is sometimes crucial in optimum controller design. Since the integrands in the performance indices are non-negative, the integrals are always nondecreasing functions. The curves will remain flat when the integrand, or the error signal $e(t)$, settles down at zero. The strategy of selecting t_f will be depicted through examples.
(2) Sometimes, the upper and lower bounds of controller parameters are also important, one may try different boundaries in the design process.
(3) The overload functions in the `@fotf` folder can be used to analyze the behaviors of the closed-loop system in MATLAB command window as well. Since the plant model and controller are denoted by the variables G and Gc, respectively, one can use the MATLAB commands `step(feedback(1,G*Gc),t)` and `step(feedback(G,Gc),t)` to draw the signals $e(t)$ and $u(t)$ in the system.

5. ILLUSTRATIVE EXAMPLES

In this section, several illustrative examples are presented. Commonly used plant models in literatures are adopted and optimal fractional-order PID controllers are designed.

[Example 1]: Consider the fractional-order model of the furnace given by Podlubny (1999a)

$$G(s) = \frac{1}{0.8s^{2.2} + 0.5s^{0.9} + 1}$$

The procedures in designing optimal controllers are

(1) The following statement can be used to enter the plant model
`>> G=fotf([0.8 0.5 1],[2.2 0.9 0],1,0)`
(2) Click the Plant model button to load the plant into the interface.
(3) Set the upper boundaries of the parameters to 15, and the terminate time to 8.
(4) Click Optimize button to start the optimization process.

The parameter vector obtained is

$$\boldsymbol{x} = [5.2440, 12.4485, 9.0996, 0.9835, 1.1954]$$

such that

$$
\begin{aligned}
G_{c1}(s) &= \frac{9.0996s^{2.1789}+5.244s^{0.98349}+12.4485}{s^{0.98349}} \\
&= 5.244+\frac{12.4485}{s^{0.98349}}+9.0996s^{1.1954}
\end{aligned}
$$

It can be seen that the order of integrator is very close to unity, the PID$^\mu$ controller can be selected from the list box and the Optimize button can be clicked again to find the PID$^\mu$ controller

$$G_{c2}(s)=8.342+\frac{14.9109}{s}+11.317s^{1.2622}$$

Also the PID item can be selected from the listbox to find the conventional PID controller

$$G_{c3}(s)=2.2685+\frac{15}{s}+10.4292s$$

The step responses under these controllers are shown in Fig. 4. It can be seen that the step responses under the PI$^\lambda$D$^\mu$ and PID$^\mu$ controllers are almost identical, and they are much better than the optimal conventional PID controller.

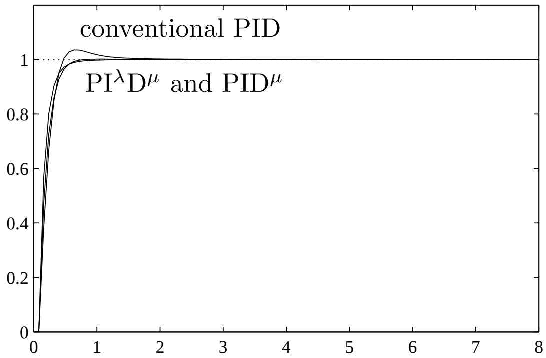


Fig. 4. Step responses under different controllers

[Example 2]: Consider another fractional-order transfer function

$$G(s)=\frac{1}{s^{2.6}+3.3s^{1.5}+2.9s^{1.3}+3.32s^{0.9}+1}$$

Again the procedures in designing the controllers with OptimFOPID interface are

(1) The plant model can be entered into MATLAB by
```
>> G=fotf([1 3.3 2.9 3.32 1],...
          [2.6 1.5 1.3 0.9 0],1,0);
```
and again, one can click the Plant model button to load the model into the interface.
(2) Select terminate time to 10
(3) Click Optimize button to start searching process.

The optimal PI$^\lambda$D$^\mu$ controller can be found

$$
\begin{aligned}
G_{c1}(s) &= \frac{10.0194s^{1.6029}+14.9959s^{1.0137}+2.5974}{s^{1.0137}} \\
&= 14.9959+\frac{2.5974}{s^{1.0137}}+10.0194s^{0.5892}
\end{aligned}
$$

The optimal PID$^\mu$ controller can be obtained as

$$G_{c2}(s)=14.9997+\frac{2.7843}{s}+10.9289s^{0.5795}$$

The conventional PID controller can also be obtained as

$$G_{c3}(s)=15.0000+\frac{2.0632}{s}+11.1509s$$

Under the three controllers, the closed-loop step responses are shown in Fig. 5. Again it can be seen that the quality of the two fractional-order controllers are almost identical and the system behaviors are satisfactory. While the optimal conventional PID controller is not as good.

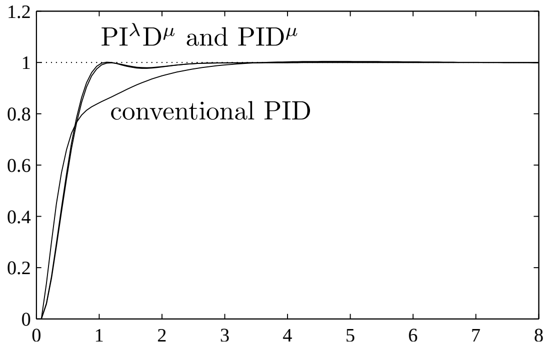


Fig. 5. Step responses under different controllers

It should be noted that the above optimal controllers are designed under the finite-time ITAE criterion. However if the ISE criterion is used instead, the optimal PI$^\lambda$D$^\mu$ controller can be obtained as

$$G_{c4}(s)=15+\frac{15}{s^{1.3\times10^{-5}}}+15s^{1.5766}$$

It almost equivalents a PD$^\mu$ controller. The parameters are on the boundaries of the pre-specified 15. Even though the value is increased to 1000, or other larger numbers, the controller parameters obtained will also be on the boundaries. This means that there is no optimal PI$^\lambda$D$^\mu$ controller for the plant model, when ISE criterion is used. Under the controller $G_{c4}(s)$, the step response of the closed-loop system is shown in Fig. 6, and it is very poor, compared with the controllers in Fig. 5. Since the integral action in ISE controller is very weak, there may exist steady-state errors in the output, at least when t is not extremely large.

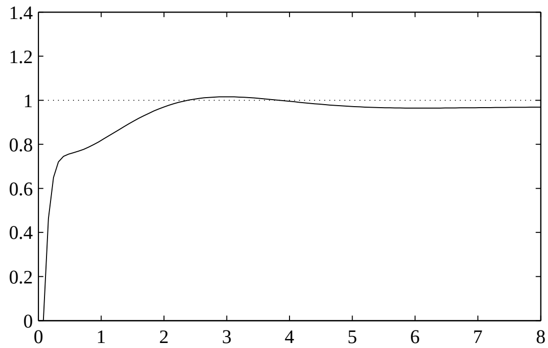


Fig. 6. Step response with ISE optimal PI$^\lambda$D$^\mu$ controller

[Example 3] For a fractional-order plant model

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

it will be very difficult to find optimal PID controllers, integer-order or fractional-order, using existing design algorithms. However with the use of numerical optimization techniques, the optimum PI$^\lambda$D$^\mu$ and conventional PID controllers can be obtained easily.

The following procedures can be used in searching for the optimal controllers.

(1) Specify the plant model as a FOTF object G
```
>> a=[2,3.8,2.6,2.5,1.5];
   na=[3.5001,2.42,1.798,1.31,0];
   G=fotf(a,na,[-2,-4],[0.63,0]);
```
(2) Load the plant model into the interface by clicking the Plant model button.
(3) Specify the terminate time to 4, or other value of user's choice, however this value should be validated after the controller design process.
(4) Set the upper and lower boundaries of controller parameters to 0 and −20, respectively.
(5) Click Optimize button to design the optimal PID$^{\mu}$ controller. For this example, we have

$$G_{c1}(s) = -20 - \frac{0.017852}{s} - 18.3742s^{0.6695}$$

and the conventional PID controller can also be found as

$$G_{c2}(s) = 8.5552 - \frac{8.5552}{s} - 9.3095s$$

The closed-loop step responses of the systems under the two controllers can be obtained as shown in Fig. 7. It can be seen that the behavior under the PID$^{\mu}$ controller is much much better than that under the optimum conventional PID controller.

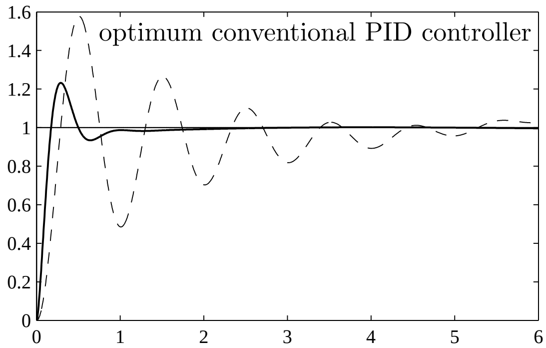


Fig. 7. Step responses with the two controllers

If conventional optimization problem solvers cannot find global optimal controllers, the Optimization algorithms listbox can be clicked, from which other global optimization solvers, including Genetic Algorithm Optimization Toolbox (GAOT), Particle Swarm Optimization Toolbox (PSOt), as well as MATLAB's Global Optimization Toolbox, in particular the simulated annealing algorithm and pattern search algorithm, can be selected.

6. CONCLUSIONS

In the paper, an optimum fractional-order PID controller design interface in MATLAB is presented. With such an interface, different types of optimum fractional-order PID controllers under different integral performance indices can be designed, when the plant model is a linear fractional-order model. The interface is easy to use, and may be used by inexperienced users directly. The ITAE performance index is recommended for the design, since it is more reasonable than the widely used ISE criterion. Global optimized controllers can be obtained with the use of relevant tools. The conclusions and advantages of the proposed OptimFOPID interface are:

(1) This interface is a numerical optimization based optimal controller design interface. It is applicable to any linear fractional-order plant models.
(2) Finite-time ITAE criterion is mainly studied and recommended, rather than the well accepted ISE and other criteria, since it is much better in describing time domain response behaviors. This criterion is meaningful in control applications, although the optimum controllers under other criteria can equally be obtained, if needed.
(3) Different optimization problem solvers are integrated in the interface, and genetic algorithm, particle swarm optimization, simulated annealing as well as pattern search algorithms may lead to global optimization results, which are likely to ensure stabilizing closed-loop behavior even for complicated plants.

REFERENCES

Åström, K.J. and Hägglund, T. (1995). *PID controllers: theory, design and tuning.* Research Triangle Park, Instrument Society of America.

Birge, B. (2003). PSOt, a particle swarm optimization toolbox for MATLAB. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, 182∼186. Indianapolis.

Chang, F.K. and Lee, C.H. (2008). Design of fractional PID control via hybrid of electromagnetism-like and genetic algorithms. *Eighth International Conference on Intelligent Systems Design and Applications*, 525∼530.

Chen, Y.Q., Bhaskaran, T., and Xue, D. (2008). Practical tuning rule development for fractional-order proportional and integral controller. *ASME Journal of Computational and Nonlinear Dynamics*, 3(2), 021403–1∼021403–8.

Houck, C.R., Joines, J.A., and Kay, M.G. (1995). *A genetic algorithm for function optimization: a MATLAB implementation.* Electronic version.

Johnson, M.A. and Moradi, M.H. (2005). *PID control — new identification and design methods.* Springer, London.

Li, H.S. and Chen, Y.Q. (2008). A fractional order proportional and derivative controller tuning algorithm. *Proceedings of Chinese Control and Decision Conference*, 4059∼4063.

MathWorks Inc (2011). *Global Optimization Toolbox user's manual.*

Monje, C., Chen, Y., Vinagre B.M., Xue D., and Feliu V. (2010). *Fractional-order systems and controls — fundamentals and applications.* Springer, London.

O'Dwyer, A. (2003). *Handbook of PI and PID controller tuning rules.* Imperial College Press, London.

Podlubny, I. (1999a). *Fractional differential equations.* Academic Press, San Diago.

Podlubny, I. (1999b). Fractional-order systems and PI$^{\lambda}$D$^{\mu}$ controllers. *IEEE Transactions on Automatic Control*, 44(1), 208∼214.

Silva, G.J., Datta, A., and Bhattacharyya, S.P. (2005). *PID controllers for time-delay systems.* Birkhäuser, Boston.

Xue, D. (2012). *OptimFOPID: an optimum fractional-order PID-type controller design interface.* MathWorks File-exchange. To be submitted.