SUBOPTIMUM $H_2$ PSEUDO-RATIONAL APPROXIMATIONS TO FRACTIONAL-ORDER LINEAR TIME INVARIANT SYSTEMS

Dingyü Xue¹ and YangQuan Chen²

¹ Institute of Artificial Intelligence and Robotics, Faculty of Information Science and Engineering, Northeastern University, Shenyang 110004, PR China, E-mail: xuedingyu@ise.neu.edu.cn
² 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; E-mail: yqchen@ece.usu.edu

Abstract

In this paper, we propose a procedure to achieve pseudo-rational approximation to arbitrary fractional-order linear time invariant (FO-LTI) systems with suboptimum $H_2$-norm. The proposed pseudo-rational approximation is actually a rational model with a time delay. Through illustrations, we show that the pseudo-rational approximation is simple and effective. It is also demonstrated that this suboptimum approximation method is effective in designing integer-order controllers for FO-LTI systems in general non-commensurate form. Useful MATLAB codes are also included in the appendix.

Keywords
Fractional-order systems, model reduction, optimal model reduction, time delay systems, $H_2$-norm approximation.

1 Introduction

Fractional order calculus, a 300-years-old topic [1, 2, 3, 4], has been gaining increasing attention in research communities. Applying fractional-order calculus to dynamic systems control, however, is just a recent focus of interest [5, 6, 7, 8, 9]. We should point out references [10, 11, 12, 13] for pioneering works and [14, 15, 16] for more recent developments. In most cases, our objective is to apply fractional-order control to enhance the system control performance. For example, as in the CRONE, where CRONE is a French abbreviation for “Commande robuste d’ordre non-entier” (which means non-integer order robust control), [17, 7, 8], fractal robustness is pursued. The
desired frequency template leads to fractional transmittance [18, 19] on which the CRONE controller synthesis is based. In CRONE controllers, the major ingredient is the fractional-order derivative $s^r$, where $r$ is a real number and $s$ is the Laplace transform symbol of differentiation. Another example is the PI$^\lambda$D$^\mu$ controller [6, 20], an extension of PID controller. In general form, the transfer function of PI$^\lambda$D$^\mu$ is given by $K_p + T_i s^{-\lambda} + T_d s^\mu$, where $\lambda$ and $\mu$ 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 $\mu = 1$, we obtain a classical PID controller. If $T_i = 0$ we obtain a PD$^\mu$ controller, etc. All these types of controllers are particular cases of the PI$^\lambda$D$^\mu$ controller. It can be expected that the PI$^\lambda$D$^\mu$ controller may enhance the systems control performance due to more tuning knobs introduced.

Actually, in theory, PI$^\lambda$D$^\mu$ itself is an infinite dimensional linear filter due to the fractional order in the differentiator or integrator. It should be pointed out that a band-limit implementation of FOC is important in practice, i.e., the finite dimensional approximation of the FOC should be done in a proper range of frequencies of practical interest [21, 19]. Moreover, the fractional order can be a complex number as discussed in [21]. In this paper, we focus on the case where the fractional order is a real number.

For a single term $s^r$ with $r$ a real number, there are many approximation schemes proposed. In general, we have analog realizations [22, 23, 24, 25] and digital realizations. The key step in digital implementation of an FOC is the numerical evaluation or discretization of the fractional-order differentiator $s^r$. In general, there are two discretization methods: direct discretization and indirect discretization. In indirect discretization methods [21], two steps are required, i.e., frequency-domain fitting in continuous time domain first and then discretizing the fit $s$-transfer function. Other frequency-domain fitting methods can also be used but without guaranteeing the stable minimum-phase discretization. Existing direct discretization methods include the application of the direct power series expansion (PSE) of the Euler operator [26, 27, 28, 29], continuous fractional expansion (CFE) of the Tustin operator [27, 28, 29, 30, 31], and numerical integration-based method [26, 30, 32]. However, as pointed out in [33, 34, 35], the Tustin operator-based discretization scheme exhibits large errors in high-frequency range. A new mixed scheme of Euler and Tustin operators is proposed in [30] which yields the so-called Al-Alaoui operator [33]. These discretization methods for $s^r$ are in IIR form. Recently, there are some reported methods to directly obtain the digital fractional-order differentiators in FIR (finite impulse response) form [36, 37]. However, using an FIR filter to approximate $s^r$ may be less efficient due to very high order of the FIR filter. So, discretizing fractional differentiators in IIR forms is preferred [38, 30, 32, 31].

In this paper, we consider the general fractional-order LTI systems (FO-LTI) with noncommensurate fractional orders as follows:

$$G(s) = \frac{b_m s^{\gamma_m} + b_{m-1} s^{\gamma_{m-1}} + \cdots + b_1 s^{\gamma_1} + b_0}{a_n s^{\eta_n} + a_{n-1} s^{\eta_{n-1}} + \cdots + a_1 s^{\eta_1} + a_0}. \tag{1}$$
Using the aforementioned approximation schemes for a single $s^r$ and then for the general FO-LTI system (1) could be very tedious, leading to a very high-order model. In this paper, we propose to use a numerical algorithm to achieve a good approximation of the overall transfer function (1) using finite integer-order rational transfer function with a possible time delay term and illustrate how to use the approximated integer-order model for integer-order controller design. In Examples 1 and 2, approximation to a fractional-order transfer function is given and the fitting results are illustrated. In example 3, a fractional-order plant is approximated using the algorithm proposed in the paper, by a FOPD (first-order plus delay) model, and using an existing PID tuning formula, an integer order PID can be designed with a very good performance.

2 True Rational Approximations to Fractional Integrators and Differentiators: Outstaloup’s Method

For comparison purpose, here we present Oustaloup’s algorithm [18, 19, 39]. Assuming that the frequency range to fit is selected as $(\omega_b, \omega_h)$, the transfer function of a continuous filter can be constructed to approximate the pure fractional derivative term $s^{\gamma}$ such that

$$G_{f,\gamma}(s) = K \prod_{k=-N}^{N} \frac{s + \omega'_k}{s + \omega_k}$$

where the zeros, poles, and the gain can be evaluated from

$$\omega'_k = \omega_b \left( \frac{\omega_h}{\omega_b} \right)^{\frac{k+N+\frac{1}{2}(1-\gamma)}{2N+1}}, \quad \omega_k = \omega_b \left( \frac{\omega_h}{\omega_b} \right)^{\frac{k+N+\frac{1}{2}(1+\gamma)}{2N+1}}, \quad K = \omega_h^{\gamma},$$

where $k = -N, \ldots, N$.

An implementation in MATLAB is given in Appendix 1. Substituting $\gamma_i$ and $\eta_i$ in (1) with $G_{f,\gamma_i}(s)$ and $G_{f,\eta_i}(s)$ respectively, the original fractional-order model $G(s)$ can be approximated by a rational function $\hat{G}(s)$. It should be noted that the order of the resulted $\hat{G}(s)$ is usually very high. Thus, there is a need to approximate the original model by reduced order ones using the optimal-reduction techniques.

3 A Numerical Algorithm for Suboptimal Pseudo-Rational Approximations

In this section, we are interested in finding an approximate integer-order model with a low order, possibly with a time delay in the following form:
\[ G_{r/m,\tau}(s) = \frac{\beta_1 s^r + \ldots + \beta_r s + \beta_{r+1}}{s^m + \alpha_1 s^{m-1} + \ldots + \alpha_{m-1} s + \alpha_m} e^{-\tau s}. \] (4)

An objective function for minimizing the $H_2$-norm of the reduction error signal $e(t)$ can be defined as

\[ J = \min_{\theta} \left\| \hat{G}(s) - G_{r/m,\tau}(s) \right\|_2 \] (5)

where $\theta$ is the set of parameters to be optimized such that

\[ \theta = [\beta_1, \ldots, \beta_r, \alpha_1, \ldots, \alpha_m, \tau]. \] (6)

For an easy evaluation of the criterion $J$, the delayed term in the reduced order model $G_{r/m,\tau}(s)$ can be further approximated by a rational function $\hat{G}_{r/m}(s)$ using the Padé approximation technique [40]. Thus, the revised criterion can then be defined by

\[ J = \min_{\theta} \left\| \hat{G}(s) - \hat{G}_{r/m}(s) \right\|_2 \] (7)

and the $H_2$ norm computation can be evaluated recursively using the algorithm in [41].

Suppose that for a stable transfer function type $E(s) = \hat{G}(s) - \hat{G}_{r/m}(s) = B(s)/A(s)$, the polynomials $A_k(s)$ and $B_k(s)$ can be defined such that,

\[ A_k(s) = a_0^k + a_1^k s + \ldots + a_k^k s^k, \quad B_k(s) = b_0^k + b_1^k s + \ldots + b_{k-1}^k s^{k-1} \] (8)

The values of $a_i^{k-1}$ and $b_i^{k-1}$ can be evaluated from

\[ a_i^{k-1} = \begin{cases} a_{i+1}^k, & i \text{ even} \\ a_i^k - \alpha_k a_{i+2}^k, & i \text{ odd} \end{cases} \quad i = 0, \ldots, k - 1 \] (9)

and

\[ b_i^{k-1} = \begin{cases} b_{i+1}^k, & i \text{ even} \\ b_i^k - \beta_k a_{i+2}^k, & i \text{ odd} \end{cases} \quad i = 1, \ldots, k - 1 \] (10)

where, $\alpha_k = a_0^k/a_1^k$, and $\beta_k = b_1^k/a_1^k$.

The $H_2$-norm of the approximate reduction error signal $\hat{e}(t)$ can be evaluated from

\[ J = \sum_{k=1}^{n} \frac{\beta_k^2}{2\alpha_k} = \sum_{k=1}^{n} \frac{(b_1^k)^2}{2a_0^k a_1^k} \] (11)

The sub-optimal $H_2$-norm reduced order model for the original high order fractional order model can be obtained using the following procedure [40]:

1. Select an initial reduced model $\hat{G}_r^\theta(s)$.
2. Evaluate an error $\left\| \hat{G}(s) - \hat{G}_r^\theta(s) \right\|_2$ from (11).
3. Use an optimization algorithm (for instance, Powell’s algorithm [42]) to iterate one step for a better estimated model \( \hat{G}_{r/m}^1(s) \).

4. Set \( \hat{G}_{r/m}^0(s) \leftarrow \hat{G}_{r/m}^1(s) \), go to step 2 until an optimal reduced model \( \hat{G}_{r/m}^*(s) \) is obtained.

5. Extract the delay from \( \hat{G}_{r/m}^*(s) \), if any.

We call the above procedure suboptimal since the Oustaloup’s method is used for each single term \( s^\gamma \) in (1), and also, Padé approximation is used for pure delay terms.

4 Illustrative Examples

Examples are given in the section to demonstrate the optimal-model reduction procedures with full MATLAB implementations. Also the integer-order PID controller design procedure is explored for fractional-order plants, based on the model reduction algorithm in the paper.

**Example 1: Non-commensurate FO-LTI system**

Consider the non-commensurate FO-LTI system

\[
G(s) = \frac{5}{s^{2.3} + 1.3s^{0.9} + 1.25}.
\]

Using the following MATLAB scripts,

\[
\begin{align*}
\text{w1} &= 1e-3; \quad \text{w2} = 1e3; \quad \text{N} = 2; \\
\text{g1} &= \text{ousta_fod}(0.3,\text{N},\text{w1},\text{w2}); \quad \text{g2} = \text{ousta_fod}(0.9,\text{N},\text{w1},\text{w2}); \\
\text{s} &= \text{tf}(\text{’s’}); \quad \text{G} = \text{5}/(\text{s}^2\text{g1}+1.3\text{g2}+1.25);
\end{align*}
\]

with the Oustaloup’s filter, the high-order approximation to the original fractional-order model can be approximated by

\[
\begin{align*}
G(s) &= 5s^{10} + 6677s^9 + 2.191 \times 10^6s^8 + 1.505 \times 10^8s^7 \\
&\quad + 2.936 \times 10^9s^6 + 1.257 \times 10^{10}s^5 + 1.541 \times 10^{10}s^4 \\
&\quad + 4.144 \times 10^9s^3 + 3.168 \times 10^8s^2 + 5.065 \times 10^6s + 1.991 \times 10^4 \\
&\quad + 7.943s^{12} + 8791s^{11} + 1.731 \times 10^6s^{10} + 8.766 \times 10^7s^9 \\
&\quad + 1.046 \times 10^9s^8 + 3.82 \times 10^9s^7 + 6.099 \times 10^9s^6 + 7.743 \times 10^9s^5 \\
&\quad + 5.197 \times 10^9s^4 + 1.15 \times 10^9s^3 + 8.144 \times 10^7s^2 + 1.278 \times 10^6s + 4987.
\end{align*}
\]

The following statements can then be used to find the optimum reduced order approximations to the original fractional order model.

\[
\begin{align*}
\text{G1} &= \text{opt_app}(\text{G},1,2,0); \quad \text{G2} = \text{opt_app}(\text{G},2,3,0); \\
\text{G3} &= \text{opt_app}(\text{G},3,4,0); \quad \text{G4} = \text{opt_app}(\text{G},4,5,0); \\
\text{step}(\text{G},\text{G1},\text{G2},\text{G3},\text{G4})
\end{align*}
\]
where the four reduced order models can be obtained

\[
G_1(s) = \frac{-2.045s + 7.654}{s^2 + 1.159s + 1.917}
\]

\[
G_2(s) = \frac{-0.5414s^2 + 4.061s + 2.945}{s^3 + 0.9677s^2 + 1.989s + 0.7378}
\]

\[
G_3(s) = \frac{-0.2592s^3 + 3.365s^2 + 4.9s + 0.3911}{s^4 + 1.264s^3 + 2.25s^2 + 1.379s + 0.09797}
\]

\[
G_4(s) = \frac{1.303s^4 + 1.902s^3 + 11.15s^2 + 4.71s + 0.1898}{s^5 + 2.496s^4 + 3.485s^3 + 4.192s^2 + 1.255s + 0.04755}
\]

The step responses for the above four reduced-order models can be obtained as compared in Fig. 1. It can be seen that the 1/2th order model gives a poor approximation to the original system, while the other low-order approximations using the method and codes of this paper are effective.

![Step Response](image)

Fig. 1. Step responses comparisons of rational approximations.

**Example 2: Non-commensurate FO-LTI system**

Consider the following non-commensurate FO-LTI system:

\[
G(s) = \frac{5s^{0.6} + 2}{s^{3.3} + 3.1s^{2.6} + 2.89s^{1.9} + 2.5s^{1.4} + 1.2}
\]

Using the following MATLAB scripts,

```matlab
N=2; w1=1e-3; w2=1e3;
g1=ousta_fod(0.3,N,w1,w2); g2=ousta_fod(0.6,N,w1,w2);
```
an extremely high-order model can be obtained with the Oustaloup’s filter, such that

\[
G(s) = \frac{317.5s^{25} + 8.05 \times 10^5 s^{24} + 7.916 \times 10^8 s^{23} + 3.867 \times 10^{11} s^{22} + 1.001 \times 10^{14} s^{21} + 1.385 \times 10^{16} s^{20} + 1.061 \times 10^{18} s^{19} + 4.664 \times 10^{19} s^{18} + 1.197 \times 10^{21} s^{17} + 1.778 \times 10^{22} s^{16} + 1.5 \times 10^{23} s^{15} + 7.242 \times 10^{23} s^{14} + 2.052 \times 10^{24} s^{13} + 3.462 \times 10^{24} s^{12} + 3.459 \times 10^{24} s^{11} + 2.009 \times 10^{24} s^{10} + 6.724 \times 10^{23} s^9 + 1.329 \times 10^{23} s^8 + 1.579 \times 10^{22} s^7 + 1.12 \times 10^{21} s^6 + 4.592 \times 10^{19} s^5 + 1.037 \times 10^{18} s^4 + 1.314 \times 10^{16} s^3 + 9.315 \times 10^{13} s^2 + 3.456 \times 10^{11} s + 5.223 \times 10^8}{7.943 s^{28} + 2.245 \times 10^4 s^{27} + 2.512 \times 10^7 s^{26} + 1.427 \times 10^{10} s^{25} + 4.392 \times 10^{12} s^{24} + 7.384 \times 10^4 s^{23} + 6.896 \times 10^6 s^{22} + 3.736 \times 10^8 s^{21} + 1.208 \times 10^{10} s^{20} + 2.343 \times 10^6 s^{19} + 2.716 \times 10^8 s^{18} + 1.896 \times 10^8 s^{17} + 8.211 \times 10^6 s^{16} + 2.268 \times 10^7 s^{15} + 4.076 \times 10^6 s^{14} + 4.834 \times 10^4 s^{13} + 3.845 \times 10^4 s^{12} + 2.134 \times 10^4 s^{11} + 8.772 \times 10^3 s^{10} + 2.574 \times 10^3 s^9 + 5.057 \times 10^2 s^8 + 6.342 \times 10^2 s^7 + 4.686 \times 10^2 s^6 + 2.16 \times 10^5 s^5 + 5.176 \times 10^4 s^4 + 6.863 \times 10^3 s^3 + 5.055 \times 10^3 s^2 + 1.938 \times 10^3 s + 3.014 \times 10^2},
\]

and the order of rational approximation to the original order model is the 28th, for \( N = 2 \). For larger values of \( N \), the order of rational approximation may be even much higher. For instance, the order of the approximation may reach the 38th and 48th respectively for the selections \( N = 3 \) and \( N = 4 \), with extremely large coefficients. Thus the model reduction algorithm should be used with the following MATLAB statements

\[
\begin{align*}
G2 &= \text{opt_app}(G,2,3,0) \quad G3 = \text{opt_app}(G,3,4,0) \\
G4 &= \text{opt_app}(G,4,5,0) \quad \text{step}(G,G2,G3,G4,60)
\end{align*}
\]

the step responses can be compared in Fig. 2 and it can be seen that the third-order approximation is satisfactory and the fourth-order fitting gives a better approximation. The obtained optimum approximated results are listed in the following:

\[
\begin{align*}
G_2(s) &= \frac{0.41056s^2 + 0.75579s + 0.037971}{s^2 + 0.24604s^2 + 0.22176s + 0.021915} \\
G_3(s) &= \frac{-4.4627s^3 + 5.6139s^2 + 4.3354s + 0.15330}{s^3 + 7.4462s^3 + 1.7171s^2 + 1.5083s + 0.088476} \\
G_4(s) &= \frac{1.7768s^4 + 2.2291s^3 + 10.911s^2 + 1.2169s + 0.010249}{s^5 + 11.347s^4 + 4.8219s^3 + 2.8448s^2 + 0.59199s + 0.0059152}
\end{align*}
\]
Example 3: Sub-optimum pseudo-rational model reduction for integer-order PID controller design

Let us consider the following FO-LTI plant model:

$$G(s) = \frac{1}{s^{2.3} + 3.2s^{1.4} + 2.4s^{0.9} + 1}.$$  

Let us first approximate it with Oustaloup’s method and then fit it with a fixed model structure known as FOLPD (first-order lag plus deadtime) model, where $G_r(s) = \frac{K}{T_s + 1} e^{-Ls}$. The following MATLAB scripts can perform this task and the obtained optimal FOLPD model is given as follows:

```
N=2; w1=1e-3; w2=1e3;
g1=ousta_fod(0.3,N,w1,w2);
g2=ousta_fod(0.4,N,w1,w2);
g3=ousta_fod(0.9,N,w1,w2);
s=tf('s'); G=1/(s^2+3.2*s+2.4*s+1);
G2=opt_app(G,0,1,1); step(G,G2)
```

The comparison of the open-loop step response is shown in Fig. 3. It can be observed that the approximation is fairly effective.

Designing a suitable feedback controller for the original FO-LTI system $G$ can be a formidable task. Now, let us consider designing an integer-order PID controller...
controller for the optimally reduced model $G_r(s)$ and let us see if the designed controller still works for the original system.

The integer order PID controller to be designed is in the following form:

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} + \frac{T_d s}{T_d / N s + 1} \right).$$  \hfill (12)

The optimum ITAE criterion-based PID tuning formula \cite{43} can be used

$$K_p = \frac{(0.7303 + 0.5307T/L)(T + 0.5L)}{K(T + L)},$$  \hfill (13)

$$T_i = T + 0.5L, \quad T_d = \frac{0.5LT}{T + 0.5L}.$$  \hfill (14)

Based on this tuning algorithm, a PID controller can be designed for $G_r(s)$ as follows:

L=0.63; T=3.5014; K=0.9951; N=10; Ti=T+0.5*L;
Kp=(0.7303+0.5307*T/L)*Ti/(K*(T+L));
Td=(0.5*L*T)/(T+0.5*L); [Kp,Ti,Td]
Gc=Kp*(1+1/Ti/s+Td*s/(Td/N*s+1))

The parameters of the PID controller are then $K_p = 3.4160$, $T_i = 3.8164$, $T_d = 0.2890$, and the PID controller can be written as

$$G_c(s) = \frac{1.086s^2 + 3.442s + 0.8951}{0.0289s^2 + s}$$
Finally, the step response of the original FO-LTI with the above-designed PID controller is shown in Fig. 4. A satisfactory performance can be clearly observed. Therefore, we believe, the method presented in this paper can be used for integer-order controller design for general FO-LTI systems.

![Step Response](image)

**Fig. 4.** Step response of fractional-order plant model under the PID controller.

5 Concluding Remarks

In this paper, we presented a procedure to achieve pseudo-rational approximation to arbitrary FO-LTI systems with suboptimum $H_2$-norm. Relevant MATLAB codes useful for practical applications are also given in the appendix. Through illustrations, we show that the pseudo-rational approximation is simple and effective. It is also demonstrated that this suboptimum approximation method is effective in designing integer order controllers for FO-LTI systems in general form.

Finally, we would like to remark that the so-called pseudo-rational approximation is essentially by cascading irrational transfer function (a time delay) and a rational transfer function. Since a delay element is also infinite dimensional, it makes sense to approximate a general fractional-order LTI system involving time delay. Although it might not fully make physical sense, the pseudo-rational approximation proposed in this paper will find its practical applications in designing an integer-order controller for fractional-order systems, as illustrated in Example 3.
Acknowledgment

We acknowledge that this paper is a modified version of a paper published in the Proceedings of IDETC/CIE 2005 (Paper# DETC2005-84743). We would like to thank the ASME for granting us permission in written form to publish a modified version of IDETC/CIE 2005 (Paper# DETC2005-84743) as a chapter in the book entitled Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and in Engineering edited by Professors Machado, Sabatier, and Agrawal (Springer).

Appendix 1  MATLAB functions for optimum fractional model reduction

- **ousta_fod.m**  Outstaloup’s rational approximation to fractional differentiator, with the syntax $G=\text{ousta}\_\text{fod}(r,N,\omega_L,\omega_H)$

  ```matlab
  function G=ousta_fod(r,N,\omega_L,\omega_H)
  mu=\omega_H/\omega_L; k=-N:N; w_kp=(mu).^((k+N+0.5-0.5*r)/(2*N+1))\*\omega_L;
  w_k=(mu).^((k+N+0.5+0.5*r)/(2*N+1))\*\omega_L;
  K=(mu)^(-r/2)*prod(w_k./w_kp); G=tf(zpk(-w_kp',-w_k',K));
  ```

- **opt_app.m**  Optimal model reduction function, and the pseudo-rational transfer function model $G_r$, i.e., the transfer function with a possible delay term, can be obtained. $G_r=\text{opt}\_\text{app}(G,r,d,\text{key},G_0)$, where $\text{key}$ indicates whether a time delay is required in the reduced order model. $G_0$ is the initial reduced order model, optional.

  ```matlab
  function G_r=opt_app(G,nn,nd,key,G0)
  GS=tf(G); num=GS.num{1}; den=GS.den{1}; Td=totaldelay(GS);
  GS.ioDelay=0; GS.InputDelay=0; GS.OutputDelay=0;
  if nargin<5,
    n0=[1,1];
    for i=1:nd-2, n0=conv(n0,[1,1]); end
  G0=tf(n0,conv([1,1],n0));
  end
  beta=G0.num{1}(nd+1-nn:nd+1); alph=G0.den{1}; Tau=1.5*Td;
  x=[beta(1:nn),alph(2:nd+1)]; if abs(Tau)<1e-5, Tau=0.5; end
  if key==1, x=[x,Tau]; end
  dc=dcgain(GS); y=opt_fun(x,GS,key,nn,nd,dc);
  x=fminsearch(’opt\_fun’,x,[],GS,key,nn,nd,dc);
  alph=[1,x(nn+1:nn+nd)]; beta=x(1:nn+1); if key==0, Td=0; end
  beta(nn+1)=alph(end)*dc;
  if key==1, Tau=x(end)+Td; else, Tau=0; end
  G_r=tf(beta,alph,’ioDelay’,Tau);
  ```
• opt_fun.m  internal function used by opt_app,

function y=opt_fun(x,G,key,nn,nd,dc)
ff0=1e10; alph=[1,x(nn+1:nn+nd)];
beta=x(1:nn+1); beta(end)=alph(end)*dc; g=tf(beta,alph);
if key==1,
tau=x(end); if tau<=0, tau=eps; end
[nP,dP]=pade(tau,3); gP=tf(nP,dP);
else, gP=1; end
G_e=G-g*gP;
G_e.num{1}=[0,G_e.num{1}(1:end-1)];
[y,ierr]=geth2(G_e);
if ierr==1, y=10*ff0; else, ff0=y; end

• get2h.m  internal function to evaluate the $H_2$ norm of a rational transfer function model.

function [v,ierr]=geth2(G)
G=tf(G); num=G.num{1}; den=G.den{1}; ierr=0; n=length(den);
if abs(num(1))>eps
    disp('System not strictly proper'); ierr=1; return
else, a1=den; b1=num(2:end); end
for k=1:n-1
    if (a1(k+1)<=eps), ierr=1; v=0; return
    else,
        aa=a1(k)/a1(k+1); bb=b1(k)/a1(k+1); v=v+bb*bb/aa; k1=k+2;
        for i=k1:2:n-1
            a1(i)=a1(i)-aa*a1(i+1); b1(i)=b1(i)-bb*a1(i+1);
        end, end, end
    v=sqrt(0.5*v);

References


