

Hybrid symbolic and numerical simulation studies of time-fractional order wave-diffusion systems

J. LIANG and Y. Q. CHEN*

Center for Self-Organizing and Intelligent Systems (CSOIS), Dept. of Electrical and Computer Engineering,
Utah State University, 4120 Old Main Hill, Logan, UT 84322-4120, USA

(Received 30 August 2005; revised 30 January 2006; in final form 22 March 2006)

Boundary control of time-fractional order diffusion-wave systems is becoming an active research area. However, there is no readily available simulation tool till now for researchers to analyze and design controllers. In this paper, a simulation method for some typical boundary control problems, combining symbolic mathematics and numerical method, is presented with two application examples. In the intermediate steps of the simulation, an important by-product, the transfer function of the controlled system, can be obtained, which makes the design of more advanced boundary controllers possible and much easier.

1. Introduction

Fractional diffusion and wave equations are obtained from the classical diffusion and wave equations by replacing the first and second order time derivative term by a fractional order derivative of an order satisfying $0 < \alpha \leq 1$ and $1 < \alpha \leq 2$, respectively. Since many of the universal phenomena can be modelled accurately using the fractional diffusion and wave equations (Nigmatullin 1986), there has been a growing interest in investigating the solutions and properties of these evolution equations (Wyss 1986, Schneider and Wyss 1989, Matignon and d'Andréa-Novel 1995, Mainardi and Paradisi 1997, Agrawal 2002; Liang *et al.* 2003, 2004, 2005). Most research has been focused on the analytical solution to the fractional diffusion and wave equations. Contrary to the progress in the theoretical analysis, simulation examples in the publications are very few, albeit simulation plays such an important role in verifying the theoretical analysis and design, identifying the potential problems, reducing the investment, and selecting the optimal solution. The reason for this, we would like to suggest, is that

the difficulty of simulating control of fractional order diffusion-wave equations, especially using a fractional order controller, is beyond the capability of most commonly available mathematical tools, such as Matlab, Maple, or even FEMLAB (<http://www.comsol.com>).

In this paper, we present an easy-to-implement, yet effective, boundary control simulation method, which combines the analytical method, the numerical method, and the modern symbolic algebra. The simulation examples show that this method can be used to solve some problems that is very hard to solve with analytical methods. The method is also much easier to implement than finite element method (FEM) or finite difference method (FDM) (Mitchell and Griffiths 1980). No extra software is needed except Matlab and the Matlab Symbolic Math Toolbox. Since the analysis and design of controllers for the fractional diffusion-wave equations is still an open problem, the simulation method proposed in this paper bears special meaning. Actually, the simulation method can be used to solve higher order problems, for which the analytical method is totally impractical.

This paper is organized as follows. Section 2 introduces the principle and the implementation procedure via an illustrative example. Section 3 shows

*Corresponding author. Email: yqchen@ieee.org

some interesting applications of this simulation method. Finally, § 4 concludes the paper.

2. Problem formulation, principle and implementation

In this section, the following example will be used to show how boundary control of the fractional order diffusion-wave equation is formulated. The same example will be used to demonstrate the basic principle and implementation details of the simulation method presented in this paper.

We consider a cable made with special smart materials governed by the fractional wave equation, fixed at one end, and stabilized by a boundary controller at the other end. Omitting the mass of the cable, the system can be represented by

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{\partial^2 u}{\partial x^2}, \quad 1 < \alpha \leq 2, \quad x \in [0, 1], \quad t \geq 0 \quad (1)$$

$$u(0, t) = 0, \quad (2)$$

$$u_x(1, t) = f(t), \quad (3)$$

$$u(x, 0) = u_0(x), \quad (4)$$

$$u_t(x, 0) = v_0(x), \quad (5)$$

where $u(x, t)$ is the displacement of the cable at $x \in [0, 1]$ and $t \geq 0$, $f(t)$ is the boundary control force at the free end of the cable, $u_0(x)$ and $v_0(x)$ are the initial conditions of displacement and velocity, respectively.

The control objective is to stabilize $u(x, t)$ using $f(t)$, given the initial conditions (4) and (5).

We adopt the following definition for the fractional derivative of order α of function $f(t)$ (Mainardi 1996, Mainardi and Paradisi 1997),

$$\frac{d^\alpha}{dt^\alpha} f(t) \doteq \begin{cases} f^{(n)}(t) & \text{if } \alpha = n \in N, \\ \frac{t^{n-\alpha-1}}{\Gamma(n-\alpha)} * f^{(n)}(t) & \text{if } n-1 < \alpha < n, \end{cases} \quad (6)$$

where the $*$ denotes the time convolution between two functions.

In this section, we will simulate the response of the following fractional order controller:

$$f(t) = -k \frac{d^\mu u(1, t)}{dt^\mu}, \quad 0 < \mu \leq 1 \quad (7)$$

where k is the controller gain, μ is the order of fractional derivative of the displacement at the free end of the cable.

When $\mu = 1$, the controller (7) is called integer order controller and has been widely used in the boundary control of wave equations and beam equations

(Chen 1979, Chen *et al.* 1987, Conrad and Morgül 1998). In the following, we will study via the simulation method to see if the fractional order control ($0 < \mu < 1$) is able to stabilize the system.

It is well-known that linear PDEs can be solved by means of the Laplace transform (Jeffrey 2002). Following is a summary of this method. We assume that the solution of a PDE is a function $u(x, t)$ of the two independent variables x and t .

- (i) Transform $u(x, t)$ with respect to t by means of the Laplace transform, so we obtain an ODE for the transformed variable $U(x, s)$:

$$f\left(U(x, s), \frac{dU(x, s)}{dx}, \dots, \frac{d^n U(x, s)}{dx^n}, x, s\right) = 0. \quad (8)$$

- (ii) Solve the ODE (8) for $U(x, s)$ as a function of x , with the transform variable s still appearing as a parameter in the solution, and use the boundary conditions of the original problem to determine the precise form of $U(x, s)$.
- (iii) Take the inverse Laplace transform of $U(x, s)$ with respect to s to find the solution $u(x, t)$.

Several problems make the above method hard to use in practice to solve the problem of boundary control of fractional order diffusion-wave equations. First, the obtained ODE is hard to solve manually due to the complicity of the fractional order equation, the fractional order controller, and the initial conditions. Second, the arbitrary constants in the general solution of the ODE are hard to determine due to the same reason. Third, even if we can determine the undefined constants, usually the inverse Laplace transform can not be performed by looking up a table of transformation pairs.

We solve the above problems using the Matlab Symbolic Math Toolbox (MathWorks Inc. 2002) and the numerical inverse Laplace transform (Duffy, 1993, Brančik, 1999b). The detailed implementation procedures are demonstrated in the following example.

The initial conditions are chosen as

$$u(x, 0) = -\sin(0.5\pi x), \quad (9)$$

and

$$u_t(x, 0) = 0. \quad (10)$$

The other parameters are chosen to be $\alpha = 1.8$, $\mu = 0.8$, and $k = 1$.

- (i) Take the Laplace transform of (1)–(5) and (7) with respect to t . Based on the definition of (6), the

Laplace transform of the fractional derivative is

$$\mathcal{L}\left\{\frac{d^\alpha f(t)}{dt^\alpha}\right\} = s^\alpha F(s) - \sum_{k=0}^{n-1} f^{(k)}(0^+)s^{\alpha-1-k} \quad (11)$$

Thus, the original PDE of $u(x, t)$ with initial and boundary conditions is transformed into an ODE of $U(x, s)$ with boundary conditions.

- (ii) Call the Matlab Symbolic Math Toolbox function `dsolve()` to symbolically solve the ODE. Although `dsolve()` is able to determine the arbitrary constants in the solution using the boundary or initial condition(s), we find that its capability is weak. Here, we feed only the ODE of $U(x, s)$ to `dsolve()`, rather than provide both the ODE of $U(x, s)$ and the boundary conditions. The expression of $U(x, s)$ with two arbitrary constants C_1 and C_2 , to be determined later, can be obtained.

$$U(x, s) = C_1 e^{-s^{9/10}x} + C_2 e^{s^{9/10}x} - 4 \frac{s^{4/5} \sin(1/2 \pi x)}{4 s^{9/5} + \pi^2} \quad (12)$$

- (iii) Using Matlab Symbolic Math Toolbox function `diff()`, differentiate $U(x, s)$ with respect to x to get the derivative of $U(x, s)$. Substituting $U(x, s)$ and its derivative into the Laplace transform of (2) and (3), we get two equations with two unknowns C_1 and C_2 .

$$C_1 + C_2 = 0 \quad (13)$$

$$\begin{aligned} & s^{9/10} e^{s^{9/10}} C_2 - s^{9/10} C_1 e^{-s^{9/10}} \\ & + s^{4/5} \left(e^{s^{9/10}} C_2 + e^{-s^{9/10}} C_1 - \frac{4s^{4/5}}{4s^{9/5} + \pi^2} \right) \\ & + \frac{1}{\sqrt[5]{s}} = 0. \end{aligned} \quad (14)$$

- (iv) Passing the two equations obtained in the last step to the Matlab Symbolic Math Toolbox function `solve()` to determine the constants C_1 and C_2 .

$$\begin{aligned} C_1 = -C_2 = & e^{s^{9/10}} \pi^2 \left(4s^{29/10} e^{2s^{9/10}} + s^{11/10} e^{2s^{9/10}} \pi^2 + 4s^{29/10} \right. \\ & \left. + s^{11/10} \pi^2 + 4s^{14/5} e^{2s^{9/10}} + s e^{2s^{9/10}} \pi^2 - 4s^{14/5} - s\pi^2 \right)^{-1}. \end{aligned} \quad (15)$$

Now, we have obtained the explicit expression of $U(x, s)$, which is usually a very long expression.

- (v) Due to the complicity of $U(x, s)$, its analytical

inverse Laplace transform is usually unavailable. We apply the numerical inverse Laplace transform (Brančik 1999b) to $U(x, s)$ to obtain the numerical solution.

Following the above steps, the solution of (1)–(5) and (7) can be obtained, which can be used to verify the effectiveness of a fractional order controller.

Although obtaining the explicit expression of $U(x, s)$ is just an intermediate step of this simulation method, it will be shown later that this is critical to designing more advanced boundary controllers. If we use a general symbol $F(s)$ for boundary controller rather than using the specific boundary controller (7), following the same procedure, we can obtain $U(x, s) = F(s)P(s)$. Divide $U(x, s)$ by $F(s)$, we obtain $P(s)$, the transfer function of this control system. The role of the transfer function in control system design cannot be over-emphasized. For other numerical methods, such as FEM or FDM, it is impossible to obtain the transfer function. This is another advantage of the simulation method developed in this paper.

To obtain $u(x, t)$, we need to take the inverse Laplace transform of $U(x, s)$. We should *not* use the Matlab Symbolic Math Toolbox function `ilaplace()`, which takes the inverse Laplace transform symbolically, since for such a complicated expression of $U(x, s)$, the explicit expression of $u(x, t)$ is usually unavailable. However, we can make use of the numerical inverse Laplace transform algorithms. There are many numerical techniques available for inverse Laplace transform (Duffy 1993). Among the existing numerical inverse Laplace transform methods, we choose the method introduced in (Brančik 1999b) for its proven accuracy and fastness. The basic idea of this method is summarized as follows (Brančik 1999b).

The inverse Laplace transform is defined as

$$f(t) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s) e^{st} ds. \quad (16)$$

After applying the trapezoidal quadrature formula with some rearrangements, an approximate formula in discrete form can be written as

$$\tilde{f}^k = C^k \left\{ 2\text{Re} \left[\sum_{n=0}^{\infty} F_n E_n^k \right] - F_0 \right\}, \quad (17)$$

for $k = 0, 1, \dots, N - 1$, with

$$\begin{aligned} \tilde{f}^k &= \tilde{f}(kT), \quad C^k = \frac{\Omega}{2\pi} e^{ckT}, \\ F_n &= F(c - jn\Omega), \quad E_n^k = e^{-jkTn\Omega}, \end{aligned}$$

where T and $\Omega = 2\pi/(NT)$ are sampling periods in original and transform domains, respectively. It can be proved that (17) corresponds to a Fourier series approximation of the original $f(t)$ when the error can theoretically be controlled in the interval $t \in (0, NT)$.

To speed up the convergence of infinite complex Fourier series, the ϵ -algorithm is applied, which makes the results from FFT more accurate as if more terms are used to compute the FFT. Interested readers can refer to Brančík (1998, 1999a) for detailed theory and Brančík (1999b) for implementation details with readily available Matlab code.

At this point, we have actually finished the time-domain simulation. The plot of tip end displacement is shown in figure 1. It is also very easy to show the displacement of the whole string as in figure 2. It is clearly seen that the fractional order boundary controller for $\mu = 0.8$ is stable.

3. Two application examples

In this section, we will use two application examples to show the usefulness of the simulation method and that more difficult boundary control problems can be simulated.

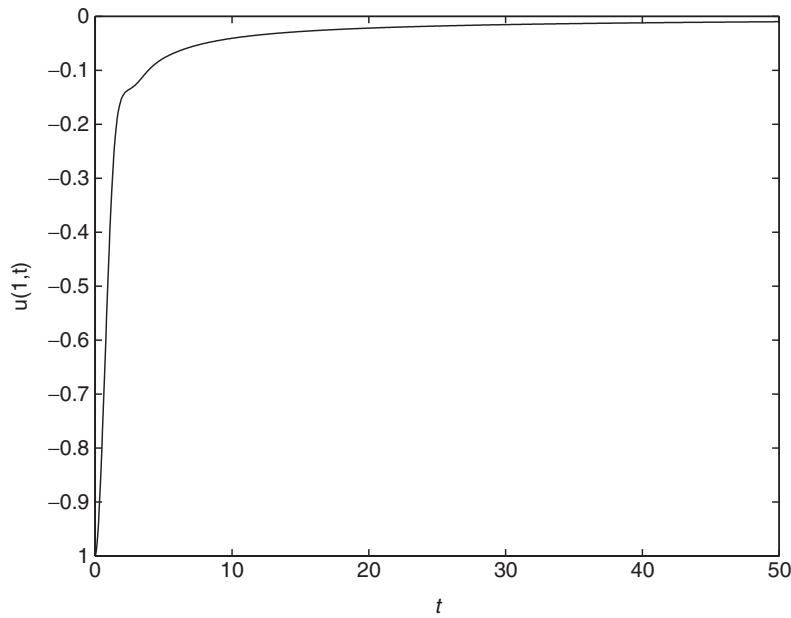


Figure 1. Tip displacement.

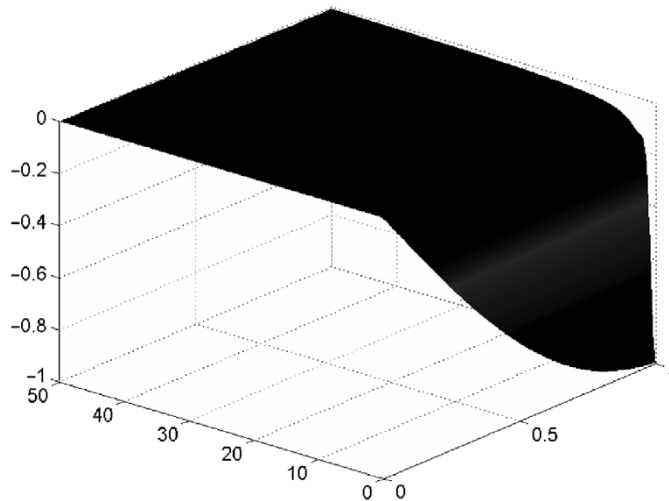


Figure 2. Displacement of the whole string.

3.1 Which optimal boundary controller: fractional order or integer order?

In this section, the following question will be answered via simulation method: why use the fractional order controller ($0 < \mu < 1$) instead of the integer order controller ($\mu = 1$)? We will show that the fractional order controller can have better performance than the integer order controller. The performance comparison of two kinds of controllers can only be achieved when both of them are optimal. In the simulation procedure described in §2, if the numerical values of k and μ are replaced by corresponding symbolic symbols, the symbolic expression of $U(x, s)$ with k and μ embedded can be obtained. The exact expression of $U(x, s)$ is shown in appendix. The availability of $U(x, s)$ allows the following optimization to be performed.

We define the following objective function, similar to comparing the settling time, yet easier to implement.

For integer order boundary controllers ($\mu = 1$), we seek the best gain k to

$$\min_k J(k) = \max(|u(1, t)|), \quad t \in [t_f - T, t_f], \quad (18)$$

subject to: $k > 0$.

For fractional order boundary controllers ($0 < \mu \leq 1$), the task is to find the best gain k and the fractional order μ to

$$\min_{k, \mu} J(k, \mu) = \max(|u(1, t)|), \quad t \in [t_f - T, t_f], \quad (19)$$

subject to: $k > 0$ and $0 < \mu \leq 1$.

In the above optimization tasks, $u(1, t)$ is the displacement of the free end of the cable; t_f is the total time of simulation; T is the time period to optimize within the time interval $[t_f - T, t_f]$ and is determined by trial-and-error. Here we choose $t_f = 5$ and $T = 1$.

With $\alpha = 1.5$, the optimization results show that the parameters of optimal fractional order controller are $k^* = 0.7608$ and $\mu^* = 0.9275$. The optimal integer order boundary controller is with gain $k^* = 1.1453$.

The comparison of the free end displacement between the optimal fractional order boundary controller and optimal integer order boundary controller is shown in figure 3. From figure 3 we can see that the response to the optimal fractional order boundary controller not only has shorter rise time and settling time, but also has no overshoot.

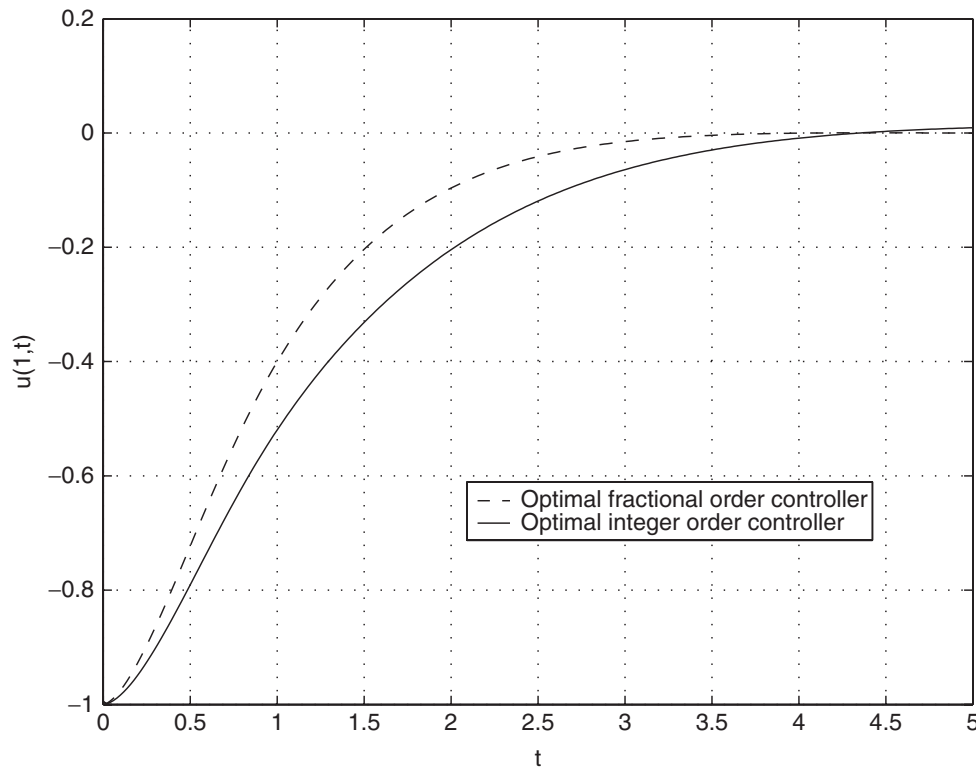


Figure 3. Comparison between two optimal boundary controllers for $\alpha = 1.5$.

3.2 Boundary control of fractional diffusion-wave equations with delays using the Smith predictor

3.2.1 Problem formulation. In this section, we consider the presence of a time delay θ in the boundary measurement, shown as follows:

$$f(t) = -ku_t^{(\mu)}(1, t - \theta), \tag{20}$$

where θ is the time delay.

The situation is also illustrated in figure 4, where $P(s)$ is the transfer function of the plant and $C(s)$ is the Laplace transform of the controller. In our case,

$$C(s) = ks^\mu. \tag{21}$$

Via simulation, the transfer function $P(s)$ can be obtained as

$$P(s) = \left(e^{2s^{9/10}} - 1 \right) s^{-9/10} \left(e^{2s^{9/10}} + 1 \right)^{-1} \tag{22}$$

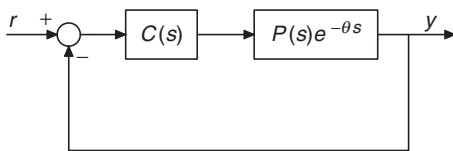


Figure 4. A feedback control system with a time delay.

It is well-known that a delay in a control system usually reduces the phase margin, or even worse, makes the system unstable. What if a delay exists in the boundary control of diffusion-wave equations using a fractional order controller? With $\theta=0.6$ and other parameters the same as in §2, the simulation results are shown in figure 5 and figure 6. We can see that the system becomes unstable with the presence of a delay.

3.2.2 A brief introduction to the Smith predictor. The Smith predictor was proposed by Smith (1957) and is probably the most famous method for control of systems with time delays (Levine 1996, Wang *et al.* 1999). Consider a typical feedback control system with a time delay in figure 4, where $C(s)$ is the controller; $P(s)e^{-\theta s}$ is the plant with a time delay θ .

With the presence of the time delay, the transfer function of the closed-loop system relating the output $y(s)$ to the reference $r(s)$ becomes

$$\frac{y(s)}{r(s)} = \frac{C(s)P(s)e^{-\theta s}}{1 + C(s)P(s)e^{-\theta s}}. \tag{23}$$

Obviously, the time delay θ directly changes the closed-loop poles. Usually, the time delay reduces the stability margin of the control system, or more seriously, destabilizes the system.

The classical configuration of a system containing a Smith predictor is depicted in figure 7, where $\hat{P}(s)$ is the assumed model of $P(s)$ and $\hat{\theta}$ is the assumed delay.

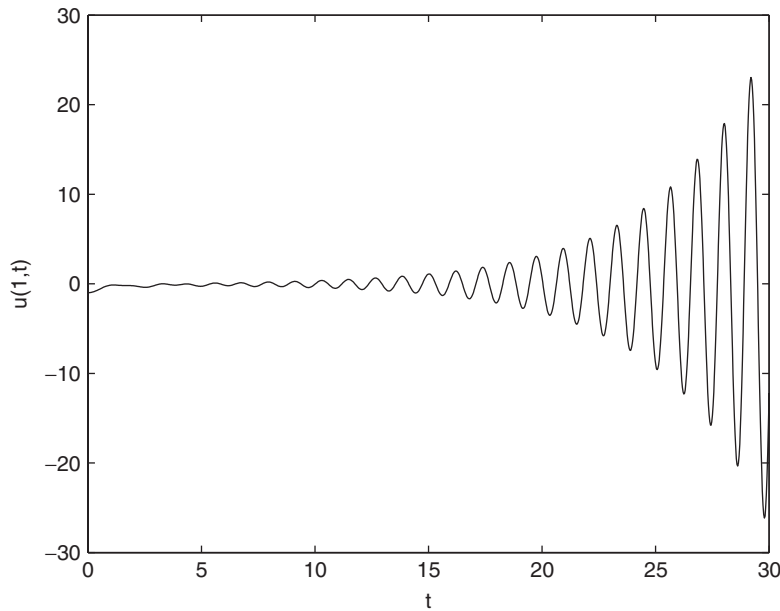


Figure 5. Tip end displacement subject to a delay.

The block $C(s)$ combined with the block $\hat{P}(s) - \hat{P}(s)e^{-\hat{\theta}s}$ is called “the Smith predictor”. If we assume the perfect model matching, *i.e.*, $\hat{P}(s) = P(s)$ and $\theta = \hat{\theta}$, the closed-loop transfer function becomes

$$\frac{y(s)}{r(s)} = \frac{C(s)P(s)e^{-\theta s}}{1 + C(s)P(s)}. \tag{24}$$

Now, it is clear what the underlying idea of the Smith predictor is. With the perfect model matching, the time delay can be removed from the denominator of the transfer function, making the closed-loop stability irrelevant to the time delay.

Based on the controller (21) as $C(s)$, we have the following expression of the boundary controller (the Smith predictor), denoted as $C_{sp}(s)$:

$$C_{sp}(s) = \frac{k_s \mu}{1 + k_s \mu P(s)(1 - e^{-\hat{\theta}s})} \tag{25}$$

In Liang *et al.* (2005), it is proven that the controller (25) is even robust against a small enough difference between θ and $\hat{\theta}$ as long as $\hat{\theta}$ is chosen as the minimal value of the possible delay and the following condition is satisfied

$$\mu < \frac{\alpha}{2} \tag{26}$$

3.2.3 Simulation results. Using the boundary controller (25) and the simulation method proposed in this

paper, the symbolic expression of $U(x, s)$ can be obtained. The exact expression of $U(x, s)$ is omitted since it is more than 1500-character long. The simulation results using the Smith predictor are shown in figures 8 and 9. We can see that the Smith predictor is able to stabilize the system subject to a delay.

4. Concluding remarks

A hybrid symbolic and numerical method based on MATLAB Symbolic Math Toolbox is developed in this paper to simulate some typical problems of boundary control of fractional order diffusion-wave equations. The proposed simulation method can be applied to a wide range of boundary control problems. Furthermore, the transfer function can be calculated in the intermediate steps of the simulation, which can be used to design more advanced boundary controllers. We hope that this method is helpful for researchers to

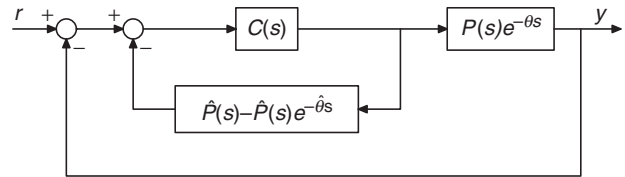


Figure 7. The Smith predictor.

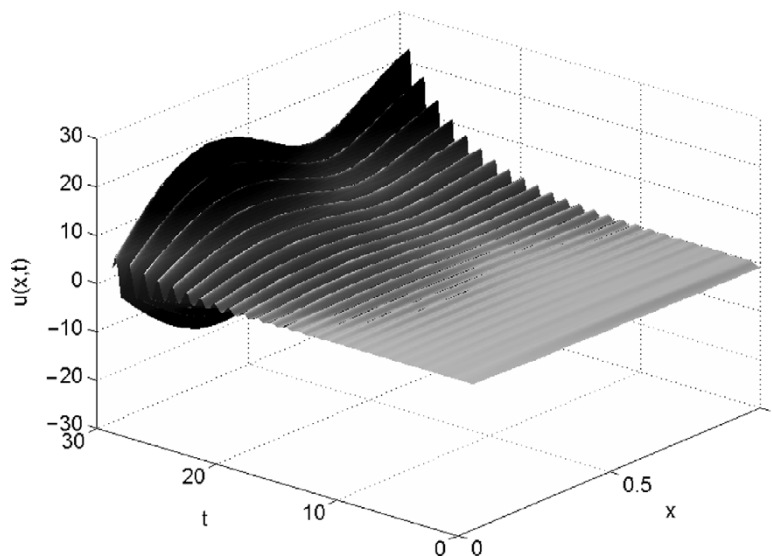


Figure 6. Displacement of the whole string subject to a delay.

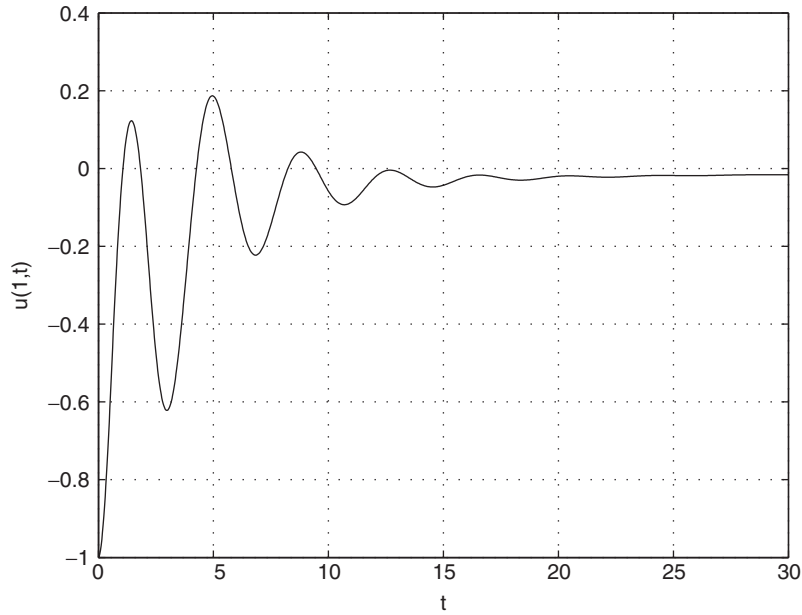


Figure 8. Tip end displacement using the Smith predictor.

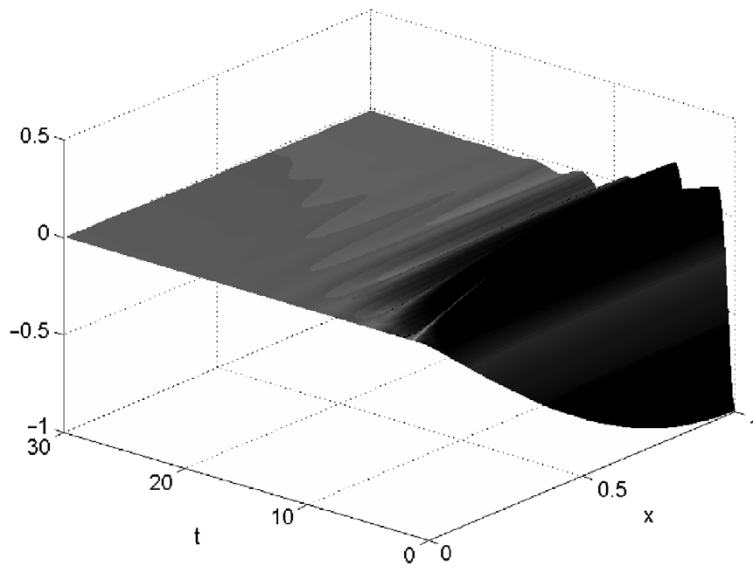


Figure 9. Displacement of the whole string using the Smith predictor.

study boundary control of fractional order diffusion-wave equations.

Acknowledgement

The authors are grateful to two anonymous reviewers' comments which clarified the presentation of this paper.

Since the expression of $U(x,s)$ in §3.1 is too long to be printed within a line, it is copied here in its original Matlab format. The purpose of these appendices is to make the results reported in this paper more easily reproducible by researchers. For reproducibility, the code used in this paper are available upon request. It is also put at http://mechatronics.ece.usu.edu/foc/code/ijc_gures.zip

$$U(x,s)=-\exp(s.^{(1./2.*alpha).*x})*k.*\exp(s.^{(1./2.*alpha)}).*(-4.*s.^{\mu}.*s.^{\alpha+4}.*s.^{(\mu-1)}).$$

$*s.*s.^{\alpha}+2778046668940015./281474976710656.$
 $*s.^{(\mu-1)}.*/s./(4.*s.^{(1./2.*\alpha)}.*exp(s.^{(1./2.*\alpha)}).^2.*s.^{\alpha}+2778046668940015./$
 $281474976710656.*s.^{(1./2.*\alpha)}.*exp(s.^{(1./2.*\alpha)}).^2+4.*s.^{(1./2.*\alpha)}.*s.^{\alpha}+27$
 $78046668940015./281474976710656.*s.^{(1./2.*\alpha)}+4.*k.*s.^{\mu}.*exp(s.^{(1./2.*\alpha)}).^2.$
 $*s.^{\alpha}+2778046668940015./281474976710656.*k.*s.^{\mu}.*exp(s.^{(1./2.*\alpha)}).^2-4.*k.*s.$
 $^{\mu}.*s.^{\alpha}-2778046668940015./281474976710656.*k.*s.^{\mu}+1./exp(s.^{(1./2.*\alpha)}.*x).$
 $*k.*exp(s.^{(1./2.*\alpha)}).*(-4.*s.^{\mu}.*s.^{\alpha}+4.*s.^{(\mu-1)}.*s.*s.^{\alpha}+277804666894$
 $0015./281474976710656.*s.^{(\mu-1)}.*/s./(4.*s.^{(1./2.*\alpha)}.*exp(s.^{(1./2.*\alpha)}).^2.*s.$
 $^{\alpha}+2778046668940015./281474976710656.*s.^{(1./2.*\alpha)}.*exp(s.^{(1./2.*\alpha)}).^2+4.*s.$
 $^{\alpha}+2778046668940015./281474976710656.*s.^{(1./2.*\alpha)}+4.*k.*s.^{\mu}.*exp$
 $(s.^{(1./2.*\alpha)}).^2.*s.^{\alpha}+2778046668940015./281474976710656.*k.*s.^{\mu}.*exp(s.^{(1./2.$
 $*\alpha)}).^2-4.*k.*s.^{\mu}.*s.^{\alpha}-2778046668940015./281474976710656.*k.*s.^{\mu}-4.*s.^{\alpha}./$
 $s.*sin(1./2.*\pi.*x)./(4.*s.^{\alpha}+2778046668940015./281474976710656)$

References

- O.P. Agrawal, "Solution for a fractional diffusion-wave equation defined in a bounded domain", *Nonlinear Dynamics*, 29, pp. 145–155, 2002.
- L. Brančík, "The fast computing method of numerical inversion of Laplace transforms using FFT algorithm", in *Proc. of 5th EDS 98 Int. Conf.*, pp. 97–100, Brno, Czech Republic, June 1998.
- L. Brančík, "An improvement of FFT-based numerical ILT procedure by application of ϵ -algorithm", in *Proceedings of the Seminar STO-7*, pp. 196–199, Brno, Czech Republic, 1999a.
- L. Brančík, "Programs for fast numerical inversion of Laplace transforms in Matlab language environment", in *Proceedings of 7th Conference MATLAB'99*, pp. 27–39, Prague, Czech Republic, 1999b.
- G. Chen, "Energy decay estimates and exact boundary value controllability for the wave equation in a bounded domain", *J. Math. Pure Appl.*, 58, pp. 249–273, 1979.
- G. Chen, M.C. Delfour, A.M. Krall, and G. Payre, "Modelling, stabilization and control of serially connected beams", *SIAM J. Contr. Optimiz.*, 25, pp. 526–546, 1987.
- F. Conrad and Ö. Morgül, "On the stability of a flexible beam with a tip mass", *SIAM Journal on Control and Optimization*, 36, pp. 1962–1986, 1998.
- D.G. Duffy, "On the numerical inversion of laplace transforms: comparison of three new methods on characteristic problems from applications", *ACM Transactions on Mathematical Software*, 19, pp. 333–359, 1993. <http://www.comsol.com>. (accessed 19 July 2006).
- A. Jeffrey, *Advanced Engineering Mathematics*, Burlington, MA, USA, Harcourt/Academic Press, 2002.
- W. Levine (Ed.) *The Control Handbook*, pp. 224–237, Boca Raton, FL, USA, CRC Press, 1996.
- J. Liang, Y.Q. Chen and R. Fullmer, "Simulation studies on the boundary stabilization and disturbance rejection for fractional diffusion-wave equation", *Proceedings of the IEEE 2004 American Control Conference*, Boston, MA, pp. 5010–5015, 2004.
- J. Liang, Y.Q. Chen, B.M. Vinagre, and I. Podlubny, "Boundary stabilization of a fractional wave equation via a fractional order boundary controller", in *The First IFAC Symposium on Fractional Derivatives and Applications (FDA'04)*, Bordeaux, France, July 2004.
- J. Liang, W. Zhang, Y.Q. Chen and I. Podlubny, "Robustness of boundary control of fractional wave equations with delayed boundary measurement using fractional order controller and the Smith predictor", in *Proceedings of 2005 ASME Design Engineering Technical Conferences*, Long Beach, California, USA, 2005.
- F. Mainardi, "Fractional relaxation-oscillation and fractional diffusion-wave phenomena", *Chaos, Solitons, and Fractals*, 7, pp. 1461–1477, 1996.
- F. Mainardi and P. Paradisi, "A model of diffusive waves in viscoelasticity based on fractional calculus", in *Proceedings of the 36th IEEE Conference on Decision and Control*, Hyatt Regency San Diego, California, 1997.
- MathWorks Inc., *Matlab Symbolic Math Toolbox User's Guide*. The Mathworks, Inc., 2002.
- D. Matignon and B. d'Andréa-Novel, "Spectral and time-domain consequences of an integro-differential perturbation of the wave PDE", in *SIAM Proc. of the Third International Conference on Mathematical and Numerical Aspects of Wave Propagation Phenomena*, Mandelieu la Napoule, France, pp. 769–771, 1995.
- A.R. Mitchell and D. Griffiths, *The Finite Difference Method in Partial Differential Equations*, New York, John Wiley & Sons, 1980.
- R.R. Nigmatullin, "Realization of the generalized transfer equation in a medium with fractal geometry", *Phys. Stat. Sol. (b)*, 133, pp. 425–430, 1986.
- W.R. Schneider and W. Wyss, "Fractional diffusion and wave equation", *J. Math. Phys.*, 30, pp. 134–144, 1989.
- O.J.M. Smith, "Closer control of loops with dead time", *Chem. Eng. Progress*, 53, pp. 217–219, 1957.
- Q.-G. Wang, T. H. Lee and K. K. Tan, *Finite Spectrum Assignment for Time-delay Systems*, London, Springer Verlag, 1999.
- W. Wyss, "The fractional diffusion equation", *J. Math. Phys.*, 27, pp. 2782–2785, 1986.