SIMULATION TOOLS IN MAPLE FOR A FRACTIONAL ORDER DISCRETE SYSTEM PERTURBED BY A SEQUENCE OF REAL RANDOM VARIABLES

Mădălina Roxana BUNECI, *University Constantin Brâncuşi of Târgu-Jiu*, ROMÂNIA
Viorica Mariela UNGUREANU, *University Constantin Brâncuşi of Târgu-Jiu*, ROMÂNIA

Abstract. *The purpose of this paper is provide a set of Maple procedures to simulate a general fractional order discrete systems perturbed by a sequence of real random variables.*

Keywords: fractional order operator; fractional order discrete system; discrete-time stochastic system.
.

1. INTRODUCTION

It is well known that fractional order discrete systems have notable applications in various areas and can be utilized in control and systems modeling (for instance, in some recent results and simulation tools (in Matlab) in the area of application of fractional order system models can be found in [1-2] and [4]). The aim of this paper is to provide a set of Maple procedures to simulate fractional order discrete systems perturbed by sequences of real random variables. We use symbolic manipulations by Statistics package of Maple. The reference indicated in documentation of the Statistics package is [5].

Let us start by presenting the mathematical model that we will use. The fractional order operator Δ^{α} is given according Grünwald-Letnikov's definition:

$$\Delta^{\alpha}x_{k+1}=\frac{1}{h^{\alpha}}\sum_{i=0}^{k+1}(-1)^{i}\binom{i}{\alpha}x_{k+1-i},$$

where h>0 is the sampling period or the time increment and

$$\binom{i}{\alpha}=\begin{cases}1, i=0\\ \dfrac{\alpha(\alpha-1)...(\alpha-i+1)}{i!}, i\in\mathbb{N}^{*}.\end{cases}$$

We consider a probability space (Ω, $\mathcal{F}$, P), where Ω is the sample space, F is the σ-algebra of the events and P is a probability measure on $\mathcal{F}$. If $(x_k)_k$ is a sequence of measurable functions $x_k=(x_{k,1}, x_{k,2},\ldots, x_{k,d}) : \Omega\to\mathbb{R}^d$ with respect to σ-algebra $\mathcal{F}$ on Ω and the Borel structure on $\mathbb{R}^d$, $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d)\in\mathbb{R}^d$ and $h=(h_1, h_2, \ldots, h_d)\in\mathbb{R}^d$ ($h_j>0$ for all $j\in\{1,2, \ldots, d\}$), then by $\Delta^{[\alpha]}x_{k+1}$ we mean the function $y=(y_1,y_2,..:, y_d): \Omega\to\mathbb{R}^d$ defined by

$$y_j(\omega)=\Delta^{\alpha_j}x_{k+1,j}(\omega)=\frac{1}{h_j^{\alpha_j}}\sum_{i=0}^{k+1}(-1)^{i}\binom{i}{\alpha_j}x_{k+1-i,j}(\omega),\ \text{for all } \omega\in\Omega \text{ and all } j\in\{1,2, \ldots, d\}.$$

Obviously, $y : \Omega \to \mathbb{R}^d$ is measurable with respect to σ-algebra $\mathcal{F}$ on Ω and the Borel structure on $\mathbb{R}^d$.

The general discrete-time fractional order system that we treat has the form:

$$\Delta^{[\alpha]} x_{k+1} = f\left(x_k, \xi_k, k\right),\ k \in \mathbb{N},$$

where $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d) \in \mathbb{R}^d$, $f=(f_1,f_2,..:, f_d)$: $\mathbb{R}^{d+2} \to \mathbb{R}^d$ is a measurable function, $(\xi_k)_k$ is a sequence of random variables on (Ω, $\mathcal{F}$, P) (real measurable functions on Ω) modelling the stochastic perturbations and the function $f\left(x_k, \xi_k, k\right):\Omega \to \mathbb{R}^d$ is defined by

$$f\left(x_k, \xi_k, k\right)(\omega) = f\left(x_k(\omega), \xi_k(\omega), k\right),\ \text{for all } \omega \in \Omega.$$

2. MAPLE PROCEDURES FOR SIMULATING DISCRETE-TIME FRACTIONAL ORDER SYSTEMS PERTURBED BY SEQUENCES OF RANDOM VARIABLES

The purpose of this section is to provide Maple procedures to simulate general discrete-time fractional order systems. As a matter of fact, the first procedure that we provide can be used to simulate more general discrete-time stochastic systems perturbed by sequences of real random variables. More precisely, the considered systems have of the form

$$x_{k+1,j} = \sum_{i=1}^{k+1} c_{i,j} x_{k+1-i,j} + f_j\left(x_k, \xi_k, k\right),\ j \in \{1,2,...,d\},\ k \in \mathbb{N}, \quad (2.1)$$

where $(c_{i,1}, c_{i,2}, \ldots, c_{i,d}) \in \mathbb{R}^d$ for all $i \in \mathbb{N}^*$, $f=(f_1,f_2,..:, f_d)$: $\mathbb{R}^{d+2} \to \mathbb{R}^d$ is a measurable function and $(\xi_k)_k$ is a sequence of random variables on (Ω, $\mathcal{F}$, P). For computational purposes it is useful to consider samples drawn from the same distributions as of the random variables ξ_k, $k \in \mathbb{N}$. Thus we will start with a structure $w=(w[i][j])_{i,j}$, where for all i,

$$\{w[i][1], w[i][1], \ldots, w[i][m]\}$$

is a random sample of size m drawn from the same distribution as ξ_i. Consequently, for each k instead of each scalar component $x_{k,j}$ in (2.1) we shall obtain a sample

$$[x[k,j,1],x[k,j,2],\ldots,x[k,j,m]]$$

In the below procedure DiscreteStochasticSyst we take an index $p \in \{1,2,\ldots,m\}$ and a natural number $n \in \mathbb{N}$ and we compute x[k,j,p] for all $k \in \{0,1,\ldots,n\}$ and all $j \in \{1,2,\ldots,d\}$. The formal parameters of the procedure DiscreteStochasticSyst have the following signification:

- p is the index in the sample.
- n is the number of iterations using (2.1)
- w is the structure containing random samples drawn from the same distributions as of stochastic perturbations ξ_i
- f=[f1,f2,…,fd] is the list of the scalar components of the function f in (2.1)
- x0 is a list containing the components of the initial state of the system (2.1) : x0[j]=x[0,j,p].

- c is an array containing the coefficients $(c_{i,1}, c_{i,2}, \ldots, c_{i,d}) \in \mathbb{R}^d$ in (2.1) for $i \in \{1,2,\ldots,n\}$ (c[j,i]= $c_{i,j}$).

```
>DiscreteStochasticSyst := proc (c, x0, f, w, n, p)
local j, k, i, x, d, halpha;
d := nops(x0);
x := array(0 .. n, 1 .. d);
for i to d do
 x[0, i] := x0[i];
 x[1, i] := c[i, 1]*x0[i]+f[i](seq(x0[j], j = 1 .. d), w[1][p], 1)
end do;
for k to n-1 do
 for i to d do
  x[k+1, i] := sum(c[i, j]*x[k+1-j, i], j = 1 .. k+1)+f[i](seq(x[k, j], j = 1 .. d), w[k+1][p], k+1)
 end do
end do;
return x
end proc;
```

Let us consider the general discrete-time fractional order system

$$\Delta^{[\alpha]} x_{k+1} = f\left(x_k, \xi_k, k\right),\ k \in \mathbb{N}, \quad (2.2)$$

where $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d) \in \mathbb{R}^d$, $f=(f_1,f_2,..:, f_d)$: $\mathbb{R}^{d+2} \to \mathbb{R}^d$ is a measurable function and $(\xi_k)_k$ is a sequence of random variables on (Ω, $\mathcal{F}$, P) modelling the stochastic perturbations. Using the definition of $\Delta^{[\alpha]} x_{k+1}$ and taking $h=(h_1, h_2, \ldots, h_d) \in \mathbb{R}^d$ ($h_j>0$ for all $j \in \{1,2, \ldots, d\}$) as the vector of sampling periods or the time increments, the system (2.2) can be written as

$$x_{k+1,j} = \sum_{i=1}^{k+1} c_{i,j} x_{k+1-i,j} + g_j\left(x_k, \xi_k, k\right),\ j \in \{1,2,...,d\},\ k \in \mathbb{N}, (2.3)$$

where for all for $i \in \mathbb{N}^*$ and $j \in \{1,2,..,d\}$,

$$c_{i,j} = (-1)^{i+1} \binom{i}{\alpha_j} = (-1)^{i+1} \frac{\alpha_j(\alpha_j - 1)...(\alpha_j - i + 1)}{i!}$$

$g_j = h_j^{\alpha_j} f_j$.

The following procedure GenFractCoeff generates the coefficients in (2.3). More precisely, it returns an array c such that c[j,k]= $c_{k,j}$ for all $k\in\{1,2,\ldots,n\}$ and $j\in\{1,2,..,d\}$. Its formal parameters are the list $[\alpha_1, \alpha_2, \ldots, \alpha_d]$ and n.

```
> GenFractCoeff := proc (alpha, n)
local i, k, x, c, d;
d := nops(alpha);
c := array(1 .. d, 1 .. n);
for i to d do c[i, 1] := alpha[i] end do;
 for k to n-1 do for i to d do c[i, k+1] := -c[i, k]*(alpha[i]-k)/(k+1) end do
end do;
return c
end proc;
```

```
> evalm(GenFractCoeff([1/2], 4));
```

$$\frac{1}{2} \quad \frac{1}{8} \quad \frac{1}{16} \quad \frac{5}{128}$$

The procedure hpower (alpha, h) computes $\left[h_1^{\alpha_1}, h_2^{\alpha_2}, ..., h_d^{\alpha_d}\right]$:

```
> hpower := proc (alpha, h)
local d, i, halpha;
d := nops(alpha); halpha := [seq(1, i = 1 .. d)];
for i to d do halpha[i] := h[i]^alpha[i] end do;
return halpha
end proc;
```

The following procedure DFStochasticSyst computes the samples

$$[x[k,j,1],x[k,j,2],\ldots,x[k,j,m]]$$

for all $k\in\{0,1,\ldots,n\}$ and all $j\in\{1,2,\ldots,d\}$ associated to $x_k=(x_{k,1}, x_{k,2},\ldots, x_{k,d})$ from the general discrete-time fractional order system (2.1) written in the equivalent form (2.3).

The formal parameters of the procedure DFStochasticSyst have the following sense:

- alpha is the list [α_1, α_2, …, α_d], where α_1, α_2, …, α_d are the fractional orders in (2.2)
- h is the list [h_1, h_2, …, h_d], where (h_1, h_2, …, h_d) is the vector of sampling periods in (2.2)
- [f1,f2,…,fd] is the list of the scalar components of the function f in (2.2)
- S is the structure containing random samples drawn from the same distributions as of stochastic perturbations ξ_i
- x0 is a list containing the components of the initial state of the system (2.2)
- m is the common size of the random samples.

```
DFStochasticSyst := proc (alpha, h, f, S, x0, n, m)
local i, j, k, c, halpha, g, gi, x, xp, p, d;
d := nops(alpha);
c := GenFractCoeff(alpha, n);
halpha := hpower(alpha, h);
g := [];
for i to d do
 gi := unapply(halpha[i]*f[i](seq(varx[j], j = 1 .. d), w, k), seq(varx[j], j = 1 .. d), w, k);
 g := [op(g), gi]
end do;
x := array(0 .. n, 1 .. d, 1 .. m);
for p to m do
 xp := DiscreteStochasticSyst(c, x0, g, S, n, p);
 for k from 0 to n do
  for i to d do x[k, i, p] := xp[k, i] end do
 end do
end do;
return x
end proc;
```

The procedure GenStP generates for each $k \in \{1,2,\ldots,n\}$ a random sample of size m drawn from the same distribution as the random variables ξ_k. It returns the list of the generated random samples. Its formal parameters are the natural numbers n, m and the list of the distributions of ξ_1, ξ_2, …, ξ_n.

```
>GenStP := proc (n, m, distribution)
 local i, j, S, Xi, Samples;
```

```
 Samples := [seq(array(1 .. m), i = 1 .. n)];
 for i to n do
Xi := Statistics:-RandomVariable(distribution[i]);
S := Statistics:-Sample(Xi, m);
for j to m do Samples[i][j] := S[j] end do
end do;
return Samples
end proc;
>GenStP(2, 5, [Normal(0, 1), Laplace(0, 2)]);
[[.229736818053089,  -2.35415622029595, -1.06135420331567}],
  [.905157006340350, .453739519868418,  -1.00238320871061]]
```

Now we can write a variant of the procedure DFStochasticSyst, using the of the distributions of $\xi_1, \xi_2, \ldots, \xi_n$ instead of the structure S:

```
DFStochasticSystD := proc (alpha, h, f, distribution, x0, n, m)
local i, j, k, c, S, halpha, g, gi, x, xp, p, d;
d := nops(alpha);
c := GenFractCoeff(alpha, n);
halpha := hpower(alpha, h); g := [];
for i to d do
 gi := unapply(halpha[i]*f[i](seq(varx[j], j = 1 .. d), w, k), seq(varx[j], j = 1 .. d), w, k);
 g := [op(g), gi]
end do;
S := GenStP(n, m, distribution);
x := array(0 .. n, 1 .. d, 1 .. m);
for p to m do
 xp := DiscreteStochasticSyst(c, x0, g, S, n, p);
 for k from 0 to n do
  for i to d do x[k, i, p] := xp[k, i] end do
 end do
end do;
return x
```

end proc;

> x := DFStochasticSystD([3/4], [1], [proc (x, w, k) options operator, arrow; -x^2/(4+k)+(1/20)*w*x end proc], [seq(Normal(0, 1), i = 1 .. 5)], [1], 5, 8):

> for i to 8 do printf("%f ", x[4, 1, i]) end do;

0.435149 0.394981 0.408017 0.351033 0.345522 0.338785 0.394070 0.370531

(the sample corresponding to x_4)

3. LINEAR CASE

Let us consider the linear discrete-time fractional order system with multiplicative noise

$$\Delta^{[\alpha]}x_{k+1} = A_k x_k + \xi_k W_k x_k,\ k \in \mathbb{N},\ (3.1)$$

where $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d)\in \mathbb{R}^d$, $A_k, W_k \in M_{d,d}(\mathbb{R})$ for all $k\in \mathbb{N}$, and $(\xi_k)_k$ is a sequence of random variables on (Ω, $\mathcal{F}$, P) modelling the stochastic perturbations.

The following procedure FLinear returns the list [f1,f2,…,fd] of the scalar components of the function f so that the system (3.1) can be equivalently written $\Delta^{[\alpha]}x_{k+1} = f(x_k, \xi_k, k),\ k \in \mathbb{N}$. Its formal parameters are: the lists: $\left[h_1^{\alpha_1}, h_2^{\alpha_2}, ..., h_d^{\alpha_d}\right]$, [$A_0$, A_1, …,A_{n-1}] and [W_0,W_1,…,W_{n-1}].

```
>FLinear := proc (halpha, A, W)
local i, j, k, f, fj, d;
d := nops(halpha); f := [];
for j to d do
  fj := unapply(halpha[j]*(sum((A[k][j, i]+w*W[k][j, i])*varx[i], i = 1 .. d)), seq(varx[i], i = 1 .. d), w, k);
  f := [op(f), fj]
end do;
return f
end proc;
```

Thus we can provide a variant of the procedure DFStochasticSystD for the linear case:

```
>DFLSyst := proc (alpha, h, A, W, distribution, x0, n, m)
local i, j, k, c, halpha, f, S, x, xp, p, d;
```

```
 d := nops(alpha);
c := GenFractCoeff(alpha, n);
halpha := hpower(alpha, h);
f := FLinear(halpha, A, W);
S := GenStP(n, m, distribution);
x := array(0 .. n, 1 .. d, 1 .. m);
for p to m do
xp := DiscreteStochasticSyst(c, x0, f, S, n, p);
for k from 0 to n do
for i to d do x[k, i, p] := xp[k, i] end do
end do
end do;
return x
end proc;
```

BIBLIOGRAPHY

[1] A. Dzielinski and D. Sierociuk, *Simulation and experimental tools for fractional order control education*, Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

[2] A. Dzielinski, D. Sierociuk and G. Sarwas, *Some applications of fractional order calculus, Bulletin of The Polish Academy of Sciences Technical Sciences*, Vol. 58, No. 4, 2010.

[3] Z. A. Karian and E. A. Tanis, *Probability and Statistics Explorations with MAPLE*, Second Edition, Pearson Prentice Hall, 2008

[4] P. W. Ostalczyk, *Fractional-order linear digital 1D and 2D filter response calculation using Matlab*, Int. J. Dynam. Control, 2016. DOI 10.1007/s40435-016-0227-0.

[5] A. Stuart and K. Ord, *Kendall's Advanced Theory of Statistics. Vol. 1: Distribution Theory*. 6th ed. London: Edward Arnold, 1998.