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

Mădălina Roxana Buneci, *University Constantin Brâncuşi of Târgu-Jiu,* ROMÂNIA

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

Keywords: visualization tools; fractional order discrete system; discrete-time stochastic system.

1. INTRODUCTION

We shall use the same notation and terminology as in [1]. The aim of this paper is to provide a suite of Maple procedures to visualize the behavior of fractional order discrete systems perturbed by sequences of real random variables:

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

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

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

In [1] we provided Maple procedures for computing x_1, x_2, ..., x_n assuming that x_0, α, h (vector of sampling periods), f and $(\xi_k)_k$ (or their distributions) are given. For computational purposes we considered in [1] samples drawn from the same distributions as of the random variables ξ_k, $k \in \mathbb{N}$. Consequently, for each k instead of each scalar component $x_{k,j}$ we obtained a sample [x[k,j,1],x[k,j,2],…,x[k,j, m]]. Thus the results of the procedures in [1] are three-dimensional arrays x. The procedures in this paper, which is a continuation of [1], will have x as formal parameter.

2. A MAPLE PROCEDURE FOR SIMULATING A DETERMINISTIC DISCRETE-TIME FRACTIONAL ORDER SYSTEM

In order to compare the behavior of a fractional order discrete system perturbed by a sequence of real random variables with the behavior of a deterministic fractional order discrete system, in this section we provide a Maple procedure to simulate (deterministic) general discrete-time fractional order systems. More precisely, the considered systems have of the form

$$\Delta^{[\alpha]}x_{k+1} = f\left(x_k, k\right),\ k \in \mathbb{N},\ (2.1)$$

where $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d) \in \mathbb{R}^d$ and $f=(f_1,f_2,..:, f_d)$: $\mathbb{R}^{d+2} \rightarrow \mathbb{R}^d$ is a function.

The formal parameters of the procedure DDFSyst are:

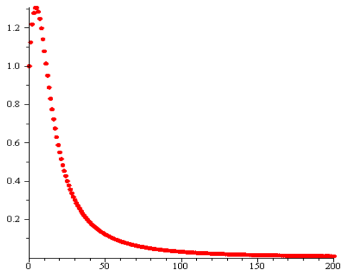
- alpha is the list [α_1, α_2, …, α_d], where α_1, α_2, …, α_d are the fractional orders in (2.1)
- h is the list [h_1, h_2, …, h_d], where (h_1, h_2, …, h_d) is the vector of sampling periods in (2.1)
- [f1,f2,…,fd] is the list of the scalar components of the function f in (2.1)
- x0 is a list, vector or one dimensional array containing the components of the initial state of the system (2.1)
- n is number of iterations performed in (2.1).

The procedure DDFSyst returns a two dimensional array x such that x[k,j]=$x_{k,i}$ for all k∈{0, 1, …, n} and all j∈{1,2,…,d}.

```
> DDFSyst := proc (alpha, h, f, x0, n)
local i, j, k, c, halpha, g, gi, x, d;
d := nops(alpha); c := array(1 .. d, 1 .. n); halpha := [seq(1, i = 1 .. d)];
for i to d do
 c[i, 1] := alpha[i]; halpha[i] := h[i]^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;
g := [];
for i to d do
 gi := unapply(halpha[i]*f[i](seq(varx[j], j = 1 .. d), t), seq(varx[j], j = 1 .. d), t);
 g := [op(g), gi]
end do;
x := array(0 .. n, 1 .. d);
for i to d do x[0, i] := x0[i]; x[1, i] := c[i, 1]*x0[i]+g[i](seq(x0[j], j = 1 .. d), 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)+g[i](seq(x[k, j], j = 1 .. d), k+1)
 end do
end do;
return x
end proc;
```

> n := 200; xd := DFDSyst([3/4], [1], [proc (x, k) options operator, arrow; x^2/(k+1)-(1/8)*x end proc], [1.], n);

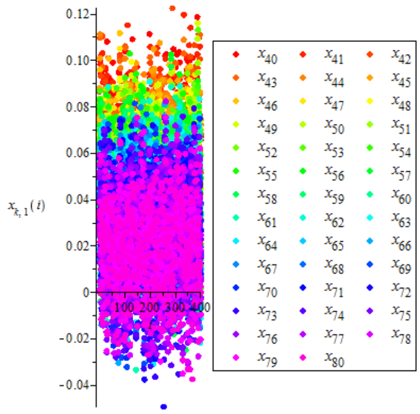
> plots:-pointplot([seq([i, xd[i, 1]], i = 0 .. n)], color = COLOR(RGB, 1, 0, 0), symbol = solidcircle, symbolsize = 12, connect = false);



3. VISUALIZATION TOOLS FOR SCALAR COMPONENTS OF SOLUTION

The purpose of this section is to provide various procedures to visualize the scalar components of x_1, x_2, ..., x_n. In particular they can be used to simulate a one-dimensional system. Their first formal parameter (ax) is the thee-dimensional array described in Section 1. In the following procedure j is the index of the scalar component. The procedure will plot $x_{k,j}$, with k=n1..n2.

```
>visScalarComponent := proc (ax, j, n1, n2)

local m;

m := op(2, op(3, [op(2, op(ax))]));

plots:-display(seq(plots:-pointplot([seq([i, ax[k, j, i]], i = 1 .. m)], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = solidcircle, symbolsize = 12, connect = false, legend = typeset('x'[k]), legendstyle = [location = right], labels = ['i', 'x'['k', j]('i')]), k = n1 .. n2), insequence = false)

end proc;

>visScalarComponent(xs, 1, 40, 80)
```



xs = array retuned by the procedure DFStochasticSystD from [1] applied to the following system:

$$\Delta^{[\alpha]} x_{k+1} = f\left(x_k, \xi_k, k\right),\ k \in \mathbb{N},\ \ x_0 \equiv 1$$

$$f\left(x, w, k\right) = \frac{1}{1+k^2} x^2 - \frac{1}{8} x + \frac{1}{80} w$$

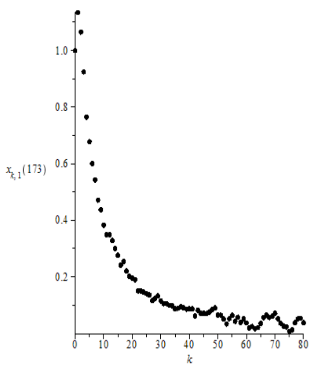
$$\alpha = \left[\frac{3}{4}\right]\ ,\ h=[1]$$

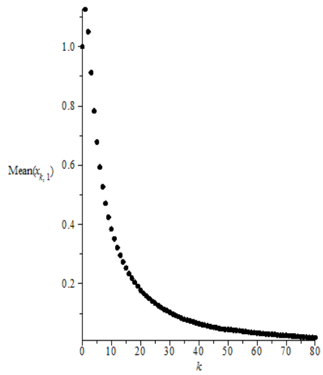
The distribution of each ξ_k is Normal(0,1).

The procedure visScalarComponentI plots $x_{k,j}(i)$ for k=n1..n2, while the procedure visScalarComponentMean plots Mean($x_{k,j}$) for k=n1..n2.

```
>visScalarComponentI := proc (ax, j, i, n1, n2)
plots:-pointplot([seq([k, ax[k, j, i]], k = n1 .. n2)], color = COLOR(RGB, 0, 0, 0), symbol = solidcircle, symbolsize = 12, connect = false, labels = ['k', 'x'['k', j](i)])
end proc;
>visScalarComponentMean := proc (ax, j, n1, n2) plots:-pointplot([seq([k, Statistics:-Mean([seq(ax[k, j, i], i = 1 .. m)])], k = n1 .. n2)], color = COLOR(RGB, 0, 0, 0), symbol = solidcircle, symbolsize = 12, connect = false, labels = ['k', typeset("Mean(", 'x'['k', j], ")")]) end proc;
```

The below left image is obtained as a result of visScalarComponentI(xs, 1, 173, 0, 80), while the right one as the result of visScalarComponentMean(xs, 1, 0, 80).





The following procedures compare the behavior of a fractional order discrete system perturbed by a sequence of real random variables versus a deterministic one.

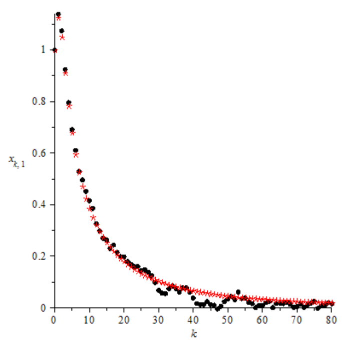
```
>visScalarComponentIVsDet := proc (ax, axd, j, i, n1, n2)
```

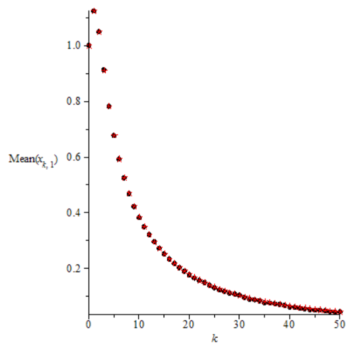
plots:-display(plots:-pointplot([seq([k, ax[k, j, i]], k = n1 .. n2)], color = COLOR(RGB, 0, 0, 0), symbol = solidcircle, symbolsize = 12, connect = false), plots:-pointplot([seq([k, axd[k, j]], k = n1 .. n2)], color = COLOR(RGB, 1, 0, 0), symbol = asterisk, symbolsize = 12, connect = false, labels = ['k', 'x'['k', j]]))

end proc;

>visScalarComponentMean := proc (ax, j, n1, n2) local m; m := op(2, op(3, [op(2, op(ax))])); plots:-pointplot([seq([k, Statistics:-Mean([seq(ax[k, j, i], i = 1 .. m)])], k = n1 .. n2)], color = COLOR(RGB, 0, 0, 0), symbol = solidcircle, symbolsize = 12, connect = false, labels = ['k', typeset("Mean(", 'x'['k', j], ")")]) end proc;

The left image is obtain as a result visScalarComponentIVsDet(xs, 1, 48, 0, 80), while the right one as the result of visScalarComponentMeanVsDet(xs, 1, 0, 80).





4. VISUALIZATION TOOLS FOR TWO AND THREE-DIMENSIONAL SYSTEMS

The below procedure visDFStochasticSyst2D (respectively, visDFStochasticSyst3D) plots $x_k \in \mathbb{R}^2$ (respectively, $\mathbb{R}^3$) with k=n1..n2.

>visDFStochasticSyst2D := proc (ax, n1, n2)
local m;

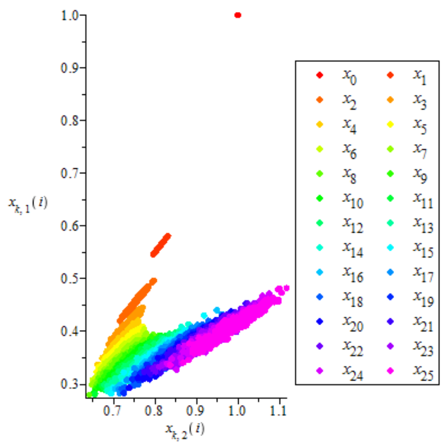
m := op(2, op(3, [op(2, op(ax))]));

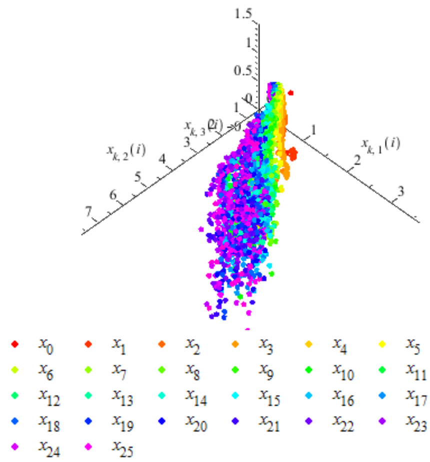
plots:-display(seq(plots:-pointplot([seq([ax[k, 1, i], ax[k, 2, i]], i = 1 .. m)], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = solidcircle, symbolsize = 12, connect = false, labels = [typeset('x'['k', 2]('i')), typeset('x'['k', 1]('i'))], legend = typeset('x'[k]), legendstyle = [location = right]), k = n1 .. n2), insequence = false)

end proc;

>visDFStochasticSyst3D := proc (ax, n1, n2) local m; m := op(2, op(3, [op(2, op(ax))])); plots:-display(seq(plots:-pointplot3d([seq([ax[k, 1, i], ax[k, 2, i], ax[k, 3, i]], i = 1 .. m)], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = solidcircle, symbolsize = 12,

connect = false), k = n1 .. n2), labels = [typeset('x'['k', 2]('i')), typeset('x'['k', 1]('i')), typeset('x'['k', 3]('i'))], axes = normal, insequence = false) end proc;
The below left image is obtained as a result visDFStochasticSyst2D(xs2, 0, 50), while the right one as the result of visDFStochasticSyst3D(xs3, 1, 0, 50).



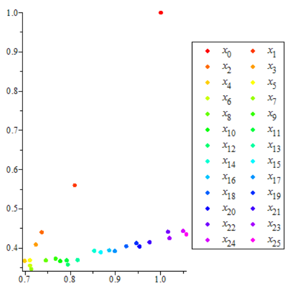


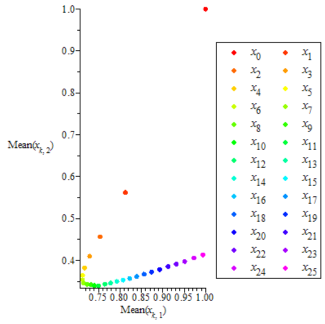
xs2 = array retuned by the procedure DFStochasticSystD from [1] applied to the following system: $\Delta^{[\alpha]}x_{k+1} = f(x_k, \xi_k, k),\ k \in \mathbb{N},\ x_0 \equiv (1,1)$ $f(x,y,w,k) = \begin{pmatrix} -\frac{1}{8}\sin(kx) + \frac{1}{k+1}y + \frac{1}{80}w \\ \frac{1}{6}x - \frac{1}{80}\cos(xy)w \end{pmatrix}^t$ $\alpha = \left[\frac{3}{4}, \frac{1}{2}\right]$, h=[1,1] The distribution of all ξ_k is Normal(0,1).	xs3 = array retuned by the procedure DFStochasticSystD from [1] applied to : $\Delta^{[\alpha]}x_{k+1} = f(x_k, \xi_k, k),\ k \in \mathbb{N},\ x_0 \equiv (1,1,1)$ $f(x,y,z,w,k) = \begin{pmatrix} -\frac{\sin(kx)}{8} + \frac{yz}{k+1} + \frac{1}{80}w \\ \frac{x}{16} - \frac{y}{4} + \frac{2z}{3} + \frac{w}{20} \\ \cos(xyz) - \frac{w}{30} \end{pmatrix}^t$ $\alpha = \left[\frac{3}{4}, \frac{1}{2}, -\frac{1}{2}\right]$, h=[1,1,1] The distribution of all ξ_k is Normal(0,1).

The following procedure visDFStochasticSystI2D plots $x_{k,j}(i)$ for k=n1..n2 and the procedure visDFStochasticSystMean2D plots Mean($x_{k,j}$) for k=n1..n2.

```
>visDFStochasticSystI2D := proc (ax, i, n1, n2)
plots:-display(seq(plots:-pointplot([ax[k, 1, i], ax[k, 2, i]], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = solidcircle, symbolsize = 12, connect = false, legend = typeset('x'[k]), legendstyle = [location = right]), k = n1 .. n2))
end proc;
>visDFStochasticSystMean2D:=proc(ax,n1,n2)
plots:-display(seq(plots:-pointplot([Statistics:-Mean([seq(ax[k, 1, i], i = 1 .. m)]),
Statistics:-Mean([seq(ax[k, 2, i], i = 1 .. m)])],color=COLOR(HSV, 7*(k-n1)/(8*(n2-n1+1)),
```

1, 1),symbol= solidcircle, symbolsize=12,labels =
[typeset("Mean(",'x'['k',1],")"),typeset("Mean(",'x'['k',2],")")],legend = typeset('x'[k]),
legendstyle = [location = right]),k=n1..n2))
end proc;



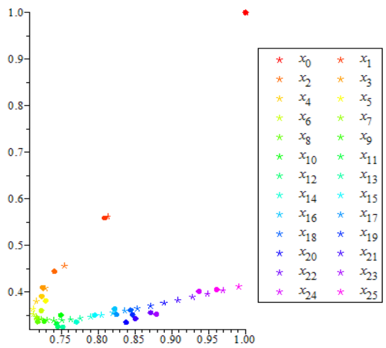


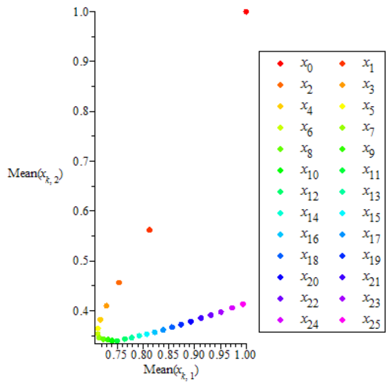
The above left image is obtained calling visDFStochasticSystI2D(xs2, 94, 0, 25), while the right one as the result of visDFStochasticSystMean2D(xs2, 0, 25).

The following procedures compare the behavior of a fractional order discrete two-dimensional systems perturbed by sequences of real random variables versus a deterministic one.

>visDFStochasticSystI2DVsDet := proc (ax, axd, i, n1, n2)
plots:-display(seq(plots:-pointplot([ax[k, 1, i], ax[k, 2, i]], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = solidcircle, symbolsize = 12), k = n1 .. n2), seq(plots:-pointplot([axd[k, 1], axd[k, 2]], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = asterisk, symbolsize = 12, legend = typeset('x'[k]), legendstyle = [location = right]), k = n1 .. n2))
end proc

>visDFStochasticSystMean2DVsDet := proc (ax, axd, n1, n2)
local m; m := op(2, op(3, [op(2, op(ax))])); plots:-display(seq(plots:-pointplot([Statistics:-Mean([seq(ax[k, 1, i], i = 1 .. m)]), Statistics:-Mean([seq(ax[k, 2, i], i = 1 .. m)])], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = solidcircle, symbolsize = 12), k = n1 .. n2), seq(plots:-pointplot([axd[k, 1], axd[k, 2]], color = COLOR(HSV, (7*k-7*n1)/(8*n2-8*n1+8), 1, 1), symbol = asterisk, symbolsize = 12, legend = typeset('x'[k]), legendstyle = [location = right]), k = n1 .. n2))
end proc;





BIBLIOGRAPHY

[1] M. Buneci and V.M. Ungureanu, Simulation tools in MAPLE for a fractional order discrete system perturbed by a sequence of real random variables, Fiabilitate şi durabilitate (Fiability & durability), SYMECH 2016.

[2] 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.

[3]***Maple User Manual, http://www.maplesoft.com/