SIMULATION TOOLS IN MAPLE FOR FRACTIONAL ORDER STOCHASTIC DISCRETE-TIME CONTROL SYSTEMS

Mădălina Roxana Buneci, University Constantin Brâncuşi of Târgu-Jiu, ROMANIA

ABSTRACT: Simulation tools (in Matlab) for linear discrete-time fractional order system as well as for discrete-time linear fractional order stochastic systems (with additive noise) can be found in [3]. In this paper we intend to approach the nonlinear case. More precisely, the purpose of this paper is to provide a small collection of procedures for simulating nonlinear fractional order stochastic discrete-time control systems. Compared with the procedures offered in [1], these procedures allow the implementation of models of discrete-time fractional order stochastic control systems with multidimensional stochastic perturbations.

KEY WORDS: fractional order operator; fractional order discrete-time system; discrete-time stochastic system; control system.

1. NOTATION AND TERMINOLOGY

We begin by presenting the necessary notation and definitions from discrete fractional calculus theory. Let Ω be a set, $(x_k)_k$ a sequence of functions $x_k=(x_{k,1}, x_{k,2},\ldots, x_{k,d}) : \Omega\to\mathbb{R}^d$, $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d)\in\mathbb{R}^d$ and $h=(h_1, h_2, \ldots, h_d)\in\mathbb{R}^d$ ($h_j>0$ for all $j\in\{1,2, \ldots, d\}$). By $\Delta^{[\alpha]}x_{k+1}$ we mean the function $\tilde{x}=(\tilde{x}_1, \tilde{x}_2,\ldots, \tilde{x}_d)$: $\Omega\to\mathbb{R}^d$, where for all $\omega\in\Omega$ and all $j\in\{1,2, \ldots, d\}$, $\tilde{x}_j(\omega)=\Delta^{\alpha_j}x_{k+1,j}(\omega)$ is defined by

$$\Delta^{\alpha_j}x_{k+1,j}(\omega)=\frac{1}{h_j^{\alpha_j}}\sum_{i=0}^{k+1}(-1)^i\binom{\alpha_j}{i}x_{k+1-i,j}(\omega),$$

and

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

(Grünwald-Letnikov's definition of the fractional order operator Δ^{α_j}).

In the subsequent considerations in this paper (Ω, F, P) is probability space (Ω, F, P) (Ω is the sample space, F is the σ-algebra of the events and P is a probability measure on F) and $\mathbb{R}^n$ is endowed with the Borel structure for all n.

We shall consider the following general fractional order stochastic discrete-time control system:

$$\begin{cases}\Delta^{[\alpha]}x_{k+1}=f(x_k,u_k,\xi_k,k)\\ y_k=g(x_k,\eta_k,k),\ k\in\mathbb{N},\end{cases}\qquad(1.1)$$

where

- $x_k \in\{x: \Omega\to X\subset\mathbb{R}^d$ measurable$\}$ is the state at time k and X is the state space.
- $u_k \in\{u: \Omega\to U \subset\mathbb{R}^p$ measurable$\}$ is the control input applied at time k and U is the control input space
- $y_k \in\{y: \Omega\to Y \subset\mathbb{R}^q$ measurable$\}$ is the output at time k, and Y is the output space.

- $\xi=(\xi_k)_k$ is a stochastic process; $\xi_k:\Omega\to W \subset \mathbb{R}^{s_1}$ is a measurable function modelling the stochastic perturbation (noise) entering the state equation at time k.
- $\eta=(\eta_k)_k$ is a stochastic process; $\eta_k:\Omega\to V \subset \mathbb{R}^{s_2}$ is a measurable function modelling the stochastic perturbation (noise) entering the observation equation at time k.
- $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d)\in\mathbb{R}^d$
- $f=(f_1,f_2,\ldots, f_d)$: $X\times U\times W\times\mathbb{N} \subset \mathbb{R}^{d+p+s_1+1}\to\mathbb{R}^d$ is a measurable function.
- $g=(g_1,g_2,\ldots, g_q)$: $X\times Y\times V\times\mathbb{N} \subset \mathbb{R}^{d+s_2+1}\to\mathbb{R}^d$ is a measurable function.

In this framework $f(x_k,u_k,\xi_k,k):\Omega\to\mathbb{R}^d$ is defined by

$$f(x_k,u_k,\xi_k,k)(\omega)= =(x_k(\omega),u_k(\omega),\xi_k(\omega),k)$$

for all $\omega\in\Omega$, and $g(x_k,\eta_k,k):\Omega\to\mathbb{R}^d$ is defined by

$$g(x_k,\eta_k,k)(\omega)=(x_k(\omega),\eta_k(\omega),k),$$

for all $\omega\in\Omega$.

We assume that x_0 is given by means of a random variable x_0: $\Omega\to X\subset\mathbb{R}^d$ (measurable function).

2. MAPLE PROCEDURES FOR SIMULATING FRACTIONAL ORDER STOCHASTIC DISCRETE-TIME CONTROL SYSTEMS

The purpose of this section is to provide Maple procedures to simulate a fractional order stochastic discrete-time control system:

$$\begin{cases}\Delta^{[\alpha]}x_{k+1}=f(x_k,u_k,\xi_k,k)\\ y_k=g(x_k,\eta_k,k),\ k\in\mathbb{N},\end{cases} \quad (2.1)$$

The scalar components of the initial state x_0: $\Omega\to X\subset\mathbb{R}^d$ as well as of the control inputs will be given as random samples of size m.

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, the system (2.1) can be written as

$$\begin{cases}x_{k+1,j}=\sum_{i=1}^{k+1}c_{i,j}x_{k+1-i,j}+\tilde{f}_j(x_k,u_k,\xi_k,k)\\ \qquad j\in\{1,2,\ldots d\},\ k\in\mathbb{N}\\ y_k=g(x_k,\eta_k,k),\ k\in\mathbb{N},\end{cases} \quad (2.2)$$

where

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

$$\tilde{f}_j = h_j^{\alpha_j} f_j$$

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

The first procedure that we provide can be used to simulate a stochastic discrete-time control system of form (2.2). For computational purposes it is useful to consider samples drawn from the same distributions as of the random variables ξ_k and η_k for $k\in\mathbb{N}$. Therefore we shall consider two data structures w=(w[i,j,k])$_{i,j,k}$ and v=(v[i,j,k])$_{i,j,k}$, where for all $i\in\mathbb{N}^*$ and $j\in\{1,2,..,s_1\}$,

{w[i,j,1], w[i,j,1], …, w[i,j,m]},

is a random sample of size m drawn from the same distribution as the scalar component j of ξ_i and for all $i\in\mathbb{N}^*$ and $j\in\{1,2,..,s_2\}$,

{v[i,j,1], v[i,j,1], …, v[i,j,m]},

is a random sample of size m drawn from the same distribution as the scalar component j of η_i. As a consequence, for every $i\in\mathbb{N}^*$ each scalar component $x_{i,j}$ (respectively, $y_{i,j}$) of x_i (respectively, y_i) in (2.2) will be obtained as a sample

{x[i,j,1],x[i,j,2],…,x[i,j, m]}

respectively,

{y[i,j,1],y[i,j,2],…,y[i,j, m]}

In the procedure DiscreteCStochasticSyst we take a natural number $n \in \mathbb{N}$, two data structures $w=(w[k,i,j])_{k,i,j,}$ and $v=(v[k,j,k])_{k,i,j,}$ for storing the samples of size m associated to the stochastic processes ξ and η in (2.2) and we compute x[k,i,j] for all $k\in\{0,1,\ldots,n\}$, $i \in\{1,2,\ldots,d\}$, $j\in\{1, 2, \ldots, m\}$ and y[k,i,j] for all $k\in\{1, 2, \ldots,n\}$, $i \in\{1, 2, \ldots, q\}$, $j\in\{1, 2, \ldots, m\}$.

```
DiscreteCStochasticSyst := proc (c, x0, u, f, g, w, v, n)
local i, j, k, x, y, d, p, s1, s2, q, m;
m := op(2, op(3, [op(2, op(w))]));
s1 := op(2, op(2, [op(2, op(w))]));
s2 := op(2, op(2, [op(2, op(v))]));
d := nops(x0);
p := nops(op(1, u));
q := nops(g);
x := array(0 .. n, 1 .. d, 1 .. m);
y := array(1 .. n, 1 .. q, 1 .. m);
for j to m do
 for i to d do
  x[0, i, j] := x0[i][j];
  x[1, i, j] := c[i, 1]*x0[i][j]+f[i](seq(x0[r][j], r = 1 ..
d), seq(u[1][r][j], r = 1 .. p), seq(w[1, r, j], r = 1 .. s1),
1)
 end do;
 for i to q do
  y[1, i, j] := g[i](seq(x[1, r, j], r = 1 .. d), seq(v[1, r,
j], r = 1 .. s2), 1)
  end do
end do;
for k to n-1 do
 for j to m do
  for i to d do
    x[k+1, i, j] := sum(c[i, r]*x[k+1-r, i, j], r = 1 ..
k+1)+f[i](seq(x[k, r, j], r = 1 .. d), seq(u[k+1][r][j], r = 1
.. p), seq(w[k+1, r, j], r = 1 .. s1), k+1)
   end do;
   for i to q do
    y[k+1, i, j] := g[i](seq(x[k+1, r, j], r = 1 .. d),
seq(v[k+1, r, j], r = 1 .. s2), k+1)
   end do
  end do
end do;
return [x, y]
end proc
```

The formal parameters of the procedure DiscreteCStochasticSyst have the following signification:

- c is an array containing the coefficients $(c_{i,1}, c_{i,2}, \ldots, c_{i,d})\in\mathbb{R}^d$ in (2.2) for $i\in\{1, 2, \ldots,n\}$ ($c[j,i]= c_{i,j}$).
- x0 is a list of lists containing samples from the components of the initial state of the system (2.2) :

[x0[i][1], x0[i][2], …., x0[i][m]]

is a sample from $x_{0,i}$ for all $i\in\{1, 2, \ldots, d\}$. Thus x0[i][j]=x[0,i,j] for $i \in\{1, 2, \ldots, d\}$ and $j\in\{1, 2, \ldots, m\}$

- u is the list of control inputs in (2.2). u is a data structure of the form (u[k][i][j])i,j,k where :

[u[k][i][1], u[k][i][2], …., u[k][i][m]]

is a sample from the i component of u_k

for all $i \in \{1, 2, \ldots, p\}$.

- f=[f1,f2,…,fd] is the list of the scalar components of the function f in (2.2).
- g=[g1,g2,…,gq] is the list of the scalar components of the function g in (2.2).
- w is the data structure storing the samples of size m associated to the stochastic processes ξ in (2.2).
- v is the data structure storing the samples of size m associated to the stochastic processes η in (2.2).
- n is the number of iterations using (2.2)

The procedure DFCStochasticSyst computes x[k,i,j] for all $k \in \{0,1,\ldots,n\}$, $i \in \{1,2,\ldots,d\}$, $j \in \{1, 2, \ldots, m\}$ and y[k,i,j] for all $k \in \{1, 2, \ldots,n\}$, $i \in \{1, 2, \ldots, q\}$, $j \in \{1, 2, \ldots, m\}$ associated to $x_k=(x_{k,1}, x_{k,2},\ldots, x_{k,d})$ and $y_k=(y_{k,1}, y_{k,2},\ldots, y_{k,q})$ from the fractional order stochastic discrete-time control system (2.1). The formal parameter alpha is the list $[\alpha_1, \alpha_2, \ldots, \alpha_d]$, where $\alpha_1, \alpha_2, \ldots, \alpha_d$ are the fractional orders in (2.1). The formal parameter h is the list $[h_1, h_2, \ldots, h_d]$, where $(h_1, h_2, \ldots, h_d)$ is the vector of sampling periods in (2.1). The remaining formal parameters f, g, w, v, x0, u and n of the procedure DFCStochasticSyst have the same meaning as the corresponding parameters in the procedure DiscreteCStochasticSyst.

The procedures GenFractCoeff and hpower called by DFCStochasticSyst were defined in [1].

```
DFCStochasticSyst := proc (alpha, h, f, g, w, v, x0, u, n)
local i, j, k, c, halpha, fh, fhi, p, d, s1;
d := nops(alpha);
s1 := op(2, op(2, [op(2, op(w))]));
p := nops(op(1, u));
c := GenFractCoeff(alpha, n); halpha := hpower(alpha, h);
fh := [];
for i to d do
   fhi := unapply(halpha[i]*f[i](seq(varf[j], j = 1 ..
d+p+s1+1)), seq(varf[j], j = 1 .. d+p+s1+1));
   fh := [op(fh), fhi]
end do;
return DiscreteCStochasticSyst(c, x0, u, fh, g, w, v, n)
end proc
```

We also provide a variant of the procedure DFCStochasticSyst, using the of the distributions of $\xi_1, \xi_2, \ldots, \xi_n$ and $\eta_1, \eta_2, \ldots, \eta_n$ instead of the data structures w, respectively, v. The formal parameters distribution1 and distribution2 of DFStochasticSystD are lists of lists of distributions: distribution1[i][j] is the distribution of the component j of ξ_i and distribution2[i][j] is the distribution of the component j of η_i. The remaining formal parameters alpha, h, f, g, x0, u, n, m of DFStochasticSystD have the same meaning as the corresponding parameters in the procedure DFCStochasticSyst.

```
DFCStochasticSystD:=proc(alpha,h,f,g,distribution1,distribution
2,x0,u,n,m)
local i,j,k,c,halpha,fh,fhi,p,d,s1,s2,w,v,RX,S;
d:=nops(alpha);
s1:=nops(op(1,distribution1));
p:=nops(op(1,u));
c:=GenFractCoeff(alpha,n); halpha:=hpower(alpha,h);
fh:=[];
for i to d do
 fhi:=unapply(halpha[i]*f[i](seq(varf[j],j=1..d+p+s1+1)),
seq(varf[j],j=1..d+p+s1+1));
 fh:=[op(fh),fhi]
end do;
s2:=nops(op(1,distribution2));
w:=array(1..n,1..s1,1..m);
v:=array(1..n,1..s2,1..m);
for i to n do
 for j to s1 do
  RX:=Statistics:-RandomVariable(distribution1[i][j]);
  S:=Statistics:-Sample(RX,m);
  for k to m do w[i,j,k]:=S[k] end do
 end do;
 for j to s2 do
  RX:=Statistics:-RandomVariable(distribution1[i][j]);
  S:=Statistics:-Sample(RX,m);
  for k to m do v[i,j,k]:=S[k] end do
 end do
end do;
return DiscreteCStochasticSyst(c,x0,u,fh,g,w,v,n)
end proc
```

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

We shall consider the following model of a fractional order discrete-time control system:

$$\begin{cases} \Delta^{[\alpha]} x_{k+1} = f\left(x_k, u_k, k\right) \\ y_k = g\left(x_k, k\right),\ k \in \mathbb{N}, \end{cases} \qquad (3.1)$$

where

- $x_k \in X \subset \mathbb{R}^d$ is the state at time k and X is the state space.
- $u_k \in U \subset \mathbb{R}^p$ is the control input applied at time k and U is the control input space
- $y_k \in Y \subset \mathbb{R}^q$ is the output at time k, and Y is the output space.
- $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_d) \in \mathbb{R}^d$
- $f=(f_1,f_2,\ldots, f_d)$: $X \times U \times \mathbb{R} \subset \mathbb{R}^{d+p+s_1+1} \rightarrow \mathbb{R}^d$ is a function.
- $g=(g_1,g_2,\ldots, g_q)$: $X \times Y \times \mathbb{R} \quad \mathbb{R}^{d+s_2+1} \rightarrow \mathbb{R}^d$ is a function.

The formal parameters of the procedure DDFCSyst (which is a extension of the procedure DDFSyst from [1] to include control inputs) are:

- alpha is the list $[\alpha_1, \alpha_2, \ldots, \alpha_d]$, where $\alpha_1, \alpha_2, \ldots, \alpha_d$ are the fractional orders in (3.1)
- h is the list $[h_1, h_2, \ldots, h_d]$, where $(h_1, h_2, \ldots, h_d)$ is the vector of sampling periods in (3.1)
- x0 is a list, vector or one dimensional array containing the components of the initial state of the system (3.1)
- u is the list of control inputs in (3.1). u is a list of lists: u[k][i] is the i component of u_k for all $i \in \{1, 2, \ldots, p\}$.
- f=[f1,f2,…,fd] is the list of the scalar components of the function f in (3.1).

- g=[g1,g2,…,gq] is the list of the scalar components of the function g in (3.1).
- n is number of iterations performed in (3.1).

The procedure DDFCSyst returns a list [x,y] of two two-dimensional arrays: x is such that x[k,i]=$x_{k,i}$ (component i of state x_k) for all k∈{0, 1, …, n} and all i∈{1,2,…,d} and x is such that y[k,i]=$y_{k,i}$ (component i of output y_k) for all k∈{1, 1, …, n} and all i∈{1,2,…,p}.

```
DDFCSyst := proc (alpha, h, x0, u, f, g, n)
local i, j, k, c, halpha, fh, fhi, x, y, d, p, q;
d := nops(alpha); p := nops(op(1, u)); q := nops(g);
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;
fh := [];
for i to d do
   fhi := unapply(halpha[i]*f[i](seq(varf[j], j = 1 .. d+p+1)),
seq(varf[j], j = 1 .. d+p+1));
    fh := [op(fh), fhi]
end do;
x := array(0 .. n, 1 .. d); y := array(0 .. n, 1 .. q);
for i to d do
  x[0, i] := x0[i];
   x[1, i] := c[i, 1]*x0[i]+fh[i](seq(x0[j], j = 1 .. d),
seq(u[1][j], j = 1 .. p), 1)
end do;
for i to q do
 y[1, i] := g[i](seq(x[1, 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)+fh[i](seq(x[k, j], j = 1 .. d), seq(u[k+1][j], j = 1 ..
p), k+1) end do;
 for i to q do
  y[k+1, i] := g[i](seq(x[k+1, j], j = 1 .. d), k+1)
 end do
end do;
return [x, y]
 end proc
```

4. MAPLE PROCEDURES FOR SIMULATING FRACTIONAL ORDER STOCHASTIC DISCRETE-TIME LINEAR CONTROL SYSTEM

We shall consider the following model of a fractional order stochastic discrete-time control system:

$$\begin{cases} \Delta^{[\alpha]}x_{k+1} = A_k x_k + B_k u_k + \\ \quad + \left(W_k^1 x_k + W_k^2 u_k + W_k^3\right)\xi_k,\ k \in \mathbb{N}, \\ y_k = C_k\, x_k + \left(V_k^1 x_k + V_k^2\right)\eta_k,\ k \in \mathbb{N}, \end{cases}$$
(4.1)

where $\alpha, \Delta^{[\alpha]}$, x_k, u_k, y_k, ξ_k and η_k has the same meaning as in the case of (2.1), A_k, B_k, W_k^3 and V_k^2 are time-varying matrices of appropriate dimension, W_k^1 W_k^2 and V_k^1 are three-dimensional arrays. The following procedure FSLinear (that the general case of the system (4.1)) returns the list [f1,f2,…,fd] of the scalar components of the function f so that the state equation in the system (4.1) can be equivalently written as $\Delta^{[\alpha]}x_{k+1} = f\left(x_k, u_k, \xi_k, k\right)$. 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}], [B_0, B_1, …, B_{n-1}] [W_0^1, W_1^1, …, W_{n-1}^1], [W_0^2, W_1^2, …, W_{n-1}^2] and [W_0^3, W_1^3, …, W_{n-1}^3].

```
FSLinear := proc (A, B, W1, W2, W3)
local i, j, k, f, fj, d, p, s1;
d := op(2, op(1, [op(2, op(1, W2))]));
s1 := op(2, op(3, [op(2, op(1, W2))]));
p := op(2, op(2, [op(2, op(1, W2))]));
f := [];
for j to d do
 fj := unapply(sum((A[k][j, i]+sum(W1[k][j, i, r]*varn[r], r =
1 .. s1))*varx[i], i = 1 .. d)+sum((B[k][j, i]+sum(W2[k][j, i,
r]*varn[r], r = 1 .. s1))*varu[i], i = 1 .. p)+sum(W3[k][j,
i]*varn[i], i = 1 .. s1), seq(varx[i], i = 1 .. d),
seq(varu[i], i = 1 .. p), seq(varn[i], i = 1 .. s1), k);
 f :=  [op(f), fj]
end do;
return f
end proc
```

is a rewrite of the procedure FLinear [1] in Similarly, the procedure GSLinear returns the list [g1,g2,…,gq] of the scalar components of the function g so that the observation equation in the system (4.1) can be equivalently written as $y_k = g\left(x_k, \eta_k, k\right),\ k \in \mathbb{N}$. Its formal parameters are the lists: [C_0, C_1, …,C_{n-1}], [V_0^1,r V_1^1, …, V_{n-1}^1] and [V_0^2, V_1^2, …, V_{n-1}^2] .

Using the procedures FSLinear and GSLinear we can easily write a procedure (such as the below procedure DFSLCSyst) to compute x_1, x_2, … and y_1, y_2, … from the system (4.1).

```
GSLinear := proc (C, V1, V2)
local i, j, k, g, gj, q, d, s2;
d := op(2, op(2, [op(2, op(1, C))]));
q := op(2, op(1, [op(2, op(1, C))]));
s2 := op(2, op(3, [op(2, op(1, V1))]));
g := [];
for j to q do
 gj := unapply(sum((C[k][j, i]+sum(V1[k][j, i, r]*varn[r], r =
1 .. s2))*varx[i], i = 1 .. d)+sum(V2[k][j, i]*varn[i], i = 1
.. s2), seq(varx[i], i = 1 .. d), seq(varn[i], i = 1 .. s2),
k);
g := [op(g), gj]
end do;
return g
end proc
```

```
DFSLCSyst := proc (alpha, h, A, B, W1, W2, W3, C, V1, V2,
distribution1, distribution2, x0, u, n, m)
local f, g;
f := FSLinear(A, B, W1, W2, W3);
g := GSLinear(C, V1, V2);
return DFCStochasticSystD(alpha, h, f, g, distribution1,
distribution2, x0, u, n, m)
end proc
```

5. EXAMPLES

Let us consider the following system
he following system:

$$\begin{cases} \Delta^{[\alpha]}x_{k+1} = f\left(x_k, u_k, \xi_k, k\right),\ k \in \mathbb{N}, \\ y_k = g\left(x_k, \eta_k, k\right),\ k \in \mathbb{N} \end{cases} \quad (5.1)$$

with

$\alpha = \left[\frac{3}{4}, -\frac{1}{2}\right]$, h=[1,1]

$$f\left(x^1, x^2, u, w_1, w_2, w_3, k\right) = \begin{pmatrix} -\frac{1}{2}\cos(kx^1+x^2)+\frac{1}{k+1}u+\frac{w_1 w_2}{75\left(k+1\right)}+\frac{w_3}{25} \\ \frac{1}{5}\sin\left(kx^1+x^2\right)+\frac{w_1}{25} \end{pmatrix}^t$$

$$g\left(x^1, x^2, v, k\right) = x^1 + x^2 + \frac{v}{50}$$

$x_0 \equiv (1,1)$, $u_k \equiv 1,\ k \in \mathbb{N}$

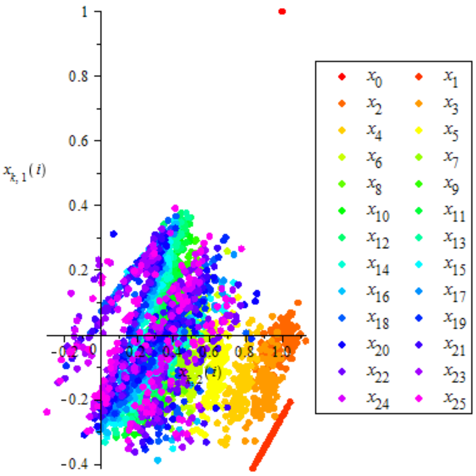
$\xi_k = \left(\xi_k^1, \xi_k^2, \xi_k^3\right)$

For all k, the distributions of ξ_k^1 , ξ_k^3 and η_k are Normal(0,1) and the distribution of ξ_k^2 is Bernoulli(1/(k+1)).

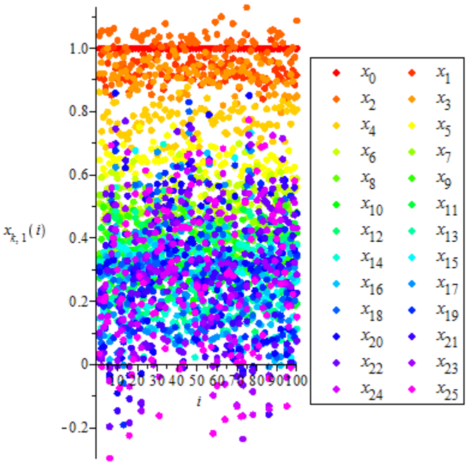
Then we can compute samples of size m=100 from x_1, x_2, …, x_{25} and y_1, y_2, …,y_{25} from the system (5.1) using

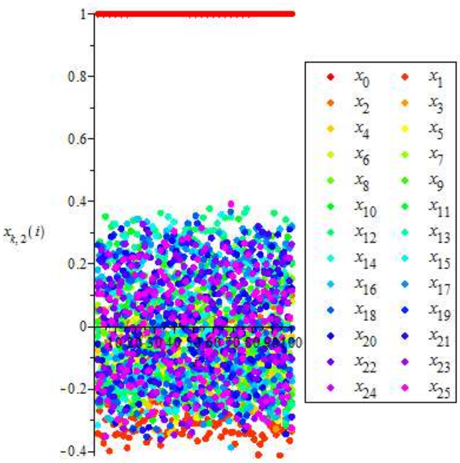
```
>n := 25: m := 100: xys := DFCStochasticSystD([3/4, -1/2], [1, 1], [(x1, x2, u, w1, w2, w3, k) -> -(1/2)*cos(k*x1+x2)+u/(k+1)+w1*w2/(75*k+75)+(1/25)*w3, (x1, x2, u, w1, w2, w3, k) -> (1/5)*sin(k*x1+x2)+(1/25)*w1 ], [(x1, x2, v, k) -> x1+x2+(1/50)*v ], [seq([Normal(0, 1), Bernoulli(1/(i+1)), Normal(0, 1)], i = 1 .. n)], [seq([Normal(0, 1)], i = 1 .. n)], [[seq(1., j = 1 .. m)], [seq(1., j = 1 .. m)]], [seq([[seq(1., j = 1 .. m)]], i = 1 .. n)], n, m):
```

We can plot $x_k \in \mathbb{R}^2$ with $k \in \{0,1, ...n\}$ using the visDFStochasticSyst2D defined in [2].

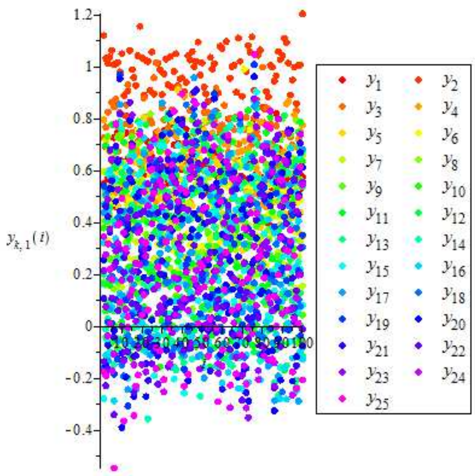


We can also use the procedure visScalarComponent to plot the scalar components of $x_k \in \mathbb{R}^2$ with $k \in \{0,1, ...n\}$





and to plot $y_k \in \mathbb{R}$ with $k \in \{1,2, ...n\}$.



Let us consider the deterministic system associated to (5.1) (noise free):

$$\begin{cases} \Delta^{[\alpha]} x_{k+1} = \tilde{f}(x_k, u_k, k) \\ y_k = \tilde{g}(x_k, k),\ k \in \mathbb{N}, \end{cases} \qquad (5.2)$$

with

$$\alpha = \left[\frac{3}{4}, -\frac{1}{2}\right],\ h=[1,1]$$

$$\tilde{f}(x^1, x^2, u, k) = \begin{pmatrix} -\frac{1}{2}\cos(kx^1+x^2)+\frac{1}{k+1}u \\ \frac{1}{5}\sin(kx^1+x^2) \end{pmatrix}^t$$

$$\tilde{g}(x^1, x^2, k) = x^1 + x^2$$

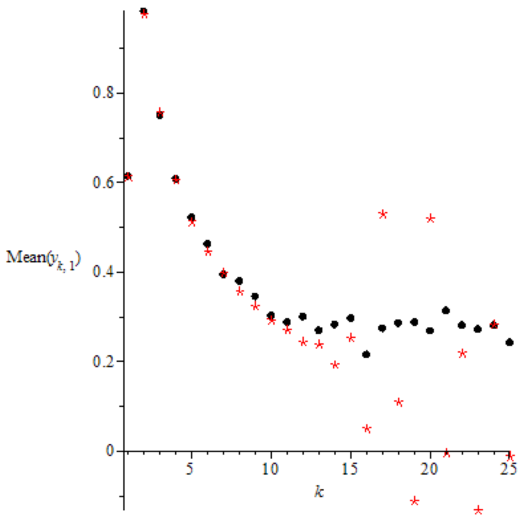
$$x_0 \equiv (1,1),\ u_k \equiv 1,\ k \in \mathbb{N}$$

We compute $x_1, x_2, \ldots, x_{25}$ and $y_1, y_2, \ldots, y_{25}$ from the system (5.2) using

```
>n := 25: xyd := DDFCSyst([3/4, -1/2], [1, 1], [1., 1.], [seq([1.], i = 1 .. n)], [(x1, x2, u, k) ->-(1/2)*cos(k*x1+x2)+u/(k+1), (x1, x2, u, k) -> (1/5)*sin(k*x1+x2)], [(x1, x2, k) ->x1+x2], n):
```

We can compare the outputs of the stochastic system with the outputs of the deterministic system using the procedure visScalarComponentMeanVsDet defined in [2] :

```
>visScalarComponentMeanVsDet(xys2[2], xyd2[2], 1, 1, n,)
```



(deterministic correspond to red asterisks and stochastic to black solid circles).

REFERENCES

[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), No 1 Supplement (2016), 244-251.

[2] M. Buneci, Maple visualization tools for a fractional order discrete system perturbed by a sequence of real random variables, Fiabilitate şi durabilitate (Fiability & durability), No 1 Supplement (2016), 252-259.

[3]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.

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