MATLAB SIMULATIONS FOR FRACTONAL DISCRETE TIME SYSTEMS WITH MARKOVIAN JUMPS

Viorica-Mariela UNGUREANU, Assoc. Professor,
"Constantin Brâncuşi" University of Tg-Jiu, Tg-Jiu, ROMÂNIA

ABSTRACT: This paper presents some simulations in MATLAB of a class of fractional discrete-time systems with control and Markovian jumps. This subject seems to be new.

KEYWORDS: fractional discrete-time systems, Markov jumps, simulations.

I. Introduction

The study of discrete-time systems of non integer order began to attract increasingly more interest from scientist due to their new applications in different areas of applied science including electrochemistry, electromagnetism, biophysics, quantum mechanics, radiation physics or control theory (see [5], [4] and the references therein). However, the case of linear discrete-time systems with finite Markovian jumps seems not to be studied yet in the scientific literature. That is not the situation of Markov jump linear systems which were intensively treated in the last decades in both finite-dimensional and infinite-dimensional frameworks. Without being exhaustive, we refer the reader to [1], [2], [3], [6], [7] and the references therein).

The aim of this paper is to present some MATLAB programs which simulate this type of systems.

For a fixed $\mathbf{N} \in N$ we define the set $\mathbf{Z} = \{1, 2,.., \mathbf{N}\}$. Let $(\Omega, \mathbf{F}, P)$ be a probability space. Assume that $\{r_n\}_{n\in N}$ is an homogeneous Markov chain on $(\Omega, \mathbf{F}, P)$ with the state space $\mathbf{Z}$ and the transition probability matrix

$$Q = \{q_{i,j} := P(r_{n+1} = j \,|_{r_n = i})\}_{(i,j)\in \mathbf{Z}\times\mathbf{Z}}, n \in N.$$

In the sequel we shall denote by $\mathbf{F}_n$ the σ -algebra generated by $\{r_k, 0 \le k \le n\}_{n\in N}$.

For any fixed $\alpha \in (0,2)$,

$$\binom{\alpha}{j} = \begin{cases} 1, & j = 0 \\ \frac{\alpha(\alpha-1)\dots(\alpha+1-j)}{j!}, & j \in \mathbf{N}^* \end{cases}$$

is the generalized binomial coefficient and

$$\Delta^{[\alpha]} x_{k+1} = \frac{1}{h^\alpha} \sum_{j=0}^{k+1} (-1)^j \binom{\alpha}{j} x_{k+1-j} \qquad (1.1)$$

is the discrete fractional-order operator that arises in the Grünwald-Letnikov definition of the fractional order derivatives (see for e.g. [Concept]). In (1.1), h is the sampling time. For the sake of simplicity we assume that $h = 1$ in the rest of the paper.

We consider the stochastic discrete-time fractional system with control

$$\begin{aligned} \Delta^{[\alpha]} x_{k+1} &= A(r_k) x_k + x_k + D(r_k) u_k \\ x_0 &= x \in R^d, \ k \in N \end{aligned} \qquad (1.2)$$

where $A(i) \in R^{d\times d}$, $D(i) \in R^{d\times m}$ for all $i \in \mathbf{Z}$ and the control $u = \{u_k\}_{k\ge 0}$ belongs

to the class of admissible controls $\mathbf{U}^a$ formed by all sequences $\{u_k\}_{k\geq 0}$ which elements $u_k \in L^2(\Omega, R^m)$ are $\mathbf{F}_k$ - measurable random variables.
Denoting $M(i) = A(i) + \alpha I_{R^d}$,

$$c_j := (-1)^j \binom{\alpha}{j+1}, k \in N \quad (1.3)$$

and $A_j = c_j I_{R^d}, i \neq 0$, system (1.2) can be equivalently rewritten as

$$x_{k+1} = M(r_k)x_0 + \sum_{j=1}^{k} c_j x_{k-j} + D(r_k)u_k, \quad (1.4)$$
$$x_0 = x \in R^d.$$

Main results

In this section we will provide a MATLAB program which plots the graphs of the components of a finite segment $x_0,.., x_k$ of the trajectory $\{x_k\}_{k\in N}$ of system (1.4). We assume that α, x_0, u, k, Q, $M(i), i \in \mathbf{Z}$ are given.

```
function [rez] = xku(alph, x0,k,u)
%computes the element xk of the trajectory
%First let us generate the Markov chain
Q = [0.3 0.7;0.8 0.2]; starting_value = 1; chain_length = k-1;
chain = zeros(1,chain_length);
 chain(1)=starting_value;
 for i=2:chain_length
 sdistrib = Q(chain(i-1),:);
 cumulative_distribution = cumsum(sdistrib);
  r = rand;
 A=find(cumulative_distribution>r)
 chain(i) = A(1);
 end
 rez(:,1)=x0;
 M(:,:,1)=[1 2; 3 2];
 M(:,:,2)=[-1 2; -3 2];
 D(:,:,1)=[1 ;4];
 D(:,:,2)=[-1 ;-3];
 rez(:,2)=M(:,:,chain(1))*x0+D(:,:,chain(1))*u(1);
 for i=2: k-1
 S=M(:,:,chain(i))*rez(:,i)+D(:,:,chain(i))*u(i);
 for l=1: i-1
 S=S+cj( alph,l )*rez(:,i-l);
 end
 rez(:,i+1)=S;
 end
end
```

Knowing the transition matrix Q of the

```
function [ r ] = cj( alph,j )
if j>1
r = -cj(alph,j - 1) * (alph-j)/(j+1);
else
r = -alph*(alph-1)/2;
end
end
```

Markov chain $\{r_n\}_{n\in N}$, we first generate a homogeneous Markov with a given starting value (in our example $r_0 = 1$ with probability 1) and a length of $k-1$. .

The MATLAB function listed below uses the function cj(.,.) which computes the coefficients (1.3).

This function has the following code

Example. Assume that $\alpha = 0.5, \mathbf{N} = 2, x_0 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, k = 9$ and $u(i) = -0.01, i \in 0,..,19$. The dynamic of system (1.4) could be described as a random jump, governed by the Markov process r_k, between the states of the following two deterministic systems

$$V_{k+1} = M(1)x_0 + \sum_{j=1}^{k} c_j V_{k-j} + D(r_k)u_k, \quad (2.1)$$

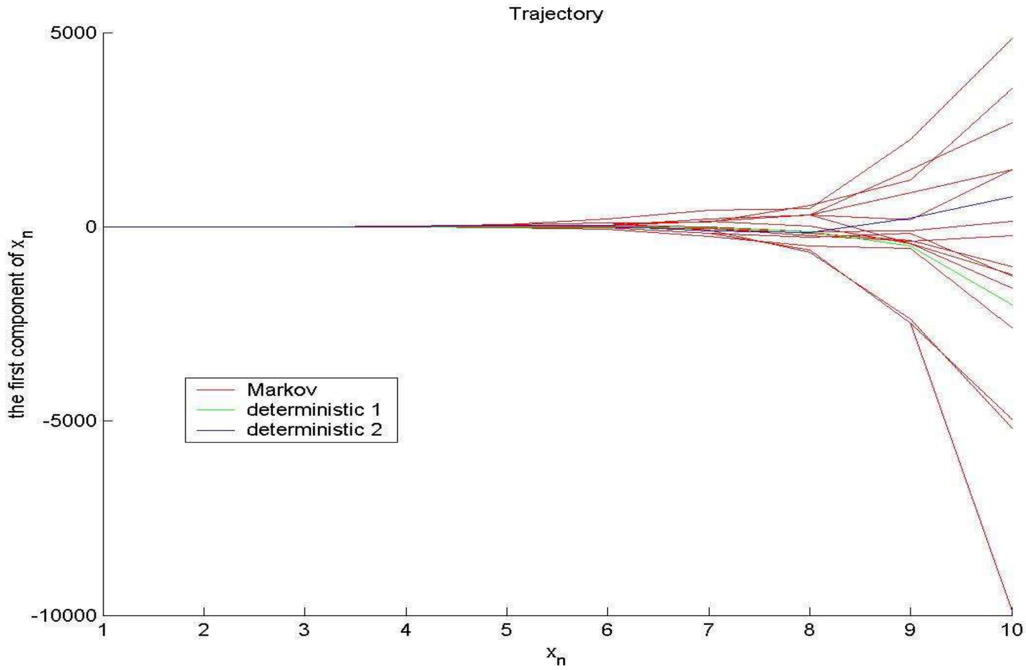
and

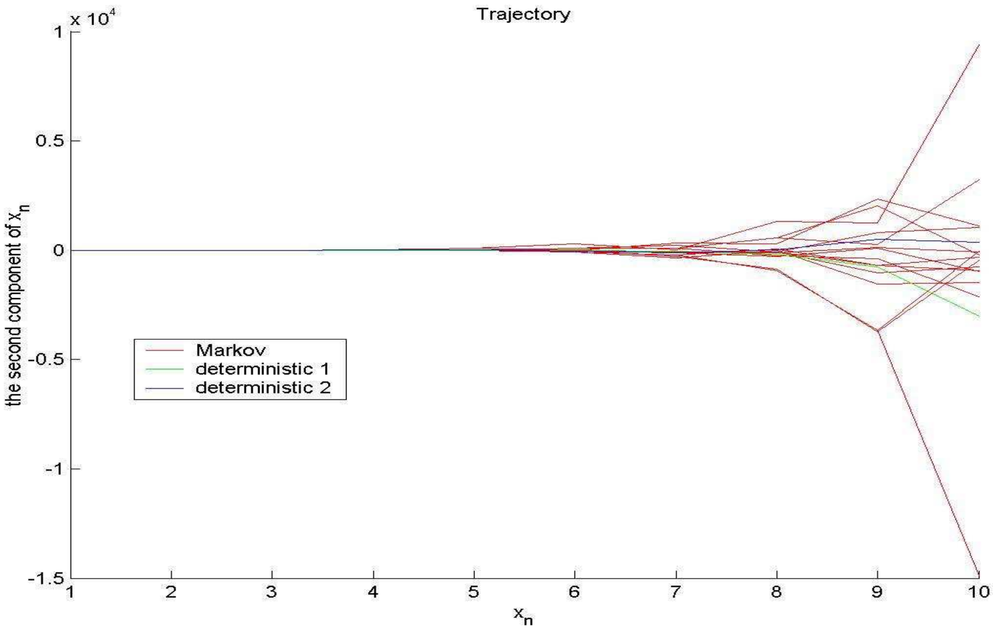
$$W_{k+1} = M(2)x_0 + \sum_{j=1}^{k} c_j W_{k-j} + D(r_k)u_k. \quad (2.2)$$

The following program plots the graph of the sequence formed by the first components of the segment $x_0(\omega),.., x_{19}(\omega)$ (for 20 instances $\omega \in \Omega$), as well as the graphs of the corresponding segments of the trajectories of (2.1) and (2.2), generates with the same initial value x_0.

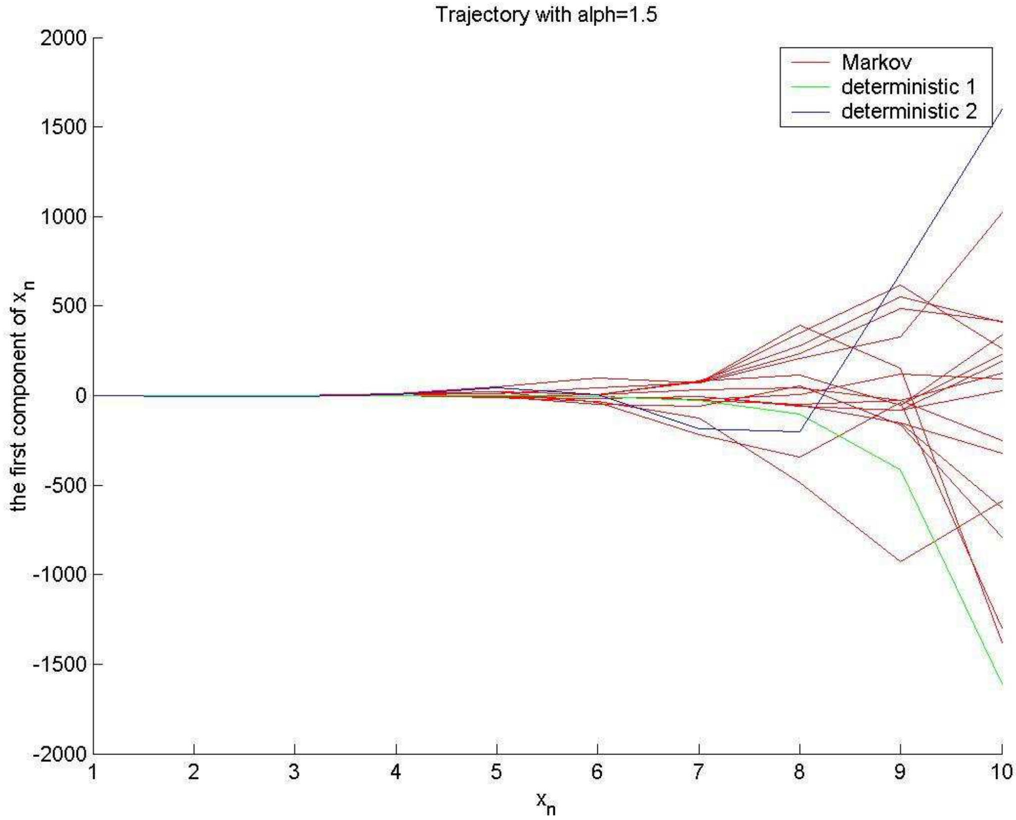
The results are illustrated by the above graphs. The green curves are associated with the trajectories of the deterministic system (2.1), while the blue ones with the trajectories of system (2.2). The red curves correspond to the Markov jump system (1.4).

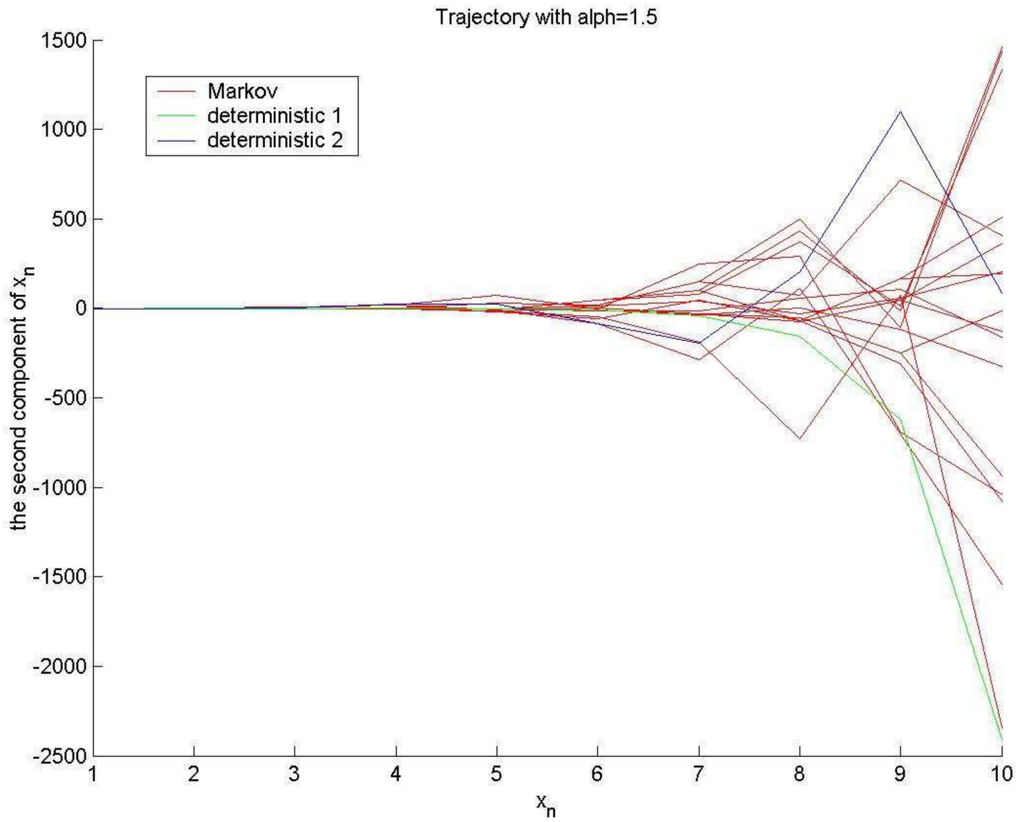
```
x0=[1;-1]
for i=1:10
u(i)=-0.01;
end
V=xkfmu( 0.5,x0,10,u);
W=xkfm2u( 0.5,x0,10,u);
for i=1:20
ZU(:,:,i)=xku( 0.5,x0,10,u);
end
figure
hold on
for i=1:20
plot(1:10,ZU(1,:,i), 'r',1:10,V(1,:), 'g', 1:10,W(1,:), 'b')
end
title('Trajectory')
   xlabel('x_{n}')
   ylabel('the first component of x_{n}')
   legend('Markov','deterministic 1','deterministic 2')
   figure
hold on
for i=1:20
plot(1:10,ZU(2,:,i), 'r',1:10,V(2,:), 'g', 1:10,W(2,:), 'b')
end
title('Trajectory')
   xlabel('x_{n}')
   ylabel('the second component of x_{n}')
   legend('Markov','deterministic 1','deterministic 2')
```





For a different value of α we obtain the following two figures.


Trajectory with alph=1.5
Markov
deterministic 1
deterministic 2
the first component of x_n
x_n
2000
1500
1000
500
0
-500
-1000
-1500
-2000
1
2
3
4
5
6
7
8
9
10


Trajectory with alph=1.5
Markov
deterministic 1
deterministic 2
the second component of x_n
x_n
1500
1000
500
0
-500
-1000
-1500
-2000
-2500
1
2
3
4
5
6
7
8
9
10

REFERENCES

[1] O. L. V. Costa, M. D. Fragoso, M.G. Todorov, Continuous-Time Markov Jump Linear Systems, Springer Berlin Heidelberg, 2013.

[2] O. L. V. Costa, M. D. Fragoso, R. P. Marques, Discrete-time Markov jump linear systems, Springer-Verlag London, 2004.

[3] V. Dragan, T. Morozan, A. Stoica, Mathematical Methods in Robust Control of Discrete Time Linear Stochastic Systems, Springer, New York, 2013.

[4] A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, Theory and Applications of Fractional Differential Equations. Elsevier Science, Amsterdam, 2006.

[5] C.A. Monje et al. Fractional-order Systems and Controls: Fundamentals and Applications, Springer Science & Business Media, London, 2010.

[6] V. M. Ungureanu, Stability, stabilizability and detectability for Markov jump discrete-time linear systems with multiplicative noise in Hilbert spaces, Optimization 63.11 (2014): 1689-1712

[7] V. M. Ungureanu, Optimal control for infinite dimensional stochastic differential equations with infinite Markov jumps and multiplicative noise, Journal of Mathematical Analysis and Applications 417.2 (2014): 694-718.