CEAI, Vol.16, No.3 pp. 46-55, 2014 Printed in Romania

Design of Fractional-order Sliding Mode Controller (FSMC) for a class of Fractional-order Non-linear Commensurate Systems using a Particle Swarm Optimization (PSO) Algorithm

Abdelhamid Djari, Toufik Bouden, Abdesselem Boulkroune*

NDT Laboratory, Automatic Department, Jijel University, BP°98, Ouled-Aissa,18000 Jijel, Algeria
(*)LAJ Laboratory, Automatic Department, Jijel University, BP°98, Ouled-Aissa,18000 Jijel, Algeria
E-mail: hamiddjari@yahoo.fr; bouden_toufik@yahoo.com; boulkroune2002@yahoo.fr

Abstract: The aim of this paper is to develop a fractional-order sliding mode control (FSMC) and to compare its application with the conventional sliding mode control (SMC) for a class of fractional-order non-linear commensurable systems using particle swarm optimization (PSO) algorithm which calculates optimum saturation gain $\boldsymbol{k}$ and the parameter $\boldsymbol{\lambda}$ of the sliding surface. According to the obtained results from the two used systems for these applications, it's clearly shown that the stability and the robustness issues using FSMC are more enhanced than those obtained using SMC. This robustness is also depending on the nature of the chosen sliding surface.

Keywords: Fractional sliding mode control; System engineering; Fractional-order non linear commensurate system; Particle swarm optimization algorithm.

1. INTRODUCTION

Extending classical integer order calculus to non-integer order case leads to the so-called fractional calculus. It has a firm and long-standing theoretical foundation and the earliest systematic studies of fractional calculus were in the 19th century by Liouville, Riemann and Holmgren see (Oldham and Spanier, 1974). At present, the number of applications of fractional calculus rapidly grows. These mathematical phenomena allow us to describe and model a real object more accurately than the classical "integer" methods. The real objects are generally fractional (Nakagawa and Sorimachi, 1992; Oustaloup, 1995; Podlubny, 1999; Westerlund, 2002). However, for many of them, the fractionality is very low. A typical example of a non-integer (fractional) order system is the voltage-current relation of a semi-infinite lossy transmission line (Wang, 1987) or diffusion of heat through a semi-infinite solid, where the heat flow is equal to the half-derivative of the temperature (Podlubny, 1999).

In motion control branch, some example applications can be found in (Xue and Li, 2006). The application of the theory of fractional calculus in sliding mode control is just beginning, but with more and more research efforts on this subject (Efe, 2008 and 2011; Yahyazadeh, 2008). Fractional sliding mode control (FSMC) is the use of the classical sliding mode control with fractional systems, or the use of sliding mode control with a sliding surface corresponding to a fractional order dynamic, or both. In this paper, we have developed a sliding mode control with fractional-order sliding surface via a class of fractional-order non-linear commensurable systems using PSO algorithm. The objective of PSO is to determinate the optimum parameters of fractional-order sliding surface and the gain of saturation. The contribution of this paper is to demonstrate that the response of the fractional-order systems under control is significantly better for the FSMC than that for the conventional integer-order SMC.

This paper is organized as follows: in section 2, the determination of the state representation of this class of fractional order systems is presented. In sections 3 and 4, the FSMC, the conventional SMC and the use of PSO algorithm are respectively discussed. In section 5, SMC and FSMC controllers optimized by PSO algorithm are applied to control fractional order non-linear commensurable systems. Finally, concluding remarks are drawn in section 6.

2. FRACTIONAL-ORDER SYSTEMS

Fractional-order calculus has been known since the development of the integer-order calculus, but for a long time, it has been considered as a sole mathematical problem. It is an area of mathematics that deals with derivatives and integrals from non-integer orders. In other words, it is a generalization of the traditional calculus that leads to similar concepts and tools, but with a much wider applicability. In recent decades, fractional calculus has become an interesting topic among system analysis and control fields and has also applied in an increasing number of other fields. The success of fractional-order controllers is unquestionable with a lot of success due to emerging of effective methods in differentiation and integration of non-integer order equations (Sabatier and Farges, 2012).

2.1 Fractional order calculus

The Fractional-order continuous integro-differential fundamental operators defined as (Petras, 2011):

$$ {}_aD_t^{\alpha} = \begin{cases} \frac{d^{\alpha}}{dt^{\alpha}}, Re(\alpha) > 0 \\ 1 \quad\quad , Re(\alpha) = 0 \\ \int_a^t (d\tau)^{\alpha} \quad , Re(\alpha) < 0 \end{cases} \quad (1) $$

where a and t are the limits of the operation, α is the order of the operation which can be a complex number and Re(α) its real part. Generally $\alpha \in R\ (Re(\alpha) = \alpha)$. In this paper, we focus on the case where the fractional order is a real number.

There are several definitions of fractional derivatives. The most frequently used are: the Grunwald-Letnikov (GL), the Riemann-Liouville (RL) and the Caputo definitions (Miller and Ross, 1993). An elementary definition of the fractional derivative of a continuous function $f(t)$ can be formalized as follows (Podlubny, 1999).

$$ D_t^m f(t) = lim_{h\to 0} \frac{1}{h^m} \sum_{k=0}^{N} (-1)^k \binom{m}{k} f(t-k.h) \quad (2) $$

where $N = \left[\frac{t-a}{h}\right]$, where $[\cdot]$ means the integer part, h is the time incremented and the coefficients $\binom{m}{k}$ are given by:

$$ \binom{m}{k} = \frac{\Gamma(m+1)}{\Gamma(k+1)\Gamma(m-k+1)} \quad (3) $$

and the Euler's gamma function (Γ) defined for positive real z is a generalized integral, given by:

$$ \Gamma(z) = \int_0^{+\infty} e^{-u} u^{z-1} du \quad (4) $$

It is also possible to generalize several results based on transforms; yielding expressions such as the Fourier transform expression:

$$ \widehat{\frac{df}{dx}}(\omega) = (j\omega).\hat{f}(\omega) \ \text{ with } \ \hat{f}(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-j\omega x} dx \quad (5) $$

For a wide class of functions which appear in real physical and engineering applications, these definitions are equivalent. The derivation operation is a multiplication by $(j\omega)$ and the derivation to the order n is a multiplication by $(j\omega)^n$. It is then natural to try to define the derivation of order α by a multiplication by $(j\omega)^{\alpha}$ in Fourier transform. For causal function, one can also use the Laplace transform in the same manner as Fourier (deriving order α corresponds to a multiplication by s^{α}, where s is a Laplace operator). There are several methods for the approximation fractional order operators to rational functions. The most approximation used is that of Oustaloup.

The Oustaloup's approximation model of a fractional order differentiator s^{α} $(\alpha \in R^+)$ can be written as:

$$ G(s) = K.\prod_{i=1}^{n} \frac{s+\omega_i'}{s+\omega_i} \quad (6) $$

where the poles, zeros and gain are evaluated from:

$$ \omega_i' = \omega_b.\omega_u^{(2i-1-\alpha)/n}, \omega_i = \omega_b.\omega_u^{(2i-1+\alpha)/n}, K = \omega_h^{\alpha} \quad (7) $$

ω_u is the unity frequencies gain and the central frequency of a band of frequencies distributed geometrically. Let $\omega_u = \sqrt{\omega_h.\omega_b}$, where ω_h and ω_b are respectively the upper and the lower frequencies. α is the order of derivative and n is the order of the approximating function (Bensafia and Ladaci, 2011).

Distinctly, the fractional-order operator has more degrees of freedom than that with integer order. It is likely that a better performance can be obtained with the proper choice of order. The long-standing discussion about the pros and cons of the different definitions are outside the scope of this paper. In short, while the Riemann-Liouville definition involves an initialization of fractional order, the Caputo counterpart requires integer order initial conditions which are easier to apply (often the Caputo initial conditions are called freely as 'with physical meaning'). So the Grünwald-Letnikov formulation is frequently adopted in numerical algorithms and control systems because it inspires a discrete-time calculation algorithm, based on the approximation of the time increment through the sampling period.

2.2 Fractional-order non-linear commensurate systems

The fractional-order linear time-invariant (LTI) system can also be represented by the following state-space model (Matignon, 1998):

$$ \begin{cases} D_t^{(q)} x(t) = Ax(t) + Bu(t) \\ \quad\quad y(t) = Cx(t) \end{cases} \quad (8) $$

where $x \in R^n$, $u \in R^r$ and $y \in R^p$ are state, input and output vectors of the system and $A \in R^{n\times n}$, $B \in R^{n\times r}, C \in R^{p\times n}$ and $q = [q_1, q_2, \dots\dots q_n]^T$ are the fractional orders.

If $q_1 = q_2 = \cdots = q_n = \alpha$, system (8) is called a commensurate-order system, otherwise it is an incommensurate-order system.

The fractional-order non-linear system can be taking the flowing fractional order state space system:

$$ \begin{cases} x_i^{(\beta_i)}(t) = x_{i+1}(t); i = 1,2,\dots,n-1. \\ \quad x_n^{(\beta_n)}(t) = f(x) + g(x)u(t) \end{cases} \quad (9) $$

where $0 < \beta i \le 1$ are the fractional differentiation orders, $f(x)$ and $g(x)$ are nonlinear continuous functions of the state variables (Jianqing and Zibin, 2009).

If $\beta_1 = \beta_2 = \cdots = \beta_n = \alpha$, this form of systems is known fractional order non-linear commensurate systems. These systems are controlled and stabilized in this work, by the conventional SMC and the proposed FSMC controllers both optimized by PSO algorithm.

3. FRACTIONAL ORDER SLIDING MODE CONTROL (FSMC)

Sliding mode control is an important and a widely studied robust scheme that has a switching nature. The state of the process under this type of control is guided towards a predefined attracting subspace of the state space such that the trajectories on it display a desired behaviour. The phase lasting until the hitting of a trajectory to the switching subspace is called reaching phase, while the motion thereafter is called sliding mode. The latter phase exhibits certain degrees of robustness against disturbances and variations in the process parameters and this result is called the invariance property. So, for a class of systems, sliding mode controller design provides a systematic approach to the problem of maintaining stability and consistent performance in the face of modelling imprecision. On the other hand, by allowing the trade offs between modelling and performance to be quantified in a simple fashion, it can illuminate the whole design process (Efe, 2011).

The purpose of the switching control law is to drive the non-linear system's state trajectory onto a pre-specified (user-chosen) surface in the state space and to maintain the system's state trajectory on this surface for subsequent time. The surface is called a switching surface. When the system state trajectory is "above" the surface, a feedback path has one gain and a different gain if the trajectory drops "below" the surface. This surface defines the rule for proper switching. This surface is also called a sliding surface (sliding manifold). Ideally, once intercepted, the switched control maintains the system's state trajectory on the surface for all subsequent time and the system's state trajectory slides along this surface. The most important task is to design a switched control that will drive the system state to the switching surface and maintain it on the surface upon interception. A Lyapunov stability approach is used to characterize this task (Mahieddine and Chrifi-Alaoui, 2008).

In this work, this control scheme type is adapted to fractional order case; we will modify the sliding surface design for a fractional order control for a class of systems with the same nature. i.e., we need to design a fractional order sliding mode controller (FSMC) for fractional order systems from a new form of the sliding surface which contains fractional order derivatives. Let a fractional-order commensurable non-linear system be defined as:

$$\begin{cases} x_i^{(\alpha)}(t) = x_{i+1}(t); i = 1,2,\dots,n-1 \\ x_n^{(\alpha)}(t) = f(x) + g(x)u(t) \end{cases}; 0 < \alpha \le 1 \tag{10}$$

where α is the order of derivation ($x^{(\alpha)} = \frac{d^{(\alpha)}x(t)}{dt^{(\alpha)}} = D^{(\alpha)}(x)$). The design procedure for a FSMC can be divided into two steps:

- Step.1: Finding the sliding surface.
- Step.2: Designing a controller $u(t)$.

3.1 Sliding surface

Consider a given reference trajectory $r(t)$ and define the state tracking errors $e_i = x_i - x_{d_i}$ with $x_{d_i} = r(t)^{(Q_i)}$ and $Q_i = \sum_{k=1}^{i-1}\alpha_k$; i=2, 3,…, n. It should be noted that, according to (10), $Q_i = \sum_{k=1}^{i-1}\alpha_k = \sum_{k=1}^{i-1}\alpha = (i-1)\alpha$; i=2, 3,4,5…, n. We choose a sliding surface (11) such that the dynamics described by S=0 is stable:

$$S(t) = e_n + \sum_{i=1}^{n-1}\lambda_i e_i \ ; \ \lambda_i \ge 0 \tag{11}$$

Now differentiate S at order α, this yields:

$$S(t)^{(\alpha)} = e_n{}^{(\alpha)} + \sum_{i=1}^{n-1}\lambda_i e_i{}^{(\alpha)} = f(x) + g(x)u(t) - x_{d_n}{}^{(\alpha)} + \sum_{i=1}^{n-1}\lambda_i e_i{}^{(\alpha)} \tag{12}$$

3.2 Designing a FSMC controller

Equating $S(t)^{(\alpha)}$ to $-k.sgn(S)$ and solving the control signal would let us have:

$$u(t) = \frac{1}{g(t)}\left[-f(x) + x_{d_n}{}^{(\alpha)} - \sum_{i=1}^{n-1}\lambda_i e_i{}^{(\alpha)} - k.sgn(S)\right] \tag{13}$$

Indeed, the application of this signal forces the reaching dynamics $S^{(\alpha)} = -\ k.sgn(S)$, which enforces $\dot{S}.S^{(\alpha)} = -k|S| < 0,\ S \ne 0$. In conventional sense, one can have the following equalities to see the closed loop stability, see (Vinagre and Calderon, 2006):

$$S^{(\alpha)} = -\ k.sgn(S) \tag{14}$$

Integrating both sides by order α yields (15), and differentiating once at order unity gives:

$$S = -\ k.(D^{(-\alpha)}sgn(S)) \tag{15}$$

$$\Rightarrow \dot{S} = -\ k.(D^{(1-\alpha)}sgn(S)) \tag{16}$$

$$\Rightarrow -\frac{1}{k}.\dot{S} = (D^{(1-\alpha)}sgn(S)) \tag{17}$$

It implies:

$$sgn(-\frac{1}{k}.\dot{S}) = sgn(D^{(1-\alpha)}sgn(S)) \tag{18}$$

According to (Vinagre and Calderon, 2006), $sgn(D^{(1-\alpha)}sgn(S)) = sgn(S)$; equation (18) yields:

$$sgn(-\frac{1}{k}.\dot{S}) = sgn(S) \tag{19}$$

with: $$sgn(S) = \begin{cases} 1 & if\ S > 0 \\ -1 & if\ S < 0 \\ 0 & if\ S = 0 \end{cases} \tag{20}$$

and: $$sgn\left(-\frac{1}{k}.\dot{S}\right) = \begin{cases} 1 & if -\frac{1}{k}.\dot{S} > 0 \\ -1 & if -\frac{1}{k}.\dot{S} < 0 \\ 0 & if -\frac{1}{k}.\dot{S} = 0 \end{cases} \tag{21}$$

Case 1:

if: $$sgn\left(-\frac{1}{k}.\dot{S}\right) = sgn(S) = 1 \Rightarrow \left(-\frac{1}{k}.\dot{S}\right).S>0 \Rightarrow \dot{S}.S< -k \Rightarrow \dot{S}.S<0;\ (k>0) \tag{22}$$

Case 2:

if: $$sgn\left(-\frac{1}{k}.\dot{S}\right) = sgn(S) = -1 \Rightarrow \left(-\frac{1}{k}.\dot{S}\right).S>0 \Rightarrow \dot{S}.S<-k \Rightarrow \dot{S}.S<0\ ;\ (k>0) \tag{23}$$

Case 3:

if: $$sgn\left(-\frac{1}{k}.\dot{S}\right) = sgn(S) = 0 \Rightarrow \left(-\frac{1}{k}.\dot{S}\right).S=0 \Rightarrow \dot{S}.S=0 \Rightarrow \dot{S}.S=0 \tag{24}$$

And this proves that the chosen form of the control signal causes $S.\dot{S} \le 0$ which verifies the sliding condition.

4. PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

Initially, the PSO algorithm chooses candidate solutions randomly within the search space. Figure 1 shows the initial state of a four-particle PSO algorithm seeking the global maximum in a one-dimensional search space. The search space is composed of all the possible solutions along the x-axis; the curve denotes the objective function. It should be noted that the PSO algorithm has no knowledge of the underlying objective function and thus has no way of knowing if any of the candidate solutions are near to or far away from a local or global maximum. The PSO algorithm simply uses the objective function to evaluate its candidate solutions and operates upon the resultant fitness values.

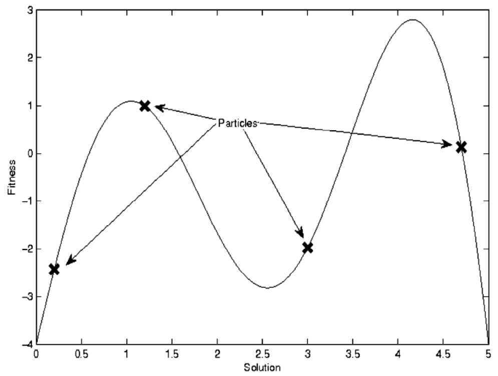

Fig. 1. Initial PSO State.

Each particle maintains its position, composed of the candidate solution, its evaluated fitness and its velocity. Additionally, it remembers the best fitness value it has achieved thus far during the operation of the algorithm, referred to as the individual best fitness and the candidate solution that achieved this fitness, referred to as the individual best position or individual best candidate solution.

Finally, the PSO algorithm maintains the best fitness value achieved among all particles in the swarm, called the global best fitness and the candidate solution that achieved this fitness, called the global best position or global best candidate solution.

The PSO algorithm consists of just three steps, which are repeated until some stopping condition is met (Frans, 2001):

1. Evaluate the fitness of each particle;
2. update individual and global best fitnesses and positions;
3. update velocity and position of each particle.

The first two steps are fairly trivial. Fitness evaluation is conducted by supplying the candidate solution to the objective function. Individual and global best fitnesses and positions are updated by comparing the newly evaluated fitnesses against the previous individual and global best fitnesses, and replacing the best fitnesses and positions as necessary. The velocity and position update step is responsible for the optimization ability of the PSO algorithm. The velocity of each particle in the swarm is updated using the following equation:

$$v_i(t+1) = w.v_i(t) + c_1.r_1.\left[p_{best_i}(t) - x_i(t)\right] + c_2.r_2.\left[g_{best}(t) - x_i(t)\right] \quad (25)$$

The index of the particle is represented by i. Thus, $v_i(t)$ is the velocity of particle i at time t and $x_i(t)$ is the position of particle i at time t. The parameters w, c_1 and c_2 ($0 \le w \le 1.2$, $0 \le c_1 \le 2$ and $0 \le c_2 \le 2$) are user-supplied coefficients. The values r_1 and r_2 ($0 \le r_1 \le 1$ and $0 \le r_2 \le 1$) are random values regenerated for each velocity update. The value $p_{best_i}(t)$ is the individual best candidate solution for particle i at time t, and $g_{best}(t)$ is the swarm's global best candidate solution at time t.

Each of the three terms of the velocity update equation has different roles in the PSO algorithm:
The first term $w.v_i(t)$ is the inertia component, responsible for keeping the particle moving in the same direction it was originally heading. The value of the inertial coefficient w is typically between 0.8 and 1.2, which can either dampen the particle's inertia or accelerate the particle in its original direction (Yuhui and Russell, 1998). Generally, lower values of the inertial coefficient speed up the convergence of the swarm to optima, and higher values of the inertial coefficient encourage exploration of the entire search space.

The second term $c_1.r_1.\left[p_{best_i}(t) - x_i(t)\right]$, called the cognitive component, acts as the particle's memory, causing it to tend to return to the regions of the search space in which it has experienced high individual fitness. The cognitive coefficient c_1 is usually close to 2, and affects the size of the step the particle takes toward its individual best candidate solution p_{best_i}.

The third term $c_2.r_2.[g_{best}(t) - x_i(t)]$, called the social component, causes the particle to move to the best region the swarm has found so far. The social coefficient c_2 is typically close to 2, and represents the sizeof the step the particle takes toward the global best candidate solution $g_{best}(t)$ the swarm has found up until that point.

The random values r_1 in the cognitive component and r_2 in the social component cause these components to have a stochastic influence on the velocity update. This stochastic nature causes each particle to move in a semi-random manner heavily influenced in the directions of the individual best solution of the particle and global best solution of the swarm.
In order to keep the particles from moving too far beyond the search space, we use a technique called velocity clamping to limit the maximum velocity of each particle (Frans, 2001). For a search space bounded by the range $[-x_{max}, x_{max}]$, velocity clamping limits the velocity to the range $[-v_{max}, v_{max}]$, where $v_{max} = k \times x_{max}$. The value k represents a user-supplied velocity clamping factor, $0.1 \le k \le 1$. In many optimization tasks, such as the ones discussed in the paper, the search space is not centered around 0 and thus the range $[-x_{max}, x_{max}]$ is not an adequate definition of the search space. In such a case where the search space is bounded by $[x_{min}, x_{max}]$, we define $v_{max} = k \times (x_{max} - x_{min})/2$.

Once the velocity for each particle is calculated, each particle's position is updated by applying the new velocity to the particle's previous position:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (26)$$

This process is repeated until some stopping condition is met. Some common stopping conditions include: a preset number of iterations of the PSO algorithm, a number of iterations since the last update of the global best candidate solution, or a predefined target fitness value. Figure 2 shows the basic PSO algorithm:

```
For each particle
    Initialize particle.
END.
Do
    For each particle
        Calculate fitness value.
        If the fitness value is better than the best fitness value
        (pbest) in history:
        Set current value as the new pbest.
    End.
Choose the particle with the best fitness value of all the
particles as the gbest
    For each particle.
        Calculate particle velocity according equation (25).
        Update particle position according equation (26).
    End.
    While maximum iterations or minimum error criteria is
not attained.
```

Fig. 2. Pseudo code of basic PSO algorithm (Rama and Sivasubramanian, 2008).

At the end of the iterations, the best position of the swarm will be the solution of the problem. It is not possible to get an optimum result of the problem always, but the obtained solution will be an optimal one. It cannot be able to an optimum result of the problem, but certainly it will be an optimal one. The convergence of the PSO algorithm toward the global optimal solution is guided by an objective function. In this work, it is defined by the following formula:

$$F(k) = \sum_{i=1}^{N} |e_1(i)| + |u(i)| \quad (27)$$

N is the number of sample, k is the iteration number, $e_1$is the state tracking error and u is the control signal.

5. SIMULATION RESULTS

We will try to control and stabilize two fractional order non-linear commensurable dynamic systems with SMC and FSMC controllers optimized by the PSO algorithm. The first system is characterized by a second order non-linear fractional system and the second system is a fractional Van der Pol oscillator (FVPO). The population size is set to 15 particles; each particle P_i has 2 elements. We set [0, 52] the range of target parameters of SMC and FSMC controllers.

The simulation is carried out using the "Matlab/Simulink" tools within 0.009 sample time. The parameters k and λ of FSMC and SMC will be calculated by using the approximations of Oustaloup to stabilize the system with approximation order $n = 5$ in a waveband $[10^{-3}, 10^{3}]\mathrm{rad/s}$.

➢ *Application to a second order non-linear fractional system*

This fractional order dynamic system is given by the following fractional order non-linear differential equations:

$$\begin{cases} x_1^{(0.3)} = x_2 \\ x_2^{(0.3)} = -x_1^3 + x_2^2 + u \end{cases} \quad (28)$$

In this simulation, both optimized controllers are tested for two control objectives, namely:

- regulation problem, i.e. for a reference signal $r(t) = 01$, and ;
- tracking problem, i.e. for a reference signal $r(t) = sin(t)$.

A. Designing of SMC

The choice of a sliding surface is given by:

$$S(t) = \dot{e}_1 + \lambda.e_1 \text{ with } e_1 = x_1 - x_{d_1} \text{and } x_{d_1} = r(t) \quad (29)$$

$$\begin{aligned}\Rightarrow S^{(0.3)} &= e_1^{(1.3)} + \lambda.e_1^{(0.3)} \\ &= \dot{x}_2 - x_{d_1}{}^{(1.3)} + \lambda.e_1^{(0.3)}\end{aligned} \quad (30)$$

$$\text{with: } \begin{aligned} e_1^{(0.3)} &= x_1^{(0.3)} - x_{d_1}{}^{(0.3)} \\ &= x_2 - x_{d_1}{}^{(0.3)}\end{aligned} \quad (31)$$

$$\Rightarrow e_1^{(1.3)} = \dot{x}_2 - x_{d_1}{}^{(1.3)} \quad (32)$$

Equating (30) to zero, we obtained:

$$\dot{x}_2 - x_{d_1}{}^{(1.3)} + \lambda.e_1^{(0.3)} = 0 \quad (33)$$

$$\Rightarrow \dot{x}_2 = x_{d_1}{}^{(1.3)} - \lambda.e_1^{(0.3)} \quad (34)$$

$$\Rightarrow D^{(-0.7)}(\dot{x}_2) = D^{(-0.7)}(x_{d_1}{}^{(1.3)} - \lambda.e_1^{(0.3)}) \quad (35)$$

$$\Rightarrow x_2^{(0.3)} = x_{d_1}{}^{(0.6)} - \lambda.e_1^{(-0.4)} \quad (36)$$

From (28), equation (36) yields:

$$-x_1^3 + x_2^2 + u_{eq} = x_{d_1}{}^{(0.6)} - \lambda.e_1^{(-0.4)} \quad (37)$$

The equivalent control signal designed is:

$$u_{eq}(t) = x_1^3 - x_2^2 + x_{d_1}{}^{(0.6)} - \lambda.e_1{}^{(-0.4)} \quad (38)$$

The global control signal designed is:

$$u(t) = x_1^3 - x_2^2 + x_{d_1}{}^{(0.6)} - \lambda.e_1{}^{(-0.4)} - k.sgn(S) \quad (39)$$

In order to reduce the chattering phenomenon in sliding mode control, it is possible to change the function $sgn(t)$ by a saturation function $sat(t)$ to replace the discontinuity in the sgn function, the control signal will be:

$$u(t) = x_1^3 - x_2^2 + x_{d_1}{}^{(0.6)} - \lambda.e_1{}^{(-0.4)} - k.sat(S) \quad (40)$$

B. Designing of FSMC

The sliding surface is given by:

$$S(t) = e_2 + \lambda.e_1 = e_1^{(0.3)} + \lambda.e_1;\ e_2 = x_2 - x_{d_2};\ x_{d_2} = r^{(0.3)} \quad (41)$$

and the control signal FSMC obtained is:

$$u(t) = x_1^3 - x_2^2 + x_{d_1}{}^{(0.6)} - \lambda . e_1{}^{(0.3)} - k. sat(S) \qquad (42)$$

λ and k parameters obtained after optimization for both controllers are given in Table.1.

Table 1. λ and k parameters of SMC and FSMC controllers obtained after optimization.

	$r(t) = 1$		$r(t) = \sin(t)$	
	λ	k	λ	k
SMC	2.2419	10.0000	9.9959	8.6824
FSMC	3.6381	6.4864	4.7288	5.0450

The optimal SMC-PSO and FSMC-PSO controller responses, control signals and fitness functions for both references cases are given in Figures (3-10).

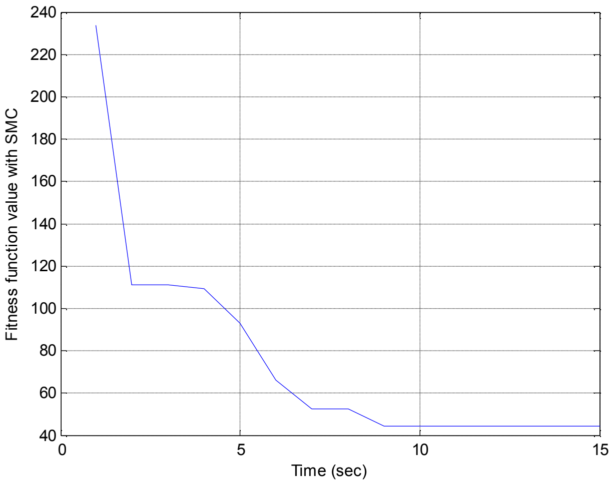

Fig. 3. Objective function value F during the optimization process with SMC for $r(t) = 1$.

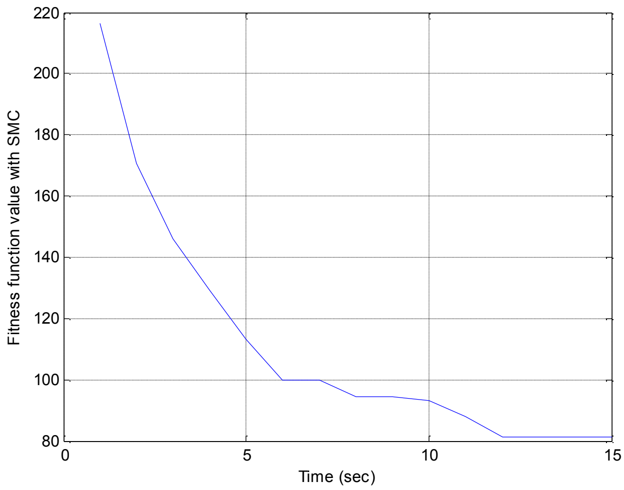

Fig. 4. Objective function value F during the optimization process with SMC for $r(t) = \sin(t)$.

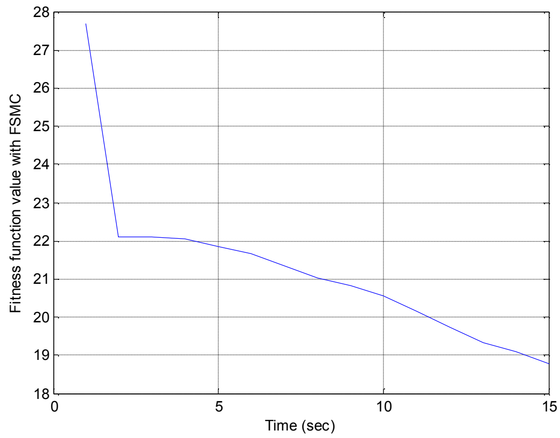

Fig. 5. Objective function value F during the optimization process with FSMC for $r(t) = 1$.

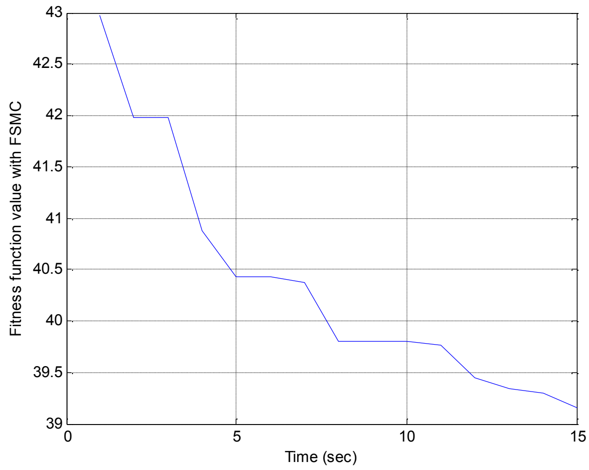

Fig. 6. Objective function value F during the optimization Process with FSMC for $r(t) = \sin(t)$

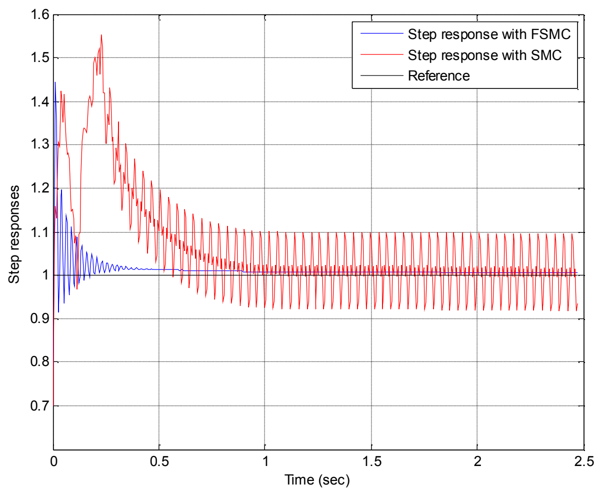

Fig. 7. Step responses of system with SMC and FSMC Controllers.

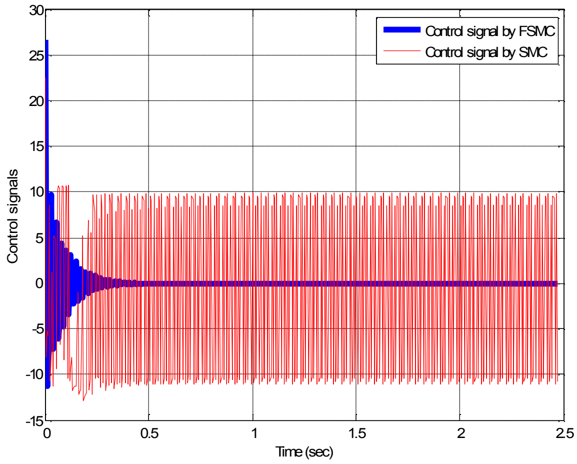

Fig. 8. Control signals of SMC and FSMC controllers with $r(t) = 1$

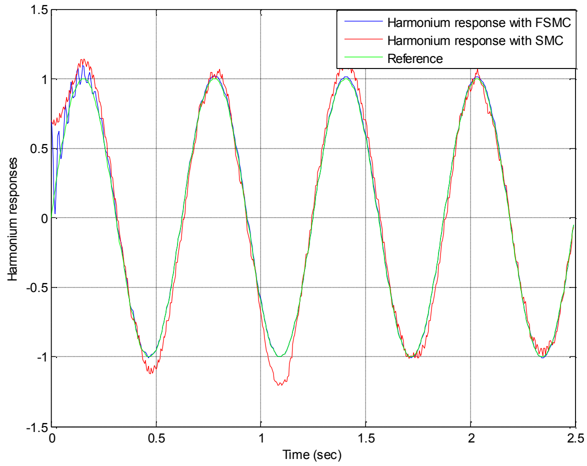

Fig. 9. Harmonium responses of system with SMC and FSMC controllers.

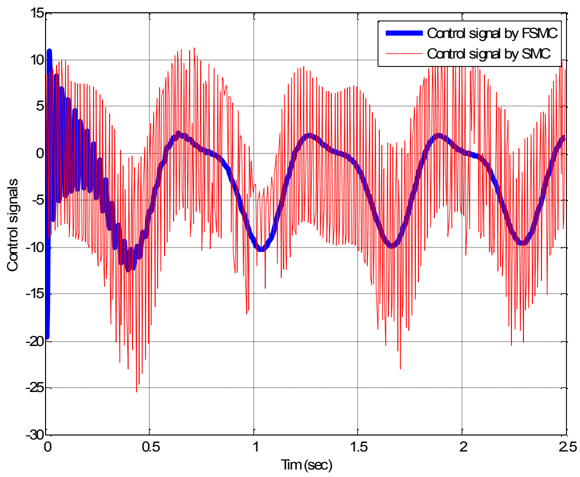

Fig. 10. Control signals of SMC and FSMC controllers with $r(t) = sin(t)$.

From those simulation results, we can easily see the superiority of the FSMC-PSO controller to the SMC-PSO controller.

➢ *Fractional Van der Pol oscillator (FVPO)*

The Van der Pol oscillator (VPO) represents a nonlinear system with an interesting behavior that exhibits naturally in several applications. It has been used for study and design of many models including biological phenomena, such as the heart beat, neurons, acoustic models, radiation of mobile phones, and as a model of electrical oscillators (implemented with a tunnel diode, memristor or operating amplifier).

The VPO model was used by Van der Pol in 1920 to study oscillations in vacuum tube circuits. In the standard form, it is given by a nonlinear differential equation of type:

$$\ddot{y}(t) + \epsilon(y(t)^2 - 1)\dot{y}(t) + y(t) = 0 \tag{43}$$

Equation (43) can be rewritten into its state-space representation as follows:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = -\epsilon({x_1}^2 - 1)x_2 - x_1 \end{cases} \text{ with } \begin{cases} x_1 = y(t) \\ x_2 = \dot{y}(t) \end{cases} \tag{44}$$

Let us consider the modified version of the VPO in the fractional order form:

$$\begin{cases} x_1^{(q)} = x_2 \\ x_2^{(q)} = -\epsilon({x_1}^2 - 1)x_2 - x_1 \end{cases} ;\text{ with } 0 < q < 1 \tag{45}$$

We inject the signal control $u(t)$ in the system (45) to stabilize the FVPO. It can write it in the following form:

$$\begin{cases} x_1^{(q)} = x_2 \\ x_2^{(q)} = -\epsilon({x_1}^2 - 1)x_2 - x_1 + f.u \end{cases} \tag{46}$$

We consider $q = 0.9$, $\epsilon = 1$, $f = 7$, initial conditions $[x_1(0) = 0.6; x_2(0) = 0]$ and the choice of a sliding surfaces is given by:

- For SMC: $S(t) = \dot{e}_1 + \lambda.e_1$ with: $e_1 = x_1 - x_{d_1}$ and $x_{d_1} = r(t)$
- For FSMC: $S(t) = e_2 + \lambda.e_1 = e_1^{(0.3)} + \lambda.e_1$; $e_2 = x_2 - x_{d_2}$; $x_{d_2} = r(t)^{(0.3)}$

We applied the same steps as in the last application, we find the following results.

The control signal SMC is given by:

$$u(t) = \frac{1}{7}.[({x_1}^2 - 1)x_2 + x_1 + {x_{d_1}}^{(1.8)} - \lambda.{e_1}^{(0.8)} - k.sat(S)] \tag{47}$$

and the control signal FSMC obtained is:

$$u(t) = \frac{1}{7}.[({x_1}^2 - 1)x_2 + x_1 + {x_{d_1}}^{(1.8)} - \lambda.{e_1}^{(0.9)} - k.sat(S)] \tag{48}$$

λ and k parameters obtained after optimization for both controllers are given in Table.2.

Table 2. λ and k parameters of SMC and FSMC controllers obtained after optimization.

	$r(t) = 1$		$r(t) = \sin(t)$	
	λ	k	λ	k
SMC	9.0000	2.2683	9.0000	3.4412
FSMC	5.0167	4.1086	8.8758	2.1570

The optimal SMC-PSO and FSMC-PSO controllers responses, control signals and fitness functions for both references cases are given in Figures (11-19).

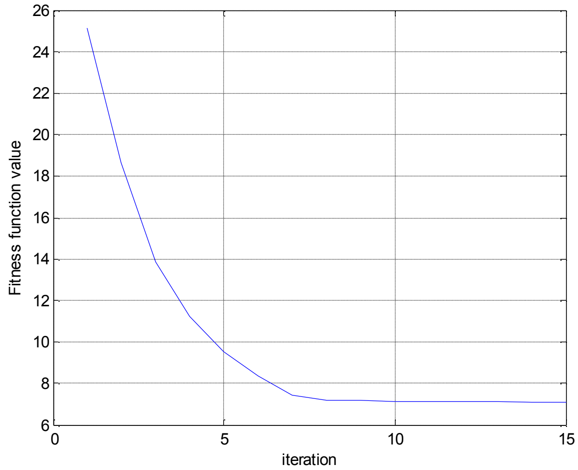

Fig. 11. Objective function value F during the optimization process of FVPO with SMC for $r(t) = 1$.

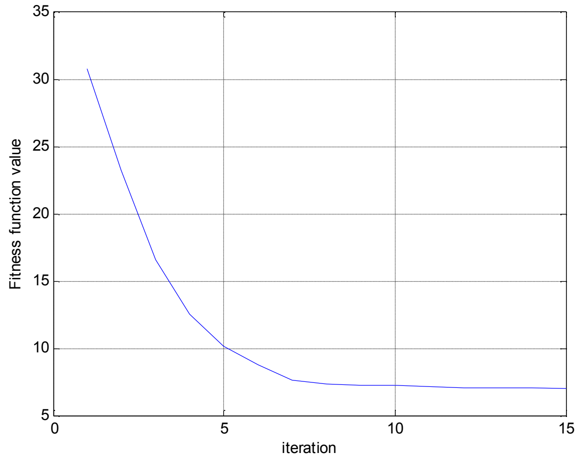

Fig. 12. Objective function value F during the optimization process of FVPO with SMC for $r(t) = \sin(t)$.

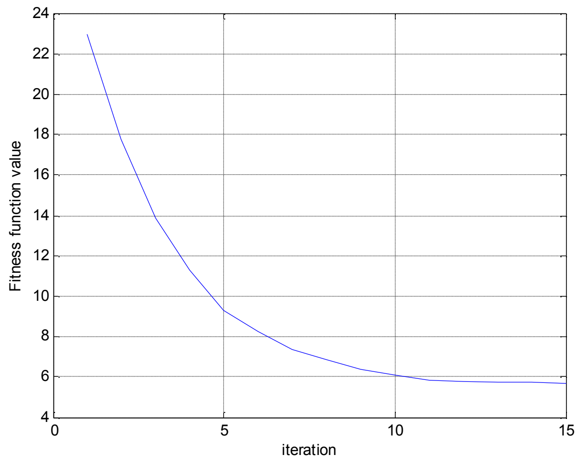

Fig. 13. Objective function value F during the optimization process of FVPO with FSMC for $r(t) = 1$.

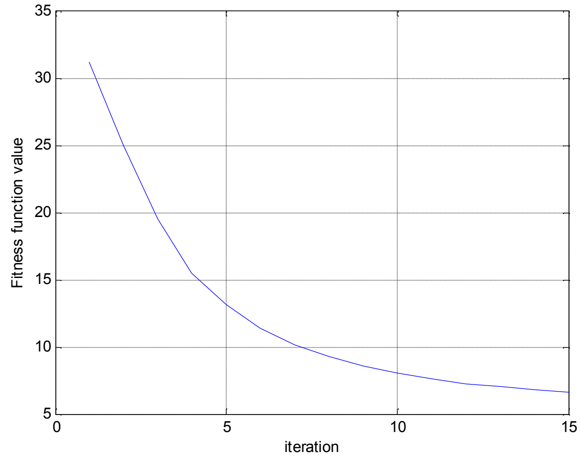

Fig. 14. Objective function value F during the optimization process of FVPO with FSMC for $r(t) = sin(t)$.

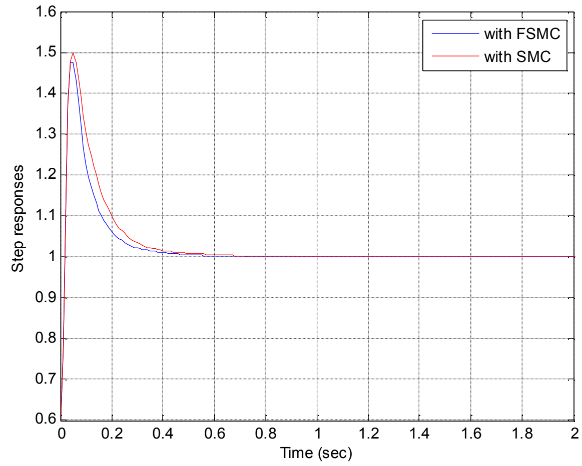

Fig. 15. Step responses of FVPO with SMC and FSMC controllers.

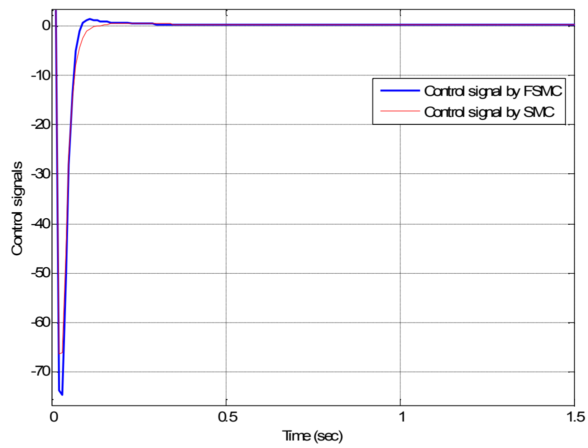

Fig. 16. Control signals of SMC and FSMC controllers with $r(t) = 1$.

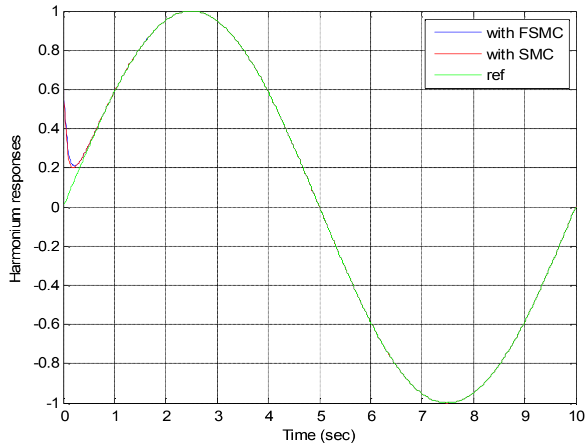

Fig. 17. Harmonium responses of FVPO with SMC and FSMC controllers.

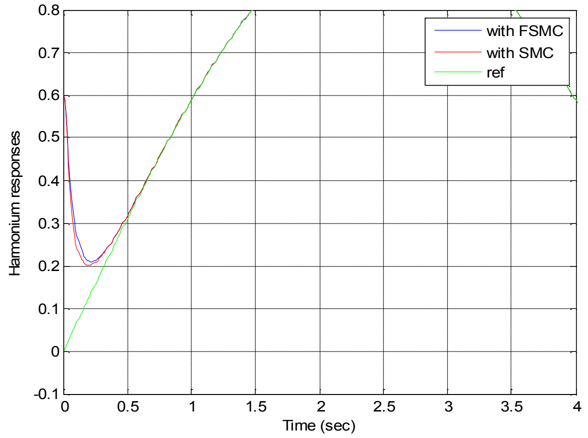

Fig. 18. Zoom of harmonium responses of FVPO with SMC and FSMC controllers.

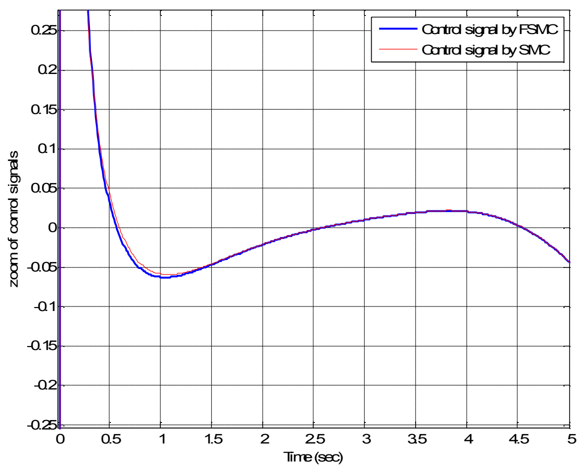

Fig. 19. Control signals of SMC and FSMC controllers with $r(t) = sin(t)$ (zoom)

We can see that the FSMC-PSO controllers stabilize the FVPO and ensure the best rapidity by comparing them with those of an integer order SMC-PSO controllers.

6. CONCLUSION

In this paper a design of SMC and FSMC controllers for controlling a class of fractional order nonlinear systems is investigated. A comparison between SMC and FSMC in two application examples based on the optimization of the two parameters λ and k by PSO algorithm is made. The simulation results show clearly that the stability and the robustness issues using FSMC are more enhanced than those obtained using SMC (a better speed and a very little error). This robustness is also depending on the nature of the chosen sliding surface. We can also conclude that the choice of a sliding surface should be the same nature with the systems to be controlled, i.e., to design a fractional order sliding mode control for stabilizing this form of fractional order systems, it must select a fractional order sliding surface also contains same derivatives (or integrals) orders of the plants to be stabilised.

REFERENCES

Bensafia, Y. and Ladaci, S. (2011). Adaptive control with fractional order reference model. *International journal of sciences and techniques of automatic control & computer engineering,* Vol. 5, pp.1614-1623.

Delavari, H., Ghaderi, R., Ranjbar, A. N., HosseinNia, S.H. and Momani, S. (2010). Adaptive Fractional PID Controller for Robot Manipulator. *Proceedings of FDA'10. The 4th IFAC Workshop Fractional Differentiation and its Applications*, Spain. Article No. FDA10-038, pp.01-07.

Duma, R., Trusca, M., and Dobra, P. (2011). Tuning and Implementation of PID Controllers using Rapid Control Prototyping. *Journal of Control Engineering and Applied Informatics, CEAI*, Vol.13, No.4, pp. 64-73.

Efe, M.O. (2008). Fractional Fuzzy Adaptive Sliding-Mode Control of a 2-DOF Direct-Drive Robot Arm. *IEEE Transactions Systems, Man, and Cybernetics, Part B: Cybernetics,* Vol.38, No.6, pp. 1561-1570.

Efe, M.O. (2011). Fractional Order Systems in Industrial Automation—A Survey. *IEEE Trans. on Industrial Informatics*, Vol. 7, No. 4, pp. 582 - 591.

Fransvanden, B. (2001). *An Analysis of Particle Swarm Optimizers*. *PhD thesis.* Faculty of Natural and Agricultural Science, pp.283, University of Pretoria.

Jianqing, M. and Zibin, X. (2009). Backstepping Control for a Class of Uncertain Systems Based on Non-singular Terminal Sliding Mode. *International Conference on Industrial Mechatronics and Automation*. pp.169 - 172

Mahieddine, M. S. and Chrifi-Alaoui, L. (2008). Sliding Mode Control of non linear SISO systems with both matched and unmatched disturbances. *International journal of sciences techniques of automatique control & computer engineering.* Vol.2, pp.350-367.

Matignon, D. (1998). Generalized fractional differential and difference equations: stability properties and modelling issues, *Proc. of the Math. Theory of Networks and Systems Symposium*, Padova, Italy. Vol. 5, pp. 145-158.

Miller, K. S. and Ross, B. (1993). An Introduction to the Fractional Calculus and Fractional Differential Equations. *New York. Conference on Evolutionary Computation, John Wiley & Sons. Inc.,* pp. 69–73.

Nakagawa, M. and Sorimachi, K. (1992). Basic characteristics of a fractance device, *IEICE Trans. fundamentals*, E75-A, pp.1814–1818.

Oldham, B. and Spanier, J. (1974). The fractional calculs. *Academic Press*, New York.

Oustaloup, A. (1995). *La Derivation Non Entiere: Theorie, Synthese et Applications*. Hermes, Paris.

Petras, I., (2011). *Fractional-Order Nonlinear Systems*. Springer, pp.09-10, Berlin Heidelberg.

Podlubny, I. (1994). Fractional-order Systems and Fractional-order Controllers. *The Academy of Sciences Institute of Experimental Physics*, UEF-03-94, Kosice, Slovak Republic.

Podlubny, I. (1999). *Fractional Differential Equations*. Academic Press, pp.38-42, San Diego.

Rama, M. R.A. and Sivasubramanian, K. (2008). Multiobjective optimal design of fuzzy logic controller using a self configurable swarm intelligence algorithm. *Computers and Structures* 86.

Sabatier, J. and Farges, C. (2012). On stability of commensurate fractional order systems. *Int. J. Bifurcation Chaos Appl. Sci. Eng.,* Vol.22, No. 4, pp.1-8.

Shi, Y. and Eberhart, R.C. (1998). A modified particle swarm optimizer. *In Proceedings of 1998 IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA. pp. 69-73

Vinagre, B. and Calderon, A. (2006). On fractional sliding mode control. *Proc. Of the 7th Portuguese Conference on Automatic Control*, Lisbon, Portugal.

Wang, J. C. (1987). Realizations of generalized Warburg impedance with RC ladder net works and transmission lines. *Journal of The Electrochemical Society*, Vol.134, pp.1915–1920.

Westerlund, S. (2002). *Dead Matter Has Memory!, Causal Consulting*, Kalmar, Sweden.

Xue, Y., Li, A., Wang, L., Feng, H. and Yao, X. (2006). *Prediction of PK-specific phosphorylation site with Bayesian decision theory*. University of Science and Technology of China, Hefei, Anhui, 230027, China.

Yuhui, S. and Russell, E. (1998). A modified particle swarm optimizer. *The 1998 IEEE InternationalConference on Evolutionary Computation Proceedings*, pp.69-73.

Yahyazadeh, M. (2008). Application of fractional derivative in control functions. *India Conference, 2008. INDICON 2008, Annual IEEE,* Vol.1, pp.252-257.