TUNING FUZZY PD\(^\alpha\) SLIDING-MODE CONTROLLER USING PSO ALGORITHM FOR TRAJECTORY TRACKING OF A CHAOTIC SYSTEM

N. BOUARROUDJ, D. BOUKHETALA, F. BOUDJEMA
Laboratoire de Commande des Processus, Département d’automatique, Ecole Nationale Polytechnique, Algiers, Algeria
E-mails: autonour@gmail.com, djamel.boukhetala@enp.edu.dz, fares.boudjema@enp.edu.dz

B. AHSENE
LAMEL, Faculté des Sciences et de la Technologie, Université de Jijel, Algeria
E-mail: ah_boubakir@yahoo.fr

Abstract: This paper deals with the comparison between fuzzy sliding mode controller (FSMC) and novel approach of FSMC using a PD\(^\alpha\) sliding surface, for a trajectory tracking of a chaotic system. In order to alleviate the chattering phenomenon due to the discontinuity in the signum function, a Takagi-Sugeno fuzzy logic controller is used, and to ensure optimal performance in the closed loop system, the PSO algorithm is also used. Finally the effectiveness of the proposed approach of FPD\(^\alpha\)-SMC-based PSO algorithm is demonstrated by simulation results.

Key words: SMC, FSMC, FPD\(^\alpha\)SMC, PSO, chaotic system.

1. Introduction

Fractional-order calculus is an area of mathematics; it has 300 years of history that deals with derivatives and integrals from non-integer orders. In the last two decades, fractional calculus has been rediscovered by scientists and engineers. It has been applied in a many number of fields, namely in the area of control theory such as, A Fractional Order PID Tuning Algorithm for A Class of Fractional Order Plants [1], A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices [2], Observer Based Control of a Class of Nonlinear Fractional Order Systems using LMI [3], Optimized wave-absorbing control: Analytical and experimental results [4].

Besides, the SMC for example was largely proved its efficiency through the reported theoretical studies [5], [6]. The first step of SMC design is to select a sliding surface that models the desired closed-loop performance in state variable space. The second step is to design the equivalent and a hitting control law such as the system state trajectories forced toward the sliding surface and slides along it to the desired attitude. In the literature, several methods for selecting sliding surface have been reported. The approach in [7, 8, 9, 10] uses a proportional-derivative type sliding surface, where the order of derivation is an integer. In [11] a nonlinear sliding surface for the coupled tanks system is adopted and given best results. Due to the fact that the fractional order controller plays an important role in various domains, a PD\(^\alpha\) sliding surface is proposed in [12] [29], and a novel fractional integral terminal sliding mode concepts for the output tracking problem of relative-degree-one systems with uncertainty and disturbance is presented in [30]. Also authors in [31] have proposed a novel fractional-order integral type sliding surface for robust stabilization/ synchronization problem of a class of fractional-order chaotic systems in the presence of model uncertainties and external disturbances.

Motivated by the above discussion this paper designs a sliding surface based on the fractional order proportional-derivative (PD\(^\alpha\)) [12], the best choice of the order of the sliding surface can results a small output response, improves settling time and stability of the system.

Then, to make the developed surface globally attractive and invariant, the control law is designed.

An advantage of these methods of control (SMC) is their robustness to parameter perturbations and bounded external disturbances. The robustness is attributed to the discontinuous term in the control input. However, this discontinuous term also causes an undesirable effect called chattering.

 Sometimes this discontinuous control action can even cause the system performance to be unstable. To reacha better compromise between small chattering and good tracking precision, various compensation strategies have been proposed.

For example, integral sliding control [13, 14, 15], a fuzzy sliding mode control strategy [16]. Though introducing a fuzzy logic controller and taking off the sign function in the hitting control law of SMC may reduce the chatter amplitude.

The selection of suitable parameters of fuzzy PD\(^\alpha\)
sliding mode controller (FPDαSMC) is a significant problem, that it can be solved either by manually changing the values or to use some optimization methods [36],[37], in this paper we are interested by the particle swarm optimization algorithm (PSO).

The rest of this article is organized as follows. Basic definitions of fractional calculus are described in Section 2. The Fuzzy PDα sliding mode controller design, in Section 3. After that the PSO approach is described in Section 4. The optimization of FPDαSMC with PSO in Section 5. And finally the simulation results and conclusion are given in Sections 6 and 7, respectively.

2. Basic definitions of fractional calculus

The fractional differ-integral operators denoted by $\alpha D^\alpha_t$ (Fractional calculus) are a generalization of integration and differentiation of the operators of a non integer order. In the literature we find different definitions of fractional differ-integral, but the commonly used are:

The Riemann-Liouville (RL) definition:

$$aD^\alpha_t f(t) = \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_{a}^{t} \frac{f(\tau)}{(t-\tau)^{1-(m-\alpha)}} d\tau$$  \hspace{1cm} (1)

The Caputo’s definition:

$$aD^\alpha_t f(t) = \frac{1}{\Gamma(m-\alpha)} \int_{a}^{t} \frac{f^{(m)}(\tau)}{(t-\tau)^{1-(m-\alpha)}} d\tau$$  \hspace{1cm} (2)

Where $m-1<\alpha<m$ and $\Gamma(\cdot)$ is the well-known Euler’s gamma function, and its definition is:

$$\Gamma(x) = \int_{0}^{\infty} e^{-t} t^{(x-1)} dt, \ x > 0$$  \hspace{1cm} (3)

Where, $x$ is the order of the integration.

On the other hand, Grunwald-Letnikov (GL) reformulated the definition of the fractional order derivative as follows:

$$aD^\alpha_t f(t) = h^{lim} \frac{1}{h^\alpha} \sum_{k=0}^{\frac{(t-a)/h}{h}} (-1)^{\frac{a}{k}} f(t-kh)$$  \hspace{1cm} (4)

Because the numerical simulation of a fractional differential equation is not simple as that of an ordinary differential equation, so the Laplace transform method is often used as being a tool for the resolution of the problems arising in engineering [17, 18].

In the following section, we give the Laplace transforms of the fractional order derivative given previously.

The Laplace transform of RL definition is as follows [17],[32]:

$$\int_{0}^{\infty} e^{-\alpha} f(t) dt = s^\alpha F(s) - \sum_{k=0}^{m-1} \alpha D^{\alpha-k-1} f(t)|_{t=0}$$  \hspace{1cm} (5)

Where $s=j\omega$ denotes the Laplace operator. For zero initial conditions, the Laplace transform of fractional derivative of Riemann-Liouville, caputo and Grunwald-Letnikov reduce to (6) [32],[33].

$$L\left(\alpha D^\alpha f(t)\right) = s^\alpha F(s)$$  \hspace{1cm} (6)

In this paper the fractional order element $s^\alpha$ is approximated with Oustaloup’s filter [19]. The Oustaloup’s filter is based on the approximation of a function of the form:

$$G(s) = s^\alpha, \ \alpha \in R^+$$  \hspace{1cm} (7)

By a rational function:

$$\hat{G}(s) = s^\alpha = K \prod_{k=-N}^{N} \frac{s+w_k}{s^2+w_k^2}$$  \hspace{1cm} (8)

Where the parameters of this function (zeros, poles, and gain) can be determined by the following formulas:

$$w_k = w_b \left(\frac{w_p}{w_b}\right)^{\left[(k+N+0.5\alpha)/(2N+1)\right]}$$ \hspace{1cm} (9)

$$w_k = w_b \left(\frac{w_p}{w_b}\right)^{\left[(k+N+0.5\alpha)/(2N+1)\right]}$$ \hspace{1cm} (9)

where $w_b$ and $w_p$ are respectively the low and high transient-frequencies. In this paper we consider the 5th order Oustaloup’s rational approximation for the FO element within the frequency range $w \in [10^{-2},10^{2}]$.

3. Fuzzy PDα sliding mode controller design

We consider the following state-space representation of the second-order nonlinear system:

$$\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f(x) + b(x)u
\end{align*}$$  \hspace{1cm} (10)

Where $x=[x_1, x_2]^T$ is the state vector, $f(x)$ and $b(x)$ are nonlinear functions, $u$ is the control input designed to track a command $x_{1d}(t)$ closely. Without losing
generally, assume \( b(x) > 0 \) for all \( x \).

For this kind of system, we find several control methods, such as, fuzzy control, PID control, sliding mode control, etc.

**A. PD sliding mode controller (PD SMC)**

For the system presented in Eq. (10), firstly we use the following PD sliding surface using Caputo’s definition as:
\[
S = D^{\alpha}_{t} (e) + k_p e
\]

(11)

**Remark:** It is clear that selecting \( \alpha = 1 \), a classical PD sliding surface can be recovered.

The fractional derivatives Caputo right hand definition (RHD) [34] of function \( f(t) \) gives,
\[
D^{\alpha}_{t} f(t) = D^{m}_{t} (e^{(m-\alpha)} \frac{d^m}{dt^m} f(t)), \text{ where } m \text{ is an integer greater than } \alpha.
\]

From this we can write the sliding surface \( S \) as follows:
\[
S = D^{\alpha-1}_{t} (\dot{e}) + k_p (e)
\]

(12)

Where \( e = x_1 - x_{id} \), and \( k_p \) is a positive constant.

It is obvious from (11) that keeping system states on the sliding surface \( S(x, t), \forall t > 0 \) will guarantee that the tracking error vector asymptotically approach to zero.

The corresponding sliding condition is:
\[
\frac{1}{2} \frac{d}{dt} S^2 = S \dot{S} \leq 0
\]

(13)

The general control structure that satisfies the stability condition of the sliding motion can be written as:
\[
u = u_{eq} + \frac{1}{b(x)} D^{(1-\alpha)}_{t} (u_h)
= u_{eq} + \frac{1}{b(x)} D^{(1-\alpha)}_{t} (-K_x, \text{sgn}(S))
\]

(14)

Where \( u_{eq} \) is called the equivalent control law that is derived by setting \( S = 0 \) and \( K_x \) is a positive constant.

We refer to [32] for more details. Differentiating both sides of Eq (12) to the order unity yields the equality in (15)
\[
\dot{S} = D^{(\alpha-1)}_{t} \dot{e} + k_p, (\dot{e}) = D^{(\alpha-1)}_{t} (\ddot{x_1} - \ddot{x}_{id}) + k_p, (\dot{e})
\]

(15)

From Eq (15) one can conclude that:
\[
D^{(1-\alpha)}_{t} (\dot{S}) = (\ddot{x_1} - \ddot{x}_{id}) + k_p D^{(1-\alpha)}_{t} (\dot{e})
\]

(16)

By setting \( \dot{S} = 0 \), and substituting \( \ddot{x}_1 = \ddot{x}_2 \), the equivalent control is obtained, and it has the flowing formula:
\[
u_{eq} = -\frac{1}{b(x)} \left[ f(x) - \dot{x}_{id} + k_p, D^{(1-\alpha)}_{t} (\dot{e}) \right]
\]

(17)

To verify the stability analysis, substituting Eq(14) into Eq(10) yields:
\[
\ddot{x}_2 = \ddot{x}_1 - k_p, D^{(1-\alpha)}_{t} (\dot{e}) - K_x, D^{(1-\alpha)}_{t} (\text{sgn}(S))
\]

(18)

Eq(18) becomes
\[
\ddot{x}_2 = \ddot{x}_1 - k_p, D^{(1-\alpha)}_{t} (\dot{e}) - K_x, D^{(1-\alpha)}_{t} (\text{sgn}(S))
\]

(19)

By using Eq (16), the Eq (19) can be rewritten as follows:
\[
D^{(1-\alpha)}_{t} (\dot{S}) = -K_x, D^{(1-\alpha)}_{t} (\text{sgn}(S))
\]

(20)

Diffe rentiate (20) to the order \( \alpha -1 \). Since \( (\alpha-1) < 0 \) this indeed corresponds to fractional order integration, corresponding to negative valued \( \alpha \) in Caputo’s definition in (2), and taking into account the property of Caputo’s derivative
\[
\frac{d}{dt} D^{(\alpha)}_{t} f(t) = f(t) - \sum_{k=0}^{m-1} \frac{t^k}{k!} \frac{d^k f(t)}{dt^k}; \text{ with }
\]
\[
\dot{S}(0) = 0; \frac{d}{dt} (\text{sgn}(S)) = 0 \text{ for } m = 1. \text{ This lets us have:}
\]
\[
\dot{S} = -K_x, (\text{sgn}(S)) + K_x, (\text{sgn}(S(0)))
\]

(21)

For \( t = 0 \), we have \( \dot{S} = 0 \) and for \( t > 0 \) we have
\[
\dot{S} = -K_x, (\text{sgn}(S)) \>

Thus by using (13) we can obtain:
\[
S \dot{S} = S (t - K_x, (\text{sgn}(S)))
= -K_x, |S| \leq 0
\]

(22)

**Lemma 1.** [28] Consider the following autonomous linear fractional-order system:
\[
D^{\alpha}_{t} x(t) = A x(t), \quad x(0) = x_0
\]

(23)

Where \( x \in \mathbb{R}^n, A = (a_{ij}) \in \mathbb{R}^{n \times n}, 0 < \alpha < 1, \) is asymptotically stable if and only if (see figure 1):
\[
\arg(\text{eig}(A)) > \frac{\pi}{2}
\]

(24)

In this case, the components of the state decay towards 0 like \( t^{-\alpha} \).
Fig. 1. Stable domain of fractional order system in \( s^{\alpha} \) plane

**Proof.** When the sliding mode occurs, system (11) can be represented as follow:

\[
D^{\alpha}_t (e) + k_p e = 0
\]  
(25)

Therefore, the sliding mode dynamics is obtained by the following equation:

\[
D^{\alpha}_t (e) = -k_p e
\]  
(26)

It can be seen that \( \arg(-k_p) = \pi \), so the sliding surface parameter \( k_p \) is selected to be positive to satisfy the stability condition of Lemma 1.

In summary, the used \( PD^{\alpha} \) sliding surface can guarantee the stability in the sense of the Lyapunov theorem.

However, a large control gain \( K_s \) often causes the chattering effect. In order to tackle this problem, a continuous fuzzy logic control term \( \Delta u \) is used to approximate \( u_c \).

**B. Fuzzy PD\(^{\alpha}\) Sliding Mode Controller (FPD\(^{\alpha}\)SMC)**

The FPD\(^{\alpha}\)SMC is a hybrid controller; it can be regarded as a fuzzy regulator that controls the variable \( S \) approach to zero.

The structure of a fuzzy controller design consists of: 1) the definition of input-output fuzzy variables; 2) decision-making related to fuzzy control rules; 3) fuzzy inference logic; and 4) defuzzification.

For the proposed FPD\(^{\alpha}\)SMC, we used the sliding surface \( S \) as the input at the fuzzy controller, and \( \Delta u \) is the fuzzy controller output. The structure is shown in figure 2:

Where:

\[
u = u_{eq} + u_f = u_{eq} + \frac{1}{b(x)} D^{(1-\alpha)}_t (\Delta u)
\]  
(27)

Assuming that the input and output of the fuzzy controller has five level language variables, its membership function is shown in figure 3. Where \( \phi \) and \( K \) are used to expand or shrink the divisions of the membership functions along the universes of discourse, \( r \) is a coefficient to be used to adjust the width of the input membership function of the linguistic variable Zero [24].

Such linguistic expressions can be used to form the fuzzy control rules as below:

Rule 1: IF \( S \) is \( NB \), THEN \( \Delta u \) is \( PB \).
Rule 2: IF \( S \) is \( NM \), THEN \( \Delta u \) is \( PM \).
Rule 3: IF \( S \) is \( ZO \), THEN \( \Delta u \) is \( ZO \).
Rule 4: IF \( S \) is \( PM \), THEN \( \Delta u \) is \( NM \).
Rule 5: IF \( S \) is \( PB \), THEN \( \Delta u \) is \( NB \).

Where \( NB \) denotes "Negative Big", \( NM \) denotes "Negative Mid", \( ZO \) denotes "Zero", \( PB \) denotes "Positive Big", and \( PM \) denotes "Positive Mid".

The FLC output (\( \Delta u \)) is determined using the weighted average method [11].

**4. Particle swarm optimization (PSO)**

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [20]. The inspiration underlying the development of this algorithm was the social behaviour of animals, such as the flocking of birds and the
schooling of fish, and the swarm theory. It has been proven to be efficient in solving optimization problems especially for nonlinearity and non differentiability, multiple optimum, and high dimensionality [21, 22].

In PSO, the velocity of each particle is modified iteratively by its individual best position \((p_{best})\), and the global best position \((gbest)\) found by particles in its neighborhood. As a result, each particle searches around a region defined by its individual best position \((p_{best})\) and the global best position \((gbest)\) from its neighborhood. Henceforth we use \(V_i\) to denote the velocity of the \(i^{th}\) particle in the swarm, \(p_i\) denote its position. At each step (or iteration) \(n\), by using the individual best position, \((p_{best})\), and global best position, \((gbest)\), the velocity and position of each particle are updated by the following two equations:

\[
V_i(n) = W[V_i(n-1) + c_1r_1(p_{best} - p_i(n-1)) + c_2r_2(g_{best} - p_i(n-1))] \\
p_i(n) = p_i(n-1) + V_i(n)
\]

Where \(r_1\) and \(r_2\) are random numbers between 0 and 1; \(c_1\) and \(c_2\) are positive constant learning rates; \(W\) is called the constriction factor [23] and is defined by (30):

\[
W = \frac{2}{2 + c - \sqrt{c^2 - 4c}} \quad c = c_1 + c_2, \quad c > 4
\]  

In each step \(n\) the position is confined within the range of \([p_{min}, p_{max}]\). If the position violates these limits, it is forced to its proper values [21].

\[
p_i = \begin{cases} 
  p_{min} & \text{if } p_i < p_{min} \\
  p_i & \text{if } p_{min} < p_i < p_{max} \\
  p_{max} & \text{if } p_i > p_{max}
\end{cases}
\]

Changing position by this way enables the \(i^{th}\) particle to search around its individual best position \((p_{best})\) and global best position, \((gbest)\).

The following shows the design step for implementing the PSO algorithm [21].

**Step 1.** Initialize particles with random position and velocity on dimension in the problem space.

**Step 2.** If a prescribed number of iterations (generations) is achieved, then stop the algorithm.

**Step 3.** For each particle, evaluate the desired optimization fitness function, and record each particle’s best previous position \((p_{best})\), and global best position \((gbest)\).

**Step 4.** Change the velocity and position according to equations (28) and (29) respectively, for each particle

**Step 5.** Check each particle’s position using (31).

**Step 6.** Go back to Step 2.

### 5. Optimization of FPD-SMC with PSO

The design problem is defined as finding the optimum values of the fuzzy PD\(^*\) sliding mode controller parameters in the closed-loop system. The Parameters vector composed by the positions of the membership functions (when the conclusions are fixed), the gain \(k_p\), and the fractional order \(\alpha\).

Let \(p_i = [\phi, r, K, k_p, \alpha]\) the vector of selective parameters of FPD-SMC, the regions of the decision variables are mentioned as follows.

\[0.1 < \phi < 10, \quad 0.1 < r < 1, \quad 0.1 < K < 20, \quad 0.01 < k_p < 20, \quad 0.1 < \alpha < 0.98\]

To converge toward the optimal solution, the PSO algorithm must be guided by the cost function. Hence, it should be properly defined before the PSO algorithm is executed.

In the present study, the used cost function \((F_1)\) is defined by the following formula:

\[
F_1 = \sum_{i=1}^{N} (\gamma_1 |e(i)| + \gamma_2 |u(i)|)
\]  

In order to compare the performances of the different controllers, we define the flow cost functions:

\[
F_2 = \sum_{i=1}^{N} (e^2(i))
\]

\[
F_3 = \sum_{i=1}^{N} (u^2(i))
\]

Where \(e(i)\) is the trajectory error of \(i^{th}\) sample, \(u(i)\) is the control signal of \(i^{th}\) sample and \(N\) is the number of samples. The weighting factors \(\gamma_1\) and \(\gamma_2\) are used to give more flexibility to the designer depending on the nature of application and relative importance of low error and control signal.

### 6. Simulation results

In this section, we shall demonstrate that the FPD-SMC tuned with PSO is applicable to the problem of trajectory tracking control of a non-linear chaotic system. Tuning process with PSO is also applied to the Fuzzy Sliding Mode Controller (FSCM) using PD sliding surface.

For the robustness evaluation of the different controllers tuned by PSO algorithm, an external disturbances \((d=0.2\sin(t))\) is added to the system, and
the parameters variation of the system is carried out, 
where the controllers’ parameters tuned by PSO are kept 
unchanged.

The simulation is carried out using 
the “Matlab/Simulink” tools within 0.01 sample time. 
The population size of PSO algorithm is set to 20 particles. 
The parameters $c_1$, $c_2$ and $W$ are set to 2.05, 2.05 and 
0.7298 respectively, and the maximum number of 
iteration $n$ is set to 40 iterations. The weighting factors 
$\gamma_1$ and $\gamma_2$ are set to 3 and 0.1 respectively. The cost 
function (32) is minimized for each of the FSMC and the 
FPD*SMC controllers with the corresponding controller 
parameters reported in Table 1.

Table 1
Optimal parameters for FSMC and FPD*SMC

<table>
<thead>
<tr>
<th>Performance index</th>
<th>FSMC</th>
<th>FPD*SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$</td>
<td>0.1881</td>
<td>0.1000</td>
</tr>
<tr>
<td>$r$</td>
<td>0.1000</td>
<td>0.3513</td>
</tr>
<tr>
<td>$K$</td>
<td>12.5932</td>
<td>4.1112</td>
</tr>
<tr>
<td>$k_p$</td>
<td>4.7645</td>
<td>10.9677</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>-</td>
<td>0.9800</td>
</tr>
</tbody>
</table>

A. Example (Chaotic system)

In recent years, chaotic systems have attracted 
considerable interest and have been extensively 
investigated. An interesting subject in chaos theory is to 
eliminate the chaotic behavior by means of control 
systems [25], [26], [35], [38]. Consider a second-order 
chaotic system such as [27].

$$
\dot{x}_1 = x_2 \\
\dot{x}_2 = -0.4x_2 + 1.1x_1 - x_1^3 + q \cos(w_c t) + u + d
$$

(35)

Where $u$ the control signal and $d$ is an unknown external 
disturbance assumed to be bounded as follows:

$$
|d| \leq D
$$

(36)

Simulation result of tracking control of the state $x_1(t)$ 
with a desired reference wave is shown in figure. 4 
where $q$ and $w_c$ are set to 2.1 and 1.8 respectively. 
Figure. 5 shows the simulation result with parameters 
variation and adding external disturbance. The expression 
of the desired reference ($x_{1d}$) is given as follows:

$$
x_{1d} = 0.12(\cos(\pi t) + \sin(\frac{2\pi t}{3}))
$$

(37)

Where the initial conditions ($x_1(0)$; $x_2(0)$) are set to (0.3, 
0).

From the simulation results, the FPD*SMC performs 
better control specification such as fast response 
compared with FSMC, also it is evident that the 
FPD*SMC outperforms the FSMC for trajectory 
tracking task.

However when compared with respect to small 
magnitude of control signal, the FSMC gives better 
results.

After adding external disturbance and changing in 
parameters of the system we could obviously find that 
the disturbance rejection ability of different controllers 
tuned with PSO.

7. Conclusion

In this paper a Fuzzy PD* Sliding Mode Controller that 
combines the advantages in term of robustness of the 
fractional calculus, fuzzy logic for its ability to express 
the amount of ambiguity in human reasoning and sliding 
mod controller in term of robustness to parameters
variation and external disturbances, is investigated for a chaotic system.

Firstly, PD* surface sliding mode controller is used. The design yields an equivalent control term with an addition of fuzzy logic control to approximate the discontinuous control term and to alleviate the chattering phenomenon. Then the application of the PSO method can perform an efficient search for the optimal parameters of both FS and FPD*SMC, and achieve good accuracy. Finally, the simulation results show the effectiveness of the proposed controller algorithm for nonlinear chaotic systems.

References


26. Chun-Fei, H., Chih-Min. L, and Rong-Quan, Y.: Supervisory adaptive dynamic RBF-based neural-fuzzy...


32. Podlubny. I.: Fractional differential equations, an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, mathematics in science and engeneering, volume 198.


