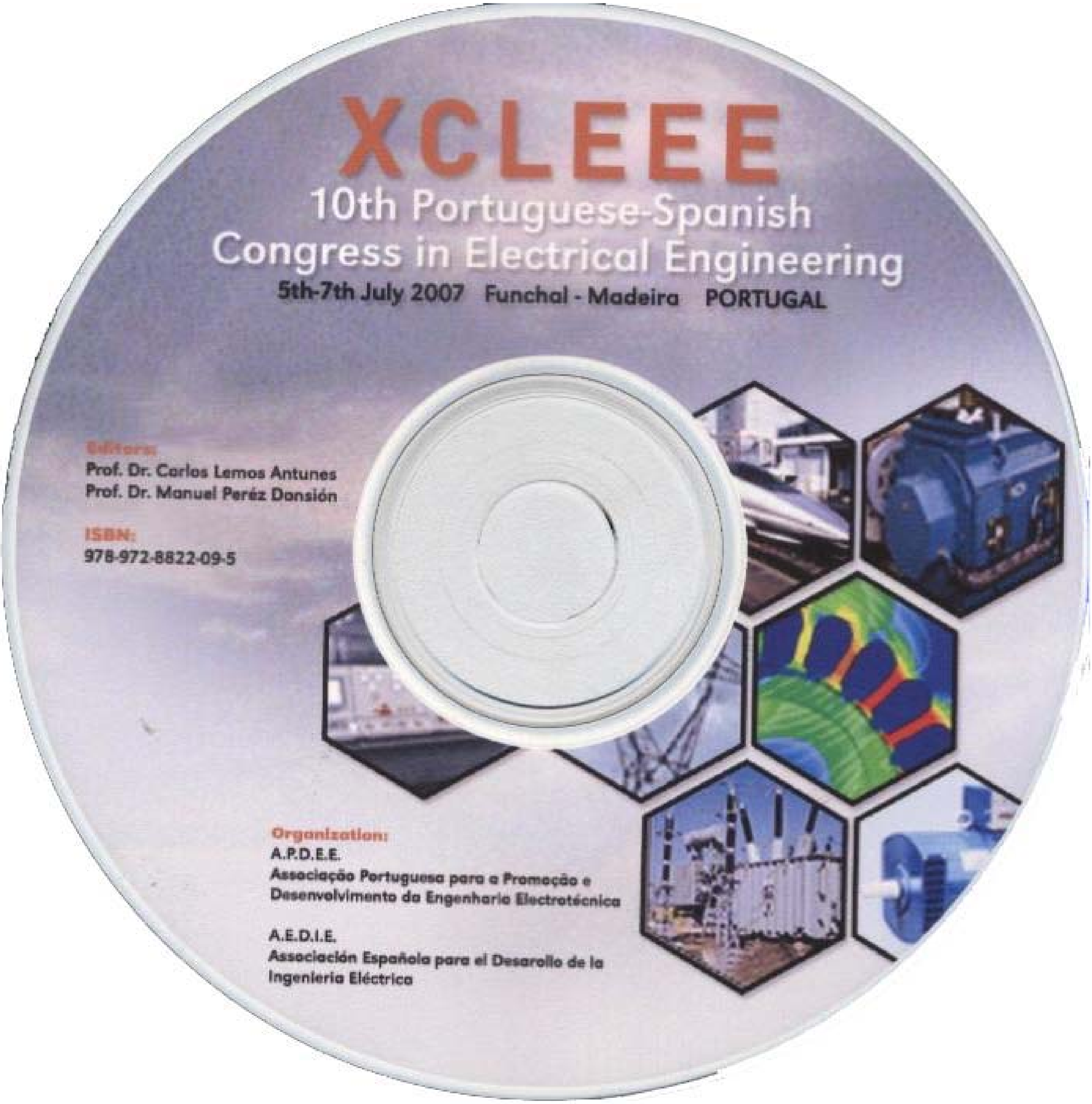
XCLEEE
10th Portuguese-Spanish
Congress in Electrical Engineering
5th-7th July 2007 Funchal - Madeira PORTUGAL
Editors:
Prof. Dr. Carlos Lemos Antunes
Prof. Dr. Manuel Peréz Donsión
ISBN:
978-972-8822-09-5
Organization:
A.P.D.E.E.
Associação Portuguesa para a Promoção e
Desenvolvimento da Engenharia Electrotécnica
A.E.D.I.E.
Associación Española para el Desarollo de la
Ingenieria Eléctrica

Application of Fractional Calculus in Genetic Algorithms, Transportation Systems and Robotics

J. A. T. Machado, Manuel Silva, Cecília Reis, Lino Figueiredo, Ramiro Barbosa, Isabel Jesus

Department of Electrotechnical Engineering, Institute of Engineering of Porto
Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal
E-mail: {jtm, mss, cmr, lbf, rsb, isj}@isep.ipp.pt

Abstract

This article illustrates several applications of fractional calculus (FC). This paper investigates the use of FC in circuit synthesis, traffic systems and robot control.

Keywords: Fractional Calculus, Swarm Optimization, Intelligent Transportation Systems, Robotics.

1. Introduction

In recent years fractional calculus (FC) has been a fruitful field of research in science and engineering. In fact, many scientific areas are currently paying attention to the FC concepts. This article presents novel results on the dynamics and control of several distinct systems. Bearing these ideas in mind, sections two to four present three case studies about the implementation of FC-based models and control algorithms.

2. Circuit synthesis using particle swarm optimization

The Particle Swarm Optimization (PSO) algorithm is a method inspired in the collective intelligence of swarms of biological populations, and was discovered through social model simulation of bird flocking, fishing schooling and swarm theory [1]. In this section we adopt a PSO algorithm to design combinational logic circuits. A truth table specifies the circuits and the goal is to implement a functional circuit with the least complexity. Four sets of logic gates have been defined, as Gset 2 ≡ {AND,XOR,WIRE}, Gset 3 ≡ {AND,OR, XOR,WIRE}, Gset 4 ≡ {AND,OR,XOR,NOT,WIRE}, Gset 6 ≡ {AND,OR,XOR,NOT, NAND,NOR,WIRE}. The logic gate named WIRE means a logical no-operation.

In the PSO scheme the circuits are encoded as a rectangular matrix **A** (*row* × *column* = $r \times c$) of logic cells. Three genes represent each cell: <*input1*><*input2*><*gate type*>, where *input1* and *input2* are one of the circuit inputs, if they are in the first column, or one of the previous outputs, if they are in other columns. The gate type is one of the elements adopted in the gate set. The chromosome is formed with as many triplets as the matrix size demands (*e.g.*, triplets = $3 \times r \times c$).

The initial population of circuits (particles) has a random generation and the initial velocity of each particle is initialized with zero. The velocities and positions of the following generations are obtained applying the PSO equations [1]. Therefore, the new positions are as many as the number of genes in the chromosome. If the new values of the input genes result out of range, then a re-insertion function is used. If the calculated gate gene is not allowed a new valid one is generated at random. The particles have memory and each one keeps information of its previous best position (*pbest*) and its corresponding fitness. The swarm has the *pbest* of all the particles and the particle with the greatest fitness is called the global best (*gbest*).

The basic concept of the PSO technique lies in accelerating each particle towards its *pbest* and *gbest* locations with a random weighted acceleration. However, in our case we also use a kind of mutation operator that introduces a new cell in 10% of the population. This mutation operator changes the characteristics of a given cell in the matrix. Therefore, the mutation modifies the gate type and the two inputs, meaning that a completely new cell can appear in the chromosome.

The calculation of the fitness function F_s in (2) has two parts, f_1 and f_2, where f_1 measures the functionality and f_2 measures the simplicity. In a first phase, we compare the output **Y** produced by the PSO-generated circuit with the required values $\mathbf{Y_R}$, according with the truth table, in a bit-per-bit basis. By other words, f_1 is incremented by one for each correct bit of the output until f_1 reaches the maximum value f_{10}, that occurs when we have a functional circuit. Once the circuit is functional, in a second phase, the algorithm tries to generate circuits with the least number of gates. Therefore, the index f_2, that measures the simplicity (the number of null operations), is increased by *one* (*zero*) for each *wire* (*gate*) of the generated circuit, yielding ($f_{10} = 2^{ni} \times no$):

$$f_1 = f_1 + 1 \text{ if } \{\text{bit } i \text{ of } \mathbf{Y}\} = \{\text{bit } i \text{ of } \mathbf{Y_R}\}\ ,\ i = 1, \ldots, f_{10} \tag{1a}$$

$$f_2 = f_2 + 1 \text{ if } \textit{gate type} = \textit{wire} \tag{1b}$$

$$F_s = \begin{cases} f_1, & F_s < f_{10} \\ f_1 + f_2, & F_s \geq f_{10} \end{cases} \tag{1c}$$

where *ni* and *no* represent the number of inputs and outputs of the circuit.

The concept of dynamic fitness function F_d results from an analogy between control systems and the GA case, where we master the population through the fitness function. The simplest control system is the proportional algorithm; nevertheless, other control algorithms can be adopted, such as, for example, the proportional and the differential scheme. In this line of thought expression (1c) is a static fitness function F_s and corresponds to using a simple proportional algorithm. Therefore, to implement a proportional-derivative evolution the fitness function needs a scheme of the type:

$$F_d = F_s + KD^{\alpha}\left[F_s\right] \tag{2}$$

where $0 \leq \alpha \leq 1$ is the differential fractional-order and $K \in \Re$ is the weight of the dynamical term. The fractional derivative is calculated through a discrete-time 4$^{\text{th}}$-order Padé fraction approximation of Euler transformation.

In this study are developed n = 20 simulations for each case under analysis. The experiments consist on running the PSO algorithm to generate a typical combinational logic circuit, namely a 2-to-1 multiplexer (*M2-1*), a 1-bit full adder (*FA1*), a 4-bit parity checker (*PC4*) and a 2-bit multiplier (*MUL2*).

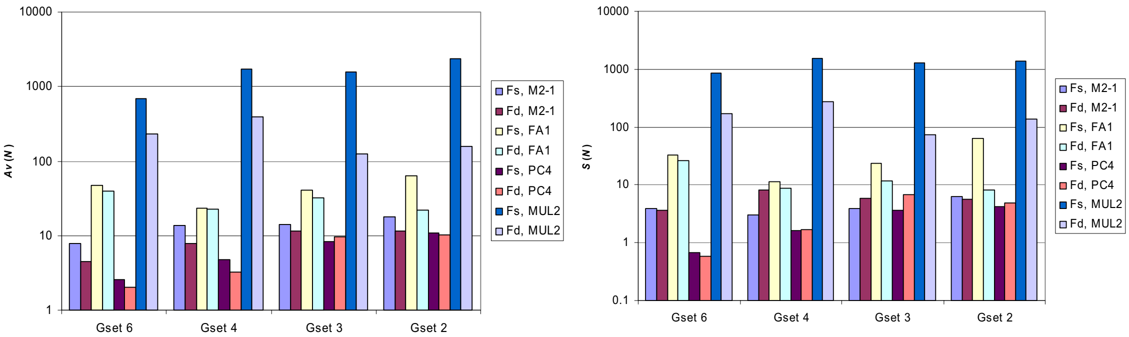

Fig. 1 – Average $Av(N)$ and standard deviation $S(N)$ of the number of generations N to achieve the solution for the PSO algorithm, $P = 3000$ using F_s and F_d

The circuits are generated with a population $P = 3000$. Figure 1 presents a comparison between F_s and F_d. Applying the F_d concept the results obtained are improved in all gate sets and in particular for the more complex circuits.

3. Simulation and dynamical analysis of freeway traffic systems

In order to study the dynamics of traffic systems it was developed the Simulator of Intelligent Transportation Systems (SITS). SITS is a software tool based on a microscopic simulation approach, which reproduces real traffic conditions in an urban or non-urban network. The program provides a detailed modelling of the traffic network, distinguishing between different types of vehicles and drivers and considering a wide range of network geometries. SITS uses a flexible structure that allows the integration of simulation facilities for any of the *ITS* related areas.

A set of simulation experiments are developed in order to estimate the influence of the vehicle speed $v(t;x)$, the road length l and the number of lanes n_l in the traffic flow $\phi(t;x)$ at time t and road coordinate x. For a road with n_l lanes the Transfer Function (TF) between the flow measured by two sensors is calculated by the expression $G_{r,k}(s; x_j,x_i) = \Phi_r(s;x_j)/\Phi_k(s;x_i)$ where k, $r = 1,2,\ldots, n_l$ define the lane number and, x_i and x_j represent the road coordinates ($0 \le x_i \le x_j \le l$), respectively.

The first group of experiments considers a one-lane road (*i.e.*, $k = r = 1$) with length $l = 1000$ m. Across the road are placed n_s sensors equally spaced. The first sensor is placed at the beginning of the road (*i.e.*, at $x_i = 0$) and the last sensor at the end (*i.e.*, at $x_j = l$). Therefore, we calculate the TF between two traffic flows at the beginning and the end of the road such that, $\phi_1(t;0) \in [1, 8]$ vehicles s^{-1} for a vehicle speed $v_1(t;0) \in [30, 70]$ km h^{-1}, that is, for $v_1(t;0) \in [v_{av} - \Delta v, v_{av} + \Delta v]$, where $v_{av} = 50$ km h^{-1} is the average vehicle speed and $\Delta v = 20$ km h^{-1} is the maximum speed variation. These values are generated according to a uniform probability distribution function.

The results obtained of the polar plot for the TF $G_{1,1}(s;1000,0) = \Phi_1(s;1000)/\Phi_1(s;0)$ between the traffic flow at the beginning and end of the one-lane road is distinct from those usual in systems theory revealing a large variability. Moreover, due to the stochastic nature of the phenomena involved different experiments using the same input range parameters result in different TFs. In fact traffic flow is a complex system but it was shown [4] that, by

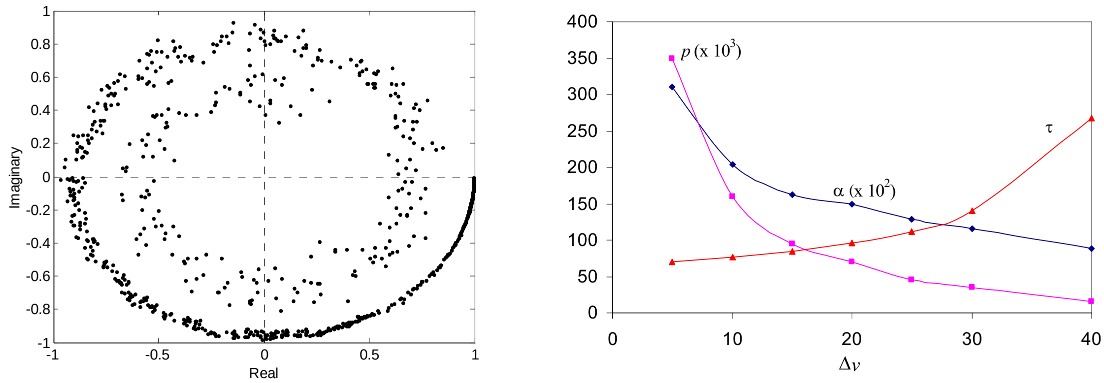

Fig. 2 – The STF $T_{1,1}(s;1000,0)$ for $n = 2000$ experiments with $\phi_1(t;0) \in [1, 8]$ vehicles s^{-1} and $v_1(t;0) \in [30, 70]$ km h^{-1}($v_{av} = 50$ km h^{-1}, $\Delta v = 20$ km h^{-1}, $l = 1000$ m and $n_l = 1$) (left) and Time delay τ, pole p and fractional order α *versus* Δv for an average vehicle speed $v_{av} = 50$ km h^{-1}, $n_l = 1$, $l = 1000$ m and $\phi_1(t;0) \in [1, 8]$ vehicles s^{-1} (right).

embedding statistics and Fourier transform (leading to the concept of Statistical Transfer Function (STF)), we could analyse the system dynamics in the perspective of systems theory.

To illustrate the proposed modelling concept (STF), the simulation was repeated for a sample of $n = 2000$ and it was observed the existence of a convergence of the STF, $T_{1,1}(s;1000,0)$, as show in Fig. 2 (left), for a one-lane road with length l = 1000 m $\phi_1(t;0) \in [1, 8]$ vehicles s^{-1} and $v_1(t;0) \in [30, 70]$ km h^{-1}.

Based on this result we can approximate numerically the STF to a fractional order system with time delay yielding the approximate expression:

$$T_{1,1}(s;1000,0) = \left(k_B \; e^{-\tau s}\right)\left(s/p+1\right)^{-\alpha} \tag{3}$$

For the numerical parameters of Fig. 2 (left) we get $k_B = 1.0$, $\tau = 96.0$ sec, p = 0.07 and $\alpha = 1.5$.

The parameters (τ, p, α) vary with the average speed v_{av} and its range of variation Δv, the road length l and the input vehicle flow ϕ_1. For example, Fig. 2 (right) shows (τ, p, α) versus Δv for $v_{av} = 50$ km h^{-1}.

It is interesting to note that $(\tau, p) \to (\infty, 0)$, when $\Delta v \to v_{av}$, and $(\tau, p) \to (l\, v_{av}^{-1}, \infty)$, when $\Delta v \to 0$. These results are consistent with our experience that suggests a pure transport delay $T(s) \approx e^{-\tau s}$ ($\tau = l\, v_{av}^{-1}$), $\Delta v \to 0$ and $T(s) \approx 0$, when $\Delta v \to v_{av}$ (because of the existence of a blocking cars, with zero speed, on the road).

4. Fractional PD$^{\alpha}$ control of an hexapod robot

Walking machines allow locomotion in terrain inaccessible to other type of vehicles, since they do not need a continuous support surface. For these robots, the control of the leg joints is usually implemented through a PID like scheme with position feedback. Recently, the application of the theory of FC to robotics revealed promising aspects for future developments. With these facts in mind, this section compares different Fractional Order (FO) robot controller tuning, applied to the joint control of a walking system.

The robot model has $n = 6$ legs, equally distributed along both sides of the robot body, having each three rotational joints (*i.e.*, j = {1, 2, 3} ≡ {hip, knee, ankle}) [3]. It is considered robot body compliance because walking animals have a spine that allows supporting the locomotion with improved stability. The robot body is divided in n identical segments and a linear spring-damper system is adopted to implement the intra-body compliance [3]. The contact of the robot feet with the ground is modeled through a non-linear system [3], being the values for the parameters based on the studies of soil mechanics [4].

The general control architecture of the hexapod robot is presented in Fig. 3 (left) [4]. In this study we evaluate the effect of different PD$^{\alpha}$, $\alpha \in \Re$, controller implementations for $G_{c1}(s)$, while G_{c2} is a P controller. For the PD$^{\alpha}$ algorithm, implemented through a discrete-time 4$^{\text{th}}$-order Padé approximation (a_{ij}, $b_{ij} \in \Re, j = 1, 2, 3$), we have:

$$G_{c1j}\left(z\right) \approx Kp_j + K\alpha_j \sum_{i=0}^{i=u} a_{ij}\, z^{-i} \Bigg/ \sum_{i=0}^{i=u} b_{ij}\, z^{-i} \tag{4}$$

where K_{pj} and $K_{\alpha j}$ are the proportional and derivative gains, respectively, and α_j is the fractional order, for joint j.

It is analyzed the system performance of the different PD$^{\alpha}$ tuning, during a periodic wave gait at a constant forward velocity. The analysis is based on the formulation of two indices measuring the mean absolute density of energy per traveled distance (E_{av}) and the hip trajectory errors (ε_{xyH}) during walking, according to:

$$E_{av} = \frac{1}{d} \sum_{i=1}^{n} \sum_{j=1}^{m} \int_0^T \left| \boldsymbol{\tau}_{ij}\left(t\right) \dot{\boldsymbol{\theta}}_{ij}\left(t\right) \right| dt \quad \left[\mathrm{Jm}^{-1}\right] \tag{5a}$$

$$\begin{gathered} \varepsilon_{xyH} = \sum_{i=1}^{n} \sqrt{\frac{1}{N_s} \sum_{k=1}^{N_s} \left({\Delta_{ixH}}^2 + {\Delta_{iyH}}^2 \right)} \quad \left[\mathrm{m}\right] \\ \Delta_{ixH} = x_{iHd}(k) - x_{iH}(k), \Delta_{iyH} = y_{iHd}(k) - y_{iH}(k) \end{gathered} \tag{5b}$$

To tune the different controller implementations we adopt a systematic method, testing and evaluating several possible combinations of parameters, for all controller implementations. Therefore, we adopt the $G_{c1}(s)$ parameters that establish a compromise in what concerns the simultaneous minimisation of E_{av} and ε_{xyH}, and a proportional controller G_{c2} with gain $Kp_j = 0.9$ ($j = 1, 2, 3$). Moreover, it is assumed high performance joint actuators, with a maximum actuator torque of $\tau_{ijMax} = 400$ Nm, and the desired angle between the foot and the ground (assumed horizontal) is made $\theta_{i3hd} = -15^{\circ}$. We tune the PD$^{\alpha}$ joint controllers for different values of the fractional order α_j while making $\alpha_1 = \alpha_2 = \alpha_3$.

Figure 3 (right) presents the best controller tuning for different values of α_j. The experiments reveal the superior performance of the PD$^{\alpha}$ controller for $\alpha_j \approx 0.5$, with $K_{p1} = 15000$, $K_{\alpha1} = 7200$, $K_{p2} = 1000$, $K_{\alpha2} = 800$ and $K_{p3} = 150$, $K_{\alpha3} = 240$.

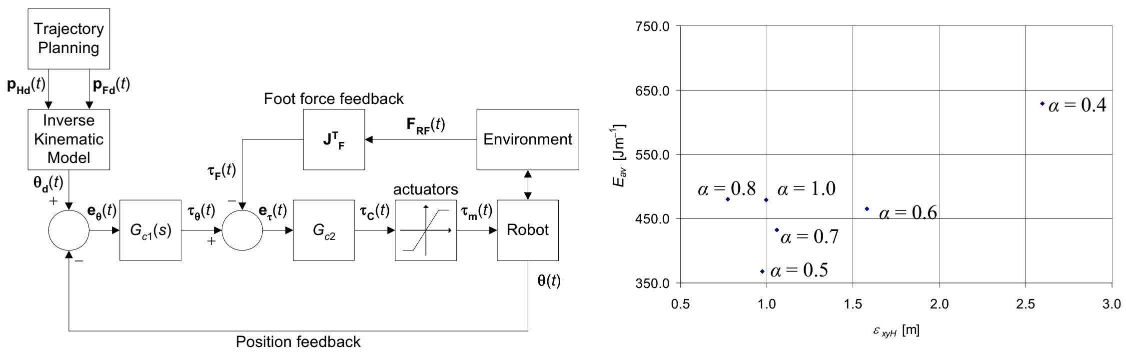

Fig. 4. Hexapod robot control architecture (left) and locus of E_{av} *vs.* ε_{xyH} for the different values of α in the $G_{c1}(s)$ tuning, when establishing a compromise between the minimisation of E_{av} and ε_{xyH}, with $G_{c2} = 0.9$ (right).

4. Conclusions

Recently FC has been a fruitful field of research in science and engineering and many scientific areas are currently paying wider attention to the FC concepts. This article presented several case studies on the implementation of FC-based models and control systems, namely in circuit synthesis, intelligent transportation systems and legged robot control.

Acknowledgments

The authors would like to thank GECAD – *Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão*, of Instituto Superior de Engenharia do Porto – Instituto Politécnico do Porto, for their financial support to this work.

References

[1] C. Reis, J. A. T. Machado, J. B. Cunha, *Fractional-Order Evolutionary Design of Digital Circuits*, FDA06 – 2nd IFAC Workshop on Fractional Differentiation and its Applications, pp. 445 – 450, July 2006, Porto, Portugal.

[2] L. Figueiredo, J. Machado, J. Ferreira, *Dynamical Analysis of Freeway Traffic*, IEEE Transactions on Intelligent Transportation Systems, 2004, Vol. 5, n. 4, ISSN: 1524-9050, pp. 259 – 266.

[3] M. F. Silva, J. A. T. Machado, I. S. Jesus, *Modelling and Simulation of Walking Robots With 3 dof Legs*, MIC 2006 – The 25th IASTED International Conference on Modelling, Identification and Control, pp. 271 – 276, February 2006, Lanzarote, Spain.

[4] M. F. Silva, J. A. T. Machado, A. M. Lopes, *Position / Force Control of a Walking Robot*, MIROC – Machine Intelligence and Robot Control, 2003, Vol. 5, n. 2, pp. 33 – 44.