

PSO to Design Decentralized Fuzzy PI Controllers Application for a Helicopter

H. Boubertakh, S. Labiod, and M. Tadjine

Abstract— In this paper, we propose the use of a Particle Swarm Optimization (PSO) for tuning decentralized typical fuzzy PI (FPI) controllers applied for the stabilization of a helicopter model. The method is used such as to minimize a square error (SE) fitness function which quantifies the performance of the whole control system. The proposed PSO method exploits all the available knowledge about the system under control since, the considered typical FPI controllers are characterized by simple and interpretable structure, which can reduce the tuning time.

I. INTRODUCTION

IN fuzzy control literature, fuzzy PID controllers (FPID) are presented as an alternative to overcome the lack of the classical PID controllers when used with nonlinear systems and/or complex systems, in particular interconnected systems [1-7]. However, the challenge to take up is how to design these controllers considering the big number of parameters to be tuned compared to the classical PID controllers which have only three parameters that can be tuned easily by using the available rules in literature [8].

Nowadays, the heuristic optimization has been gained a special interest to solve the FPID controllers design problem. Among these methods we can find, genetic algorithms (GA) [9-12], ant colony optimization (ACO) [13] and particle swarm optimization (PSO) [12], [14].

The above listed methods are used for single input and single output (SISO) systems. In this paper, we propose a PSO tuning method to design decentralized typical fuzzy PI (FPI) controllers applied for the stabilization of a helicopter model, which is essentially nonlinear, with two inputs and two outputs (TITO), and open loop instable. The performance of the proposed method is evaluated by a fitness function associated with the whole control system performance. Without loss of generality, we use a square error fitness function (SE) to quantify the quality of two step responses of closed-loop system.

We consider in this paper, the typical FPI controllers presented in [1]. This type of FPI can be linear or nonlinear depending on the membership functions positions. The linear FPI controller is obtained by modal positions of the

membership functions. However, a nonlinear behavior can be reached by handling these quantities from their modal positions or to use some existing optimization methods. Thus, these typical FPI controllers can deal with complex and nonlinear processes and combine the advantages of classical and fuzzy control namely; simplicity, nonlinearity and interpretability. Moreover, the tuning time will be reduced by the exploiting the available knowledge and the human operator experience with the plant under control.

The remainder of this paper is organized as follows. Section II presents the design settings of typical FPI controllers. Section III presents the helicopter model. Section IV gives the decentralized control structure. Section V gives a brief overview the PSO and presents the proposed tuning method. The simulation results are given in section VI. Section VII concludes de the paper.

II. THE FPI CONTROLLERS

The incremental FPI control law is given by

$$u_{FPI}(k) = u_{FPI}(k-1) + \Delta u_{FPI}(k) \quad (1)$$

$$\Delta u_{FPI}(k) = f(e(k), \Delta e(k)) \quad (2)$$

The error e and the change of error Δe are defined as

$$e(k) = y_r(k) - y(k) \quad (3)$$

$$\Delta e(k) = e(k) - e(k-1) \quad (4)$$

Where f is the function of the fuzzy inference system, u_{FPI} is the control signal, Δu_{FPI} is the change of error y_r is the reference signal and y is the controlled output.

According to the typical structural design of FPI controllers in [1], we use FPI controllers with 3 membership functions for input variables and 5 singletons for the output variable as shown on figure 1.

Now, unless otherwise specified, we consider that the input membership functions and output singletons are evenly distributed on symmetrical universes of discourse (i.e. $U_2 = 2U_1 = U$). Moreover, and to avoid saturation we consider that the universes of discourse are large enough such that all possible inputs are always within their limits.

The rule base of the controllers is donated by table 1.

The crisp change of control action is determined by the average sum:

$$\Delta u_{FPI}(k) = \frac{\sum_{i=1}^9 V_i C_i}{\sum_{i=1}^9 V_i} \quad (5)$$

where C_i is the conclusion of rule i and V_i is its truth value calculated by the algebraic product method given by

$$V_i = \mu_{e_i}(e(k)) \cdot \mu_{\Delta e_i}(\Delta e(k)) \quad (6)$$

Manuscript received January 26, 2012.

H. Boubertakh is with LAJ-Laboratoire d'Automatique de Jijel, FST, University of Jijel, BP. 98, Ouled Aissa, 18000, Jijel, Algeria (email : boubert_hamid@yahoo.com)

S. Labiod is with LAJ, FST, University of Jijel, BP. 98, Ouled Aissa, 18000, Jijel, Algeria (email: labiod_salim@yahoo.fr)

M. Tadjine is with LCP, Ecole Nationale Polytechnique, 10 av. Hassen Badi, BP.182, El Harrach, Algiers, Algeria. (email : tadjine@yahoo.fr)

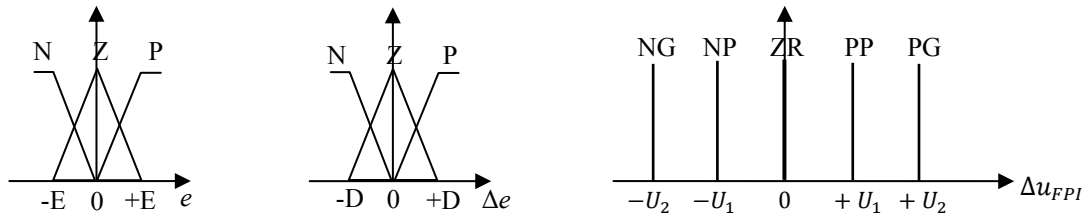


Fig. 1. Membership functions of the FPI controllers

$\mu_{y_i}(y)$ is the membership grade of the input variable y evaluated in rule i by the corresponding membership function.

TABLE I

FPI CONTROLLERS' RULE BASES			
e	Δe		
	N	Z	P
N	$C_1=NB$	$C_2=NS$	$C_3=ZR$
Z	$C_4=NS$	$C_5=ZR$	$C_6=PS$
P	$C_7=ZR$	$C_8=PS$	$C_9=PB$

The labels NB, NS, ZR, PS, and PB refer respectively to the linguistic terms; Negative Big, Negative Small, around Zero, Positive Small and Positive Big.

The inferred output is calculated by (5) and (6). After calculation [1], we find that:

$$\Delta u_{FPI}(k) = \frac{u}{2E}e(k) + \frac{u}{2D}\Delta e(k) \quad (7)$$

Equation (7) shows clearly that the typical FPI controllers are identical to the classical PI controllers with the following proportional and integral gains: $K_p = \frac{u}{2D}$ and $K_I = \frac{u}{2E}$.

We note here that a nonlinear behavior of the above designed FPI controllers can be reached by moving the input membership functions and/or the singletons from their modal positions.

We can conclude that the FPI controllers presented here combine the advantages of classical PI controllers and fuzzy controllers; the simplicity, the physical interpretability, and the possible nonlinearity of their structures.

III. HELICOPTER MODEL

The CE150 Helicopter Model of Humusoft Ltd [15] consists of a body carrying two DC motors. These motors drive the propellers. The body has two degrees of freedom. The axes of the body rotation are perpendicular as well as the axes of the motors. Both body position angles, i.e. azimuth angle Ψ in horizontal and elevation angle Φ in vertical plane are influenced by the rotating propellers simultaneously. The DC motors for driving propellers are controlled proportionally to the output signal of the computer. The helicopter model is a MIMO dynamical system with two manipulated inputs u_1 and u_2 , and two measured outputs (Φ, Ψ). All inputs and outputs are coupled. The user of the simulator communicates with the

system via the data-processing interface, the entries (u_1 and u_2) and outputs are scaled in the interval $[-1, +1]$, where "1" is called *Machine Unit* [MU].

The mathematical model of the helicopter is given by the following differential equations system:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 0.8764x_2 \sin x_1 + 3.4325x_4 u_1 \cos x_1 + 0.4211 x_2 \\ &\quad - 0.0035x_5^2 - 46.35x_6^2 - 0.8076x_5x_6 \\ &\quad - 0.0259x_5 - 2.9749x_6 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= 21.4010x_4 - 31.8841x_8^2 - 14.2029x_8 \\ &\quad - 21.7150x_9 + 1.4010u_1 \\ \dot{x}_5 &= -6.6667x_5 - 2.7778x_6 + 2u_1 \\ \dot{x}_6 &= 4x_5 \\ \dot{x}_7 &= -8x_7 - 4x_8 + 2u_2 \\ \dot{x}_8 &= 4x_7 \\ \dot{x}_9 &= -1.3333x_9 + 0.0625u_1 \end{aligned}$$

where: $x_1 = \Psi$, $x_2 = \frac{d\Psi}{dt}$, $x_3 = \Phi$, $x_4 = \frac{d\Phi}{dt}$, and x_5 to x_9 are state variables representing the two DC motors and the coupling effects.

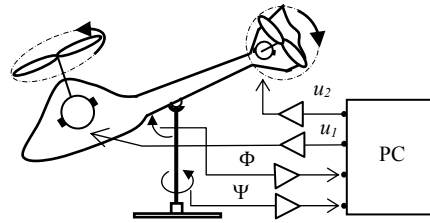


Fig. 2. Helicopter Configuration

IV. CONTROL STRUCTURE

The helicopter simulator can be seen like an interconnection of two subsystems; the elevation subsystem characterized by the input u_1 and the output Ψ and the azimuth subsystem characterized by the input u_2 and the output Φ .

The problem is to stabilize the system around a setpoint (Ψ_r, Φ_r). For that we use decentralized control method; the two subsystems are controlled by two FPI controllers with structure design presented in section III. The control structure is presented on figure 3.

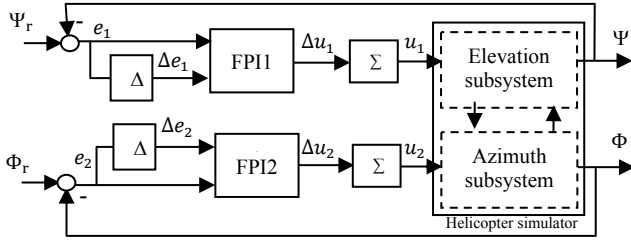


Fig. 3. Control structure

V. TUNING THE FPI CONTROLLERS USING PSO

A. Overview of the particle swarm optimization

PSO is nature-inspired heuristic optimization method which first proposed by Kennedy and Eberhart [16]. It belongs to the category of Swarm Intelligence methods. Its development was based on mimicking the movement of individuals within a swarm (i.e., fishes, birds, and insects) in an effort to find the optima in the problem space. It has been noticed that members of the swarm seem to share information among them. This communication fact leads to increase efficiency of the swarm. The PSO algorithm searches in parallel using a group of individuals similar to other population-based heuristic optimization techniques. PSO technique conducts search using a population of particles, corresponding to individuals. Each particle represents a candidate solution to the problem at hand. In a PSO system, particles change their positions by “flying” around in a multidimensional search space. Particle in a swarm adjust its position in search space using its present velocity, own previous experience, and that of neighboring particles. Therefore, a particle makes use of best position encountered by itself and that of its neighbors to steer toward an optimal solution. The performance of each particle is measured using a predefined fitness function, which quantifies the performance of the optimization problem.

The mathematical expressions for velocity and position updates are given by (9) and (10) respectively.

$$v_{ij}^{k+1} = wv_{ij}^k + c_1r_1(pb_{est_{ij}} - v_{ij}^k) + c_2r_2(gb_{est_j} - v_{ij}^k) \quad (9)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^k \quad (10)$$

Where: $1 \leq i \leq I$, $1 \leq j \leq J$, $1 \leq k \leq K$. I, J and K are respectively the number of particles in the swarm, the dimension of particle, and the maximum iterations.

x_{ij}^k and v_{ij}^k are respectively the position the velocity of particle i in the dimension j at iteration k , $pb_{est_{ij}}$ is a personal best of particle i in the dimension j , gb_{est_j} is a global best of all particles in the dimension j , w is inertia weight factor, c_1 and c_2 are acceleration constants, r_1 and r_2 are random numbers in interval $[0, 1]$.

B. The FPI Controllers tuning problem using PSO

In the proposed PSO tuning method for decentralized FPI controllers, the tuning vector is composed by the parameters of the two controllers.

The components of the tuning parameters vector can be defined according to the available knowledge and the control objective. In the aim to get nonlinear controllers, we consider the tuning parameters' vector composed by the positions of the membership functions over the input spaces and the positions of singletons over the output spaces for both the two controllers.

To simplify, we consider a symmetrical fuzzy PI controller such as the positive and negative parameters absolute values are equal. Thus, the parameter's vector is

$$P = [E_1 D_1 U_{11} U_{12} E_2 D_2 U_{21} U_{22}]^T \quad (10)$$

In order to implement the proposed method, we need the minimum and the maximum bounds for each tuning parameter. The initial parameters values are randomly distributed between these bounds. A best choice of these bounds can drastically reduce the tuning time. This is possible, since the FPI controllers have interpretable parameters, and can incorporate human operator knowledge about the system under control.

A squared error fitness function (SE) (11) is used to quantify the effectiveness of a given FPI controller; it is evaluated at the end of a step-response of the closed-loop system under control.

$$SE = \frac{1}{NT - N_1T} \sum_{k=N_0}^N e_1(k)^2 + \frac{1}{NT - N_2T} \sum_{k=N_2}^N e_2(k)^2 \quad (11)$$

N is the total number of samples, N_0 is the number of transient samples system output.

The fitness function SE defined in (11) takes into account the error over the time interval $[N_0T, NT]$. By setting $t_0 = N_0T$ to approximate the peak time of the system at which the step response of the closed-loop system reaches the corresponding first peak.

In the FPI controllers tuning problem using PSO, a particle is one value of parameters vector. A particle swarm is group of parameters vector values which are randomly initialized in search space defined by limits of the components of the tuning parameters vector. So, a the particle position x is the FPI parameter value vector, pb_{est} the best parameters values vector regarding the fitness function, gb_{est} is the global best parameters values vector of all particles.

VI. SIMULATION RESULTS

The proposed method is used to tune simultaneously the two decentralized FPI controller such as to minimize square error sum cost function (11) at the end of step-responses of the two subsystems of the helicopter model. In the simulation, we take the following parameters: the sample time $T = 0.1s$, the peak-time for the two subsystems $t_{peak} = N_1T = N_2T = 10s$, and one step episode time is 50s.

The premise parameters bounds are given by table 2 and the initial rule base for the two FPI controllers is given by table 3.

TABLE 2
INITIAL PREMISES PARAMETERS BOUNDS

Parameter	Min	Max
E1	0	1
D1	0	0.1
E2	0	1
D2	0	0.1

TABLE 3
THE INITIAL RULE BASES

e	Δe		
	N	Z	P
N	[-0.1, -0.01]	[-0.01, 0.0]	[0.0, 0.0]
Z	[-0.01, 0.0]	[0.0, 0.0]	[0.0, 0.01]
P	[0.0, 0.0]	[0.0, 0.01]	[0.01, 0.1]

The figure 4 shows the fitness function evolution with iterations. Figures 5 to 12 show respectively the evolution with iterations of parameters E1, D1, U11, U12, E2, D2, U21 and U22. The figures 13 and 14 show respectively the elevation angle and azimuth angle using the tuned parameters values. The figures 15 and 16 show respectively the control signal for the elevation subsystem and the control signal for the azimuth subsystem using the tuned parameters values. The tuned premise parameters values are given by table 4. The tuned fuzzy rule bases for the two FPI controllers are presented on tables 5 and 6.

TABLE 4
THE FINAL PREMISE TUNED PARAMETERS VALUES

Parameter	Final tuned value
E1	0.4727
D1	0.0431
E2	0.8711
D2	0.0197

TABLE 5
THE FINAL RULE BASE OF FPI 1

e	Δe		
	N	Z	P
N	-0.0881	-0.0120	0.0000
Z	-0.0120	0.0000	+0.0120
P	0.0000	+0.0120	+0.0881

TABLE 6
THE FINAL RULE BASE OF FPI 2

e	Δe		
	N	Z	P
N	-0.0238	-0.0029	0.0000
Z	-0.0029	0.0000	+0.0029
P	0.0000	+0.0029	+0.0238

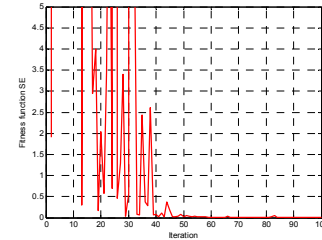


Fig. 4. Fitness function

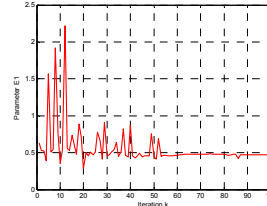


Fig. 5. Parameter E1

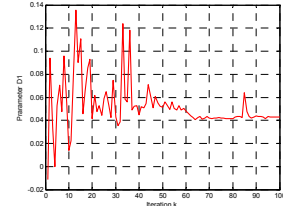


Fig. 6. Parameter D1

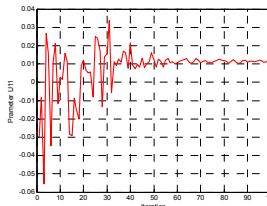


Fig. 7. Parameter U11

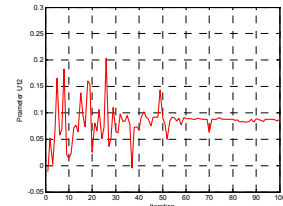


Fig. 8. Parameter U12

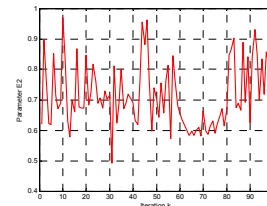


Fig. 9. Parameter E2

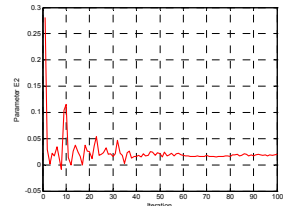


Fig. 10. Parameter D2

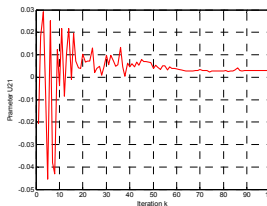


Fig. 11. Parameter U21

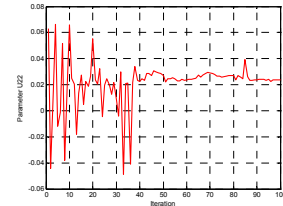


Fig. 12. Parameter U22

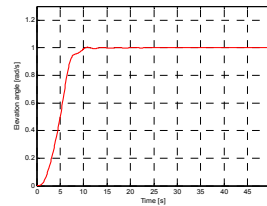


Fig. 13. Elevation angle

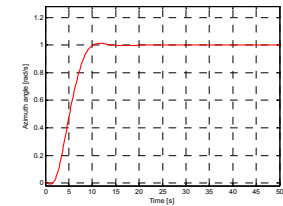


Fig. 14. Azimuth angle

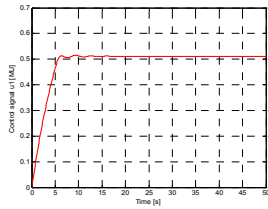


Fig. 15. Control signal u_1

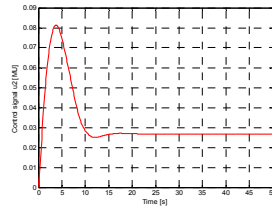


Fig. 16. Control signal u_2

VII. CONCLUSION

In this paper, we proposed a PSO-based tuning method for typical FPI controllers in a decentralized control scheme. The method exploits all the available knowledge about the system under control. The available knowledge is incorporated in the FPI structure and is used for a good choice of the parameters space search by defining the limits of tuning parameters which reduce drastically the tuning period, which is the draw-back of the most used heuristic methods. The proposed method is used for control helicopter by two decentralized FPI controllers. The efficiency of the proposed method is illustrated through simulation results.

REFERENCES

- [1] H. Boubertakh, M. Tadjine, P.Y. Glorennec and S. Labiod, "Tuning fuzzy PD and PI controllers using reinforcement learning", *ISA Transactions*, Vol.49, No. 4, pp. 543-551, 2010.
- [2] Y. Ding, H. Ying, and S. Shao, "Typical Takagi-Sugeno PI and PD fuzzy controllers: analytical structures and stability analysis," *Fuzzy Sets and Systems*, vol.151, pp. 245-262, 2003.
- [3] B. M. Mohan, and A.V. Patel, "Analytical Structures and Analysis of the Simplest Fuzzy PD Controllers," *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Cybernetics, Vol. 32, pp.239-248, 2002.
- [4] B. M. Mohan, and A. Sinha, "Analytical structure and stability analysis of a fuzzy PID controller," *Applied Soft Computing*, Vol. 8, pp.749-758, 2008.
- [5] B. S. Moon, "Equivalence between fuzzy logic controllers and PI controllers for single input systems," *Fuzzy Sets and Systems*, Vol. 69, pp. 105-113, 1995.
- [6] M. Muzimoto, "Realization of PID controls by fuzzy control methods," *Fuzzy Sets and Systems*, Vo. 105, pp. 171-182, 1995.
- [7] R. R. Yager, and D. P. Filev, *Essential of fuzzy modeling and control*. John Wiley & Sons Inc, 1995.
- [8] K. J. Åström, and T. Hägglund, *PID Controllers, Theory, Design and Tuning*, Instrument Society of America, Research Triangle Park, North Carolina, 1995.
- [9] J. M. Adams, and K. S. Rattan, "A genetic multi-stage fuzzy PID controller with a fuzzy switch," *In Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, Tucson, AZ, USA, 4, 2001, pp. 2239-2244.

- [10] C.-N. Ko, T.-L. Lee, H.-T. Fan, and C.-J. Wu, "Genetic Auto-Tuning and Rule Reduction of Fuzzy PID Controllers," *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, Taipei, Taiwan, 2006, pp. 1096-1101.
- [11] K. S. Tang, K. M. Man, and G. Chen, "A G-A optimized Fuzzy PD+I Controller for Nonlinear Systems," *Proc. of Int. Conf. on Industrial Electronics, Control, and Instrumentation (IECON'01)*, Denver, USA, 2006, pp. 718-723.
- [12] I. Pan, S. Das, and A. Gupta, "Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay", *ISA Transactions*, Vol. 50, No 1, pp. 28-36, 2011.
- [13] H. Boubertakh, M. Tadjine, P.-Y. Glorennec, S. Labiod, "Tuning fuzzy PID controllers using ant colony optimization". In: *Proc. of the 17th Mediterranean Conference on Control and Automation (MED'09)*, Thessaloniki, Greece, 2009, pp. 13-18.
- [14] C-N KO, and C-J WU, "A PSO-Tuning Method for Design of Fuzzy PID Controllers", vol. 14, No 3, pp. 375-395, 2008.
- [15] Humusoft, CE 150 helicopter model: User's manual, Humusoft, Prague (2002).
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc IEEE Int Conf on Neural Networks*, pp. 1942-1948, 1995.