International Conference on Control, Engineering & Information Technology (CEIT'14)
Proceedings - Copyright IPCO-2014, pp.329-334
ISSN 2356-5608

Differential Search Algorithm-based approach for PID-type Fuzzy Controller tuning

F. Toumi, S. Bouallègue, J. Haggège
Research Laboratory in Automatic Control LA.R.A, National Engineering School of Tunis, BP 37, Le Belvédère, 1002 Tunis, Tunisia.
fatma_toumi_enit@yahoo.fr, soufiene.bouallegue@issig.rnu.tn, joseph.haggege@enit.rnu.tn

F. Toumi, P. Siarry
Signals, Images and Intelligent Systems Laboratory, LiSSi-EA-3956, University Paris-Est Créteil, 61 av. du Général de Gaulle, 94010 CRETEIL Cedex, Bâtiment P2, Bureau 239-B, France.
fatma_toumi_enit@yahoo.fr, siarry@univ-paris12.fr.

***Abstract*—In this paper, a new improved Differential Search Algorithm optimization approach is proposed and successfully applied to the design and tuning of a PID-type Fuzzy Logic Controller. The scaling factors tuning problem of the Fuzzy Logic Controller structure is formulated and systematically resolved, using the improved constrained Differential Search Algorithm-based method. A comparison, with the standard Genetic Algorithm Optimization and Particle Swarm Optimization meta-heuristics, is investigated in order to show the superiority and the effectiveness of the developed Differential Search Algorithm-based method. Simulation results, for an electrical DC drive benchmark, show the advantages of the proposed Differential Search Algorithm-tuned PID-type Fuzzy control structure in terms of performance and robustness.**

Keywords*—Differential Search Algorithm; PID-type fuzzy control; scaling factors tuning; Particle Swarm Optimization; Genetic Algorithm; DC drive benchmark

I. INTRODUCTION

The Fuzzy logic control approach has been widely used in many successful industrial applications. This control strategy, with the Mamdani fuzzy type inference, demonstrated high robustness and effectiveness properties [10,11,13]. The known PID-type Fuzzy Logic Controller (FLC) structure, firstly proposed in [14], is especially established and improved within the practical framework [5,7,15]. This particular fuzzy controller retains characteristics similar to the conventional PID controller and can be decomposed into the equivalent proportional, integral and derivative control components [14]. In this design case, the dynamic behaviour depends on the adequate choice of the fuzzy controller scaling factors. The tuning procedure depends on the control experience and knowledge of the human operator, and it is generally achieved based on a classical trials-errors procedure. There isn't, until now, a systematic method to guide such a choice. This tuning problem becomes harder and more delicate as the complexity of the controlled plant increases.

In order to further improve the performance of the transient and steady state responses of the PID-type fuzzy structure, various strategies and methods are proposed to tune theirs parameters. In [14], Qiao and Mizumoto proposed a peak observer mechanism-based method to adjust the PID-type FLC parameters. This self-tuning mechanism decreases the equivalent integral control component of the fuzzy controller gradually with the system response process time. On the other hand, Woo et al. [15] developed a method based-on two empirical functions evolved with the system's error information. In [7], the authors proposed a technique that adjusts the scaling factors, corresponding to the derivative and integral components, using a fuzzy inference mechanism. However, the major drawback of all these PID-type FLC tuning method is the difficult choice of their relative parameters and mechanisms.

In this paper, a new approach based on the Differential Search Algorithm (DSA) meta-heuristic technique is proposed for systematically tuning the scaling factors of the PID-type FLC. This work can be considered as a contribution to the results given in [1]. The fuzzy control design is formulated as a constrained optimization problem which is efficiently solved based on an improved DSA. In order to specify more robustness and performance control objectives of the proposed DSA-tuned PID-type FLC, different optimization criteria such as Integral Square Error (ISE) and Maximum Overshoot (MO) are considered and compared. The remainder of this paper is organized as follows. In Section 2, the studied PID-type FLC structure, is presented and formulated as a constrained optimization problem. A constrained DSA algorithm, used in solving the formulated problem, is described in section 3. The DSA-based simulation results are compared with those obtained by the classical Genetic Algorithm Optimization

(GAO) and PSO based approach. Section 4 is dedicated to apply the proposed DSA-based fuzzy control approach on an electrical DC drive benchmark.

II. PID-TYPE FLC TUNING PROBLEM FORMULATION

In this section, the PID-type fuzzy controller synthesis problem is formulated as a constrained optimization problem which will be resolved by the means of the developed DSA algorithm.

A. PID-type fuzzy control structure

The particular PID-type fuzzy controller structure, originally proposed by Qiao and Mizumoto within the continuous-time formalism [14], retains characteristics similar to the conventional PID controller. This result remains valid while using a type of FLC with triangular uniformly distributed membership functions for the fuzzy inputs and a crisp output, the product-sum inference and the center of gravity defuzzification methods [1, 5, 7, 14, 15].

Under these conditions, the equivalent proportional, integral and derivative control components of such a PID-type FLC are given by $\alpha K_e \mathcal{P} + \beta K_d \mathcal{D}$, $\beta K_e \mathcal{P}$, and $\alpha K_d \mathcal{D}$, respectively, as shown in [7,14,15]. In these expressions, $\mathcal{P}$ and $\mathcal{D}$ represent relative coefficients, K_e, K_d, α and β denoted the scaling factors associated to the inputs and output of the FLC. When approximating the integral and derivative terms within the discrete-time framework, we can consider the closed-loop control structure for a digital PID-type FLC, as shown in Fig. 1. The dynamic behaviour of this PID-type FLC structure is strongly depending on the scaling factors, difficult and delicate to tune.

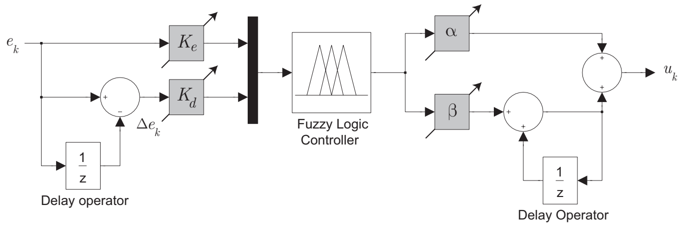

Fig. 1. Proposed discrete-time PID-type FLC structure.

As shown in Fig.1, this particular structure of PID-type fuzzy controller uses two inputs: the error e_k and the variation of error Δe_k, to provide the output u_k that describes the control law.

B. Optimization-based control problem formulation

The choice of the adequate values for the scaling factors of the described PID-type FLC structure is often done by a trials-errors hard procedure. This tuning problem becomes difficult and delicate without a systematic design method. To deal with these difficulties, the optimization of these control parameters is proposed as a promising solution. This tuning can be formulated as the following constrained optimization problem:

$$\begin{cases} \underset{\boldsymbol{x}=(K_e,K_d,\alpha,\beta)^T \in \mathbb{S} \subset \mathbb{R}_+^4}{\text{minimize}} \; f(\boldsymbol{x}) \\ \text{subject to} \\ g_1(\boldsymbol{x}) = D - D^{\max} \le 0 \\ g_2(\boldsymbol{x}) = t_s - t_s^{\max} \le 0 \\ g_3(\boldsymbol{x}) = E_{ss} - E_{ss}^{\max} \le 0 \end{cases} \tag{1}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ the cost function, $\mathbb{S} = \{\boldsymbol{x} \in \mathbb{R}^m; \boldsymbol{x}_{\min} \le \boldsymbol{x} \le \boldsymbol{x}_{\max}\}$ the initial search space, that is supposed to contain the desired design parameters, and $g_l : \mathbb{R}^m \to \mathbb{R}$ the nonlinear problem's constraints.

The optimization-based tuning problem (1) consists in finding the optimal decision variables, representing the scaling factors of a given PID-type FLC structure, which minimize the defined cost function, chosen as the Maximum Overshoot (MO) and the Integral of Square Error (ISE) performance criteria. These cost functions are minimized, using the proposed constrained DSA meta-heuristic, under various time-domain control constraints such as overshoot D, steady state error E_{ss}, rise time t_r and settling time t_s of the system's step response, as shown in the equation (1). Their specified maximum values constraint the step response of the DSA-tuned PID-type FLC controlled system, and can define some time-domain templates.

III. PROPOSED DIFFERENTIAL SEARCH ALGORITHM-BASED APPROACH

In this section, the basic concepts as well as a pseudo code of the differential search algorithm meta-heuristic are described. An external penalty method is introduced in order to improve the DSA tool in terms of handling constraints.

A. Survey of Differential Search Algorithm

The Differential Search Algorithm is a recent population-based meta-heuristic optimization algorithm invented by P. Civicioglu in 2012 [3]. This global and stochastic algorithm simulates the Brownian-like random-walk movement used by an organism to migrate.

Migration behavior allows the living beings to move from a habitat where capacity and diversity of natural sources are limited to a more efficient habitat. In the migration movement, the migrating species of living beings constitute a superorganism containing a large number of individuals. Then it starts to change its position by moving towards more fruitful areas using a Bownian-like random-walk model. The population made up of random solutions of the respective problem corresponds to an artificial-superorganism migrating. The artificial superorganism migrates to a global minimum value of the problem. During this migration, the artificial-superorganism tests whether some randomly selected positions are suitable temporarily during the migration. If such a position tested is suitable to stopover for a temporary time during the migration, the members of the artificial-superorganism that made such discovery immediately settle at the discovered position and continue their migration from this position on [3].

In DS algorithm, the artificial-organisms (i.e., X_i, $i=1,\ldots,N$) making up an artificial-organism (i.e., $Superorganism_g$, $g=1,\ldots,\text{maxgeneration}$) contain members as much as the size of the problem (i.e., x_{ij}, $j=1,\ldots,D$). Here, N signifies the number of elements in the superorganism and D indicates the size of the respective problem. A member of an artificial-organism in an initial position is defined by using Equation (2):

$$x_{ij} = rand\left(up_j - low_j\right) + low_j \tag{2}$$

In such case, the artificial-organisms are defined by $X_i = \left[x_{ij}\right]$ and Superorganism$_g$ indicates the artificial-superorganism made up of the artificial-organisms $Superorganism_g = \left[X_i\right]$. Randomly selected individuals of the artificial-organisms move towards the targets of donor $X_{random-suffling(i)}$, in order to discover stopover sites, which are very important for a successful migration. The size of the change occurred in the positions of the members of the artificial-organisms is controlled by the scale value produced by using a gamma random number generator. The stopover site position is produced by using Equation (3):

$$StopoverSite = Superorganism + Scale\left(donor - Superorganism\right) \tag{3}$$

B. DSA pseudo code

A basic pseudo code for the DSA is presented in APPENDIX as described by P. Civicioglu in [3]. There are only two control parameters in DSA (p_1 and p_2). The author of the technique indicated that the random values in [0, 0.3] usually provide best solutions for a given problem. So, the algorithm is not much sensitive to the values of p_1 and p_2.

C. Handling constraints method

Similarly to other global meta-heuristics, the DSA technique is originally formulated as an unconstrained optimizer. Several techniques have been proposed to deal with constraints. One useful approach is augmenting the cost function of unconstrained problem with penalties proportional to the degree of constraint infeasibility. In this paper, the following external static penalty technique is used:

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{l=1}^{n_{con}} \lambda_l \max\left[0, g_l(\boldsymbol{x})^2\right] \tag{4}$$

where λ_l is a prescribed scaling penalty parameters, chosen as a constant equal to 10^4, and n_{con} is the number of problem constraints g_l.

IV. DSA-TUNED PID-TYPE FLC OF A DC DRIVE

This section is dedicated to apply the proposed DSA-tuned PID-type FLC for the variable speed control of a DC drive. All the obtained simulations results are presented and discussed.

A. Plant model description

The considered benchmark is a 250 watts electrical DC drive. The machine's speed rotation is 3000 rpm at 180 volts DC armature voltage. The motor is supplied by an AC-DC power converter. The considered electrical DC drive can be described by the following model [2]:

$$G(s) = \frac{0.05}{(1+0.3s)(1+0.014s)} \tag{5}$$

B. Simulation results

For the proposed DSA-tuned PID-type FLC structure, product-sum inference and center of gravity defuzzification methods are adopted for the FLC block. Uniformly distributed and symmetrical membership functions, are assigned for the fuzzy input and output variables. The associated fuzzy rule-base is given in Table I.

The linguistic levels assigned to the input variables e_k and Δe_k, and the output variable xx are given as follows: N (Negative), Z (Zero), P (Positive), N (Negative), NB (Negative Big) and PB (Positive Big). The view of this rule-base is illustrated in Fig. 2.

TABLE I. FUZZY RULE-BASE FOR THE STANDARD FLC

Error\Error variation	N	Z	P
N	NB	N	Z
Z	N	Z	P
P	Z	P	PB

For our design, the initial search domain of PID-type FLC parameters is chosen as $\mathbb{S} = \left\{\boldsymbol{x} \in \mathbb{R}^4; [1 \quad 5 \quad 2 \quad 25] \leq \boldsymbol{x} \leq [5 \quad 10 \quad 10 \quad 50]\right\}$. We use a population size equal to **30** and run all used algorithms under **100** iterations. The size of optimization problem is equal to 4. The control parameters of the DS algorithm are fixed as $p_1=p_2=0.3rand$.

In order to get some statistical data on the quality of results and validate the proposed approach, we run the proposed DS algorithm 20 times and feasible solutions are found in 100 % of trials and within an acceptable CPU computation time. The obtained optimization results are summarized in Tables II, and III. Besides, the fact that the algorithm's convergence always takes place in the same region of the design space, whatever is the initial population, indicates that the algorithm succeeds in finding a region of the interesting research space to explore. The performance of DSA is compared with PSO [1,2,4,8,11] and GAO [6,12] based techniques.

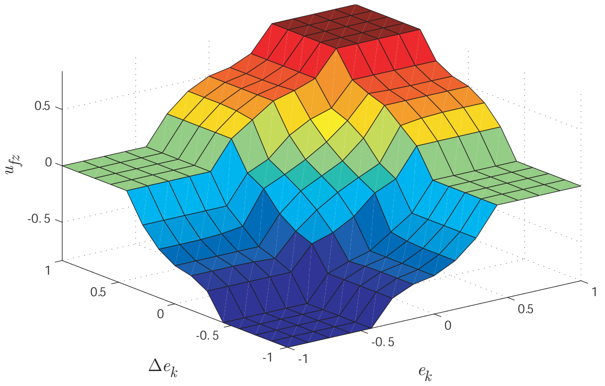

Fig. 2. View of the fuzzy rule-base for the standard FLC.

According to the statistical analysis of Tables II and III, we can conclude that the proposed DSA-based approach produces better, or sometimes near, results in comparison with the PSO- and GAO-based methods for the considered MO and IAE criteria. The best performance of the DS algorithm is its fastness in comparison with the used techniques in term of CPU computation time.

TABLE II. OPTIMIZATION RESULTS FROM 20 TRIALS OF PROBLEM (1): ISE CRITERION.

Cost function	Algo	Best	Mean	Worst	STD
ISE	DSA	0.1621	0.1691	0.1760	0.0045
	PSO	0.1600	0.1715	0.1802	0.0140
	GAO	0.1643	0.1722	0.1799	0.0086

TABLE III. OPTIMIZATION RESULTS FROM 20 TRIALS OF PROBLEM (1): MO CRITERION.

Cost function	Algo	Best	Mean	Worst	STD
MO	DSA	0.0365	0.0722	0.1307	0.0277
	PSO	0.0422	0.0936	0.1420	0.0511
	GAO	0.0441	0.0913	0.1300	0.0373

The robustness of the proposed DS algorithm convergence, under variation of the p_1 and p_2 control parameters, is analysed based on numerical simulations as shown in Fig. 3. The DS algorithm's convergence is guaranteed for a large variation of these parameters. So, we can say that the DS is a free control parameters algorithm.

The time-domain performances of the proposed DSA-tuned PID-type FLC structure are illustrated in Fig. 4. All results, for various obtained decision variables, are acceptable and show the effectiveness of the proposed DSA-based method. The specified overshoot ($\mathbf{D^{max}=20\%}$), settling time ($\mathbf{ts^{max}=0.9}$ **sec**) and steady state error ($\mathbf{Ess^{max}=0.00}$) performance criteria are respected.

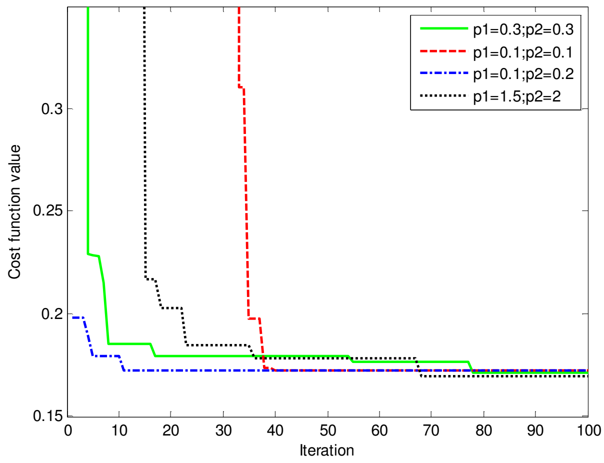

Fig. 3. Robustness convergence of the proposed DSA-based approach: ISE criterion case.

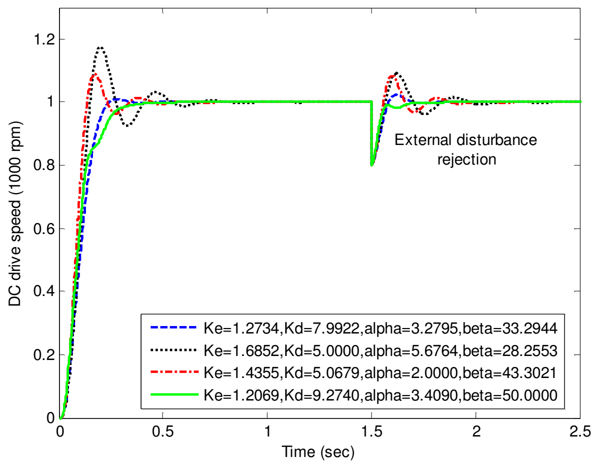

Fig. 4. Step responses of the DSA-tuned PID-type fuzzy controlled system: ISE criterion case.

V. CONCLUSION

A new method for tuning the PID-type FLC scaling factors, based on an improved DSA meta-heuristic, is proposed and successfully applied to an electrical DC drive speed control. This efficient DSA-based tool leads to a robust and systematic PID-type fuzzy control design approach. The performances comparison, with the standard GAO- and PSO-based methods, shows the efficiency and superiority of the proposed DSA-based approach in terms the convergence speed and the quality of obtained solutions. This hybrid DSA and PID-type fuzzy design methodology is systematic, practical and simple without the need to exact analytic plant model description. The obtained simulation results show efficiency in terms of performance and robustness and prepare to an experimental implementation within a real-time framework. Application of the proposed DSA-based fuzzy approach for more complex systems will be envisaged.

APPENDIX

Pseudo code of Differential Search Algorithm

```
Require:
N: Size of the population, where i = {1,2,3,…,N}
D: Dimension of the problem
G: Number of maximum generation
Superorganism = [ArtificialOrganism_i]
Superorganism = initialize ()
y_i = Evaluate (ArtificialOrganism_i)
for cycle = 1: G do
donor = SuperorganismRandom_Shuffling(i)
Scale= randg [2rand_1](rand_2-rand_3)
StopoverSite=Superorganism+Scale(donor-Superorganism)
    p_1= 0.3rand and p_2= 0.3.rand
    if rand_6 < rand_7 then
     if rand_8 < p1 then
       r = rand (N, D)
       for Counter1 = 1: N do
       r (Counter1, :) = r (Counter1, :) rand_9
       end for
     else
       r = ones (N, D)
       for Counter2 = 1: N do
      r (Counter2, randi(D))= r(Counter2,randi(D))rand_10
       end for
    end if
   else
       r = ones (N, D)
     for Counter3 = 1: N do
     d=randi(D,1,[p2randD])
     for Counter4=1: size (d) do
     r (Counter3, d (Counter4)) =0
     end for
     end for
     end if
     individuals_I,J ← r_I,J
StopoverSite(individuals_I,J):=Superorganism (individuals_I,J)
if StopoverSite_i,j < low_i,j or StopoverSite_i,j > up_i,j then
     StopoverSite_i,j :=rand(up_j-low_j)+low_j
     end if
y_StopoverSite,i=evaluate (StopoverSite_i)
y_Superorganism,i := y_StopoverSite,i if y_StopoverSite,i < y_Superorganism,i
else y_Superorganism,i := y_Superorganism,i
ArtificialOrganism_i:=StopoverSite_i
if y_StopoverSite,i< y_Superorganism,i
else ArtificialOrganism_i:=:=ArtificialOrganism_i
```

REFERENCES

[1] S. Bouallègue, J. Haggège, M. Ayadi and M. Benrejeb, "PID-type fuzzy logic controller tuning based on particle swarm optimization", Engineering Applications of Artificial Intelligence, vol. 25, pp. 484-493, 2012.

[2] S. Bouallègue, J. Haggège and M. Benrejeb, "Particle Swarm Optimization-Based Fixed-Structure H∞ Control Design", International Journal of Control, Automation, and Systems, vol. 9, no.2, pp. 258-266, 2011.

[3] P. Civicioglu, "Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm", Computers & Geosciences, vol. 46, pp. 229-247, 2012.

[4] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", 6th International Symposium on Micro Machine and Human Science, pp. 39-43, Nagoya, 1995.

[5] İ. Eker and Y. Torun, "Fuzzy logic control to be conventional method", Energy Conversion and Management, vol. 47, pp. 377-394, 2006.

[6] D.E. Goldberg, "Algorithmes génétiques : exploration, optimisation et apprentissage automatique", Addison-Wisley, France, 1994.

[7] M. Güzelkaya, I. Eksin and E. Yesil, "Self-tuning of PID type Fuzzy Logic Controller Coefficients via Relative Rate Observer", Engineering Applications of Artificial Intelligence, vol. 16, pp. 227–236, 2003.

[8] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", IEEE International Joint Conference on Neural Networks, pp. 1942-1948, Perth-Australia, 1995.

[9] C.C Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller–Part I", IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no. 2, pp. 404-418, 1998.

[10] C.C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller–Part II", IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no. 2, pp. 419-435, 1998.

[11] R. Madiouni, S. Bouallègue, J. Haggège and P. Siarry, "Particle Swarm Optimization-based Design of Polynomial RST Controllers", 10th IEEE International Multi-Conference on Systems, Signals & Devices, Hammamet-Tunisia, 2013.

[12] MathWorks Inc., "Genetic Algorithm and Direct Search Toolbox", User's Guide. Natick, 2009.

[13] K.M. Passino, S. Yurkovich, "Fuzzy Control", Addison Wesley Longman, Menlo Park, California 1998.

[14] W.Z. Qiao and M. Mizumoto, "PID type Fuzzy Controller and Parameters Adaptive Method", Fuzzy Sets and Systems, vol. 78, pp. 23–35 1996.

[15] Z.W. Woo and H.Y. Chung and J.J. Lin, "A PID type Fuzzy Controller with Self-tuning Scaling Factors", Fuzzy Sets and Systems, vol. 115, pp. 321–326, 2000.