An Approach of Fault Diagnosis Using Meta-Heuristics: a New Variant of the Differential Evolution Algorithm

Lídice Camps Echevarría[1], Orestes Llanes Santiago[1], Antônio José da Silva Neto[2], and Haroldo Fraga de Campos Velho[3]

[1]Instituto Superior Politécnico José Antonio Echeverría, CUJAE, Havana, Cuba

[2]State University of Rio de Janeiro, IPRJ-UERJ, Rio de Janeiro, Brazil

[3]National Institute of Spatial Researchs, INPE, São José dos Campos, SP, Brazil

lidice@cemat.cujae.edu.cu, orestes@electrica.cujae.edu.cu, ajsneto@iprj.uerj.br, haroldo@lac.inpe.br

Abstract. This paper presents an application of meta-heuristics to fault diagnosis. The idea behind this application is to develop methods for fault diagnosis that should be robust, sensitive and with an adequate computational cost. Applications of meta-heuristics are possible based on the formulation of fault diagnosis as an optimization problem. The results indicate the suitability of the use of meta-heuristics for fault diagnosis. In particular, this study shows an application of meta-heuristic termed Differential Evolution to diagnosing a DC Motor benchmark. This allowed developing a new variant of Differential Evolution, namely, Differential Evolution with Particle Collision. This new algorithm was validated with some benchmark functions for continuous optimization, showing that it over-performed the behavior of Differential Evolution.

Keywords. Differential evolution, meta-heuristics, fault diagnosis, particle collision, robustness, sensitivity.

Un enfoque al diagnóstico de fallos aplicando meta heurísticas: nueva variante del algoritmo Evolución Diferencial

Resumen. Este trabajo presenta un estudio de la aplicación de meta heurísticas al diagnóstico de fallos, con el fin de desarrollar métodos que sean robustos ante perturbaciones, sensibles ante fallos incipientes y con adecuado costo computacional. La aplicación de las mismas es posible a partir de la formulación del diagnóstico de fallos como un problema de optimización. Los resultados indican la factibilidad del uso de meta heurísticas. En este estudio se aplicó la meta heurística, Evolución diferencial al diagnóstico de fallos en el sistema de prueba Motor CD. El estudio permitió desarrollar un nuevo algoritmo que se ha llamado Evolución diferencial con colisión de partículas. Este fue validado con funciones de prueba de optimización continua mostrando su superioridad sobre Evolución diferencial.

Palabras clave. Colisión de partículas, diagnóstico de fallos, evolución diferencial, meta heurísticas, robustez sensibilidad.

1 Introduction

Fault Diagnosis (FDI) includes detection, isolation and identification of faults that can eventually affect a system [5, 11]. Faults should be quickly diagnosed in order to take further decisions that avoid economic losses as well as damages to human capital or environment.

FDI methods are generally divided into two groups: model-based methods [5, 11, 21, 26] and non-model-based methods [27]. The quantitative

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

model-based methods make use of an analytical or computational model of the system. The great variety of proposed model-based methods is brought down to a few basic concepts such as parity space, observer approach, and parameters identification or estimation approach [5, 21].

Due to disturbances that can be affecting the system, FDI methods can constantly produce false alarms. The FDI methods that handle this situation in an appropriate way are called robust [5, 11, 21]. Furthermore, in order to gain in robustness, many methods lose capability for diagnosing incipient faults. This is called lack of sensitivity. Thus, for a successful design of FDI methods it is required to achieve an adequate balance between robustness and sensitivity, as well as a computational cost that allows diagnosing online processes [5, 21, 22].

Despite the fact that many robust and sensitive FDI methods have been developed, the development of new FDI methods is still considered as an open problem [21, 22].

A detailed description of each model-based approach and their limitations can be found in [5, 21, 30]. It is recognized that observer scheme and parity space do not always allow isolation of actuator faults [30]. For non-linear models, the complexity of the design of observer increases while it is necessary to build an exact model of the system for parity space [14, 30].

Parameter estimation approach requires the knowledge of relationships between such parameters and physical coefficients of the system, as well as the influence of the faults on these coefficients [11]. This approach does not provide a good diagnosis for sensor faults. Furthermore, it usually demands a high computing time, which makes it unfeasible for most situations [30].

Recent studies have shown that with the use of soft computing techniques, FDI methods with appropriate characteristics can be obtained [14, 29, 30]. Within them, Neural Networks and Fuzzy Logic achieve the greatest number of applications to the FDI. In contrast, application of meta-heuristics to optimization has not been widely studied in FDI area [29, 30], and most of such applications are related to non-model-based FDI methods [12, 28, 29, 30].

It is also known that FDI methods based on observers or parity space put a lot of effort in generation of robust and sensitive residual which is related to the detection part. Thus, the residual generation is highly dependent on the model that describes the system [5, 11].

Considering reported applications of meta-heuristics to other problems that also deal with environments under disturbances, as well as uncertainties on models or measurements [2, 13], this article shows an application of meta-heuristics to the development of FDI methods which do not need to invest a lot of effort in generation of robust and sensitive residuals.

In this study, we considered the algorithm called Differential Evolution (DE) [23, 20] which has been successfully applied in other areas of engineering [13] subject to disturbances and noisy environments. This algorithm has not been previously applied to FDI like other meta-heuristics such as Genetic Algorithms, Particle Swarm Optimization or Ant Colony Optimization [5, 12, 19, 28, 31], but it has received recognition due to its simpler structure and better results in comparison with other methods [4]. Our study also allowed the development of Differential Evolution with Particle Collision (DEwPC) method. DEwPC is a new variant of DE that modifies its Selection operator based on the scheme of another meta-heuristic, Particle Collision Algorithm [2]. DEwPC was validated for continuous optimization problems.

The main contributions of this article can be summarized as follows: study of application of meta-heuristics for developing robust and sensitive FDI methods, and development of a new variant DEwPC for improving the computational cost required by DE. The viability of the proposal is demonstrated by diagnosing simulation data from a DC Motor [5].

This article is organized as follows. In Section 2, the main aspects related to FDI are introduced. In Section 3, the algorithm of DE is briefly described. DEwPC is presented in Section 4. Section 5 details the case study, DC Motor benchmark. In Section 6, the experimental methodology is explained. Section 7 shows the results. Finally, we give some concluding comments and remarks.

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

2 FDI as an Optimization Problem

In FDI methodologies, there are three kinds of faults, depending on the part of the system that they directly affect: actuator faults, process faults and sensor faults [5, 7, 11].

Faults affecting the system may eventually change one or several parameters in the model that describes it. FDI based on model parameters is divided into two steps. The first is meant for estimation of the model parameters vector $\theta(t)$. The second is meant for detecting and isolating faults basing on known relationships between the model parameters, physical coefficients of the system, and faults [11].

The main drawback of this approach is that fault isolation may become extremely difficult when the model parameters do not uniquely correspond to those of the system. In such cases, it is usually difficult to distinguish a fault from a change in the parameters vector $\theta(t)$ [11]. It should be also pointed out that detection of faults in sensors and actuators is possible but rather complicated [30].

Instead of estimating the model parameters vector $\theta(t)$, let's consider faults in a model explicitly. This approach is widely used in other model-based FDI methods such as Diagnostic Observer or Parity Space [5, 7].

In the case of a SISO (single input single output) system in a closed loop that is described by a Lineal Time Invariant (LTI) model in transfer function, the model can be represented as [5, 11]

$$y(s) = G_{yw}(s)w(s) + G_{yf_u}(s)f_u(s) + G_{yf_y}(s)f_y(s) + G_{yf_p}(s)f_p(s) \tag{1}$$

where $w(s) \in \mathbb{R}$ is the reference signal of the control system, $f_p, f_a, f_y \in \mathbb{R}$ are faults in process, actuator and output sensor, respectively. The transfer function $G_{yw}(s)$ represents the dynamics of the system while $G_{yf_u}(s)$, $G_{yf_p}(s)$ and $G_{yf_y}(s)$ are the transfer functions that represent the faults in the actuator, process and sensor, respectively [5].

This proposed approach considers estimation of the faulty parameters vector $\theta_f = [f_u \quad f_y \quad f_p]^T$. Estimation of θ_f allows diagnosing the system. It can be obtained from the minimization of the sum of the squares of output errors:

$$\min \; F(\hat{\theta_f}) = \sum_{t=1}^{I} \left[y(\theta_f, t) - \hat{y}(\hat{\theta_f}, t) \right]^2$$
$$\text{s.t} \;\; \theta_{f(min)} \leq \hat{\theta_f} \leq \theta_{f(max)} \tag{2}$$

where S is the number of sampling instants, $\hat{y}(\hat{\theta_f}, t)$ is the estimated vector output in each instant of time and it is obtained from the model (1); $y(\hat{\theta_f}, t)$ is the output vector measured by the sensors at the same time instant t.

For solving this optimization problem in (2), we apply meta-heuristics due to the fact that they have shown to be robust in other areas of application.

3 Differential Evolution

Differential Evolution (DE) was proposed in 1995 for optimization problems [23]. DE is based on three operators: Mutation, Crossover and Selection [20, 23] from Genetic Algorithm (GA) [9].

These operators are based on vector operations. This is the main difference comparing with GA. This difference provides DE with a more simple structure and computational implementation than GA. Two other most important advantages of DE are speed and robustness [20, 23].

The algorithm generates at each iteration $Iter$ a new population of Z feasible solutions $X^1_{Iter}, X^2_{Iter}, ..., X^Z_{Iter}$ at each iteration. For that purpose the three operators are applied on the current population. This mechanism can be summarized with the notation

$$DE/X^{\delta}_{Iter-1}/\gamma/\lambda \tag{3}$$

where γ indicates the number of pairs of solutions to be used for perturbations of the current solution X^{δ}_{Iter-1}; λ represents the distribution function to be used during the crossover. In this work we applied the scheme $DE/X^{best}/2/bin$, bin being a notation for a binomial distribution function. Mutation is described by

$$\hat{X}^z_{Iter} = X^{best} + C_{scal}(X^{\alpha_1}_{Iter-1} - X^{\alpha_3}_{Iter-1} + X^{\alpha_2}_{Iter-1} - X^{\alpha_4}_{Iter-1}) \tag{4}$$

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

where $X^{best}, X^{\alpha_1}_{Iter-1}, X^{\alpha_2}_{Iter-1}, X^{\alpha_3}_{Iter-1}, X^{\alpha_4}_{Iter-1} \in \mathbb{R}^n$ are solutions of the current population and C_{scal} is a parameter called *Scaling factor*. On the other hand, the Crossover and Selection operator can be described as

— Crossover:

$$\hat{x}^z_{(Iter)n} = \begin{cases} \hat{x}^z_{(Iter)n} & \text{if } q_{rand} \leq C_{cross} \\ x^\delta_{(Iter-1)n} & \text{otherwise} \end{cases} \tag{5}$$

where $\hat{x}^z_{(Iter)n}$ are components of the vector $\hat{X}^z_{Iter}$; $0 \leq C_{cross} \leq 1$ is another parameter called *Crossover factor*; and q_{rand} is a random number generated by means of a distribution represented by λ.

— Selection: the vector X^z_{Iter}, which will be part of the new population, is selected following the rule:

$$X^z_{Iter} = \begin{cases} \hat{X}^z_{Iter} & \text{if } F(\hat{X}^z_{Iter}) \leq F(X^\delta_{Iter-1}) \\ X^\delta_{Iter-1} & \text{otherwise} \end{cases} \tag{6}$$

A general description of the algorithm for DE is given in Fig. 1.

Data: $Z, MaxIter, C_{cross}, C_{scal}$
Result: X^{best}
Generate an initial population of Z solutions;
Select best solution X^{best};
for $Iter = 1$ *to* $Iter = MaxIter$ **do**
  Apply *Mutation*;
  Apply *Crossover*;
  Apply *Selection*;
  Update X^{best};
  Verify stopping criteria;
end
Solution: X^{best}

Fig. 1. Algorithm of DE

Some modifications over the original version of DE have been made in order to improve its ability for escaping from local minimum. The more successful variants of DE are focused on variations of Mutation operator and the self-adaptation of parameters C_{cross} and C_{scal} [1, 3, 15, 17, 25, 32, 33, 34]. DE has also been hybridized with other meta-heuristics [10].

4 New Algorithm: Differential Evolution with Particle Collision

The new Differential Evolution with Particle Collision algorithm (DEwPC) has the objective to improve the performance of DE based on the incorporation of some ideas from Particle Collision Algorithm (PCA) [2, 18, 20].

PCA is inspired on the interaction of particles inside a nuclear reactor [2, 6, 18, 20]. It is a recent algorithm with successful application to other problems in engineering [2] and it has been extended to a based population version [16].

DEwPC keeps the same structure as Mutation and Crossover operators in DE, while introducing a modification in Selection operator. This modification adds a new parameter $MaxIter_c$ with the objective to improve the capacity of DE for escaping from local minimum.

The new Selection operator takes the ideas of Absorption and Scattering from PCA. The adaptation of this operator to DEwPC has been called *Selection with Absorption - Scattering with probability* and it has been established as

Selection with Absorption - Scattering with probability

— If $F(\hat{X}^z_{Iter}) \leq F(X^\delta_{Iter-1})$ then operator *Absorption* is applied to $\hat{X}^z_{Iter}$.

— If $F(\hat{X}^z_{Iter}) > F(X^\delta_{Iter-1})$ then operator *Scattering with probability* is applied to $\hat{X}^z_{Iter}$.

The Absorption and Scattering with probability operators are represented in Figs 2 and 3, respectively.

In Fig. 4 the algorithm DEwPC is also represented. Small Search operator indicates a small stochastic perturbation around a solution. Search indicates a stochastic perturbation around a solution [20].

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

Data: $\hat{X}_{Iter}$
Result: X_{Iter}
$X_{Iter} = \hat{X}_{Iter}$;
SmallSearch$(X_{Iter}, MaxIter_c)$;

Fig. 2. Absorption Operator

Data: $\hat{X}_{Iter}, F(X^{best})$
Result: X_{Iter}
Compute $F(\hat{X}_{Iter})$;
Compute $p_{r(Iter)} = 1 - \frac{F(X^{best})}{F(\hat{X}_{Iter})}$;
if $rand < p_{r(Iter)}$ **then**
  $X_{Iter} = \hat{X}_{Iter}$;
  Search$(X_{Iter}, MaxIter_c)$;
  else;
  $X_{Iter} = X_{Iter-1}$;
end

Fig. 3. Scattering with probability Operator

Data: C_{cross}, C_{scal}, Z, $MaxIter$, $MaxIter_c$
Result: X^{best}
Generate an initial population of Z solutions;
Select best solution X^{best};
for $Iter = 1$ *to* $Iter = MaxIter$ **do**
  Apply *Mutation*;
  Apply *Crossover*;
  for $j = 1$ *to* $j = Z$ **do**
    if $rand < 0.7$ **then**
      Apply *Absorption-Scattering with probability* to $\hat{X}^{(j)}_{Iter}$;
      else;
      Apply *Selection* to $\hat{X}^{(j)}_{Iter}$;
    end
  end
  Update X^{best};
  Verify stopping criteria;
end
Solution: X^{best}

Fig. 4. Algorithm for DEwPC

4.1 Validation of DEwPC

In order to validate our proposal DEwPC for continuous optimization problems, five test functions have been considered. They are taken from the inventory of functions recommended in the literature [4, 24]. A comparison against DE was made.

The test functions used in this article are F1 Shifted Sphere Function, F2 Shifted Schwefel's Problem 1.2; F4 Shifted Schwefel's Problem 1.2 with Noise in Fitness; F6 Shifted Rosenbrock's Function and F7 Shifted Rotated Griewank's Function without Bounds [8, 24]. The experiments and evaluation criteria follow the indications recommended in [24].

— Experiment A. The stopping criteria are the maximum number of evaluations of the objective function $Eval_{max} = 10000n$ or the maximum error $ErrTer = 10^{-8}$. The following quantities are separately computed for each problem:

  – Success Rate, $SR = \frac{EE}{ET}$: EE being the number of successful runs and ET the number of total runs.
  – Success Performance, $SP = \frac{\bar{Eval}_{EE}}{SR}$, where $\bar{Eval}_{EE}$ is the average of the number of function evaluations for successful runs.

  A successful run is understood as a run during which the algorithm achieves the fixed accuracy level $ErrTer$ within $Eval_{max}$.

— Experiment B. We considered $Eval_{max}$ as the only one stopping criterion. Different values were established: $Eval_{max} = 100n, 1000n, 10000n$, respectively. The final error was determined for each case.

For each test function the algorithms ran 25 times, for each experiment. This number allows getting statistically valid conclusions concerning the evaluations of algorithms [4, 8, 24].

In the experiments, we considered $n = 10$ or $n = 30$. In order to suggest values for the parameters of DEwPC, some experiments with

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

different set of values for Z and $MaxIter_c$ were made. We considered only these parameters since we are interested in decreasing the computational cost. For that reason we decreased the size of population Z, with respect to DE, and made $MaxIter_c$ dependent on Z. We are also interested in making Z dependent on the size of the problem n.

In Table 1, a different set of values are shown. In all cases, the parameters $C_{cross} = 0.9$ and $C_{scal} = 0.5$ were kept constant. DE parameters were assigned the values recommended in [20, 23].

Table 1. Set of values for DEwPC parameters

Set	Z	$MaxIter_c$
1	10n	0.3Z
2	10n	0.2Z
3	10n	0.1Z
4	5n	0.3Z
5	5n	0.2Z
6	5n	0.1Z
7	2n	0.3Z
8	2n	0.2Z
9	2n	0.1Z

Concerning function F6, it is well known that the global optimum is inside a long, narrow, parabolic shaped valley. To search the valley is trivial but convergence to the global optimum is difficult. Hence this problem is often used in assessing the performance of the optimization algorithms. Therefore, this function was used for establishing the set of values of DEwPC throughout the experiments.

In Figs 5 and 6 we show a comparison of the results from minimization of function F6 for $n = 10$ and $n = 30$, respectively, for the sets of values in Table 1. In all cases $Eval_{max} = 10000n$ was considered as a stopping criterion. For Sets 1, 2 and 3, which have the biggest Z, the error is not proportional to the increase in $MaxIter_c$. In this case, Set 3 is closer to DE. The results are better for Cases 4, 5 and 6, which have a half of the recommended population in DE, and these results were obtained with $MaxIter_c = 0.2Z$. The best result was observed for Set 5.

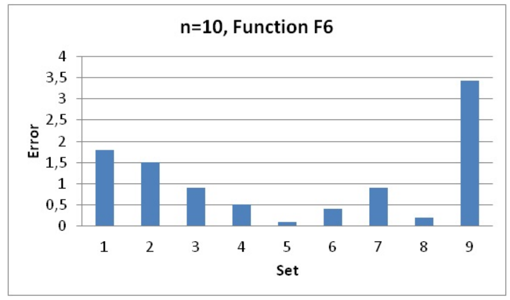


Fig. 5. Final error obtained in the minimization of F6, $n = 10$

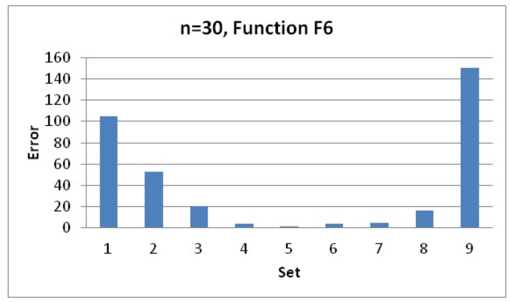


Fig. 6. Final error obtained in the minimization of F6, $n = 30$

The values for the parameters in DE and DEwPC that were set during the experiments are shown in Table 2. On the other hand, the values for the DEwPC parameters were those corresponding to Set 5. The values for the parameters of DEwPC as a function of the problem dimension n are also shown in Table 2.

Table 2. Parameter values for DE and DEwPC

alg	Z	C_{scal}	C_{cross}	$MaxIter_c$
DE	10n	0.5	0.9	—
DEwPC	5n	0.5	0.9	20%Z

4.1.1 Results of Experiment A

In Tables 3 and 4 we present a summary of the evaluation criteria, for $n = 10$ and $n = 30$, respectively. Furthermore, the normalized SP with respect to the best, SP_{best}, between both algorithms is given. In the case when some of

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

the indicators cannot be computed due to a lack of successful runs, it is marked by means of $- - --$ and the average of the final error is put in the brackets.

Table 3 shows that the DEwPC algorithm, for $n = 10$, has the best SP in four of five cases; in the left case no one of both algorithms achieved successful runs. In this case the final error of DE was higher than the final error of DEwPC.

Table 3. Comparison between SR and SP, $n = 10$

fun	SR	SP	SP/SP_{best}
DE			
F1	100	23148	1.41
F2	100	24312	1.26
F4	100	42700	1.34
F6	84	61031	1.02
F7	0	——	[0.2737]
DEwPC			
F1	100	16431	1
F2	100	19363	1
F4	100	31826	1
F6	92	59828	1
F7	0	——	[0.0643]

Table 3 shows that for function F6, the SR achieved by DEwPC is better than the SR of DE. This indicates that the new Selection operator provides DEwPC with a better local search capability and better searching diversity.

In Table 4, the results for the case $n = 30$ are presented. DEwPC achieved successful runs in four of the five test functions while DE achieved successful runs only in two of the cases. These results allow concluding that in a more complex search space (with higher dimension), our proposal DEwPC allows reaching better results than DE. In other words, the modification in the Selection operator provides DEwPC with a more effective way to avoid staying in a local minimum.

4.1.2 Results of Experiment B

The results of the experiments for functions F4, F6 and F7 are shown in Figs 7, 8 and 9. We have chosen these functions for representing the results due to the fact that they describe functions with

Table 4. Comparison between SR and SP, $n = 30$

fun	SR	SP	SP/SP_{best}
DE			
F1	100	252560	1.46
F2	100	114744	1.01
F4	0	——	[441.3]
F6	0	——	[300.0]
F7	0	——	[0.109]
DEwPC			
F1	100	172752	1
F2	100	113600	1
F4	4	6405000	1
F6	0	——	[25.87]
F7	56	460960	1

noise or are multimodal functions, in other words, they represent more complex sceneries within the five functions. It can be observed that the final error of DEwPC is always lower than the final error of DE, no matter the dimension.

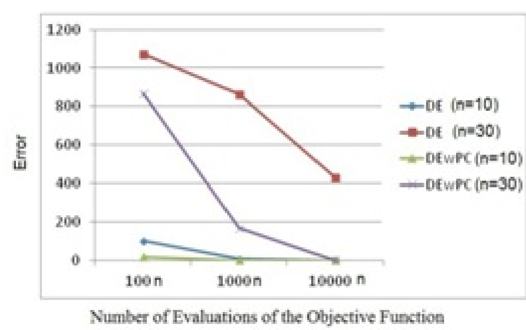


Fig. 7. Error achieved by DE and DEwPC (Experiment B, F4, $n = 10$ and $n = 30$)

Furthermore, the Wilcoxon's test showed that the error achieved by DEwPC, when the algorithm has executed the 10% of the maximum number of the objective function evaluations, is lower than the error achieved by DE under the same conditions with p-value $p = 0.0020$.

The results indicate the suitability of our proposal DEwPC. Furthermore, the results show that DEwPC over-performs DE.

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

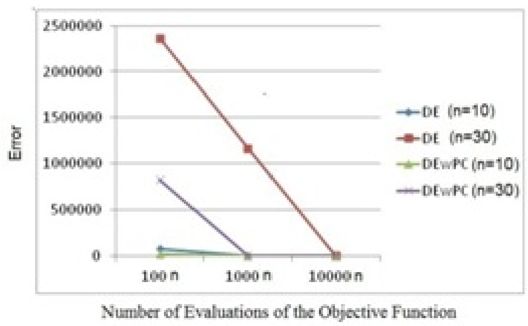


Fig. 8. Error achieved by DE and DEwPC (Experiment B, F6, $n = 10$ and $n = 30$)

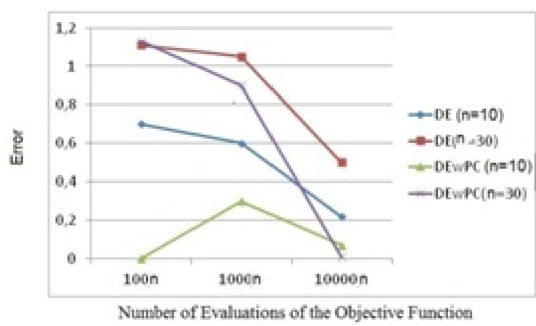


Fig. 9. Error achieved by DE and DEwPC (Experiment B, F7, $n = 10$ and $n = 30$)

5 DC Motor

DC Motor control system DR300 [5] is a benchmark for FDI. The system is formed by a permanent magnet that is coupled to a DC generator. The main function of this generator is to simulate the effect of a fault that results when a load torque is applied to the axis of the motor. The speed is measured by a tachometer that feeds the signal to a PI (proportional-integral controller) speed controller. Fig. 10 shows a block diagram for the DC Motor control system AMIRA DR300.

5.1 Mathematical Model

For this study, the internal current loop formed by the DC Motor and the tachometer has been considered as a single block that is the process to be controlled. The block diagram of the closed loop is formed by the process and the PI controller, see Fig. 10. The parameters of the laboratory DC Motor DR300 are reported in Table 3.1 from [5].

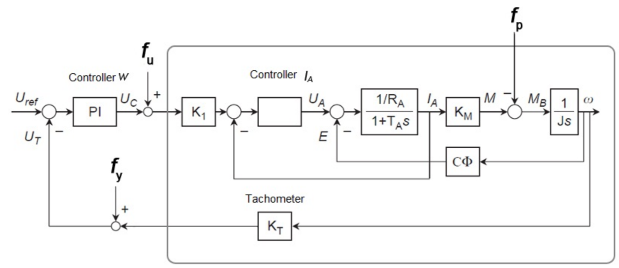


Fig. 10. Block diagram of the DC Motor control system AMIRA DR300

This analysis considers that the system can be affected by three additive faults f_u, f_p and f_y. Fault f_u represents a fault in actuator and it is modeled as a deviation of the control signal. Fault f_p represents a fault in the process itself due to a load torque applied to the axis of the motor. Fault f_y represents a fault in the measurement of the motor speed. The dynamics of the control system in the open loop is described in frequency domain by

$$U_T(s) = G_{yu}(s)U_C(s) + Gyf_p(s)f_p(s) \tag{7}$$

$$G_{yu}(s) = \frac{8.75}{(1 + 1.225s)(1 + 0.03s)(1 + 0.005s)} \tag{8}$$

$$G_{yf_p}(s) = -\frac{31.07}{s(1 + 0.005s)} \tag{9}$$

where the voltage U_T (volts) is the controlled variable; U_C (volts) is the control signal; $G_{yu}(s)$ represents the dynamics of the process in the open loop and $G_{yf_p}(s)$ is the transfer function of fault f_p.

The transfer function of the PI speed controller is

$$G_c(s) = \frac{U_C(s)}{E(s)} = 1.96 + \frac{1.6}{s} \tag{10}$$

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

Considering the other faults which may affect the system, the equation that describes it by means of the closed loop transfer function is

$$\begin{aligned} U_T(s) &= G_{yw}(s)U_{ref}(s) + G_{yf_u}(s)f_u(s) \\ &\quad + G_{yf_p}(s)f_p(s) + \ G_{yf_y}(s)f_y(s) \end{aligned} \tag{11}$$

where U_{ref} is the voltage reference.

For this study, the faults were considered to be time invariant and the following restrictions were established:

$$\begin{aligned} &f_u, f_y \in \mathbb{R} : -1\text{Volts} \le f_u, f_y \le 1\text{Volts} \\ &f_p \in \mathbb{R} : 0 \text{ Nm} \le f_p \le 1 \text{ Nm} \end{aligned} \tag{12}$$

5.2 Simulation of DC Motor Benchmark

Simulations of the closed loop of the speed control system were made. In all test cases it was considered that the system is affected by a noise within 2$\%$ up to 8$\%$ of magnitude. The addition of noise has the objective to simulate more realistic conditions. Noise affecting the systems is one of the recognized causes of a wrong diagnosis and leads to the necessity of robust FDI methods. All implementations were made in MATLAB R2008a. The speed reference is 3000 rpm. This corresponds to 15 Volts.

5.3 FDI Proposal Based on Estimation of Faults with DE and DEwPC: DC Motor

Direct estimations of faults allow diagnosing the system. These estimations can be obtained by the solution of a minimization problem:

$$\begin{aligned} \min \ & F(\hat{f}) = \sum_{t=1}^{I} \left[U_{T(t)}(f) - \hat{U}T(t)(\hat{f}) \right]^2 \\ \text{s.t } \ & f_u, f_y \in \mathbb{R} : -1\text{Volts} \le f_u, f_y \le 1\text{Volts} \\ & f_p \in \mathbb{R} : 0 \text{ Nm} \le f_p \le 1 \text{ Nm} \end{aligned} \tag{13}$$

where $\hat{U}_{T(t)}(\hat{f})$ are computed based on the model from equation (11) and Laplace antitransform.

6 Experimental Methodology

With the aim of analyzing the merit of the diagnosis based on faults estimation with DE and DEwPC, three aspects were considered: robustness, sensitivity and computational cost.

With this goal in mind, many faulty situations were considered. For analyzing the characteristics of the diagnosis, the numerical experiments were divided into two parts:

— First Part. Robust Performance: for an analysis of robustness. Situations with multiple faults are considered and the output of the system is corrupted with up to 8$\%$ level noise. The faulty situations that were considered are shown in Table 5 (Cases 1 up to 4).

— Second Part. Sensitive Performance: for an analysis of sensitivity. Here we study the diagnosis of faulty situations that include simple and incipient faults (see Cases 6-8 in Table 5); as well as multiple and incipient faults (see Case 5 in Table 5). All measurements are corrupted with up to 8$\%$ level noise.

Table 5. Faulty situations in numerical experiments

Case	f_u	f_y	f_s
1	0.87	-0.2	0.53
2	-0.27	0.96	0
3	0.63	0	0.29
4	0	0.47	0.86
5	-0.08	0.09	0.2
6	0.15	0	0
7	0	-0.1	0
8	0	0	0.12

Abbreviations in tables are $\hat{f}_u, \hat{f}_p, \hat{f}_y$ for average of estimations of faults f_u, f_p, f_y, respectively. The computational effort of the algorithm is analyzed based on the number of objective function evaluations. Comparisons between the DE and DEwPC are based on the Wilcoxon's test [4, 8].

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-015

— Implementation of DE and DEwPC: the parameters were set following the recommendations in Table 2. The stopping criteria were $MaxIter = 100$ or $F(\hat{\theta}_f) \leq 0.1$.

7 Results

In Tables 6 and 7, the results of estimations of the faults described in cases from Table 5 are shown. In most cases the most accurate estimations are achieved by DEwPC. This was expected taking into consideration the results of a comparison between these two algorithms, see Section 3.

Table 6. Results of faults estimations with DEwPC

Case	$\hat{f}_u$	$\hat{f}_y$	$\hat{f}_s$
1	0.8644	-0.189	0.549
2	-0.299	0.957	0.0011
3	0.6304	0.020	0.3001
4	0.0016	0.4683	0.8675
5	-0.0801	0.092	0.204
6	0.150	0.006	0.0041
7	0.0004	-0.14	0.002
8	0.0005	-0.0051	0.1226

Table 7. Results of faults estimations with DE

Case	$\hat{f}_u$	$\hat{f}_y$	$\hat{f}_s$
1	0.8836	-0.167	0.5603
2	-0.2984	0.958	-0.07
3	0.6427	0.095	0.3834
4	0.0067	0.4551	0.8826
5	-0.084	0.085	0.199
6	0.1427	0.01	-0.0084
7	0.0017	-0.085	0.0059
8	0.007	0.0083	0.1102

On the other hand, in all cases for which one of the faults were not affecting the system (Cases 2-4, 6-8), the diagnosis provides false alarms (see estimations in Tables 6 and 7). The algorithms should estimate 0 values but in no cases this value was achieved. Instead, values such as $\left|\hat{f}_i\right| \leq 0.1$ were obtained but they are not small enough for deciding that these faults are not affecting the system. In such situations we adopted an alternate procedure:

— If when the algorithm reaches its stopping criteria, any of the faults has value $\left|\hat{f}_i\right| \leq 0.1$, then fix the other faults and execute the algorithm again, but only for estimating the faults such that $\left|\hat{f}_i\right| \leq 0.1$.

The results after applying the alternate procedure are shown in Table 8. It was applied only in DEwPC due to its better results during the other faults' estimations. The results indicate that estimations are closer to zero.

Table 8. Results after applying the alternate procedure

Case	$\hat{f}_u$	$\hat{f}_y$	$\hat{f}_s$
2	—-	0.00002	0.000107
3	0.00009	—-	0.0002
4	0.00005	-0.00001	—-
6	—–	0.00001	0.00009
7	0.0003	—–	0.00002
8	0.00001	-0.00006	—-

In order to compare the diagnosis taking the computational cost as a criterion, Figs 11 and 12 represent the average number of evaluations of the objective function. The results indicate that DEwPC executes a lower number of function evaluations than DE. This implies a lower processing time than DE.

8 Conclusions

This article presents an application of meta-heuristics to FDI, after formulating it as an optimization problem. Furthermore, we propose a new algorithm called DEwPC which was validated for continuous optimization problems. The validation of DEwPC for continuous optimization problems showed that DEwPC allows obtaining statistically better results than DE, especially in a noise environment or a complex search space. The last characteristic is desirable for FDI methods.

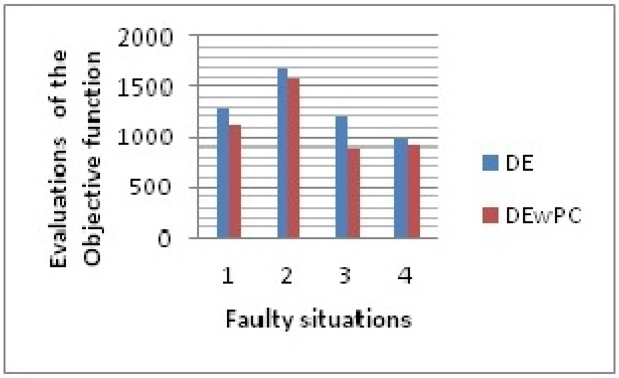


Fig. 11. Comparison between the numbers of evaluations of the objective function achieved by DE and DEwPC (Cases 1-4 from Table 5)

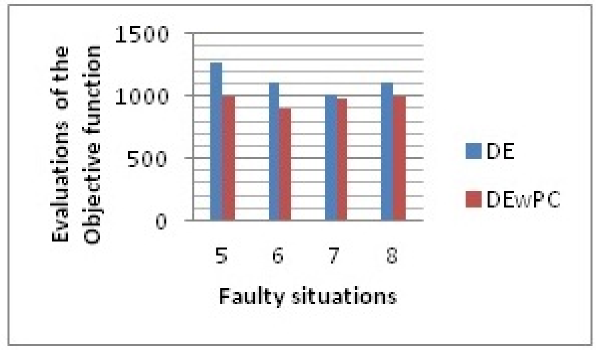


Fig. 12. Comparison between the numbers of evaluations of the objective function achieved by DE and DEwPC (Cases 5-8 from Table 5)

Another attractive property of DEwPC is that it does not introduce complex operations into the original DE framework. The only difference from the original DE is the introduction of a new Selection operator. Thus, it is also simple and easy to implement like the original DE. Therefore, DEwPC is as simple as DE, but allows decreasing the number of function evaluations for achieving a fixed accuracy.

The results of the numerical experiments with DC Motor benchmark indicate the suitability of meta-heuristics for obtaining robust and sensitive diagnosis. The use of meta-heuristics in FDI avoids spending efforts for generating robust residuals which are dependent on the kind of model that describes the system. The application of meta-heuristics and their hybridization or combination also allows decreasing the computational cost, which means a lower diagnosis time. This a desirable condition for online processes. In this case DEwPC achieved better results than DE.

Acknowledgements

The authors are grateful for the support provided by the Brazilian Agencies FAPERJ, Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro; CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico, CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior; and the MES/CUBA, Ministerio de Educación Superior de Cuba.

References

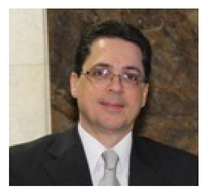
1. **Brest, S. G. B. B. M. M., J. & Zumer, V.** (**2006**). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(8), 646 – 657.
2. **Campos-Knupp, D., Silva-Neto, A. J., & Figuereido-Sacco, W.** (**2007**). Estimation of radiactive properties with the particle collision algorithm. In *Inverse Problems, Design and Optimization Symposium*. Miami, Florida, USA.
3. **Das, A. A. K. C. U., S. & Konar, A.** (**2009**). Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation*, 13(3), 526 –553.
4. **Derrac, J., García, S., Molina, D., & Herrera, F.** (**2011**). A practical tutorial on the use of nonparametric statistical test as a methodology for comparing evolutionary and swarm intellgence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3–18.
5. **Ding, S. X.** (**2008**). *Model- based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer.

6. **Figuereido-Sacco. W., C. R. E., Oliveira & Pereira, C. M. N. A.** (**2006**). Two stochastic optimization algorithms applied to nuclear reactor core design. *Progress in Nuclear Energy*, 48(6), 525 – 539.
7. **Frank, P. M.** (**1996**). Analytical and qualitative model- based fault diagnosis- a survey and some new results. *European Journal of Control*, 26(3), 459–474.
8. **García, S., Molina, D., Lozano, M., & Herrera, F.** (**2009**). A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC 2005 Special Session on Real Parameter Optimization. *J. Heuristics*, 15(6), 617– 644.
9. **Goldberg, D. E.** (**1989**). *Genetic Algorithms in search , optimization, and machine learning*. Reading, MA: Addison-Wesley.
10. **Gong, W., Cai, Z., & Ling, C. X.** (**2010**). DE / BBO a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Computing - A Fusion of Foundations, Methodologies and Applications Archive*, 15(4), 645 – 665.
11. **Isermann, R.** (**2005**). Model based fault detection and diagnosis. Status and applications. *Annual Reviews in Control*, 29(1), 71–85.
12. **Liu, Q. & Wenyuan, L.** (**2009**). The Study of Fault Diagnosis Based on Particle Swarm Optimization algorithm. *Computer and Information Science*, 2(2), 87 – 91.
13. **Lobato, F. S., Steffen, V., & Silva-Neto, A. J.** (**2009**). Solution of inverse radiative transfer problems in two-layer participating media with differential evolution. *Inverse Problems in Science and Engineering*, 18(2), 183–195.
14. **Metenidin, M. F., Witczak, M., & Korbicz, J.** (**2011**). A novel genetic programming approach to nonlinear system modelling: application to the DAMADICS benchmark problem. *Engineering Applications of Artificial Intelligence*, 24, 958 – 967.
15. **Mezura-Montes, E., Velázquez-Reyes, J., & Coello-Coello, C.** (**2006**). A comparative study of differential evolution variants for global optimization. In *GECCO 06, Seattle, Washington, USA*.
16. **Pacheco da Luz, E. F., Becceneri, J. C., & De Campos Velho, H. F.** (**2008**). A new multiparticle collision algorithm for optimization in a high-performance environments. *Journal of Computational Interdisciplinary Sciences*, 1(1), 3–10.
17. **Qin, V. L. H., A.K. & Suganthan, P. N.** (**2009**). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2), 398 – 417.
18. **Sacco, W. F. & Oliveira, C. R. E.** (**2005**). A new stochastic optimization algorithm based on particle collisions. In *2005 ANS Annual Meeting, Transactions of the American Nuclear Society*.
19. **Samanta, B. & Nataraj, C.** (**2009**). Use of particle swarm optimization for machinery fault detection. *Engineering Applications of Artificial Intelligence*, 22(2), 308 – 316.
20. **Silva-Neto, A. J. & Becceneri, J. C.** (**2011**). *Inteligênca Computacional Alicada a Problemas Inversos em Transferência Radiativa*. SBMAC.
21. **Simani, S., Fantuzzi, C., & Patton, R. J.** (**2002**). *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Springer-Verlag.
22. **Simani, S. & Patton, R. J.** (**2008**). Fault diagnosis of an industrial gas turbine prototype using a system identification approach. *Control Eng. Practice*, 16(7), 769–786.
23. **Storn, R. & Price, K.** (**1997**). Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341 – 359.
24. **Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y. P., Auger, A., & Tiwari, S.** (**2005**). Problem definitions and evaluation criteria for the cec 2005 special session on real parameter optimization. Technical report, Nanyang Technological University.
25. **Tvrdik, J.** (**2009**). Adaptation in differential evolution: A numerical comparison. *Applied Soft Computing*, 9(3), 1149 – 1155.
26. **Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N.** (**2003**). A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27(3), 293–311.
27. **Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N.** (**2003**). A review of process fault detection and diagnosis Part II: Qualitative model-based methods and search strategies. *Computers and Chemical Engineering*, 27(3), 313–326.

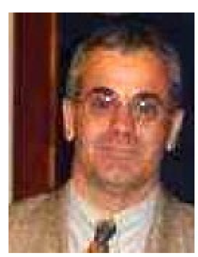
28. **Wang, L., Niu, Q., & Fei, M.** (**2008**). A novel quantum ant colony optimization algorithm and its application to fault diagnosis. *Transactions of the Institute of Measurement and Control*, 30(3/4), 313–329.
29. **Witczak, M.** (**2006**). Advances in model based fault diagnosis with evolutionary algorithms and neural networks. *Int. J. Appl. Math. Comput. Sci.*, 16(1), 85–99.
30. **Witczak, M.** (**2007**). *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems From Analytical to Soft Computing Approaches*. Springer.
31. **Yang, E., Xiang, H., Guand, D., & Zhang, Z.** (**2007**). A comparative study of genetic algorithm parameters for the inverse problem-based fault diagnosis of liquid rocket propulsion systems. *International Journal of Automation and Computing*, 4(3), 255–261.
32. **Yang, K. T., Z. & Yao, X.** (**2008**). Self-adaptive differential evolution with neighborhood search. In *IEEE Congress on Evolutionary Computation (CEC2008)*. Hong Kong, 1110 – 1116.
33. **Zaharie, D.** (**2009**). Influence of crossover on the behavior of differential evolution algorithms. *Applied Soft Computing*, 9(3), 1126–1138.
34. **Zhang, J.** (**2009**). Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5), 945–958.

Lídice Camps Echevarría received her Ph.D. in 2012. She is a Professor at the Superior Polytechnic Institute José Antonio Echeverría, Cujae, in Havana, Cuba. Her research interests are meta-heuristics, inverse problems and fault diagnosis.

Orestes Llanes Santiago received his Ph.D. from the University of Los Andes in 1994. He is a Full Professor at the Superior Polytechnic Institute José Antonio Echeverría (Cujae), a Titular Member of the Cuban National Academy of Sciences, Coordinator of the Cuban Network for Automatic Control, and a Member of the National Commission for Scientific Degrees in Automatic Control and Computer Sciences. His research interests are fault diagnosis, non-linear control and intelligent control.



Antônio J. Silva Neto received his Ph.D. from North Carolina State University in 1993. He is now a Full Professor at the State University of Rio de Janeiro, President of the Brazilian Society of Mechanical Sciences and Engineering (ABCM), and Vice-President of the Brazilian Society of Computational and Applied Mathematics (SBMAC).



Haroldo Fraga de Campos de Velho received his D.Sc. in Mechanical Engineering from the Federal University of Rio Grande do Sul - UFRGS, Brazil, in 1992. Currently, he is a Senior Researcher at the National Institute for Space Research (INPE), Associated Laboratory for Computing and Applied Mathematics (LAC). He was the Associate Director for Space and Environment of the INPE (March-2008 up to November-2010).

Article received on 03/04/2013, accepted on 27/06/2013.