CEAI, Vol. 8, No. 1, pp. 59-69, 2006 Printed in Romania

CONVENTIONAL AND NON-CONVENTIONAL METHODS FOR NONLINEAR MULTI OBJECTIVE PREDICTIVE CONTROL

Faouzi Bouani [1], Kaouther Laabidi [2], Mekki Ksouri [3]

[1] National Institute of Applied Sciences and Technology, Tunis, Tunisia.
[2] High Institute of Applied Sciences and Technology, Mateur, Tunisia.
[3] National Institute of Engineering of Tunis, Tunis, Tunisia.
e-mail adresses: Faouzi.Bouani@insat.rnu.tn, Kaouther.Laabidi@enit.rnu.tn , mekki.ksouri@mrstdc.gov.tn

Abstract: *This paper describes constrained multi objective predictive control of nonlinear systems. A nonlinear model based on the Artificial Neural Networks (ANNs) is used to characterize the process at each operating point. The control law is provided by minimizing a set of control objective which is function of the future prediction output and the future control actions. Three aggregative methods are used to compute the control law. The first and the second methods are non-conventional methods based on Genetic Algorithms (GAs) and the third method is a conventional method which is a combination between the weighted sum method and the ellipsoid algorithm. The proposed control scheme is applied to a numerical example to illustrate the performance of the proposed predictive controller.*

Keywords: *Non linear predictive control, Genetic algorithms, Neural networks, Multi objective optimization, Multi model.*

1. INTRODUCTION

The Model Based Predictive Control (MBPC) is a general control methodology based on the explicit use of a model to predict the process output over a specified predictive horizon [1]. The objective of the MBPC strategy is to compute the future incremental control sequence by minimizing a given single objective performance criterion, under given operating constraints [2], [3].

The complexity of industrial processes makes difficult their representation by only one model [4], [5]. For a system that presents several modes of working different models can be built, which are specific to every mode of particular working of the system, this permits to structure the priori qualitative and/or quantitative knowledge of the studied system.

In presence of a set of models, two approaches can be used to calculate the optimal control law. The first is the multi-model control approach which is based on the determination of models

validities and the control law is computed by the fusion of local controls [6], [7]. Each local control is obtained by minimizing a single criterion. Although, the closed loop performances of this control strategy depend on the precision of the estimated coefficients validities. The second approach is the multi criteria control strategy based on the minimization of all performance criteria at every sample time [8], [9]. This strategy of control leads to a set of optimal solutions, i.e. the Pareto optimal solutions or the non dominated solutions [10].

Genetic algorithms are stochastic search techniques used in the field of optimisation because they are more robust compared to conventional optimisation techniques [11]. The main difficulties of the multi-objective optimization methods lie in the guidance of a research process towards the Pareto surface and the maintenance of a diversity of the solutions to assure a good distribution on the Pareto border.

Methods of multi-objective optimization are divided into two groups [10], [12]. The first group includes the non aggregative methods. In these methods, there is no fusion of the different objective functions, which does not lead to a mono-objective optimization. The second group includes approaches based on the transformation of the problem into a mono-objective one.

In [13], a GA for multi objective optimization problems that appears in the design of robust controllers is presented. In [14], the optimal feedback control design is formulated as a multiple criteria problem. Then, the use of weighting matrices is avoided. The nonlinear predictive control using GA as optimizer has been studied by many authors, but the selection mechanisms of the GA in these applications work on a single valued scalar fitness function [3], [15].

In this work, we propose the constrained multi objective predictive control of nonlinear systems. The particularities of this work consist in (i) the use of a nonlinear model i.e. the neural networks model to characterize the operating point of the plant and (ii) the goal of the control system design task is thus redefined as finding a control input such that the set of Pareto optimal solutions minimizes all performance criteria simultaneously in tacking in account constraints in input signal. Since local models are nonlinear, the performance criterion is non convex in the controller parameters and can not be efficiently solved by conventional methods.

In this work, three methods will be considered to solve the constrained multi criteria nonlinear predictive control. Two methods are based on GAs and the third method is the weighted sum method.

The paper is organized as follows. The problem is formulated in section two where Neural Networks models are determined and the predictive control principle is given. The control and design are presented in section three. The GA optimizer used in multi criteria optimization is performed. The obtained results are presented in section four. Conclusions are given in the last section.

2. PROBLEM FORMULATION

2.1. Modeling of non linear systems

We consider a Single Input Single Output (SISO) nonlinear system that presents (n) modes of working. The system is characterized by the following relation:

$$y(k)=g_i\left(y(k-1),\ldots,y(k-m_1),u(k-1),\ldots,u(k-m_2)\right) \quad (1)$$

where g_i (i=1, …, n) are unknown nonlinear functions, $y(k)$ and $u(k)$ represent, respectively, the output and the input of the plant. Integers m_1 and m_2 define the range of delays in the output and the input signals, respectively. Artificial Neural Networks are an efficient tool used to characterize non linear systems [16], [17], [18]. In this work, the multilayer perceptron is used to approximate nonlinear functions g_i (i=1, …, n).

For each operating point of the system, an (ANNs) local model is determined, which output is given as follows:

$$y_{m_i}(k)=NN(\theta_i) \quad (2)$$

where θ_i (i=1, …., n) represent the neural network model parameters (weights), and n is the number of neural network models.

2.2. Performance criterion

The Model Based Predictive Control (MBPC) is a general control methodology based on the explicit use of a model to predict the process output over a specified predictive horizon [2]. The objective of the MBPC strategy is to

compute the future incremental control sequence $\Delta U=[\Delta u(k)\ldots\Delta u(k+N_u-1)]$ (N_u is the control horizon) by minimizing a given objective performance criterion, under given operating constraints [2]:

$$J_i=\sum_{j=1}^{N_y}\left(y_c(k+j)-y_{m_i}(k+j/k)\right)^2+l\sum_{j=1}^{N_u}\left(\Delta u(k+j-1)\right)^2 \quad (3)$$

where N_y is the prediction horizon, N_u is the control horizon ($N_u \le N_y$), l is the control weighting factor, $y_c(k)$ is the reference signal and $y_{m_i}(k+j/k)$ is the j-step ahead predictor computed with the local model $NN(q_i)$, $\Delta u(k+j-1)$ is the future control increments sequence and $\Delta u(k+j)=0$, if $j \ge N_u$.
Constraints, which limit the range of the control signal and the gradient of the control signal, are defined as follows [2]:

$$\begin{cases} u_{\min} \le u(k+j) \le u_{\max}, \\ \Delta u_{\min} \le \Delta u(k+j) \le \Delta u_{\max}, \\ \forall\ j=0,\ldots\ldots,N_u-1. \end{cases} \quad (4)$$

These constraints can be reformulated as follows:

$$\Omega=\left\{\Delta U / f_j(\Delta U)\le 0\right\},\ j=1,\ldots,4N_u \quad (5)$$

where $\Delta U=[\Delta u(k),\ldots,\Delta u(k+N_u-1)]$ is a N_u dimensional vector of parameter.

2.3. Multi objective predictive control

The multi objective problem seeks to minimize n objectives J_i, i=1, …, n, where each objective depends on ΔU. The multi criteria optimization problem can be described in most cases under the following formulation [10].

$$\min_{\Delta U\in\Omega}(J_1,J_2,\ldots,J_n) \quad (6)$$

where Ω is the set of constraints given by relation (5).

The multi criteria controller structure is given by figure 1. The Neural Networks Models $NN(q_i)$ are used to predict the system output over the prediction horizon. The Neural Networks Models are applied for the recursive prediction of the future process outputs as given in figure 2. The control law sequence is computed by minimizing the multi performance criteria. Since local models are non linear, the criterion J_i is non convex in the controller parameters and can not be efficiently solved by conventional methods.

The multi objective optimization leads to a set of solutions, ie: the Pareto set, so the problem is which solution can be applied to the process. Consider a two performance criteria, $J(\Delta U)=\min_{\Delta U\in\Omega}(J_1,J_2)$ and a set $\Omega=\{\Delta U_1,\Delta U_2,\Delta U_3,\Delta U_4,\Delta U_5\}$ whose function values are shown in figure 3. The vectors $\Delta U_1,\Delta U_2,\Delta U_3$ and ΔU_4 are Pareto optimal because no solution in Ω dominates them. However, ΔU_5 is dominated by ΔU_3 because $J_1(\Delta U_3)< J_1(\Delta U_5)$ and $J_2(\Delta U_3)< J_2(\Delta U_5)$.
For each solution, we can compute the following norm:

$$d_i=\sqrt{\left(J_1(\Delta U_i)\right)^2+\ldots+\left(J_n(\Delta U_i)\right)^2} \quad (7)$$

The best solution, from the non dominated solutions of the Pareto set, is the solution that gives the minimal norm d_i. Consequently, the optimal control is selected based on the norms d_i of all elements of the Pareto set.

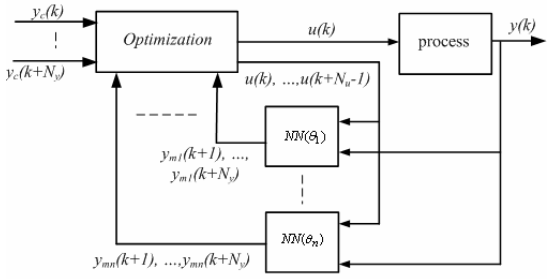

Fig. 1. Multi criteria controller structure.

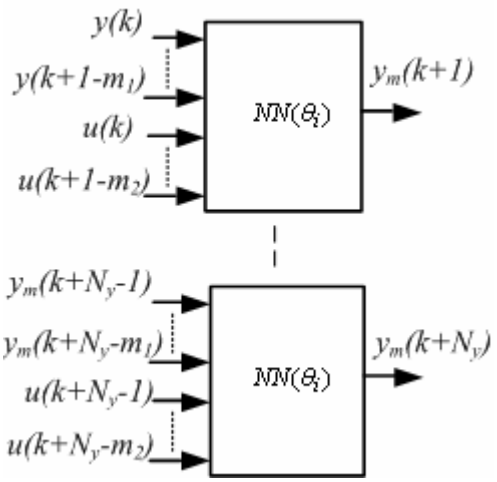

Fig. 2. The neural network predictor.

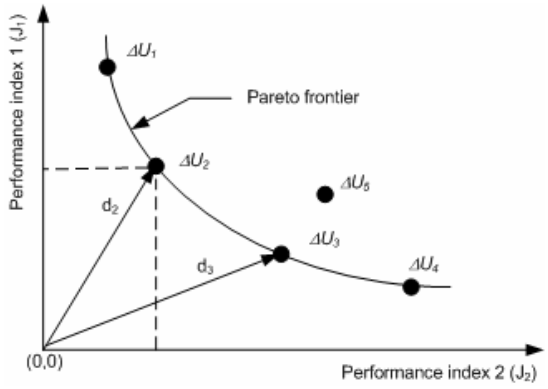

Fig. 3. The Pareto optimal set.

3. CONTROL AND DESIGN

In our work, we compare three methods to solve a constrained multi objective nonlinear predictive control. Two methods based on GAs and a method based on the hybridization between the weighted sum method and a numerical algorithm of optimization such that the ellipsoid algorithm.

3.1. Weighted sum method

It is one of the first methods used for the generation of Pareto optimal solutions [10]. It is based on the principle of combination of the various performance criteria in a single criterion. The single criterion is obtained by the sum of the weighted criteria. The n objective functions are aggregated into one criterion, as follows [10]:

$$J = \sum_{i=1}^{n} w_i J_i, \quad 0 \le w_i \le 1 \tag{8}$$

where the weights are such that:

$$\sum_{i=1}^{n} w_i = 1 \tag{9}$$

At each sample time, a set of weights (w_1, …, w_n) that respect the constraint given by relation (9) is generated. For every set of weights, the performance criterion given by relation (8) is minimized in taking in account constraints on input signal as given by relation (4). The optimal solution is computed by using a numeric method i.g. the ellipsoid method [19], [20]. The set of weights is modified and the procedure is repeated. Consequently, the points of the Pareto set are determined. The size of the Pareto set depends on the number of time that the procedure is executed. The steps of the algorithm which allows the determination of the Pareto set are given below.

1- Give N_y, N_u and l . Put $k=1$.
2- Compute the system output,
3- Introduce the weights (w_1, …, w_n) that respect the relation (9),
4- Compute the solution of the constraints problem by using for example the ellipsoid method.
5- Compute the norms d_i using relation (7).
6- Return to step 3,
7- Take the best vector ΔU of the future control increments which assures the minimal norm d_i,
8- Compute the control from the first element of the vector ΔU . Increment the sample time k and return to step 2.

3.2. GAs solution technique

The genetic algorithm is a method which may be used to solve a system of non linear equations, their most important application is in the field of optimization because of their ability to search efficiently in search spaces, which makes them more robust compared to the conventional optimization techniques.

3.2.1. Genetic algorithms:

Genetic algorithms (GAs) are stochastic search techniques inspired by the principles of natural

selection and natural genetics [11]. In genetic algorithms, each parameter is represented by a string structure. This is similar to the chromosome structure in natural genes. A group of strings are called population. It should be notice that GAs evaluate a set of solutions in the population at each iteration step. The Real Coded Genetic Algorithm (RCGA) is based on the same operators as binary coded GA. However, the individuals in RCGA are represented with real code. It has been also demonstrated that the (RCGA) worked well for optimisation functions compared to binary coding [21], [22]. In this work, we propose the RCGA to find the optimal control law. A number of genetic operators are available to generate new individuals in next generation.

- *Selection:* the selection operator produces copies of elements from actual population according to the selection probability of each element. In this work, we have considered the roulette wheel to select individuals for the new population [11].

- *Crossover:* The crossover includes the goodness of the parents for generating the offspring based on the crossover probability (c_p). The new elements can be computed by several crossover mechanisms, we note flat crossover, a simple crossover, arithmetical crossover and BLX-a crossover [23]. With the crossover operation, genetic algorithms are able to acquire more information with the generated individuals. The genetic search space is thus extended and more complete.

- *Mutation:* The mutation consists to modify some alleles of each element with the mutation probability (m_p).

The genetic algorithm has been correctly implemented if the population has been evolving over successive generations so that the fitness of the best and the average in each generation increase towards the global optimum.

With GAs, constraints have been handled mainly in two ways. The first way consists of using additive penalty functions which are known to be very problem dependent and, thus, difficult to set [24]. The second way is found in embedding constraints in the coding of the chromosomes [15]. The idea is based on tacking the population individuals while respecting constraints. First, the set of possible changes of the gradient of the control is defined in tacking in account the relation (4), as follows [15]:

$$\begin{cases} \Delta u_{\sup}(j) = \min[u_{\max} - u(k+j-2), \Delta u_{\max}] \\ \Delta u_{\inf}(j) = \min[u(k+j-2) - u_{\min}, \Delta u_{\min}] \\ \forall\ j = 1,....., N_u. \end{cases} \quad (10)$$

Second, the set of possible changes of the gradient of the control is discredited in ($2h$) values as given by the following relations:

$$S_{\inf} = \left\{ \Delta u_{\inf}(j) \frac{h-i}{h},\ i = 0,1,...,h-1 \right\}. \quad (11)$$

$$S_{\sup} = \left\{ \Delta u_{\sup}(j) \frac{h-i}{h},\ i = 0,1,...,h-1 \right\}. \quad (12)$$

The initial population of the genetic algorithm is then formed by $(N_u, 2h+1)$ values:

$$P = \left\{ S_{\inf}, 0, S_{\sup} \right\}. \quad (13)$$

Genetic algorithm operators are also defined so that the new individuals respect constraints in input signal. The new individuals (ind_1, ind_2) of the RCGA crossover are computed from the selected parents (p_1, p_2) using the arithmetic crossover [23]:

$$ind_1 = a\, p_1 + (1-a)\, p_2, \quad (14)$$

$$ind_2 = (1-a)\, p_1 + a\, p_2,\ 0<a<1. \quad (15)$$

The result of real coded mutation is given by the following relation:

$$ind_i = a_m(ind_i),\ i=1,\ 2 \text{ and } 0<a_m<1. \quad (16)$$

In order to reduce the error in permanent mode between the process output and the predicted output, we use the following relation [3]:

$$d(k) = je(k) + (1-j)\, d(k-1) \quad (17)$$

where $e(k) = y(k) - y_{m_i}(k)$ and $j \in [0,1]$.

The filtered discrepancy $d(k)$ is then used for the correction of the predictions of the model.

3.2.2. The Non Dominated Sorting Genetic Algorithm (NSGA):

Genetic algorithms are adapted very well to the treatment of a multiobjective optimization problem [25]. The NSGA algorithm is based on the classification of the individuals in categories according to the concept of Pareto set dominance [26]. All non dominated individuals of the population are assigned on rank 1. The remaining individuals are classified and non dominated points are assigned on rank 2. The procedure of classification is finished when all individuals are assigned [10].
The efficiency (ef_i) is calculated by the following relations:

$$ef_i = \frac{F}{\sum_{j=1}^{n_r} h(i,j)} \tag{18}$$

$$h(i,j) = \begin{cases} 1 - \left[\frac{d(i,j)}{\sigma}\right]^2 & if\ d(i,j) < \sigma \\ 0 & otherwise \end{cases} \tag{19}$$

where n_r is the number of individuals in the considered category, s is an initial Euclidean distance, d is the euclidean distance between individuals i and j and F is the inverse of every individual rank.

3.2.3. The Weighted Average Ranking Genetic Algorithm (WARGA):

This method ensues from the method "Multiple Objective Genetic Algorithm". The main difference lies in the way the relation of dominance is established between two solutions. With the WARGA, the relation of dominance is given as explained by the following example. Consider n objective functions to minimize $\{J_1,...,J_n\}$ and a discrete search set $\Omega = \{\Delta U_1,...,\Delta U_m\}$. Let B is a vector formed by the values of the objective functions at the solution ΔU_1 and S is a matrix that regroups the values of the objective functions at solutions $\Delta U_2,...,\Delta U_m$.

$$B = \{J_1(\Delta U_1),...,J_n(\Delta U_1)\} = \{b_1,...,b_n\} \tag{20}$$

$$S = \begin{Bmatrix} J_1(\Delta U_2)\dots J_n(\Delta U_2) \\ \dots \\ J_1(\Delta U_m)\dots J_n(\Delta U_m) \end{Bmatrix} \tag{21}$$

The following procedure permits the computation of the rank of each solution.

Step 1) - Initialize the counter of individuals, $i=1$.
Step 2) -Repeat
- Initialize the counter of functions, $j=1$.
Step 3) Repeat
- Take the j^{th} element b_j of the vector B.
- Determine the number of solutions (NS_j) of the j^{th} column of S better than b_j.
- Incrementation of j ($j=j+1$).
Until $j>n$.
Step 4) - Compute the rank of ΔU_i:

$$rank_i = \sum_{j=1}^{n} NS_j \tag{22}$$

Step 5) - Permutation between B and the i^{th} row of S.
Step 6) - Incrementation of i ($i=i+1$).
Until $i>m$.

After the determination of the rank of each individual, we compute the efficiency according to the following relation:

$$ef_i = 1 - \frac{1}{m} rank_i \tag{23}$$

The GAs approach is resumed by the following steps [27]:

Step 1) - Create an initial population according to the relation (13),
Step 2) - Evaluation of objective functions,
Step 3) - Assigned a rank for every individual in the population,
Step 4) - Compute the individual efficiency (ef_i),
Step 5)
Repeat
- selection proportional to the efficiency,
- crossover,
- mutation,
- evaluation of objective functions,
- assigned a rank for every individual,
- calculate the individual efficiency (ef_i).
Until a termination condition is reached.

The termination condition is used to stop the genetic algorithm runs. There are some criteria which can be used to terminate calculations. In this work, the termination condition is the pre-assigned number of generations to be considered.

4. SIMULATION RESULTS

In order to evaluate closed loop performances obtained by the proposed approach for non linear multi objective predictive control, we consider a SISO plant with two operating points. In the first operating point, the plant is described by the following relation [28]:

$$y(k)=\frac{y(k-1)y(k-2)y(k-3)u(k-2)(y(k-3)-1)}{1+y^2(k-1)+y^2(k-3)} + \frac{u(k-1)}{1+y^2(k-1)+y^2(k-3)}, \qquad 0\le k\le 400 \quad (24)$$

In the second operating point, the plant is characterized by the following non linear model [29]:

$$y(k)=0.2y(k-1)\left(y(k-1)+1\right)+1.2u(k-1) +0.4\sin\left(0.5y(k-1)+0.5y(k-2)\right). \cos\left(0.5y(k-1)+0.5y(k-2)\right), \quad k>400 \quad (25)$$

We suppose that the process is described by the relation (24) in the 400 first iterations then the behaviour of the process changes and it will be described by the relation (25) for remaining iterations.

4.1. Identification

Some practical problems have to be solved for training the NN models. First of all, the training patterns have to be carefully selected [17]. The size and the structure of the NN are also of practical importance. Large networks consume a great deal of computing power. Small nets are, in some cases, unable to provide enough approximation capabilities, producing large approximation errors. Trial and error is normally necessary to fix the structure of the NN model. A set of tests have been carried out with 1000 samples and 100 epochs, in order to fix the number of hidden layer neurons and the number of inputs of the NN model. We have found that neural networks model formed by one hidden layer with 10 neurons and five inputs {*y(k-1), y(k-2), y(k-3), u(k-1), u(k-2)*} assures a low performance criterion in the training phase. The activation function is the tangent sigmoid function. The training rate of the back propagation algorithm is equal to 0.08. In Figure 4 is shown the evolution of the training criterion versus the number of the neural networks inputs.

4.2. Control

The control signal is limited between 0 and 1. The gradient of the control $\Delta u_{\min}$ and Δu_{max} are taken respectively equal to -0.01 and 0.01. The prediction horizon N_y, the control horizon N_u, and the control weighting factor l are fixed, respectively, to 5, 2 and 0.

4.2.1. Nonlinear predictive control based on weighted sum method:

Figure 5 presents the evolutions of the output, the set point and the control signals. The first model is considered for the first 400 iterations and the second model is considered for the iterations that remain. In this simulation, the ellipsoid method is used with an initial ellipse $A=\begin{bmatrix}10 & 0\\ 0 & 10\end{bmatrix}$ and an initial vector $\Delta U=[0.020.02]$. The stopping criterion of the algorithm is $e=10^{-5}$. It can be observed from this figure that the controller provides the suitable control signal allowing the output to track the set point in spite of the change of the plant operating point after the iteration 400.

The Pareto surface obtained with the weighted sum method is shown in Figure 6. We note from this figure that the solutions are not uniformly distributed on the Pareto surface because the performance criteria are non convex.

4.2.2. Nonlinear predictive control based on GA:

The choice of *h=15* in relations (11) and (12), leads to a GA population size equal to 31; the crossover probability and the mutation probability are fixed respectively to $c_p=0.7$; and $m_p=0.8$. The Euclidean distance s in the NSGA is fixed to 0.2.

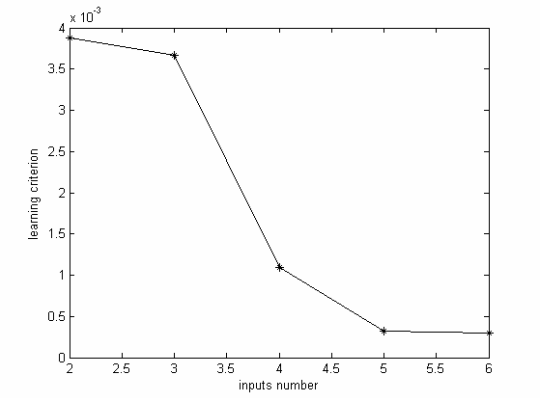

Fig. 4. Evolution of the training criterion versus neural network inputs

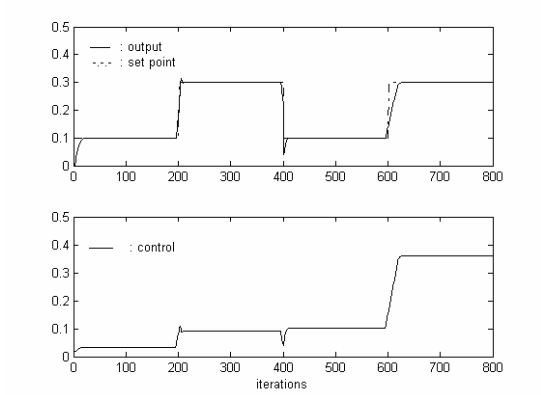

Fig. 5. Evolution of the set point, output and the control

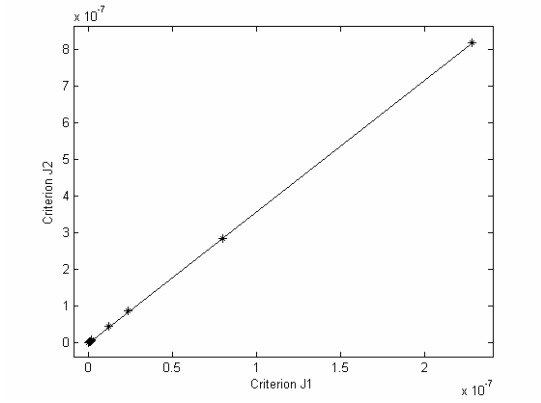

Fig. 6. Pareto surface after 100 measures

In order to compare the Pareto surface obtained by the NSGA and the WARGA, we have considered a population with 10 individuals.
The Pareto surfaces obtained by the NSGA and the WARGA with only one generation are shown respectively in figures 7 and 8. The Pareto surfaces obtained by the NSGA and the WARGA with 10, 30, 50 and 100 generations are shown respectively in figures 9 and 10. It's clear from these figures that non dominated solutions are not uniformly distributed on this surface because the performance criteria J_1 and J_2 are non convex.
It can be observed, from figures 7 and 8, that the GA approach can leads to a low performance criteria by using only one generation. The increase of the maximal number of generations, at each simple time, leads to the closeness of the solutions to the ideal solution. Therefore, many points in the Pareto optimal set are replicated.
Evolution of the set point, the output and the control signals obtained with the NSGA and the WARGA are shown respectively in figure 11 and 12. In these cases, thirty generations are used at each sampling period time to carry out the optimal control.
The Number of non dominated solutions versus generations obtained with the NSGA and the WARGA is given in figure 13. It can be observed that the number of non dominated solutions increases as the number of generations increases. One can see also from figure 13, that the WARGA method has out performed the NSGA method in terms of the number of non dominated solutions at each generation.

5. CONCLUSION

This paper described the predictive control of nonlinear systems. The system is represented by a set of non linear models, where each model corresponds to a possible operating point of the system. The control law is computed by optimizing a set of performance criteria. Each criterion is given in taking into account the local ANNs model and the input constraints. The comparative study of three methods used to optimize the multi objective problem, has demonstrated the superiority of the WARGA method. Indeed, non dominated solutions located in concave regions of the trade-off surface cannot be obtained by the weighted sum method, because their cost is sub-optimal, and the NSGA method can be particularly sensitive to the setting of the s parameter, which depends on the problem. The WARGA method can be easily implemented and it can find non dominated solutions in a single run. Moreover, one can use a non standard cost function because the GA optimizers don't require the calculation of the derivative of the performance criteria.

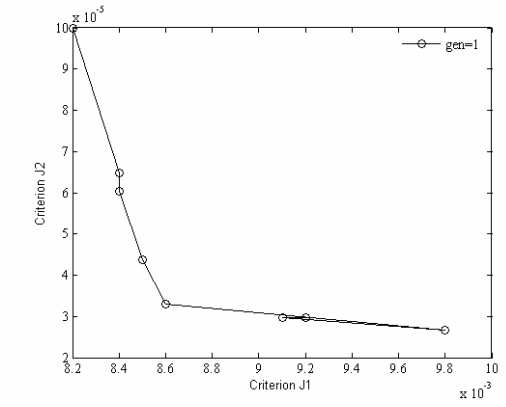

Fig. 7. Pareto surface obtained by the NSGA

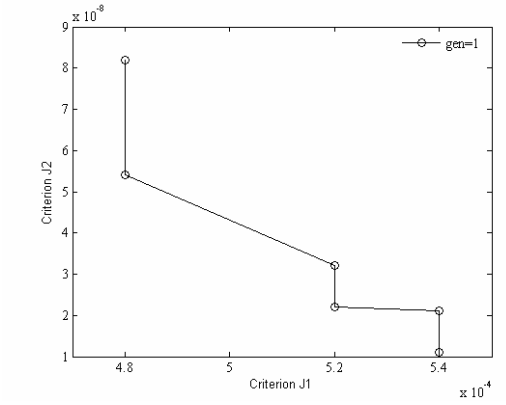

Fig. 8. Pareto surface obtained by the WARGA

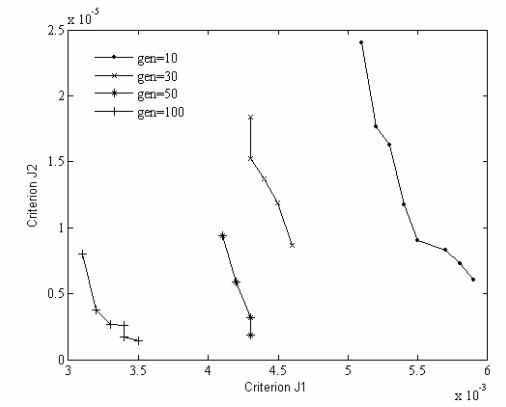

Fig. 9. Pareto surface obtained by the (NSGA)

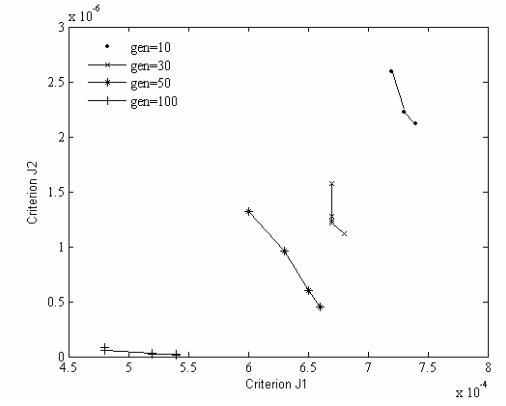

Fig. 10. Pareto surface obtained by the (WARGA)

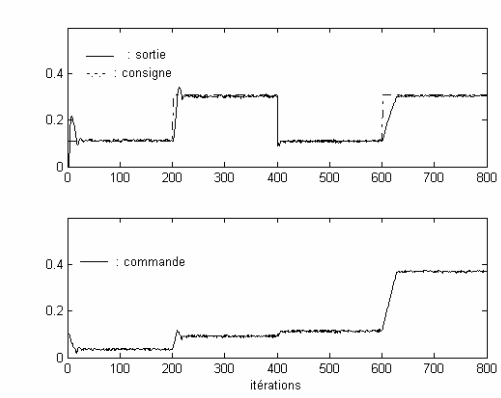

Fig. 11. Evolution of the set point, output and the control (NSGA)

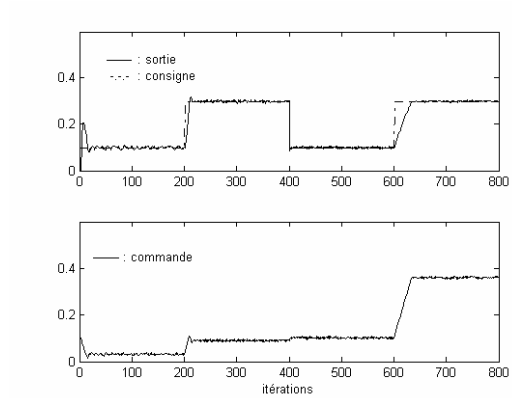

Fig. 12. Evolution of the set point, output and the control (WARGA)

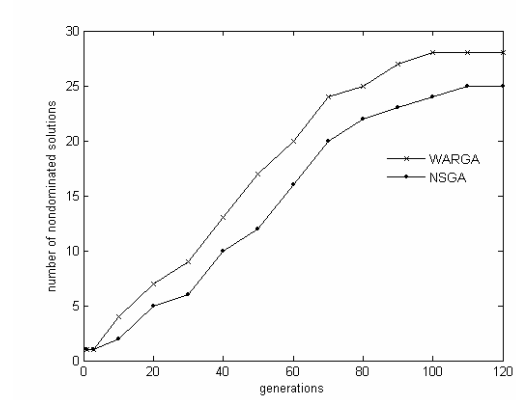

Fig. 13. Number of non dominated solutions versus generations

REFERENCES

[1] Richalet J., “Industrial applications of model based predictive control”, Automatica, Vol. 29, pp. 1251-1274, 1993.

[2] Martin P., and E. S. Dale, "A nonlinear predictive control strategy based on radial basis function models”, Computers chem. Engng, Vol. 21, No. 9, pp. 965-980, 1997.

[3] Primoz P. and G. Igor, “Nonlinear model predictive control of a cutting process”, Neurocomputing, Vol. 43, pp. 107-126, 2002.

[4] Dougherty D. and D. Cooper, “A practical multiple model adaptive strategy for single-loop MPC”, Control engineering practice, vol. 11, pp 141-159, 2003.

[5] C.R. Porfirio, E. Almeida Neto, and D. Odloak, “Multi model predictive control of an industrial C3/C4 splitter”, Control engineering practice, vol 11, pp 765-779, 2003.

[6] F. Delmotte., “Analyse Multi-Modèle”. Thèse de doctorat, université de lille, lille 1997.

[7] M. Ksouri-Lahmari, “Contributions à la commande multimodèle des processus complexes”. Thèse de doctorat, université de lille, lille 1999.

[8] C. A. Coello Coello, D. A. Van Veldhuizen and G. B. Lamont, “Evolutionary Algorithms for Solving Multi-Objective Problems”, Kluwer Academic Publishers, New York, March 2002.

[9] C. M. Silvia, E. C. Biscaia, and J. r. "Genetic algorithm development for multiobjective optimization of batch free radical polymerization reactors", Computers and Chemical Engineering, Vol. 27, pp. 1329-1344, 2003.

[10] Y. Collette and P. Siarry, "Optimisation multiobjectif", Editions Eyrolles, 2002.

[11] D. E. Goldberg, "Genetic Algorithms in search, optimization and machine learning", Addison-Wesley, Massachusetts, 1991.

[12] EL. G. Talbi, “Métaheurstiques pour l’optimisation combinatoire multi-objectif : Etat de l’art", Rapport technique, Université des Sciences et Technologies de Lille, 2001, http://www.lifl.fr~talbi/paper.html.

[13] A. Herrors, E. Baeyens and J. R. Peran, “MRCD: A genetic algorithm for multiobjective robust control design”, Engineering Applications of Artificial Intelligence, vol. 15, pp. 285-301, 2002.

[14] K. Sourav and K. Seiichi, "Genetic Algorithms for Optimal Feedback Control Design", Engineering Applications of Artificial Intelligence, vol. 9, No. 4, pp. 403-411, 1996.

[15] C. Onnen, R. Babuska, U. Kaymak, J. M. Soussa, H.B. Verbruggen, and R. Isermann, “Genetic Algorithms for Optimization in Predictive Control”, Control Engineering Practice, Vol. 5, N° 10, 1997, pp. 1363-1372.

[16] K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop, "Neural networks for control systems - a survey", Automatica, Vol. 28, no. 6, pp. 1083-1112, 1992.

[17] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks”. Part II: Observability, identification, and control, IEEE Trans. Neural Networks, Vol. 7, No. 1, pp. 30-42, 1996.

[18] O. Nelles, “Nonlinear system identification: From classical approaches to neural networks and fuzzy models”. Springer-Verlag, 2001.

[19] S. Boyd, L. EL Ghaoui, E. Feron and V. Balakrishnan, “Linear matrix inequalities in systems and control theory”, Studies In Applied Mathematics, Philadelphia, 1994.

[20] S. Shah, J. E. Mitchell and M. Kupferschmid,"An ellipsoid algorithm for equality-constrained nonlinear programming", Computers and Operations Research, Vol. 28, No. 1, pp. 85-92, 2001.

[21] K. Laabidi, B. Faouzi, R. Ben abdennour and M. Ksouri, "Binary and real coded genetic algorithm for predictive control", IMACS multiconference on Computational Engineering in Systems Applications (CESA’2003), Lille, France, July 9-11, 2003.

[22] R. Sarker, T. P. Runarsson, and C. Newton, “Evolutionary computation and constrained optimization”. In New frontiers in computational intelligence and its applications, IOS Press, Amsterdam, pp. 144-157, 2000.

[23] F. Herrera, M. Lozano, and J. L. Vergegay, “Tackling real coded genetic algorithms: operators and tools for behavioural

analysis", Proc. Congress on intelligent techniques and soft computing , pp. 1-55, 1996.

[24] J. T. Richardson, Palmer, M. R., liepins, G., and Hilliard, M., "Some guidelines for genetic algorithms with penalty functions", In Proceedings of the Third International Conference on Genetic algorithms, Morgan Kaufmann, San Mateo, CA, pp. 191-197., 1989.

[25] D. F. Jones, S. K. Mirrazavi and M. Tamiz, "Multi-objective meta-heuristics: An overview of the current state-of-the-art". European Journal of Operational Research, Vol. 137, pp. 1-9, 2002.

[26] N. Srinivas and K. Deb, "Multiobjective optimization using Nondominated sorting in genetic algorithms", Evolutionary Computation, Vol. 2, No. 3, pp. 221-248, 1994.

[27] K. Laabidi, F. Bouani, "Genetic Algorithms for Multiobjective Predictive Control", Proc. of IEEE International Symposium on Control, Communications and Signal Processing ISCCSP'04, March 21-23, 2004, Hammamet, Tunisia.

[28] H. Khabou, H. Trabelsi and M. Chtourou, "Sur la commande multimodéle neuronale", Conférence Tunisienne en Génie Electrique CTGE, pp. 132-137, 2004.

[29] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control", IEEE Trans. Neural Networks, Vol. 6, No. 1, pp. 144-156, 1995.