"ELECO'99 INTERNATIONAL CONFERENCE ON ELECTRICAL AND ELECTRONICS ENGINEERING"

E02.47/D-04

OPTIMIZED CONTROL OF PARALYSIS AND DEPTH OF UNCONSCIOUS IN ANESTHESIA

Barkat Aicha & Benmahammed Khier

Department of Electronic
University of Ferhat Abbes, Setif
19000 Setif, Algeria
e.mail : BenKhier @ hotmail.com

Abstract:

This paper investigates an objective approach to the design of fuzzy controllers. The approach uses an improving genetic algorithm, a recent search and optimization technique to optimize the parameters of supervised fuzzy controller. Such parameters as the membership functions and the rule base can not be objectively elicited in many cases. A supervised controller is a controller which can guarantee that the state of system is uniformly bounded, so it guarantee stability and that we can release it when we determine the objective function of genetic algorithm. To achieve this design, we have applied the approach to a multivariable problem for a nonlinear biomedical process, namely, control of anesthesia. The simulation results are included to demonstrate that incorporating the control engineering methods in biomedicine results in superior accurate supervising performance.

Key words:
Fuzzy logic, Fuzzy logic control, Genetic algorithms, Biomedical control, Anesthesia

1. Introduction

Several studies have shown fuzzy logic control [1], [2] to be an appropriate method for the control of complex continuous unidentified or partially identified processes, many of which cannot easily be modeled in a mathematical way. This is because unlike a conventional process controller such as a PID controller, no rigorous mathematical model is required to design a good fuzzy controller (FLC) but a set of linguistic information is incorporated from humans experts. For that reason, the FLC has been successfully applied in many complex processes [3], [4], however, they experience a deficiency in knowledge acquisition, and rely to a great extent on empirical and heuristic knowledge which generally, cannot be elicited. Among the problems to be resolved in fuzzy controller design, are the determination of the linguistic state space, definition of the membership functions and the derivation of the control rules, which are in traditional method obtained by heuristic trial-and-error approach based on analyzing process behavior, and consequent iterative modification to obtain acceptable performance, and such methods are laborious and time consuming. These shortcomings have motivated the use of an evolutionary algorithm [5] to design the FLC enabling to create and modify its knowledge base.

In this paper, we investigate the use of Genetic Algorithms (GAs) to the complete design of a multivariable control of anesthesia with its two areas: unconsciousness and muscle relaxation in surgical operation, attaining and maintaining in such way, the control of depth of anesthesia.

This paper is structured as follows: section 2 reviews the GA functioning principle in the search space. In section 3 and 4 we present the general concept of the FLC, and how GA is applied to the design of the FLC. Section 5 describes anesthesia system. while section 6 and 7 include algorithm description and simulation results. Finally we summarize the paper in section 8.

2. Brief Review of GA Optimizer

2.1 Principle of GA:

Recently, research has emphasized in optimization methods, which employ principles of evolution and heredity from nature [6]. These algorithms search the problem space with a population of points and probabilistic decisions.

The GA of which the pseudocode is presented in Fig.1 performs the optimization process with a population of individuals, each of which represents a search point in the space of potential solutions to the problem [7], [8].

```
 Initialize P(t) :population at time t
 evaluate P(t)
 while (not terminate condition)do
 begin
  t=t+1        :increment generation
  select P(t) from P(t-1)
  recombine P(t)
:apply genetic operator
  evaluate P(t)
 end
 end
```

Fig.1 Pseudocode of GA

The population is randomly created and then, evolved toward better regions of the search space by means of simulating some of the process observed in nature evolution, like selection, crossover and mutation operations. First the GA maintains a population of individuals, P(t)=x1(t),x2(t),x3(t),.....xn(t) for iteration t, each solution xi(t) is evaluated to give measure of its fitness. The new population (iteration t+1) is formed by selecting the more fit individuals to reproduce more. Some members of the new population undergo transformation like crossover operation (one or more crossing sites) which create new individuals, by combining parts from several individuals and unary mutation operator which create new individuals by a change in a single individual. After some number of generations, the search converges and is successful, if the best individual represents the global optimum solution. Despite that GAs require only information concerning the quality of the solution, not derivative information or complete knowledge of the problem, they suffer from the time taken in evolution process, which imposes restriction on the size of the population and also the number of generations required to run the GA to a final solution. To alleviate this problem, parallel processing [9], [10] can be employed to reduce the execution time; by using one-population and multiple processors to divide the evaluation task, or by separating subpopulations on each processor which develop individual solutions.

3. General Concept of the FLC

The basic configuration of the logic system considered in this paper is shown in fig.2. In what follows, a brief description of each component and the basic fuzzy operations that it performs is presented.

3.1 Knowledge Base constructed with fuzzy rules

The knowledge base for the fuzzy logic system contains a collection of fuzzy IF-THEN rules. The MISO IF-THEN rule(s) are of the form

$$R^{(j)}: \text{IF } x_1 \text{ is } A_j^1 \text{ and} \dots x_n \text{ is } A_n^j, \text{ THEN } y \text{ is } C^j \quad (1)$$

where $\underline{x} = (x_1, \dots, x_n)^T \in V \subset R^n$ and $y \in W \subset R$ denote the linguistic variables associated with the inputs and output of the fuzzy logic system [1]. A_i^j and C^j are labels of the fuzzy sets in V and W, respectively, and i denotes the number of input/state of fuzzy logic system, i.e, i=1,2,n, and j =1,2,...,M. Fuzzy rule (1) can be implemented using fuzzy implication, which gives

$$A_1^j \times \dots \times A_n^j \rightarrow C^j \quad (2)$$

which is a fuzzy set defined in the product space V × W. Based on generalizations of implications in multivalued logic, many fuzzy implication rules have been proposed in the fuzzy logic literature. In this paper, we define the implication rule using the t-norm operator, given by

$$\mu_{A_1^j \times \dots \times A_n^j \rightarrow C^j}(\underline{x}, y) = \mu_{A_1^j}(x_1) * \dots * \mu_{A_n^j}(x_n) * \mu_{C^j}(y) \quad (3)$$

where * denotes the t-norm, which in general corresponds to the conjunction 'min' or 'product'.

3.2 Fuzzy Inference Engine

The fuzzy inference engine performs a mapping from fuzzy sets in V to fuzzy sets in R, based on the fuzzy IF-THEN rules in the fuzzy rule base and the compositional inference rule.

Let B be a fuzzy set in V, then the fuzzy relational equation $B \circ R^j$ were "$\circ$" is the sup-star composition, results in M fuzzy sets. Using the t-norm operator yields

$$\mu_{B \circ R^j}(y) = \sup_{\underline{x}} [\mu_B(\underline{x}) * \mu_{A_1^j \times \dots \times A_n^j \rightarrow C^j}(\underline{x}, y)] \quad (4)$$

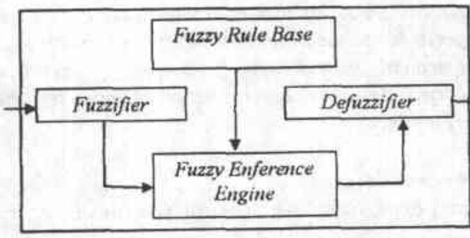


Fig.2 A block diagram of basic fuzzy logic system

In order to combine the M fuzzy sets into one fuzzy set, the t-norm can be employed and it results in

$$\mu_{B \circ (R^1, \dots, R^M)}(y) = \mu_{B \circ R^{(1)}}(y) \dot{+} \dots \dot{+} \mu_{B \circ R^{(M)}}(y) \quad (5)$$

where $\dot{+}$ denotes the t-conorm, the most commonly used operation for $\dot{+}$ is 'max'. If we use the product operation and choose * in (3) and (4) to be an algebraic product, then the inference is called product inference. Using the product inference, the equation (4) becomes

$$\mu_{B \circ R^j}(y) = \sup_{\underline{x} \in V} [\mu_B(\underline{x}) \mu_{A_1^j}(x_1) \dots \mu_{A_n^j}(x_n) \mu_{C^j}(y)] \quad (6)$$

3.3 Fuzzifier

The fuzzifier maps a crisp point $\underline{x}$ into a fuzzy set B in V. In general, there are two possible choices of this mapping [2], namely, singleton, or nonsingleton. In this paper, we use the singleton fuzzifier mapping, i.e.,

$$\mu_B(\underline{x}') = \begin{cases} 1 & \text{for } \underline{x} = \underline{x}' \\ 0 & \text{for otherwise} \end{cases} \quad (7)$$

3.4 Defuzzifier

The defuzzifier maps fuzzy sets in W to a crisp point in R. In general; there are different possible choices of this mapping [2], among others, maximum, center gravity, and center-average defuzzifier. In this paper, we use the center-average defuzzifier mapping, i.e.,

$$y = \frac{\sum_{j=1}^{M} y^j (\mu_{B \circ R^j}(y^j))}{\sum_{j=1}^{M} (\mu_{B \circ R^j}(y^j))} \qquad (8)$$

where y^j is the point in R at which μ_{C^j} achieves its maximum value (assume that $\mu_{C^j}(y^j) = 1$).

3.5 Fuzzy Basis Functions

The fuzzy logic system with *center-average defuzzifier (8), product inference (6), and singleton fuzzifier (7)*, is of the following form

$$y(\underline{x}) = \frac{\sum_{j=1}^{M} y^j (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))} \qquad (9)$$

if we fix the $\mu_{A_i^j}(x_i)$'s and view the y^j's as adjustable parameters, then the equation (9) became :

$$y(\underline{x}) = \theta^T \delta(\underline{x}) \qquad (10)$$

where $\theta = (y^1, ..., y^M)^T$ is a parameter vector, and $\delta(\underline{x}) = (\delta^1(\underline{x}), ..., \delta^M(\underline{x}))^T$ is a regressive vector with the regressor $\delta^j(\underline{x})$ defined as

$$\delta(\underline{x}) = \frac{\prod_{i=1}^{n} \mu_{A_i^j}(x_i)}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))} \qquad (11)$$

which are called fuzzy basis functions (FBF's) and these FBF's are universal approximators [2]. We can fix all the parameters in $\delta^j(\underline{x})$ at the beginning of the FBF expansion design procedure, so that the only free design parameters are θ_i.

4. Interaction Of The FLC Module With GA Optimizer

Fig.3 shows the interconnection of the FLC, simulation model, and GA optimizer [12], [9]. The FLC operates the simulation model of the plant to be controlled. An individual of the GA population represents one trial set of fuzzy membership functions and rules [11]. So the GA optimizer sends a parameter assignment (an individual) to the FLC which determines its knowledge base. The model is reset to its initial conditions. During control operation, the values of state variables are sampled with some sampling time, while the FLC issues control commands to the simulation model.

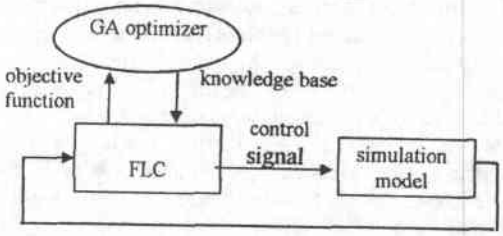


Fig.3 GA optimization of the module

5. Anesthesia System Description

In order to identify the muscle relaxation process, and the unconsciousness process associated with drugs [13], Pharmacology comprises two main areas known as pharmacoknetics and pharmacodynamics [14], [15]. Pharmacoknetics studies the relationship that exists between drug dose and drug concentration in the blood plasma as well as other parts of the body. Pharmacodynamics, however, is concerned with the drug concentration and the effect produced. In light of these considerations, the linear pharmacoknetics, which describe distribution of drugs into the blood, are given by the following equation,

$$\begin{bmatrix} Paralysis \\ MAP \end{bmatrix} = \begin{bmatrix} G11(s) & G12(s) \\ 0 & G22(s) \end{bmatrix} \begin{bmatrix} U1(s) \\ U2(s) \end{bmatrix}$$

with

$$G11(s) = \frac{1.e^{-s}(1+10.64s)}{(1+3.08s)(1+4.81s)(1+34.36s)}$$

$$G12(s) = \frac{0.27.e^{-s}}{(1+2.83s)(1+1.25s)}$$

$$G22(s) = \frac{-15.e^{-0.42s}}{1+2s}$$

where paralysis (EMG electromyogram signal response for muscle relaxation) is normalized to unity, MAP (Mean arterial Pressure) in mmHg, U1(s) atracurium and U2(s) isoflurane infusion rate are normalized to unity, and time units are minutes [15].
In addition, the pharmacodynamic effect of atracurium is usually modeled by a Hill equation:

$$Y = \frac{V^a}{V^a + (V_{50})^a}$$
$$a = 2.98 \pm 0.29$$
$$V_{50} = 0.404 \pm 0.017$$

6. Algorithm Description

In this paper we investigate the use of an additional factor to the procedure design of fuzzy controllers which is the number of partitions within the fuzzy universe, limiting in such way the intervention of an human-expert merely, to define the boundaries of the fuzzy universe of discourse. The parameter sets in this study, which consist of an entire set of fuzzy membership functions describing the space of the input variables and rule sets, are coded as concatenating strings of digits. The length of chromosome is variable, i.e. the numbers of inputs fuzzy sets and theirs parameters were determined by the GA, while the number of output fuzzy sets and theirs parameters were fixed to decrease the execution time. The membership functions used here are Gaussian of the following form:

$$\mu_i(x) = e^{-((x-c)^2/2\delta^2)}$$

Therefore, to code this function we need two parameters: the center c and the deviation δ, these values are mapped linearly between determined minimum (Pmin) and maximum (Pmax) values according to the following

$$A = P_{min} * b\,(P_{max} - P_{min})$$

where A is the value of the parameter being coded and b is the allele integer-base value.

To illustrate this method, consider a system with two inputs I1, I2, and one output O, and assume that GA give five Gaussian fuzzy sets for the first input I1, and three for the second I2 to divide their spaces respectively. We fix five fuzzy sets for the output space partition. Hence negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), for I1 and O variable spaces, and negative (N), zero (ZE), positive (P) for I2 variable space. The rule set contains (5*3) rules to account for every possible combination of input fuzzy sets. Here, we assume that the peak of the membership functions associated to the extremes are fixed, so, eight alleles are reserved for the I1 variable and four alleles for the I2 variable. The string representing the controller is integer-based, thus, the first 15 alleles which representing the rules set have values in the set {1,2,..., Number of output sets (5)}, while the alleles representing the membership functions have values in the set {1,2,...,5}, see fig.4

I2 \ I1	MN	SN	ZE	MP	SP
N	5	1	4	3	2
Z	1	5	2	1	3
P	3	2	1	2	3

Fig.4 The rule base encode.

However, the objective of the controller is to minimize the error and reduce the size of the rule base, which has more impact on the FLC performance. Indeed, these factors are weighted and summed to assess the expression of the fitness function, which is taken by the GA to optimize the solutions.

7. Simulation Results for Anesthesia Controller

This paper investigate the development of a suitable GA technique for fuzzy design, with a smooth manner in choosing the typical partition in the input and/or output spaces. During simulations, we have noticed that due to the random choice of the initial population, genetic algorithms do not behave in the same way for every run. We run the algorithm 8 times and according to these runs we have noticed that the error decreases and the performance of the FLC gets better as the number of iterations and population size increase, fig.5 show that, in wich the fitness function evolves througth the generation number. In light of these considerations, the values for population size, maximum number of generations, probability of crossover (two point crossover is used here), and probability of mutation are 100, 50, 0.3 and 0.03 respectively.

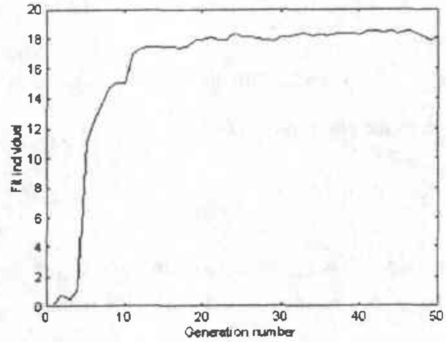
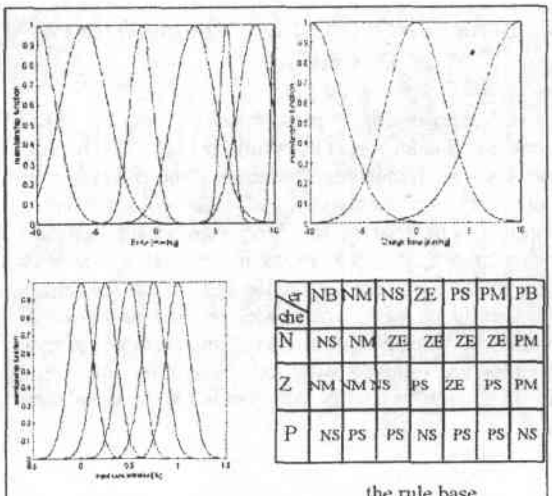


Fig.5 GA performance

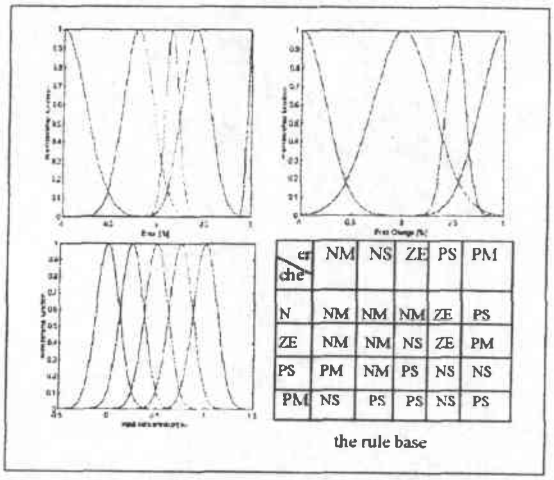
The GA is used to lead the knowledge base for a multivariable fuzzy controller. The fuzzy controller is then applied to the control of multivariable anesthesia, including the simultaneous regulation of muscle relaxation MR (expressed as a % of total paralysis) and the depth of unconscious (controlled by MAP mesure) in patient undergoing surgical operation. The model is described clearly (section 5). A fourth-order Runge-Kutta integration method was used with a sampling interval of 1 minute.



er \ che	NB	NM	NS	ZE	PS	PM	PB
N	NS	NM	ZE	ZE	ZE	ZE	PM
Z	NM	NM	NS	PS	ZE	PS	PM
P	NS	PS	PS	NS	PS	PS	NS

the rule base

a



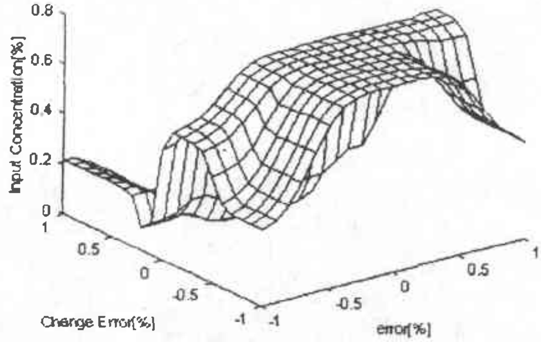
er / che	NM	NS	ZE	PS	PM
N	NM	NM	NM	ZE	PS
ZE	NM	NM	NS	ZE	PM
PS	PM	NM	PS	NS	NS
PM	NS	PS	PS	NS	PS

the rule base

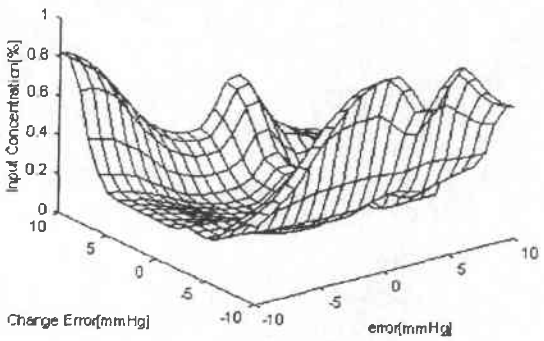
b

Fig.6 Optimized knowledg bases
a. for MAP
b. for MR

By using the best knowledge bases found by the GA optimizer, which are described for both MAP and MR in figure.6, the control surfaces within contineous and smooth transitions which are provided from a certain amount of overlap of the fuzzy sets, are portrayed in figure.7.



a



b

7. control surfaces,
a. for MR
b. for MAP

The inputs MAP and paralysis with the corresponding drugs are illustrated in figure.8, where the controller can attain and maintain the control of anesthesia depth. by adjusting the blood pressure and the paralysis rate. adequately. To check robusteness, disturbances are applied (exemple, a skin incision can lead to rapid changes in blood pressure of more than 10 mmHg).

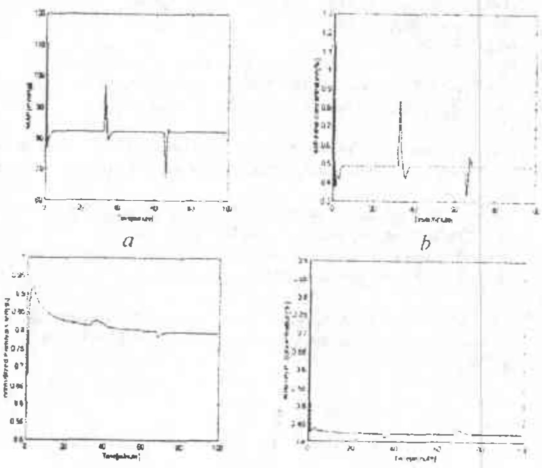


Fig 7. Multivariable response.
a and b MAP response.
c and d MR, showing interactions from MAP

8. Conclusion

The objective of this paper was to investigate the use of genetic algorithms as a tool for the design of fuzzy controllers. The simulation results presented here, have demonstrated the effectiveness of the proposed control system to insure that the patient's hemodynamics (MAP, paralysis) remain stable and the patient remains sufficiently anesthetized.
Based on these results, one can conclude that GAs are valuable tools for the design of an FLC with exellent robusteness and performance which improve by automating the number of membership functions. By introducing such additional degree of freedom, the user will have more flexibility in the design.

References

[1] C. C. Lee, «Fuzzy Logic in Control Systems: Fuzzy Logic Controller Part I and II,» IEEE Trans. Syst., Man, Cyben, vol. 20, No. 2, pp. 404-485, 1990

[2] J. M. Mendel, «Fuzzy Logic Systems for Engineering. A Tutorial. » Proc. IEEE, vol. 83, No. 3, March 1995

[3] Byungkook Yoo and Woonchul Ham, «Adaptive Fuzzy Sliding Mode Control of Nonlinear System. » IEEE Transaction of fuzzy systems. vol. 6, No. 2, May 1998.

[4] Can Yi Su and Yuruy Stepanenko, «Adaptive Control of a Class of Nonlinear Systems with Fuzzy Logic, » IEEE Transaction on fuzzy systems, vol. 2, No. 4, Nov 1994.

[5] Jimwoo Kim and Bemard P. Zeigler. «Designing Fuzzy Logic Controllers Using a Multiresolutional Search Paradigm,» IEEE Transaction on fuzzy systems, vol. 4, No. 3, August 1996.

[6] J. H. Holland, «Addaptation in natural and Artificial Systems. » Ann Arbor, University of Michigam Press, 1975

[7] D. Goldberg, «Genetic Algorithms in Search, Optimisation, and Machine Learning,» Addison-Wesley Publishing Company. Inc., Princeton. 1989.

[8] Z. Michalwicz, «Genetic Algorithms + data structure = evolution programms,» In AI Series, Spring-Verlay, New York, 1994.

[9] D. A. Linkens and H. O. Nyongesa, «Genetic algorithm for Fuzzy control Part 1: Offline system developement and application,» IEE Proc. Control Theory, Appl., vol. 142, No. 3, May 1995.

[10] Marco Russo, «FuGeNeSys, a Fuzzy Genetic Neural System for Fuzzy Modeling,» IEEE transactions on fuzzy systems, vol. 6, No. 3, August 1998.

[11] A. Homaifar and Ed. Mc. Mcormick, «Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms,» IEEE Transactions on fyzzy systems, vol. 3, No. 2, May 1995.

[12] Jinwoo Kim, Yookeon Moon and Bernard P. Zeigler, «Designing Fuzzy Net Controllers Using Genetic Algorithm,» IEEE Control systems, Tucson, AZ, March 1994.

[13] D. A. Linkens, «Adaptive and Intelligent Control of Anesthesia,» IEEE Control systems, vol. 12, No. 6, Dec 1992.

[14] M. Mahfouf, D. A. Linkens, «Generalised predictive control and bioengineering, » The Taylor & françis Systems and Control Book Series, 1998.

[15] M. Mahfouf, D. A. Linkens, «A nonlinear GPC algorithm (NLGPC) for muscle relaxant anesthesia,» International journal of control, proc. of the WAC 1998.