A MULTIOBJECTIVE OPTIMISATION SYSTEM FOR A GLASS FURNACE

João Pina *, Pedro Lima †

* Instituto Superior Técnico,
Av. Rovisco Pais, 1096 Lisboa Codex, Portugal
Fax: 351 1 8496156
e-mail: jpina@in3.dem.ist.utl.pt
† Instituto de Sistemas e Robótica,
Instituto Superior Técnico,
Av. Rovisco Pais, 1096 Lisboa Codex, Portugal
e-mail: pal@isr.ist.utl.pt

Keywords: Expert control, glass furnace, genetic algorithms, multiobjective optimisation, Pareto optimality.

Abstract

An Expert Controller (EC) for the Operation of a recuperative glass furnace is introduced in this paper. The EC is part of an Integrated Operation and Control System, whose hierarchical architecture is also described, and is divided in two sub-systems, a Fault Detection System, and a Process Optimisation System (POS). The POS architecture and operation, detailed in this paper, is based on a Genetic Algorithm, for solution search, and a learning algorithm, based on fuzzy set theory, to interpolate a cost function, which quantifies one of the objectives.

1 Introduction

Intelligent Control Systems are becoming more important in Process Industry due to higher quality product demands, stricter laws on pollutant emissions and sensor technological developments (increasing the amount and quality of available information). As an example, we find cement, [hasler], and gas production industries, [mclean]. Glass industry can not be an exception. The existence of sophisticated control systems, concerning furnace lifetime and environment conditions optimisation, is referred as a crucial aspect in the improvement of productivity and glass quality [heitor]. Glass furnaces Operation and Control are still performed only on the basis of the operator's experience. An Operation and Control System that is intended to be an on-line aid to the operator, namely through a Process Optimisation System, is described in this paper. The long-term goal is to implement a completely automatic operation system. This system will be implemented in a recuperative furnace, with two groups of four natural gas burners, each controlling a temperature zone measured by a thermocouple in the side wall. The furnace has also an electrical booster system to increase its capacity and the glass quality. The furnace and the whole factory are being built under the project NOVOVIDRO, and are located at Marinha Grande, Portugal.

The paper is organised as follows: in section 2 it is described the operation and control hierarchical architecture under development, as well as its conceptual levels; in section 3 the Process Optimisation System and its theoretical support are explained, including Pareto optimality concepts and the evolutionary approach for multiobjective optimisation; in section 4, conclusions and future trends are referred.

2 Operation and Control Hierarchical Architecture

The functional hierarchic architecture of the Operation and Control System under development is presented in Figure 1. The conceptual levels shown there are the **Process** itself, the **Goals**, the **Execution**, the **Analysis** and the **Operation** level.

2.1 Process

The purpose of a glass furnace is to melt, using a temperature distribution, a composition, called batch. As a result, some chemical reactions occur, conditioning glass quality. To achieve natural gas energy maximum utilisation, combustion in the furnace must take place with a small excess air factor. At high temperatures glass becomes electrically conductor, due to the mobility of certain ions (e. g. lithium). By Joule effect glass temperature raises, leading to the formation of melting glass currents. Pressure inside furnace must be maintained at a certain level, above the atmospheric pressure, to avoid energy losses and refractory corrosion, and to assure good furnace operation.

2.2 Goals

Glass Industry has five main control goals: **Glass Quality Maximisation**, **Thermal Efficiency Maximisation**, **Refractory and Furnace Lifetime Maximisation**, **Pollutants Production and Emission Minimisation** and

Energetic Consumption Minimisation. These objectives are interrelated, so the use of multicriteria or multiobjective techniques will be a natural approach. Some of the relations between control goals and process variables are identified in [farmer].

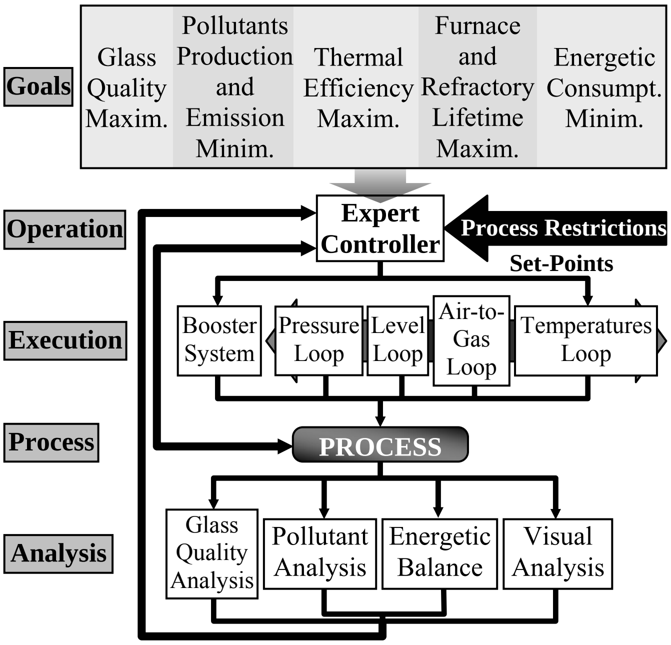

Fig. 1 – Operation and Control System Hierarchical Architecture.

2.3 Execution

This level includes low-level control loops and the operation of systems that actuate the process directly. There are, typically, four low-level control loops. These are the **furnace pressure**, **glass level**, **air-to-natural gas ratio** and **zone temperatures** control loops, whose actuation is interrelated. They are currently implemented in Programmable Logic Controllers (PLC's).

2.4 Analysis

This block will supply quantitative information about process performance. It includes **glass quality analysis**, **waste gas analysis**, **energetic balance** and inside furnace **visual analysis** provided by an automatic vision system.

2.5 Operation

Furnace Operation is based on an Expert Controller divided in two sub-systems: a **Fault Detection System**, implemented by an expert system, and a **Process Optimisation System** (POS). In Figure 2 we depict the relationships between process state variables and also POS structure and information flows. In the figure, **Cost Parameters** specify the information related to the cost vector used to quantify goals (referred later), which is specified initially by the operator, and later by the POS; **x** is the Process State, described by a vector of feature values (e. g., air and gas flows, temperatures, flame length); $\mathbf{x_A}$ is the state vector used in the analysis block; $\mathbf{x_O}$ is the state vector used for process optimisation; $\mathbf{f'(x_O)}$ is the cost vector computed from process data, and is used to complement and correct the used models (see 3.3); $\mathbf{x_{SPC}}$ is an optimal solution candidate; $\hat{\mathbf{f}}(\mathbf{x_{SPC}})$ is the evaluation estimate of a candidate solution, $\mathbf{x_{FD}}$ is the state vector used for fault detection read from sensors; $\mathbf{x'_{FD}}$ is the state vector used for fault detection computed from process analysis; $\mathbf{x_C}$ is the state vector used for low-level control. The output of the POS is the vector $\mathbf{x_{SP}}$ of set-points for the low level controllers and sub-systems. The POS is described in the sequel.

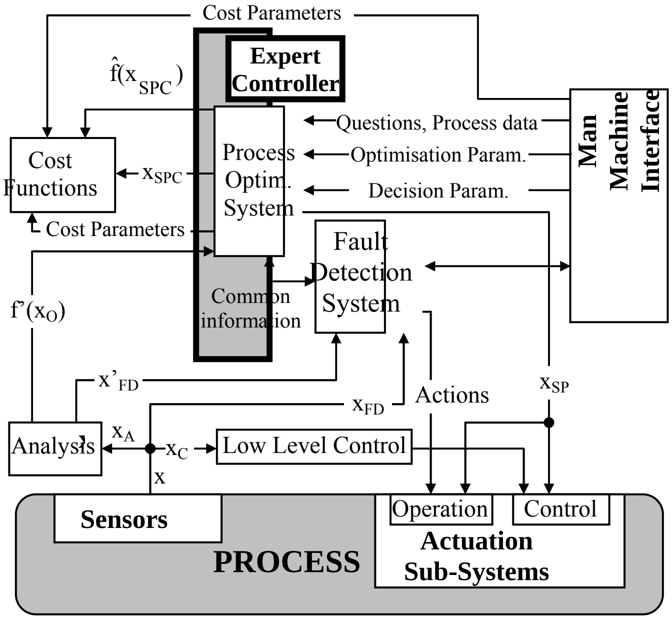

Fig. 2 – Process state variables and POS structure and information flows.

Some of the information provided by the Analysis block, namely glass quality analysis and energetic balance, will be used directly by the POS (see next section). Pollutant analysis goal is mainly for air-to-gas ratio calibration, while visual analysis will be used by the Fault Detection System, to monitor flame status. This information will be shared with POS, so that in case of burner fail, the electric boosting can compensate this. Multiobjective optimisation is next introduced as a tool for POS design.

3 Process Optimisation System

3.1 The Multiobjective Optimisation Problem

A Multiobjective Optimisation (MO) problem can be stated as: given an objective or cost vector function $\mathbf{f(x_O)}$ and a decision space Ω^m, a solution $\mathbf{x_O}^* \in \Omega^m$ is to be found, such that $\mathbf{x_O}^*$ is the argument minimising $\mathbf{f}$ [fons1]. In our work,

$\mathbf{x_O}$ has three components (m = 3), which are zone 1 temperature, zone 2 temperature and applied electric voltage. **f** represents the control goals. Next, we describe the analysis proposed to estimate how close is the process from the control goals.

Thermal efficiency maximisation: An on-line efficiency model is being designed, based on the furnace energetic balance, and will be used to predict performance changes, for different operating conditions. Being $\eta \in [0, 1]$ the output of this model, a cost function can be obtained by:

f_1(temperature set points, electric voltage, recuperators efficiency) $\triangleq 1 - \eta$

Furnace and refractory lifetime maximisation: Although it is not possible to quantify how many years lifetime a furnace still has, some factors are known to increase refractory corrosion process, for instance, higher furnace temperatures or pressures. To achieve this objective we only have to ensure that these variables are maintained within some safety limits, and no other function must be defined.

Pollutants production and emission minimisation: Pollutants production is mainly due to inefficient combustion (CO, CO_2) or to thermal mechanisms (NO_X). In the first case, the oxygen percentage in the waste gases can be monitored and the air-to-natural gas relation adapted. NO_X production is related with flame temperatures, and since these depend on the temperature set points, we define two more cost functions:

$f_2 \triangleq$ zone 1 temperature set point;
$f_3 \triangleq$ zone 2 temperature set point;

Energetic consumption minimisation: It is desirable to make a balance, under an economic point of view, between natural gas and electric energy consumption. We then define

f_4(furnace load, temperature zones set points, electric voltage) $\triangleq \alpha$.(electric energy consumption) + β.(natural gas consumption)

α will vary according to daily changes in electric energy cost.

Glass quality maximisation: We can quantify quality through the amount of defects in glass. These can be of three types, namely, **Cord**, due to faulty glass gathering or batch heterogeneity, **Stone**, due to refractory mixing with glass, caused by corrosion processes, and **Blister**, due to inadequate temperature distribution. The only defect that has a direct relationship with process variables, at furnace level, is the blister. Another function, to quantify blisters, can be defined:

f_5(batch recipe, furnace load, temperature zones set points, electric voltage) $\triangleq$ Glass defects percentage

which has to be less then a certain value, P_{max}, defined according to the needs.

In summary, and taking into account only the variables affected by the Process Optimisation System, the glass furnace MO problem can be stated as follows:

$x_1 \triangleq T_1$; zone 1 temperature
$x_2 \triangleq T_2$; zone 2 temperature
$x_3 \triangleq V_E$; applied electric voltage
$\mathbf{x_O} \triangleq (x_1, x_2, x_3)$
$\mathbf{f} \triangleq (f_1, f_2, f_3, f_4, f_5)$

Minimise $\mathbf{f}(\mathbf{x_O})$
subject to $f_5(\mathbf{x_O}) < P_{max}$.

3.2 Genetic Algorithms applied to Multiobjective Optimisation

MO problems usually do not have a unique solution, rather a set of alternative ones, whose elements are such that they cannot have all its components simultaneously improved. This is known as Pareto Optimality concept [fons1], and is based on the following definitions:
Given a vector function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_N(\mathbf{x}))$, $\mathbf{x} \in \Omega^m$,

Definition **Pareto Dominance**: a vector $\mathbf{u} = \mathbf{f}(\mathbf{x_u})$ is said to dominate another vector $\mathbf{v} = \mathbf{f}(\mathbf{x_v})$, and is denoted **u** p< **v**, iff **u** is partially less a **v**, which means:

$$\forall i \in \{1, ..., N\}, u_i \leq v_i \quad \wedge \quad \exists i \in \{1, ..., N\}, u_i < v_i \; \blacksquare$$

Definition **Pareto Optimality**: a given decision vector, $\mathbf{x_u}$, is said to be a Pareto-optimal, iff there is no other $\mathbf{x_v}$, for which $\mathbf{v} = \mathbf{f}(\mathbf{x_v})$ dominates $\mathbf{u} = \mathbf{f}(\mathbf{x_u})$ ■

The set of all Pareto-optimal vectors is called non-inferior or non-dominated set of the problem. Genetic Algorithms (GA), [gold], are frequently used for MO, [fons1]. They begin with a population of P randomly initialised individuals (set-points solution candidates) to which they apply selection, crossover, mutation and elitism operators. These and other concepts will be briefly reviewed under the particular framework of this application.

3.2.1 Population

Each problem variable is represented by a string of bits, and an individual is formed by the concatenation of these ones, forming a chromosome. Since a string of l bits represents values in the range $[0, 2^l - 1]$, a linear mapping for an interest interval, $[U_{min}, U_{max}]$, must be performed. Variables restrictions can be handled by the appropriate choice of U_{min} and U_{max}. These limits, the discretization step (corresponding to the desired accuracy) of each variable, q, and the number of bits used are shown in Table 1.

3.2.2 Selection and Reproduction

Since in natural selection only the fittest survive, in GA's an individual is selected to generate offspring based on its fitness (see 3.2.7). Each selected individual is reproduced for crossover.

Variable	Set-Point	U_{min}	U_{max}	q	l
x_{SP_1}	zone 1 temperature	1500 ºC	1600 ºC	1 ºC	7
x_{SP_2}	zone 2 temperature	1500 ºC	1600 ºC	1 ºC	7
x_{SP_3}	Electric voltage	50 V	122 V	7 V	4

Tab. 1 – MO parameters

3.2.3 Crossover

Crossover, between two selected individuals, consist of interchanging, with some probability, chromosome parts, as is exemplified next:

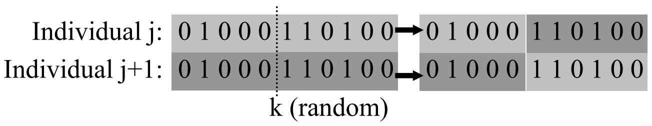

Fig. 3 – Crossover

3.2.4 Mutation

Mutation consists in changing, with some probability, each chromosome bit value, after crossover.

3.2.5 Elitism

Elitism is an operator that intends to preserve best individuals over the run of the algorithm. Giving generations n and n+1, the M% worse individuals in n+1 will be substituted by all individuals in n with highest fitness.

3.2.6 Niche Formation

Share: to avoid an uncontrolled growth of a dominant specie, i. e., individuals with the same genetic characteristics (genetic drift [gold]), some techniques (called niche formation) that favour population distribution by the several peaks of the cost function, are used. In one of those techniques, each individual fitness is weighted by the contribution of all its neighbours closer a σ_{share}. To process that contribution, between two individuals, $\mathbf{x}$ and $\mathbf{x_i}$, the sharing function of Figure 4 is used.

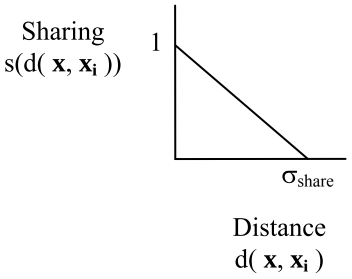

Fig. 4 – Sharing function

It was established that an individual, $\mathbf{x_i}$, is closer a σ_{share} from another, $\mathbf{x}$, if all parameters in $\mathbf{x_i}$ are at a distance (in an Euclidean way) shorter a σ_{share}%, relatively to each range, from the correspondent parameters in $\mathbf{x}$. Sharing s(.) is then computed based on the average distance between parameters of both individuals. The derated fitness is then given by:

$$h_d(\mathbf{x}) = \frac{h(\mathbf{x})}{\sum_{i=1}^{P} s(d(\mathbf{x}, \mathbf{x}_i))}$$

where h($\mathbf{x}$) is the fitness without sharing. Selection is performed based on $h_d(\mathbf{x})$.

Mating restrictions: As in nature, it’s unlike that individuals from different species attempt to mate. It was decided to restrict mate only between individuals less a σ_{mate} from each other, if they exist.

3.2.7 Fitness assignment – population ranking

The evolution of a population in a run of a GA is guided by individuals fitness. A strategy of fitness assignment, based on population ranking, is used, [fons1].

Rank: a preference vector, $\mathbf{g}$ = [$\mathbf{g_1}$, ..., $\mathbf{g_p}$], assigns goals to priorities, in the objectives. For $\mathbf{g}$ = $[(g_{1,1}, \ldots, g_{1,n_1}), \ldots, (g_{p,1}, \ldots, g_{p,n_p})]$, which implies $\sum_{i=1}^{p} n_i = N$, and, for an objective vector, $\mathbf{f}$, in the same way, each $\mathbf{g_i}$ = $(g_{i,1}, \ldots, g_{i,n_i})$ assigns priority i and goal $g_{i,j}$ to the objective $f_{i,j}$. This implies, without loss of generality, that objective functions might be permuted. For a vector $\mathbf{u}$, the components that achieve goals in priority i, are denoted by $\overset{\mathbf{u}}{\mathbf{u}_i^{\cup}}$, and the other ones by $\overset{\mathbf{u}}{\mathbf{u}_i^{\cap}}$. In another vector, $\mathbf{v}$, the correspondent components will be $\overset{\mathbf{u}}{\mathbf{v}_i^{\cup}}$ and $\overset{\mathbf{u}}{\mathbf{v}_i^{\cap}}$. Ranking is based on the preference relation defined by Preferability [fons1].

Definition **Preferability**: A given objective vector $\mathbf{u}$ is said to be preferable to another, $\mathbf{v}$, for a preference vector $\mathbf{g}$, $\mathbf{u} \underset{\mathbf{g}}{\subset} \mathbf{v}$, iff:

p = 1:

$$\left(\overset{\mathbf{u}}{\mathbf{u}_p^{\cap}}\ p< \overset{\mathbf{u}}{\mathbf{v}_p^{\cap}}\right) \vee \left\{\left(\overset{\mathbf{u}}{\mathbf{u}_p^{\cap}} = \overset{\mathbf{u}}{\mathbf{v}_p^{\cap}}\right) \wedge \left[\left(\overset{\mathbf{u}}{\mathbf{v}_p^{\cup}} \not\leq \overset{\mathbf{u}}{\mathbf{g}_p^{\cup}}\right) \vee \left(\overset{\mathbf{u}}{\mathbf{u}_p^{\cup}}\ p< \overset{\mathbf{u}}{\mathbf{v}_p^{\cup}}\right)\right]\right\}$$

p > 1:

$$\left(\overset{\mathbf{u}}{\mathbf{u}_p^{\cap}}\ p< \overset{\mathbf{u}}{\mathbf{v}_p^{\cap}}\right) \vee \left\{\left(\overset{\mathbf{u}}{\mathbf{u}_p^{\cap}} = \overset{\mathbf{u}}{\mathbf{v}_p^{\cap}}\right) \wedge \left[\left(\overset{\mathbf{u}}{\mathbf{v}_p^{\cup}} \not\leq \overset{\mathbf{u}}{\mathbf{g}_p^{\cup}}\right) \vee \left(\mathbf{u}_{1,\ldots,p\text{-}1} \underset{\mathbf{g}_{1,\ldots,p\text{-}1}}{\subset} \mathbf{v}_{1,\ldots,p\text{-}1}\right)\right]\right\}$$

with $\mathbf{u}_{1,\ldots,p\text{-}1}$ = [$\mathbf{u_1}$, ..., $\mathbf{u}_{p\text{-}1}$], $\mathbf{v}_{1,\ldots,p\text{-}1}$ = [$\mathbf{v_1}$, ..., $\mathbf{v}_{p\text{-}1}$] and $\mathbf{g}_{1,\ldots,p\text{-}1}$ = ($\mathbf{g_1}$, ..., $\mathbf{g}_{p\text{-}1}$) ■

Preferability establishes that between two objective vectors high priority goals must be achieved first. The next level is then tested, until the lowest priority is reached, when the values are compared according to Pareto notions. With these definitions, and as in [fons1], we have in our case $\mathbf{g} = [\mathbf{g_1}, \mathbf{g_2}] = [\ (0, 1600, 1600, 0), (P_{max})\ ]$. Since f_5 presents restrictions, and it represents an important goal, it is assigned the highest probability. The individuals ranking is done according to the rule [fons1]: being $r_u^{(t)}$ the number of preferable individuals to an element **u**, in iteration t, then rank(**u**, t) = $r_u^{(t)}$. This way, preferable individuals have rank 0. Fitness is calculated by an exponential transformation of the rank. Once all individuals objectives are evaluated, its ranks and fitness are computed, then niche formation techniques, selection, crossover, mutation and elitism operators are used to proceed to the next generation. This process continues until maximum generation number is achieved, when one solution is selected. If more than one solution has the maximum fitness, then the selection is done according to f_5 and/or with the one that assures stability of operation, an important factor in this kind of process.

3.3 Learning unknown functions – a *fuzzy* approach

In each GA iteration, a number of individuals, at most the size of the population must be evaluated. This means that cost functions must be determined. For functions f_l to f_4 this can be measured. On the other hand, only some initial points of function f_5 (glass defects percentage) are known, due to *a priori* tests. It should not be expected that individuals fall always on these points. Besides, these values may change in time. Although there is not an analytical model for function f_5, and this function is time variant, there is some operator's heuristic knowledge on the relationship between glass defects and process variables. This way, for each type of batch recipe, tables are built whose inputs are fuzzy variables (furnace load) and crisp variables (temperature set points, applied voltage). For each table the output is a fuzzy variable, representing a prediction of glass defects. This approach, fuzzy based, is justified because it allows, not only the integration of the referred heuristic knowledge, but also to overcome the high complexity of modelling these kind of systems, whose parameters are hardly determinable and time variant.

Using the typical rules whose syntax is

$$R^{(l)} = \mathbf{IF} \left(\mathop{2}_{i=1}^{5} \text{attribute}(i) \text{ is } A_i^{(l)} \right) \mathbf{THEN}\ \text{Defects is } B^{(l)}$$

where attribute is respectively recipe, furnace load, zone 1 temperature set-point, zone 2 temperature set-point and electric voltage, tables may be initialised by operator interaction and furnace tests. The remaining points are interpolated. New acquired information is used to complete and correct old one. For this purpose, the simplified fuzzy algorithm, described by [branco2], it's used for learning by examples. The membership functions used are triangular, symmetric and uniformly distributed.

3.4 Multiobjective Optimisation Algorithm

The MO algorithm has two main tasks:

- **Solution search**: provides the result of the GA running, a Pareto-optimal set-points vector, using the tables to provide estimates of f_5.
- **Learning**: new information, resulting from process data, is used to update the tables that implement function f_5.

The algorithm runs as long as the furnace is active, permanently trying to improve its performance. At time t we denote by $\mathbf{x}_{\mathbf{SP}}^{(t)}$ the vector of set-points being used in low level control, by $\mathbf{x}_{\mathbf{SP}}$ the new vector of set-points calculated by the GA, by $\mathbf{x}_{\mathbf{O}}^{(t)}$ the state vector used for optimisation and by $\mathbf{x}'_{\mathbf{O}}$ the same vector but with measured variables replaced by new set-points, $\mathbf{x}_{\mathbf{SP}}$. There are two important time intervals in the algorithm: T_{PROC}, defined as the estimated time delay required for the effect of the inputs to become visible (which has the magnitude of some hours) and T_{MOS}, the wait time delay if no set-point change was made (of about tenths of minutes).

Each iteration of the algorithm can be described as follows:

```
1. $\mathbf{x}_{\mathbf{OLD}} \leftarrow \mathbf{x}_{\mathbf{SP}}^{(t)}$                                   ; Store information
   $\mathbf{f}_{\mathbf{OLD}} \leftarrow \mathbf{f}(\mathbf{x}_{\mathbf{O}}^{(t)}) = (f_1(\mathbf{x}_{\mathbf{O}}^{(t)}), \ldots, f_5(\mathbf{x}_{\mathbf{O}}^{(t)}))$
2. $\mathbf{x}_{\mathbf{SP}} \leftarrow$ Solution search
3. $\hat{\mathbf{f}}(\mathbf{x}'_{\mathbf{O}}) = (f_1(\mathbf{x}'_{\mathbf{O}}), \ldots, \hat{f}_5(\mathbf{x}'_{\mathbf{O}})) \underset{\mathbf{g}}{\subset} \mathbf{f}_{\mathbf{OLD}}$ ?   ; Checks if new
      3.1 Yes: $\mathbf{x}_{\mathbf{SP}}^{(t)} \leftarrow \mathbf{x}_{\mathbf{SP}}$                        ; solution is
      3.2 No:  Wait $T_{MOS}$                                ; expected to be
            3.2.1 Go to 1.                           ; preferable to the
                                                     ; current one
4. Wait $T_{PROC}$
5. $\mathbf{x}_{\mathbf{NEW}} \leftarrow \mathbf{x}_{\mathbf{SP}}^{(t)}$
   $\mathbf{f}_{\mathbf{NEW}} = (f_{1_{NEW}}, \ldots, f_{5_{NEW}}) \leftarrow$
                  $\mathbf{f}(\mathbf{x}_{\mathbf{O}}^{(t)}) = (f_1(\mathbf{x}_{\mathbf{O}}^{(t)}), \ldots, f_5(\mathbf{x}_{\mathbf{O}}^{(t)}))$
6. $\sqrt{(f_{5_{NEW}} - \hat{f}_5(\mathbf{x}'_{\mathbf{O}}))^2} < \varepsilon$ ?                 ; Checks if the
                                                     ;predicted $f_5$ is
                                                     ;similar to the real
                                                     ; one
      6.1 No:  Add $(\mathbf{x}_{\mathbf{NEW}}, f_{2_{NEW}})$ to training set
            6.1.1 Learning
            6.1.2 $\mathbf{f}_{\mathbf{NEW}} \underset{\mathbf{g}}{\subset} \mathbf{f}_{\mathbf{OLD}}$ ?              ; If not, checks if
                                                     ; the current state
                                                     ; is preferable
                                                     ; to the previous
                                                     ; one
                  6.1.2.1 No: $\mathbf{x}_{\mathbf{SP}}^{(t)} \leftarrow \mathbf{x}_{\mathbf{OLD}}$
            6.1.2.2 Wait $T_{PROC}$
            6.1.2.3 Go to 1.
7. Wait $T_{MOS}$
8. Go to 1
```

4. Conclusions and Future Trends

The concepts presented in this paper are currently being implemented in a real furnace. Its construction has been only recently terminated, and there are not, up to date, software and hardware means that allow data gathering for testing the proposed algorithms. This is actually our major goal. Future work will include the integration in the EC of other types of information, such as image data, so that a more accurate relation between control goals, state variables and manipulated set-points can be determined. The integration with the factory information system (namely BannIV [perrea]) is also being studied, to allow automatic access to production data like glass defects.

5. Acknowledgements

The Ministry of Industry project NOVOVIDRO sponsors one of the authors, J. Pina.

References

[branco1] Branco P., *Aprendizagem por Exemplos Utilizando Lógica “Fuzzy” na Modelização e Controlo de um Accionamento Electro-Hidráulico*, PhD Thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, (1998).

[branco2] Branco P., Dente J., “Learning from Examples to Compensate Nonlinearities in Electromechanical Drives – A Fuzzy Logic Approach”, *Electromotion*, **vol. 4**, no.3, pp.101-108, (1997).

[correia] Correia D., Ferrão P., Heitor M., Silva T., “Glass Furnace Technology for Reduced Emissions Based on Advanced Control and Monitoring”, *Glastechnische Berichte Glass, Science and Technology*, **n. 10**, pp.305-310, (1996).

[farmer] Farmer D., Heitor M., Sentieiro J., Vasconcelos A., “New Methodology for Furnace Monitoring, Analysis and Control”, *Glass Industry*, (1992).

[fons1] Fonseca C., Fleming P., “Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms I: A Unified Formulation”, *Technical Report of the Dept. Automatic Control and Systems Eng. University of Sheffield*, (1995).

[fons2] Fonseca C., Fleming P., “Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms II: Application Example” *Technical Report of the Dept. Automatic Control and Systems Eng. University of Sheffield*, (1995).

[gold] Goldberg D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, (1989).

[hasler] Hasler R., Dekkiche E., “Experience in Kiln Optimisation with LinkMan Graphic”, *Proceedings of the Thirty-Sixth IEEE Cement Industry Technical Conference*, Seattle, pp.271-287, (1994).

[heitor] Heitor M., Ferrão P., Câmara J., “Análise Energética do Sector da Cristalaria em Portugal”, *Internal Report of Research Group on Experimental Thermofluids and Combustion*, Lisbon, (1993).

[lima] Lima P., Oliveira P., Ferreira C., Sentieiro J., “The Architecture of a Knowledge Based Controller for the Operation of Glass Melting Furnaces”, Pre-prints of IFAC *Workshop on Computer Software Structures Integrating AI/KBS Systems in Process Control*, 29 – 30 May, Bergen, Norway, (1991).

[mclean] McLean D., “Gas Well Production Optimisation using Expert Systems Technology”, *Kenonic White Paper*, Kenonic Controls Ltd., Calgary, Canada.

[perrea] Perreault Y., Vlasic T., *Implementing BaanIV*, Que Corporation, (1998).

[rajkov] Rajkovic V., Efstathiou J., Bohanec M., “A Concept of Rule-Based Decision Support Systems”, *Optimization Models Using Fuzzy Sets and Possibility Theory*, D. Reidel Publishing Company, (1987).

[trier] Trier W., *Glass Furnaces Design, Construction and Operation*, Society of Glass Technology, Sheffield, (1987).