Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002 Lisbon, Portugal, July 9-12, 2002.

AN OPERATION SYSTEM FOR INDUSTRIAL PROCESSES: APPLICATION TO A GLASS FURNACE

J.M. Pina*, P.U. Lima†

*Departamento de Engenharia Electrotécnica
Faculdade de Ciências e Tecnologia
Monte de Caparica, 2825-114 Caparica, PORTUGAL
e-mail: jpina@dee.fct.unl.pt

†Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa, PORTUGAL
e-mail: pal@isr.ist.utl.pt

Keywords: Process Operation Architecture, Glass Furnaces, Fuzzy Logic, Genetic Algorithms, Multiobjective Optimisation.

Abstract

An architecture for the operation of industrial processes is presented in this paper. It is based on an expert controller whose main functions are process optimisation and fault detection. Only process optimisation is detailed here. The operation system has two main sub-systems: a Multiobjective Optimisation System, based on genetic algorithms, and a Learning System, based on fuzzy rules, which are both described. A glass furnace application is described as a case study, including some results with real data.

1 Introduction

There are two major difficulties in automatic process operation, and these are generally interrelated. One is due to the multiplicity of criteria when it comes to its performance optimisation. Another depends on the absence or complexity of process models. Multiple criteria, or objectives, some of them concurrent, may be transformed into a single one, by means of an aggregate function [4]. Complex problems generally present multiple parameters as arguments so, e. g., the hill-climbing method [2] may be used to find optimal solutions. However, aggregate functions may not exist, due to incompatibilities in the nature of the objectives, and the latter method does not make a distinction between local and global optima. The approach used in this paper to solve the multiobjective optimisation problem is based on genetic algorithms (GAs) [5]. GAs perform parallel search, so the local and global optima distinction problem is reduced. Together with the preferability relation [4] the difference on the nature of the objectives is overcome. The need to employ process models, in order to evaluate different furnace operation points, involves knowledge acquisition on the most relevant process features. Since these models are, in most cases, unavailable, one possible solution is to learn them from real data. Several learning methods are described that make use of Neural Networks (NNs) [9]. NNs generalisation ability depends on the network structure and, in order to interpolate a general function, the number of units in each layer may grow exponentially. Learning systems based on Fuzzy Logic are able to emulate human knowledge and to deal with uncertainty. Under certain non-restrictive conditions they are universal interpolators [11].

All these concepts were integrated in an industrial process operation system, which was applied to a glass furnace. The major contributions of this paper are the description of a general hierarchical architecture for the operation of industrial processes, the development of an expert controller for glass furnaces, and the formalisation of a model learning methodology for a glass furnace. The paper is organised as follows: in Section 2 the architecture of the industrial process operation system is described; the algorithms used are detailed in Section 3; in Section 4, the case study (the operation of a glass furnace) is introduced, and finally, in Section 5, the experimental results are presented. Conclusions are drawn in Section 6.

2 An Architecture for the Operation of Industrial Processes

The architecture proposed for the operation of industrial processes is based on a hierarchical scheme, whose levels are denominated as **Operation Goals**, **Organisation/Coordination**, **Execution** and **Analysis**. In the next sections those levels will be briefly explained.

2.1 Operation Goals

Operation goals may be seen as the principles that guide process operators, most of the time translated in statements such as “maximise final product quality” or “minimise energy costs”. The achievement of a goal may be seen as the resolution of an optimisation problem, or, equivalently, as the minimisation of a cost function *f*. Several goals may lead to the presence of concurrent solutions, where the improvement in one objective will give rise to the degradation in another. This justifies the need of multiobjective optimisation techniques for the attainment of trade-off solutions.

2.2 Organisation/Coordination

This level generates process set points and parameters (e. g., furnace temperature, valve opening in a gas duct, geometric parameters in vision control systems) from the operation goals. On the other hand, it is also responsible for checking the process safeguard. It is composed of an expert controller split into a **process multiobjective optimisation system** and a **fault detection system**. This paper will concentrate only in process optimisation. The information flow between the different levels of the operation system is represented in Figure 1.

Process Multiobjective Optimisation System

This system receives as inputs the operation goals and generates process set points and parameters. The analysis block carries out the feedback of its actions. Set-point generation corresponds to the resolution of a multiobjective optimisation problem.

Fault Detection System

Due to external causes, equipment malfunctioning or human errors, process performance may be degraded or get over safety limits. In industrial processes, the automatic detection, diagnosis and identification of faults may be a crucial factor for the adequate response of operation systems. However, this is not under the scope of this paper.

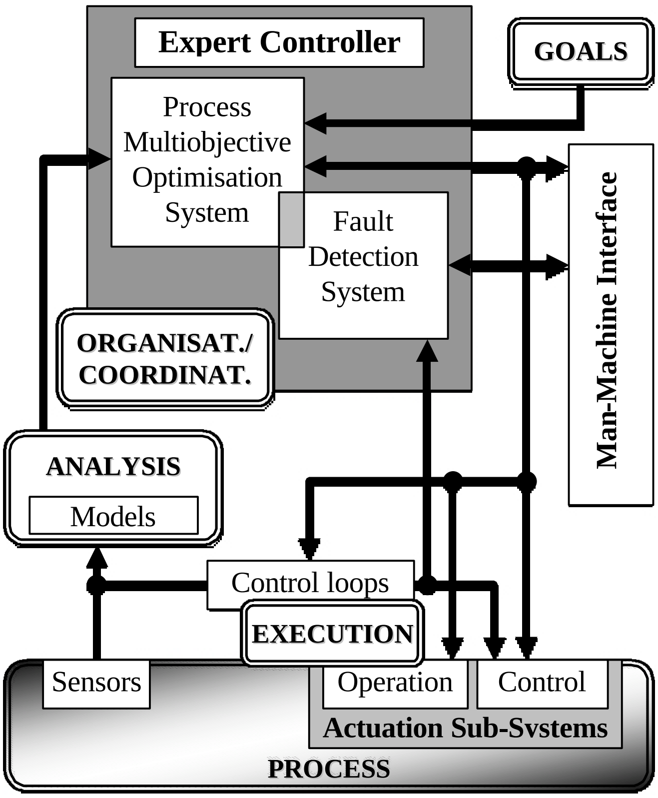

Figure 1: Information flow between different levels of the operation system

2.3 Execution

The Execution level is composed of the sub-systems through which operators act on the process. These are the control loops or directly the actuators. Control loops are generally implemented using programmable logic controllers (PLC's) or process controllers.

2.4 Analysis

The analysis block is responsible for closing the information loop with the expert controller of the organisation/coordination level. It includes all the existing models of the process. The main difficulties in the automatic control and operation of industrial processes are due to errors in process parameterisations and variables measurements, coupling of manipulated variables, presence of non linearities and time constants of different orders of magnitude. It is natural to expect the absence of analytical models, or, if these exist, to expect them to be so complex that they are of no practical use. This motivates the need to endow the analysis block with a learning system, in order to build process models iteratively from actual data.

3 Methods and Algorithms

3.1 Multiobjective Optimisation with Restrictions Based on Genetic Algorithms

An algorithm that solves the multiobjective optimisation problem, using Gas, is introduced in this section. This algorithm is based on the MOGA (Multiobjective Genetic Algorithm) [4].

Pareto Formalism

The approach used is based on the Pareto formalism, which relates objective vectors, such as:

$$\mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_N(\mathbf{x})\right)$$

where $\mathbf{x}$ is a decision vector in the universe Ω^m:

$$\mathbf{x} = (x_1, x_2, \ldots, x_m)$$

Without loss of generality, it will be assumed that the multiobjective optimisation problem will correspond to the minimisation of the function $\mathbf{f}$, i. e., the minimisation of each of its components. It is now possible to state Pareto definitions:

Definition 1: Pareto Dominance A vector $\mathbf{f}^*$ dominates another vector $\mathbf{f}'$ iff $\mathbf{f}^*$ is partially smaller than $\mathbf{f}'$, i. e.:

$$\forall i \in \{1, \ldots, m\} : \mathbf{f}_i^* \le \mathbf{f}_i' \wedge \exists i \in \{1, \ldots, m\} : \mathbf{f}_i^* < \mathbf{f}_i'$$

and this is denoted by $\mathbf{f}^* \prec \mathbf{f}'$.

Definition 2: Pareto Optimality The solution vector $\mathbf{x}^* \in \Omega^m$ is optimal-Pareto iff there is no other solution $\mathbf{x}'$, such that:

$$\mathbf{f}' = \mathbf{f}(\mathbf{x}') \prec \mathbf{f}^* = \mathbf{f}(\mathbf{x}^*)$$

The set of optimal-Pareto solutions is named non-dominated or non-inferior set. In real problems, a set of solutions, rather than a single solution, exists, defining the so-called trade-off surface.

Restrictions, Priorities and the Preference Vector

Restrictions may appear at two different levels: in decision vectors, where they can be easily satisfied, or at the level of objective functions. For N objectives, they can be defined as:

$$\mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_N(\mathbf{x})\right) \le (h_1, h_2, \ldots, h_N)$$

meaning $f_i(\mathbf{x}) \le h_i, i = 1 \ldots N$, where h_i is the restriction value for objective f_i. If an objective is unrestricted then it is possible to set h_i to $-\infty$. In the process industry, some objectives normally present different priority levels, e. g., quality or ambient concerns may be disregarded by the need to fulfil a great number of orders. In the present case, restrictions in objective functions will be seen as high priority goals [4], or "hard objectives", while the unrestricted objectives, or "soft objectives", will have lower priority.

It is now possible to define the *preference vector* as:

$$\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2)$$

The component $\mathbf{g}_1$ corresponds to the soft objectives, while $\mathbf{g}_2$ corresponds to the hard ones. Each of these components relates priorities with goals in the objectives and is defined as follows:

$$\mathbf{g}_i = \left(g_{i,1}, g_{i,2}, \ldots, g_{i,n_i}\right), i = 1,2$$

such that $n_1 + n_2 = N$. Finally, **g** is defined as:

$$\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2) = ((-\infty, -\infty, \ldots, -\infty), (g_{2,1}, g_{2,2}, \ldots, g_{2,n_2}))$$

where $g_{2,i}$ is equal to the restriction value of the *i-th* objective that presents restrictions. An objective vector **u**, for a particular solution $\mathbf{x_u}$, may be rewritten as $\mathbf{u} = \mathbf{f}(\mathbf{x_u}) = (\mathbf{u}_1, \mathbf{u}_2)$. For the sake of simplicity, it is assumed that the order of the elementary components of **u** is interchangeable.

Application of Genetic Algorithms

A detailed description of GAs can be found in [5]. In this paper, the individuals are formed by the concatenation of the arguments of the optimisation problem, $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. The genetic operators used are selection/reproduction, crossover, mutation and an elitism strategy. The evolution of the populations is guided by a fitness function that works on the individuals rank in a population. This is defined by the preferability relation [4], which plays a major role in the optimisation algorithm. Niche formation techniques are also considered, namely fitness sharing and mate restrictions.

3.2 Automatic Learning by Examples Based on Fuzzy Rules

The algorithm presented in this section builds process models from real examples. These are composed of data sets of inputs and desired output, $(x'_0, x'_1, \ldots, x'_n \mid y')$. The algorithm used is based on the learning by clusters algorithm [1].

Introduction

The models are based on IF-THEN rules whose syntax is:

$$R^{(l)}: \text{IF} \underbrace{\left(\bigcap_{i=1}^{n} x_i \text{ is } A_i^{(l)} \right)}_{\text{antecedent part}} \text{THEN } \underbrace{y = w^{(l)}}_{\text{consequent part}}$$

where $R^{(l)}$ is the *l-th* rule from *c* possible ones; x_i is the *i-th* fuzzy variable from the *n* that compose the antecedent part, defined in some universe of discourse (UoD); $A_i^{(l)}$ is the linguistic term defined by the fuzzy set assigned to variable x_i in the *l-th* rule, and characterised by the membership function $m_{A_i^{(l)}}(x_i)$; y is the model output; $w^{(l)}$ is a numeric value, learned from data over time.

All the membership functions used are Gaussian and uniformly distributed over the UoD. The inference mechanism applied is the centroid method [1,11].

Algorithm

The recursive version of the algorithm in [1] is summarised in the sequel. First of all, the original algorithm initialises the rules. When a new example $(x'_0, x'_1, \ldots, x'_n \mid y')$ is obtained:

1. Start in the first rule, $l \leftarrow 1$
2. Evaluate the membership degree of all input variables in the linguistic terms that build rule *l*, $m_{A_i^{(l)}}(x'_i), i = 1, \ldots, n$.
3. Evaluate the membership degree of the new example in rule *l*: $S1^{(l)} \leftarrow \prod_{i=1}^{n} m_{A_i^{(l)}}(x'_i)$
4. Weight the output with $S1^{(l)}$: $S2^{(l)} \leftarrow S1^{(l)}.y'$
5. With $\omega^{(l)} = (Num^{(l)} / Den^{(l)})$, where $Num^{(l)}$ and $Den^{(l)}$ were obtained in the last iteration, make the updates:
$$Num^{(l)} \leftarrow Num^{(l)} + S2^{(l)}$$
$$Den^{(l)} \leftarrow Den^{(l)} + S1^{(l)}$$
$$w^{(l)} \leftarrow Num^{(l)} / Den^{(l)}$$
6. Proceed into the next rule, $l \leftarrow l + 1$
7. Go to step 2.

4 Case Study: Operation of a Glass Furnace

The process operation architecture introduced before was applied to a real furnace, under the project NOVOVIDRO [6]. The glass furnace built under NOVOVIDRO is of the recuperative type, cross-fired, with a pull of about 11ton/day, and works with natural gas. It has two recuperators, and two firing zones in the melting chamber.

4.1 Process

The glass production process can be briefly summarised as follows [10]: the selected raw materials are mixed and introduced in the glass furnace. After their melting, the resulting glass is gathered and worked. Finally, it is cooled in a controlled way, so that it can be finished.

4.2 Operation System

It is now possible to design an operation system for the glass furnace, which is depicted in Figure 2. This system is detailed in the sequel.

Operation Goals

In the glass industry, five criteria can be defined to optimise the performance of furnace operation [3,8]:

- Glass quality maximisation;
- Thermal efficiency maximisation;
- Furnace and refractory lifetime maximisation;
- Pollutant production and emission minimisation;
- Energy consumption cost minimisation.

Cost functions are defined to quantify these goals.

Glass quality maximisation Glass quality is quantified by the amount of defects in the glass. These may be of three types: blister, stone and cord [7]. The cost functions are defined as:

$$f_1(\mathbf{x}) \equiv D_B(\mathbf{x}),\ \ f_2(\mathbf{x}) \equiv D_S(\mathbf{x}),\ \ f_3(\mathbf{x}) \equiv D_C(\mathbf{x})$$

where D_B, D_S and D_C are, respectively, the glass percentage of blister, stone and cord, and **x** is the vector that characterises the furnace operation point, to be described later. Defining the maximum admissible amount of defects as g_B, g_S and g_C, these values will act as restrictions to these objectives.

Thermal efficiency maximisation There are models for furnace efficiency, depending on the flows of gas and air used in combustion, $\dot{q}_G$ and $\dot{q}_A$, which are related by a constant, $K_{A/G}$. The less the temperature required by the glass, the more efficient will be the heat transfer, and the less fuel will be required [8]. Therefore:

$$f_4(\mathbf{x}) \equiv \dot{q}_G$$

Furnace and refractory lifetime maximisation This goal is achieved ensuring the correct balance of the pressure, glass level and temperature control loops, provided with correct set points [10]. No additional objective function is needed.

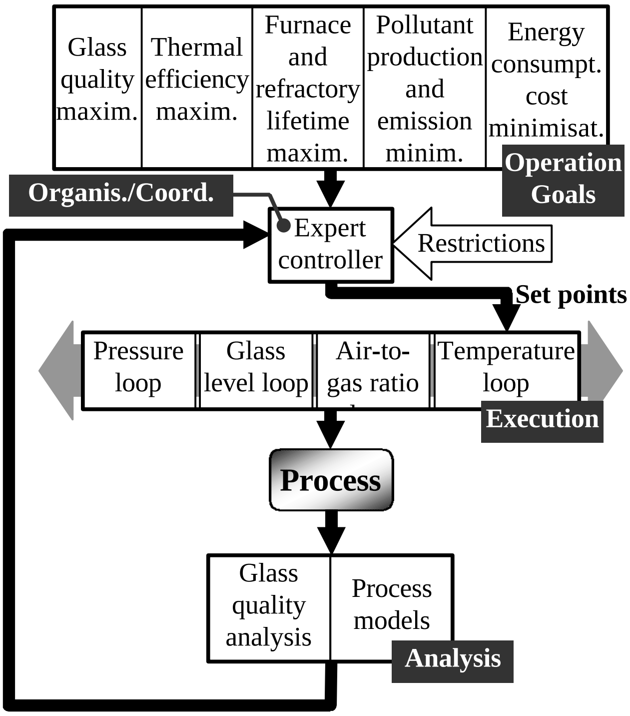

Figure 2: Glass furnace operation system hierarchical architecture

Pollutant production and emission minimisation At the furnace level, the main pollutant is NO_X, which is directly related with flames temperature. Cost functions are:

$$f_5(\mathbf{x}) \equiv T_1,\ f_6(\mathbf{x}) \equiv T_2$$

Energy consumption cost minimisation Defining á as the natural gas tariff, another cost function would be á.$\dot{q}_G$. However, this is the same as f_4, scaled by a constant factor.

The vector **x** has the following components:

$x_1 \equiv N$, batch recipe number
$x_2 \equiv P$, furnace draft $\left[\text{kg h}^{-1}\right]$
$x_3 \equiv T_{Rec1}$, recuperator 1 output air temp.$\left[°\text{C}\right]$

$x_4 \equiv T_{Rec2}$, recuperator 2 output air temp. $[°C]$
$x_5 \equiv T_{Crown}$, furnace crown temperature $[°C]$
$x_6 \equiv T_{BRear}$, furnace bottom temp. (rear) $[°C]$
$x_7 \equiv T_{BFront}$, furnace bottom temp. (front) $[°C]$
$x_8 \equiv \dot{q}_G$, combustion gas flow $[m^3 h^{-1}]$
$x_9 \equiv T_1$, zone 1 temperature $[°C]$
$x_{10} \equiv T_2$, zone 2 temperature $[°C]$

The restriction vector is defined as $(g_B, g_S, g_C, -\infty, -\infty, -\infty)$.

Organisation/Coordination

This level generates firing zones temperature set points, as the result of the Process Multiobjective Optimisation System (PMOS). An empirical analysis of the process led to the definition of the following furnace models, according to the **x** components [8]:

$$x_3 = m_{x_3}(\mathbf{x}) = m_{Rec1}(x_1, x_2, x_9, x_{10})$$
$$x_4 = m_{x_4}(\mathbf{x}) = m_{Rec2}(x_1, x_2, x_9, x_{10})$$
$$x_5 = m_{x_5}(\mathbf{x}) = m_{Crown}(x_1, x_2, x_3, x_4, x_9, x_{10})$$
$$x_6 = m_{x_6}(\mathbf{x}) = m_{BRear}(x_1, x_2, x_3, x_4, x_5, x_9, x_{10})$$
$$x_7 = m_{x_7}(\mathbf{x}) = m_{BFront}(x_1, x_2, x_3, x_4, x_5, x_6, x_9, x_{10})$$
$$x_8 = m_{x_8}(\mathbf{x}) = m_{Gas}(x_1, x_2, x_3, x_4, x_5, x_9, x_{10})$$

The following relations were also considered, as explained before:

$$f_1 = D_B = f_1(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$$
$$f_2 = D_S = f_2(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$$
$$f_3 = D_C = f_3(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$$
$$f_4 = \dot{q}_G = x_8$$
$$f_5 = T_1 = x_9$$
$$f_6 = T_2 = x_{10}$$

It is assumed that control loops are able to achieve the imposed set points. The PMOS must solve the following multiobjective optimisation problem:

Compute x_9^* and x_{10}^* such that $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), f_4(\mathbf{x}), f_5(\mathbf{x}), f_6(\mathbf{x}))$, subject to the furnace models and to the restriction vector $(g_B, g_S, g_C, -\infty, -\infty, -\infty)$, is minimised.

Application of the multiobjective optimisation algorithm The PMOS receives as inputs the actual process parameters, (x_1', x_2'), the process models, m_{Rec1}, m_{Rec2}, m_{Crown}, m_{BRear}, m_{BFront}, m_{Gas}, the cost function vector, $\mathbf{f}(\mathbf{x})$, and the restriction vector $(g_B, g_S, g_C, -\infty, -\infty, -\infty)$, which, as seen before, is converted in an equivalent preference vector given by $((-\infty, -\infty, -\infty), (g_B, g_S, g_C))$. The PMOS output is the solution vector (x_9^*, x_{10}^*), i.e., the temperature set points. At each stage of the genetic algorithm a population of candidate solutions is generated. A generic solution, denoted by (x_9', x_{10}'), is evaluated according to the values of x_1' and x_2' and follows the steps:

1. Evaluate recuperators temperatures,
   $\hat{x}_3 = m_{Rec1}(x_1', x_2', x_9', x_{10}')$
   $\hat{x}_4 = m_{Rec2}(x_1', x_2', x_9', x_{10}')$
2. Estimate crown temperature,
   $\hat{x}_5 = m_{Crown}(x_1', x_2', \hat{x}_3, \hat{x}_4, x_9', x_{10}')$
3. Estimate bottom rear temperature,
   $\hat{x}_6 = m_{BRear}(x_1', x_2', \hat{x}_3, \hat{x}_4, \hat{x}_5, x_9', x_{10}')$
4. Estimate bottom front temperature,
   $\hat{x}_7 = m_{BFront}(x_1', x_2', \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, x_9', x_{10}')$
5. Estimate natural gas flow,
   $\hat{x}_8 = m_{Gas}(x_1', x_2', \hat{x}_3, \hat{x}_4, \hat{x}_5, x_9', x_{10}')$
6. Evaluate solution through the computation of the cost functions,
   $f_1 = f_1(x_1', x_2', \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8, x_9', x_{10}')$
   $f_2 = f_2(x_1', x_2', \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8, x_9', x_{10}')$
   $f_3 = f_3(x_1', x_2', \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8, x_9', x_{10}')$
   $f_4 = \hat{x}_8$
   $f_5 = x_9'$
   $f_6 = x_{10}'$

The algorithm generally provides a set of non-dominated solutions. The selection of one particular solution is based on the fact that stable operation is a requisite for quality glass production. Then, the picked solution is the one closest to the current one, in an Euclidean sense, for smoothness of operation.

Analysis

This level has two tasks, namely, to supply production data for optimisation purposes, and to build and update process models. The Learning System carries out the latter.

Data for optimisation The data that the optimisation system needs is the batch composition number and the furnace expected average glass draft. The amount of glass expected to be produced in one day is given by:

$$\hat{Q}_{Day} = \sum_{i=1}^{N_{Day}} n_i p_i \qquad (1)$$

where n_i is the amount of type i products to be produced, N_{Day} is the number of different products, and p_i is the average weight of type i products. The expected average glass draft is:

$$\hat{\bar{P}}_{Day} = \frac{\hat{Q}_{Day}}{\Delta T_{Day}} \qquad (2)$$

where ΔT_{Day} is the duration of furnace labouring.

Learning System This system builds and updates furnace models, namely, m_{Rec1}, m_{Rec2}, m_{Crown}, m_{BRear}, m_{BFront}, m_{Gas}, and cost functions f_1, f_2 and f_3. The examples received by the system consist of data vectors, $\mathbf{y}^e$, relative to pre-defined production periods (one shift or half shift). The components of $\mathbf{y}^e$ are the batch composition number, N, the estimated average furnace draft, $\hat{\bar{P}}$, the recuperator 1 and recuperator 2 average output air temperatures, $\bar{T}_{Rec1}$ and $\bar{T}_{Rec2}$, the furnace average crown temperature, $\bar{T}_{Crown}$, the furnace average rear and front bottom temperature, $\bar{T}_{BRear}$ and $\bar{T}_{BFront}$, the combustion average gas flow, $\bar{\dot{q}}_G$, the zone 1 and zone 2 average temperatures, $\bar{T}_1$ and $\bar{T}_2$, and the estimated percentage of blister, stone and cord, namely $\hat{D}_B$, $\hat{D}_S$ and $\hat{D}_C$. For each model, only the corresponding components of $\mathbf{y}^e$ will be used. Real data coming form the furnace is first low pass filtered, and then its mean value and standard deviation are taken. The mean value is used as an example if the standard deviation is lower than a defined threshold. The values that need to be estimated are the following:

- Shift average furnace draft, $\hat{\bar{P}}$:

$$\hat{\bar{P}} = \frac{\hat{Q}_{Shift}}{\Delta T_{Shift}} = \frac{\sum_{i=1}^{N_{Shift}} n_i p_i}{\Delta T_{Shift}} \qquad (3)$$

  where N_{Shift} is the number of different shift products and ΔT_{Shift} is the shift duration.

- Percentage of glass defects, $\hat{D}_B$, $\hat{D}_S$ and $\hat{D}_C$:

$$\hat{D}_B = \frac{\hat{Q}_B}{\hat{Q}_{Shift}} \cdot 100 = \frac{\sum_{i=r_1}^{r_{N_B}} n_i p_i}{\hat{Q}_{Shift}} \cdot 100 \qquad (4)$$

  where $r_l, l = 1 \ldots N_B$, are the different products that were marked with blister defect, in the end of the production process. These values are supplied, at the end of each shift, by an information system. The other estimates are processed in the same way.

4.3 Information Integration

The operation system receives furnace variables through a process monitoring integrated system, based on SCADA software (supervisory control and data acquisition) Omron SCS-Sysmac V2.0 that interfaces with a programmable logic controllers network. The information related with the final product (amount of glass produced and glass defects) is received through an information system that monitors production.

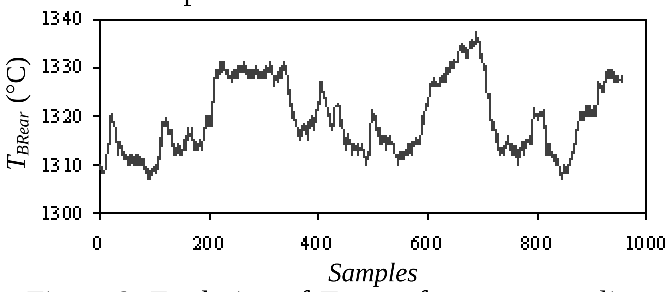

Figure 3: Evolution of T_{BRear}, after over sampling

5 Experimental Results

The furnace was monitored from 26 May to 4 June 2000. Since the sample period used was 1h, the original signals were linearly over sampled and corrupted with noise, in order to simulate a sample period of 15min, more adequate (see Figure 3 for

T_{BRear}). The difficulties that a human operator would have to find out the correlations between several process variables are evident from Figure 4. However, some correlation seems to exist between T_{Crown} and T_1. By the time this work was carried out, the production information system was not fully working. Due to this fact, and in order to test the operation system, some values had to be artificially assigned, namely, the amount of glass produced, Q_T, and the amount of glass with defects, Q_B, Q_S and Q_C. The first one is randomly generated, taking into account a production between 400 and 1300 kg in each shift. The estimate of the average draft, P, is then taken dividing this value by the shift duration, ΔT_{Shif}.

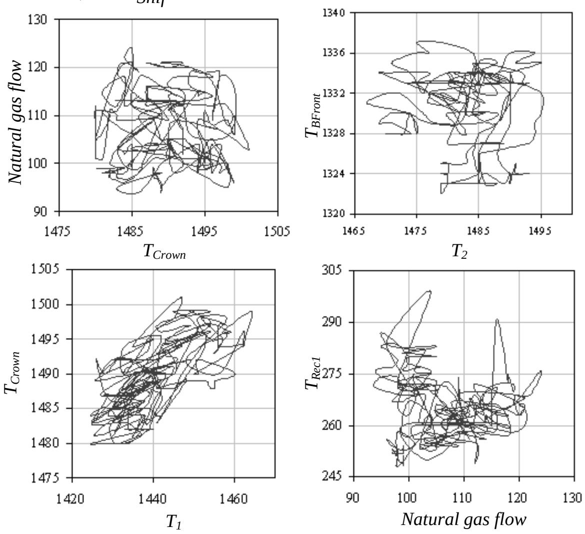

Figure 4: Variables used to build examples

The amount of glass with the three distinct defects is generated according to:

$$Q_B = 10\cdot\left(\frac{P-P_{\min}}{\Delta P}\right)+10\cdot e^{\frac{\ln(0.0001)}{\Delta T_1^2}\left(T_1-\frac{T_{1\min}+T_{1\max}}{2}\right)^2}+ +10\cdot\left(\frac{T_{2\min}-T_2}{\Delta T_2}+1\right) \quad (5)$$

$$Q_S = 15\cdot\left(\frac{P-P_{\min}}{\Delta P}\right)+10\cdot\left(\frac{T_{1\min}-T_1}{\Delta T_1}+1\right)+ +10\cdot\left(\frac{T_{2\min}-T_2}{\Delta T_2}+1\right) \quad (6)$$

$$Q_C = 10\cdot\left(\frac{P_{\min}-P}{\Delta P}+1\right)+10\cdot\left(\frac{T_{1\min}-T_1}{\Delta T_1}+1\right)+ +5\cdot\left(\frac{T_{2\min}-T_2}{\Delta T_2}+1\right) \quad (7)$$

These simple models, represented in Figure 5 for P = 414 kg h^{-1}, establish some basic relations, as the increase in the amount of blister with glass draft, or its decrease with zone two temperature. The estimate of blister percentage, for instance, is then taken dividing Q_B by the ΔT_{Shif}.

5.1 Furnace Models Learning

The UoDs of the different variables are indicated in Table 1. The thresholds used to validate examples, determined by signal observation, are indicated in the same table.

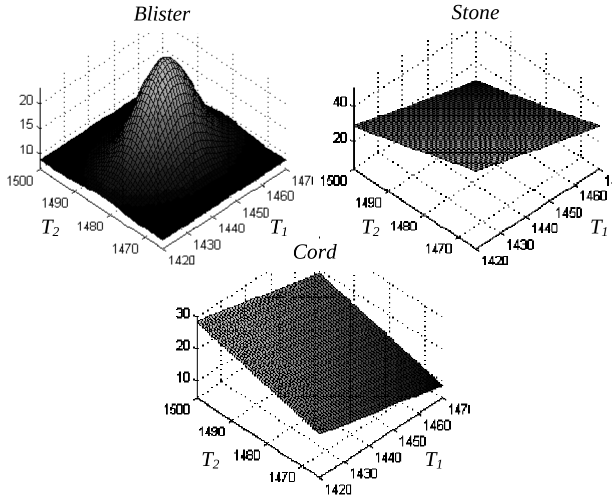

Figure 5: Artificial models of glass defects, for a draft P = 414 kg h^{-1}

Variable	Lower limit	Upper limit	Unit	Threshold (std. dev.)
P	0	473	Kg h^{-1}	-
T_{Rec1}	245	305	°C	5.5
T_{Rec2}	230	325	°C	5.5
T_{Crown}	1475	1505	°C	3
T_{Brear}	1300	1340	°C	2
T_{Bfront}	1320	1340	°C	2
$\dot{q}_G$	90	130	m^3 h^{-1}	2.5
T_1	1420	1470	°C	3.5
T_2	1465	1500	°C	3.5

Table 1: Universes of Discourse

The number of membership functions is empirically determined according to the variation coefficient (ratio between standard deviation and mean value, in percentage) of the available data. The smaller the variation coefficient, the less membership functions assigned to a variable. The dimension of each model is defined by the product of the number of membership functions of all the variables involved times the number of batch compositions (the only crisp variable defined).

From the 50 possible examples, only 16 were valid. The models built by the learning by examples algorithm are represented in Figure 6. It is clear that some of the trends of the mathematical models were captured. However, some discrepancies still exist, especially in the blister model. In fact, for T_2 small, the model does not reflect the original one. This is essentially due to the small number of examples. This method makes possible to get some physical insight of the furnace behaviour under different conditions, just by looking at the surfaces defined by the learned models. See, e. g., the influence of the firing zones temperature in the recuperator 2 temperature, under a certain draft (Figure 7).

Variable	Variation Coefficient	Nº of Membership Functions
P	54.39	7
T_{Rec1}	3.00	5
T_{Rec2}	4.25	5
T_{Crown}	0.28	3
T_{Brear}	0.58	3
T_{Bfront}	0.31	3
$\dot{q}_G$	7.11	5
T_1	0.55	3
T_2	0.40	3

Table 2: Membership functions assignment

5.2 Furnace Performance Optimisation

With furnace models available, it is possible to optimise furnace performance. This is done whenever production planning is available. This implicitly defines the amount of glass to be produced, in a certain day, Q_T. Once again, the average draft will be the ratio of Q_T by the total furnace labour period.

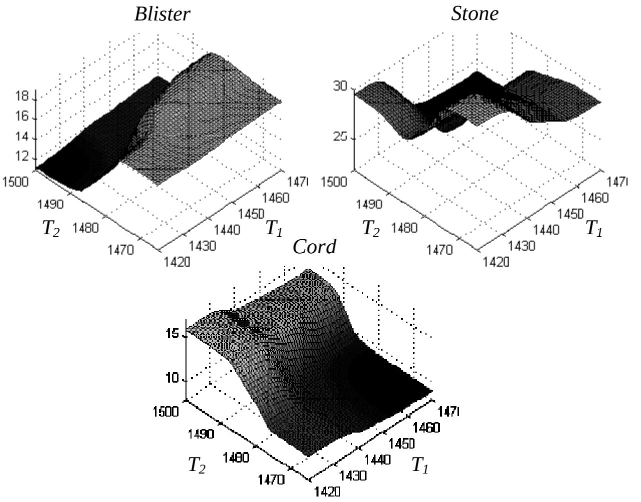

Figure 6: Learned models of glass defects, for a draft P = 414 kg h^{-1}

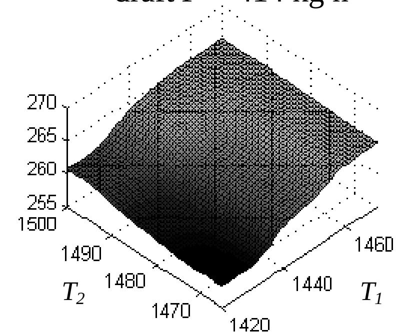

Figure 7: Learned model of recup. 2 temp., P = 414 kg h^{-1}

The experiences presented in Table 3 were performed in order to test the multiobjective optimisation algorithm, and determine furnace temperature set points. The results obtained are listed in Table 4. The different experiences were:

1. Optimise blister percentage in glass. The value of f_1 thus expresses the minimum of this function.
2. Global optimisation of glass quality. The decrease in stone and cord implied an increase in blister and in gas consumption.
3. Same as before, but with stone restricted to 25%. Blister and gas consumption were improved and stone degraded.
4. Simultaneous optimisation of blister percentage and gas consumption. The result is the same as 1.
5. Same as before, with the additional minimisation of T_1. The increase in f_1 and f_4 made possible the decrease in f_5.
6. Global optimisation with restrictions in all the objectives. Objectives f_1, f_3, f_5 and f_6 achieve goals, while the others do not. This is justified by the fact that, most plausibly, there is no solution for this problem.

	Used Cost Functions						**Restrictions**					
	f_1	f_2	f_3	f_4	f_5	f_6	f_1	f_2	f_3	f_4	f_5	f_6
1	√											
2	√	√	√									
3	√	√	√					25				
4	√			√								
5	√			√	√							
6	√	√	√	√	√	√	20	25	15	105	1450	1475

Table 3: Multiobjective optimisation problems

6 Conclusions

In this paper, an architecture for the operation system of industrial processes, with application to a glass furnace, is proposed. This architecture is based on an expert controller with two main sub-systems: process optimisation and fault detection. This paper is only focused on process optimisation. The experimental results presented are based on a blending of real and artificial data. In spite of this fact, and in the available data exiguity, the system is able to capture the main trends and relations between process variables, enabling a multiobjective process optimisation algorithm to determine the optimal set points. Future work will focus on the improvement of the learning algorithms and the development of the fault detection system.

	Results					
	f_1 (D_B)	f_2 (D_S)	f_3 (D_C)	f_4 ($\dot{q}_G$)	f_5 (T_1)	f_6 (T_2)
1	11.3	29.6	15.9	97.6	1470	1465
2	17.9	28.8	9.0	115.4	1421	1500
3	12.4	23.9	13.7	104.7	1468	1500
4	11.3	29.6	15.9	97.6	1470	1465
5	12.3	28.4	15.2	104.9	1450	1465
6	13.0	27.9	14.5	106.0	1449	1473

Table 4: Results of the algorithm runs

References

[1] P. C. Branco, J. A. Dente. "The Application of Fuzzy Logic in Automatic Modelling Electromechanical Systems", Fuzzy Sets and Systems, **95**, nº3 (1998).

[2] B. S. Everitt. *Introduction to Optimization Methods and their Application in Statistics*, Chapman and Hall (1987).

[3] D. Farmer, M. Heitor, J. Sentieiro, A. Vasconcelos. "New Methodology for Furnace Monitoring, Analysis and Control", *Glass Industry*, pp. 10-16, October (1992).

[4] C. M. Fonseca, P. J. Fleming. "Multiple Optimization and Multiple Constraint Handling with Evolutionary Algorithms I: a Unified Formulation", Technical Report of the Dept. of Automatic Control and Systems Eng., University of Sheffield (1995).

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

[6] M. Heitor, P. Ferrão, J. Pina, A. Diogo, R. Mimoso, "Análise do Estado de Desenvolvimento Técnico-Científico do Projecto Novovidro", Internal Report of IN+, Centro de Estudos em Inovação, Tecnologia e Políticas de Desenvolvimento, Instituto Superior Técnico (1999).

[7] J. Navarro. *El Vidrio, 2ª Edicion*, Consejo Superior de Investigaciones Científicas, Fundacion Centro Nacional del Vidrio, Madrid (1991).

[8] J. Pina. *Operação e Controlo de um Forno de Vidro*, MSc. Thesis, Electrical and Computers Engineering Dept., Instituto Superior Técnico (2002).

[9] S. Russel, P. Norvig, *Artificial Intelligence, a Modern Approach*, Prentice-Hall International Editions (1995).

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

[11] L. X. Wang. "Fuzzy Systems are Universal Approximators", *Proc. of IEEE International Conference on Fuzzy Systems*, San Diego, USA (1992).