utad Universidade de Trás-os-Montes e Alto Douro

Workshop on Management, Identification and Control of Agriculture Buildings

JANUARY 26, 2001
VILA REAL, PORTUGAL

Proceedings

Solar Radiation Prediction using Classical and Evolutionary Techniques†

João Paulo Coelho[1], J. Boaventura Cunha[2,3] and P.B. de Moura Oliveira[2,3]

[1] ITIDAI- Instituto de Trás-os-Montes para a Investigação e Desenvolvimento Agro-Industrial
[2] UTAD- Universidade de Trás-os-Montes e Alto Douro, Departamento de Engenharias
[3] ICETA - UTAD- Instituto de Ciências e Tecnologias Agrárias e Agro-Alimentares da UTAD
5000 Vila Real
Portugal
Tel. +351259350352
e-mail. jpcoelho@pol.pt

ABSTRACT

Modern greenhouse climate controllers are based on models to simulate and predict greenhouse environment behaviour. These models must be able to describe indoor climate process dynamics, which is as a function of both the control actions taken, and outside climate. Moreover, if predictive or feedforward control techniques are to be applied, it will be needed to employ models to describe and predict the weather (solar radiation, air temperature, etc.). In the present paper, classical and evolutionary techniques are applied to solar radiation time-series prediction within a Greenhouse located in UTAD campus. Experimental results are provided and some comparisons between the two types of methods drawn.

Keywords: Greenhouse, System Identification, Evolutionary Algorithms, Regressive Methods.

1. INTRODUCTION

The goal of identification is to build mathematical models of dynamic systems from observed data based on prior physical or subjective understanding about the process. System identification techniques are not a collection of *ad-hoc* procedures. In the past decades a strong theoretical framework was developed and, in conjunction with digital computer's trivialization, it is possible to apply these techniques in many fields of science such as: control, economics and biology in order to predict the response of unknown systems.

The identification procedure involves: the selection of a class of structures, appropriate compilation of data, and the choice of a criterion to evaluate the degree of fitness of the model. The following basic steps will be repeated iteratively until a satisfactory model has been found, [Åström and Wittenmark (1997)]

- Selection of a model structure
- Parameter estimation
- Model Validation

Model structure is often derived from prior knowledge of the process and its choice is usually a trade off between complexity and representativeness of the system dynamics. A process should be represented by a hierarchy of models, which can be used for prediction and control of "Black-Box" systems.

† This work was sponsored by the INTERREG II European Program

Parameter estimation is the core of system identification. The objective is to find the parameters a_α and b_β of a parametric model such as of the discrete model described by equation (1):

$$y(k) = \sum_{\alpha=1}^{n} a_\alpha \cdot y(k-\alpha) + \sum_{\beta=0}^{m} b_\beta \cdot u(k-\beta) \quad (1)$$

In which $y(k),u(k)$ represent the system output and input at the present time k and $y(k-\alpha)$ and $u(k-\beta)$ the past outputs and inputs respectively. In the case of a time-series model, the parameters b_β and the signal $u(k-\beta)$ are equal to zero.

There are many tools to find these parameters, some of them based on classic approaches like gradient-based techniques, and others supported by evolutionary algorithms. The former will be described in section 2 and the latest in section 3.

Model validation is usually made by comparison between the model response to independently observed data (records that were not used for parameter estimation) and the real response. Complementarily there is a collection of statistical properties that can be employed to compute the degree of representativeness of the model (ex. Akaike's Final Prediction-Error Variance (*FPE*)) [Akaike (1974)].

2. SYSTEM IDENTIFICATION USING LEAST SQUARES METHODS

In system identification, the mathematical model is build based on the input/output data collected from the process. Data records are of enormous importance and represent the first "stone" in dynamic model construction. The measured signals must have some effect on the system, *i.e.*, they must be persistently exciting. This implies that the order of the input signal must be at least equal to the order of the process to be identified. The spectral contents of the signals should have the necessary components to stimulate the internal dynamics of the system. Usually measured signals are pre-processed before estimation in order to remove noise, mean or trends.

The model order selection is a difficult issue than can be established by some prior knowledge of the process (for example based on physical laws), or by some model order assessment method such as Akaike's Information Criterion (*AIC*) [Akaike (1974)], Minimum Description Length (*MDL*) [Rissanen (1978), Schwarz (1978)] or Subset Selection [Miller (1990)] that select the variables that have the largest weight in the system representation.

2.1 OFF-LINE PARAMETER ESTIMATION

Considering the auto-regressive with exogenous inputs (ARX) model represented by equation (2):

$$y(k) = a_1 \cdot y(k-1) + \ldots + a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m) + e(k) \quad (2)$$

where y is the system output (dependent variable), u is the system input (independent variable) and e is the noise or modelation error, which is assumed to have some statistical properties (ex. white noise with zero mean, normal distribution and variance σ^2) [Ljung (1987)].

This model is a representation of a linear regression. Linear regression concerns to a parametric representation in which the parameters a_α, $1 \le \alpha \le n$ and b_β, $0 \le \beta \le m$ are linear and can be estimated by an off-line *Least Squares* method.

Equation (2) may be rewritten in a matrix form by:

$$Y = \Phi \cdot \theta + E \tag{3}$$

where Φ is the regression matrix that is build using the available data and knowledge about the model structure, θ is the parameter vector formed by the coefficients to be estimated and E is the error vector associated to the model.

If we apply the same input u to the process and to the mathematical model (Figure 1), both the outputs y and $\hat{y}$ can be compared by evaluating the model error e. This is then used to infer the degree of approximation of the model to the real process as a cost function in a minimization problem.

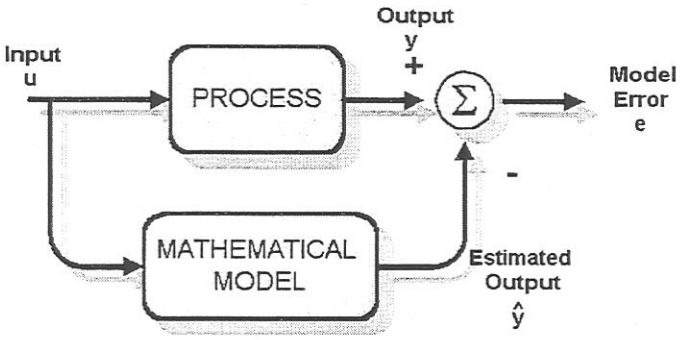


Figure1: Estimation Error.

The estimation by the *Least Squares* method is based on the minimization of the function expressed by equation:

$$J(\theta) = \frac{1}{2} \cdot \sum_{i=0}^{n-1} [y(k-i) - \hat{y}(k-i)]^2 \tag{4}$$

In witch n is the number of data samples and

$$y(k-i) - \hat{y}(k-i) = e(k-i) \tag{5}$$

is the error model evaluated in time $t = (k-i) \cdot T_s$ where T_s is the sampling interval.

The objective function (4) may be rewritten as:

$$J(\theta) = [Y - \Phi \cdot \theta]^T [Y - \Phi \cdot \theta] \tag{6}$$

By differencing equation (6) in order to θ and equalizing it to zero we find that the estimated parameter vector, $\hat{\theta}$, can be obtained by:

$$\hat{\theta} = \left(\Phi^T \cdot \Phi\right)^{-1} \cdot \Phi^T \cdot Y \tag{7}$$

The quality of the estimation using the *Least Squares* method depends on noise statistical properties and on the richness of the information enclosed in Φ (if the information are rich then $\Phi^T\Phi$ will be non-singular). In addition, the error must be uncorrelated with Φ otherwise the estimated parameters will not converge to the real coefficients. Correlation-error methods such as *Instrumental Variables* [Söderström, et al. (1989)] or *Koopmans-Levin* [Fernando, *et al.* (1985)] can be used to address this problem.

2.2 ON-LINE PARAMETER ESTIMATION

The least squares method illustrated above requires a block of data for estimation, which is a disadvantage because at each sampling time the size of the block will grow giving rise to more complex and time consuming machine implementation. To bypass this problem and taking into account that at each sampling time only a new line is added to Φ and Y a recursive approach to the *Least Squares* algorithm can be used.

At sample *k* the process model defined in (2) can be written as:

$$[y(k)] = [\Phi(k)] \cdot \theta(k) + [e(k)] \tag{8}$$

When a new sample is available equation (8) is updated as shown in (9)

$$\begin{bmatrix} y(k) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} \Phi(k) \\ \varphi^T(k+1) \end{bmatrix} \cdot \theta(k+1) + \begin{bmatrix} e(k) \\ e(k+1) \end{bmatrix} \tag{9}$$

where

$$\varphi(k+1) = [y(k\text{-}1) \ \ldots \ y(k\text{-}n) \ \ldots u(k) \ u(k\text{-}1) \ldots u(k\text{-}m)] \tag{10}$$

Likewise to (7) in the off-line least squares algorithm, $\theta(k+1)$ can be estimated by:

$$\hat{\theta}(k+1) = \left[\begin{pmatrix} \Phi(k) \\ \varphi^T(k+1) \end{pmatrix}^T \begin{bmatrix} \Phi(k) \\ \varphi^T(k+1) \end{bmatrix} \right]^{-1} \begin{bmatrix} \Phi(k) \\ \varphi^T(k+1) \end{bmatrix}^T \begin{bmatrix} y(k) \\ y(k+1) \end{bmatrix} \tag{11}$$

and with proper simplification it get the form of the recursive *Least Squares* estimation law as follows:

$$K(k+1) = \frac{P(k) \cdot \varphi(k+1)}{1 + \varphi^T(k+1) \cdot P(k) \cdot \varphi(k+1)} \tag{12}$$

$$\mathrm{E}(k+1) = y(k+1) - \varphi^T(k+1) \cdot \hat{\theta}(k) \tag{13}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1) \cdot \mathrm{E}(k+1) \tag{14}$$

$$P(k+1) = \left[\mathrm{I} - K(k+1) \cdot \varphi^T(k+1)\right] \cdot P(k) \tag{15}$$

where K is the Kalman estimator gain, P is the covariance of the parameter-estimation error, and E is the one-step ahead error. At each sampling instant a measurement and a parameter vector update takes place.

An important advantage of on-line identification techniques is that they can handle the case in which the process parameters are time varying, but due to the persistency of the incoming information, the algorithm gain tends to zero, and so the capacity of tracking parameters variation is gradually lost. The solution is discarding old samples as they become obsolete. This procedure belongs to a class of algorithms named forgetting methods. A well known method is based on the exponential forgetting principle, where the information discount take place uniformly in all directions of the parameter space. This approach is not suitable if the information is non-persistent, because the covariance matrix P grows without bond (covariance windup), leading to large estimation errors. Other methods are available to avoid this and others problems, like directional forgetting [Parkum (1992)] and selective forgetting methods [Parkum (1992)].

3. EVOLUTIONARY BASED ALGORITHMS

3.1. GENETIC ALGORITHMS

Genetic Algorithms (GAs) [Holland (1975)] [Goldberg (1989)] are search techniques based on natural selection mechanisms. The natural selection theory has his foundations on the deviations between organisms of the same population. Darwin's theory of evolution (1859) states that the population elements with characteristics that best adapt to the surrounding environment will be those who's probability of surviving and procreate will be higher, passing those characteristics that turn they best to their offspring. Generation after generation the accumulation of little variations will consequently change the population characteristics (on other words: evolution).

The information about the characteristics of an individual is coded in genes and the descendents will have in his genome a combination of the genetic material of the two progenitors. The basic genetic algorithm is illustrated by the flowchart of Figure 2. In the first stage a population with N elements is initialised randomly. In the basic algorithm each element of the population is characterized by a genotype and a phenotype. The genotype is the genetic code and may be represented by a binary string, and the phenotype represents the

degree of fitness of an individual and it is closely related to the genotype. At each generation, the fitness of every chromosome is computed, and this indicates how well the corresponding population element performs. The next step is to select $N/2$ pairs of progenitors. The selection operator is the direct reflection of the survival of the fittest. In this stage individuals with higher fitness are more likely to be selected and consequently to disseminate their genetic information. The application of this concept is normally carried out by a *roulette-wheel* selection scheme. In this approach, a pseudo roulette is divided into slashes proportional to the fitness of each individual of the population. The roulette-wheel is then "turned" N times and individuals with highest fitness have higher probability of being chosen to match.

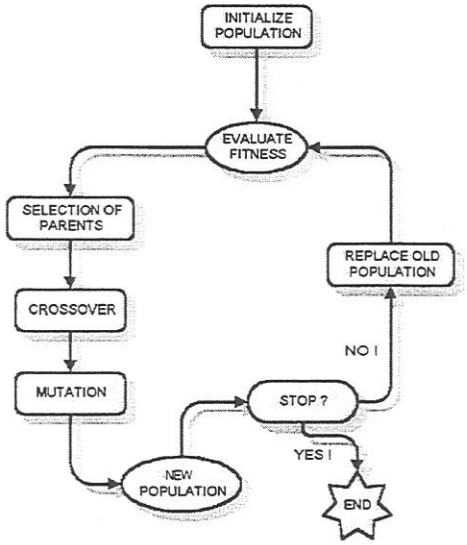


Figure 2: Basic GA cycle.

The lucky couples chosen to mate will pass to the next stage in which their genetic material is interchanged to originate a new set of individuals. This operator is known as crossover. The crossover allows information between couples of parents to be shared given place to two new elements (children). The way crossover operator works depend on the genotype codification (binary, integer or real). Using binary codification the crossover procedure begins by randomly choose a set of crossover points (usually two) and then the genetic information is crossed over as shown in figure 3.

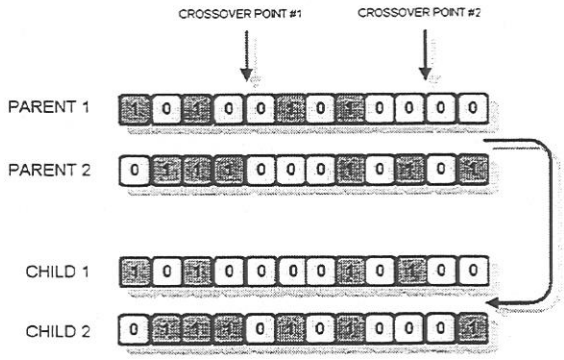


Figure 3: Crossover.

Not all the couples selected by the roulette-wheel are susceptible to cross. There is a probability associated to the crossover operator and normally a pair selected for mating has a

probability between 60 and 80 percent to interchange their information and give "birth" to two new individuals.

Mutation is the operator that follows crossover and has an important role in the GAs. It allows the population to jump out of a local minimum, contributes to maintain population diversity and sometimes it is responsible for a fine-tuning of the solution. The mutation has a very low probability associated, mainly for two reasons; If the mutation is high there is a chance that this operator will destroy good solutions obtained from crossover, the interchange of information will be meaningless and the GA will turn to a random walk search.

If a binary codification is used for the genotype then the mutation is done simply by swapping genes in the chromosome as it is shown in figure 4.

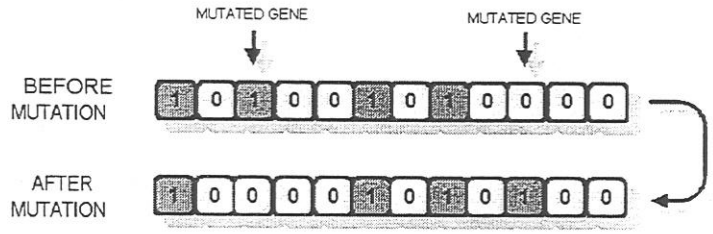


Figure 4: Mutation.

When creating a new population through genetic operators, there is a big chance of losing the best chromosome. A possible way to prevent this to happen is to maintain always the best chromosome across the generations. This strategy is called *Elitism* and it can very rapidly increase the performance of GAs.

After the mutation stage, a stop criterion is tested. It can be the elapsed time, the number of generations or the fitness degree of the solution. If this criterion is not satisfied then the new population will replace the existing one, giving rise to a new cycle.

3.2. POPULATION BASED INCREMENTAL LEARNING (*PBIL*)

Population Based Incremental Learning (PBIL) is a probabilistic search technique that mingles *GAs* and competitive learning, and was introduced by Baluja [Baluja (1994)]. One fundamental characteristic of *PBIL* is the peculiarity of learning based on the fittest member of the population. The *PBIL* is an evolutionary algorithm as *GAs*, but the resemblances almost stop here; *PBIL* doesn't incorporate a crossover operator and all the operations are made on an entity called *Probability Vector (PV)*, thus it is simpler to implement.

The *PV* is composed by an array of K float elements, in which K is the genotype's length of the population. Each element has a value between *0.0* and *1.0* representing the probability of the gene associated to be '0' or '1'. The basic algorithm of *PBIL* is shown in Figure 5.

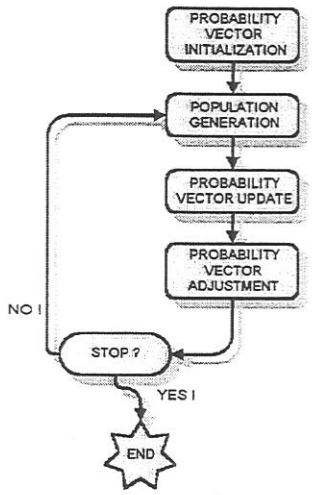


Figure 5: Basic PBIL algorithm proposed by Baluja.

The *PV* initialisation is done by setting all elements to 0.5. This value represents the ignorance concerning the optimal solution of the problem. At each cycle, individuals fitness is computed, and the *PV* is actualised toward the best solution found so far. *PV* updating is governed by expressions (16) and (17)

$$PV[i] = (1 - LR) \cdot PV[i] + LR \cdot BEST[i] \tag{16}$$

If $BEST[i] \neq WORST[i]$ then

$$PV[i] = (1 - NLR) \cdot PV[i] + NLR \cdot BEST[i] \quad 1 \leq i \leq K \tag{17}$$

In which *LR* is the learning rate, *NLR* is the negative learning rate, *BEST* is the best solution and *WORST* is the worst solution found in the present generation. Equation (16) updates the *PV* toward the best solution, and equation (17) drive it away from the worst. As long as the process goes on, *PV* values will diverge from his equilibrium point towards *0.0* or *1.0*.

PV is the key to generate the new population. The value of each of its elements is used as probabilities to create the genotype's pattern of each individual. Suppose a 4 element *PV* as shown below in figure 6:

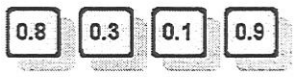


Figure 6: Example of a Probability Vector.

That information tell us that the population will be generated with an 80% probability that the first gene on the chromosome will be a '1', the second gene have a 70% probability to be a '0' and so on. To prevent population premature convergence, and to reduce the algorithm probability of being stuck in a local minimum, a mutation operator is added. The mutation in *PBIL* unlike that on *GAs* is not applied directly to the population individuals but on the *PV*. The following expression adapts that concept to the PBIL algorithm.

$$PV[i] = PV[i] \cdot (1 - ML) + MD * ML \quad 1 \le i \le K \tag{18}$$

Where, ML is the mutation level and represents the amount by which the PV is disturbed and, MD is the mutation direction and defines if the disturbance is to be carried towards 0.0 or 1.0.

After some stop criterion is reached, the algorithm either continues the cycle or exits reporting in his population an element that possess, encoded in his chromosome, the best solution found so far.

3.3. SYSTEM IDENTIFICATION WITH EVOLUTIONARY ALGORITHMS (E.A.).

Evolutionary algorithms such as *GAs* have been successfully applied to system identification [Kristinn and Guy (1992)], [Iba and Kurita (1993)], [De Moura Oliveira and Jones (1998)].

Application of evolutionary algorithms to an optimisation problem begins by choosing the objective function. For system identification the criterion normally used is the sum of squares of the differences between the actually observed data and the computed values. The objective function can be mathematically described by:

$$f_0 = \frac{1}{N} \cdot \sum_{i=1}^{N} {e_i}^2 \tag{19}$$

were N is the number of data samples. There is two important issues that must be retrieved from expression (19).

- As f_0 approaches zero, the closest the mathematical model is to the true process;
- The cost function, unlike the parameters, is non-linear and usually describes a very complex search space.

Taking into consideration the first item, system identification appears as a minimization problem. Because evolutionary algorithms are usually maximization procedures, there is a need to rescale the objective function in order to convert the minimization into a maximization statement. Two basic approaches are possible. The first involve the subtraction of an error supreme M by the equation (19) resulting in expression (20).

$$f_0 = M - \frac{1}{N} \cdot \sum_{i=1}^{N} {e_i}^2 \tag{20}$$

The other possibility is the use of an exponential transformation as described by equation (21)

$$f_0 = e^{-a \cdot \sqrt{\sum_{i=1}^{N} \frac{{e_i}^2}{N}}} \tag{21}$$

In whish a is a scaling factor. The second approach has an important advantage over the first one: a small variation in the error implies a large deviation in the objective function. This characteristic is of extreme importance when the population converges and the differences

between individual fitness become lower. In E.A. system's parameters are translated into chromosomes and usually describe a system in one of two basic topologies:

It can characterize the poles, zeros and gain of the process described generically by the transfer function (22).

$$H(z) = K \cdot \frac{\prod_{\beta=1}^{m} (z - c_\beta)}{\prod_{\alpha=1}^{n} (z - d_\alpha)} \tag{22}$$

Moreover the solution may represent the following transfer function coefficients:

$$H(z) = \frac{\sum_{\beta=1}^{m} b_\beta \cdot z^{-\beta}}{1 + \sum_{\alpha=1}^{n} a_\alpha \cdot z^{-\alpha}} \tag{23}$$

In the first case, it is possible to use the knowledge from modern control theory to reduce the search space in which the algorithm must prospect (ex. stability, causality and minimum-phase concepts). Two major drawback of this approach are:

- The dependency between gain K and the zeros position,
- The use of objective function (22) for poles/zeros identification usually leads to biased zero estimation.

In equation (23) there is no *a priori* information about the limits of the search space, it must be initially guessed and then reduced as the algorithm converge to the best solution. Usually the knowledge of the physical insight can make possible to specify intervals wherein the unknown parameters can be expected to lie.

One advantage of evolutionary algorithms compared with gradient search methods lie in the fact that the objective functions to be optimised are very complex and the gradient search methods are more likely to be stuck in a local minimum missing the global one.

4. PROBLEM STATEMENT

Figure 7 illustrates a schematic diagram of a greenhouse climate control system located at the Universidade de Trás-os-Montes e Alto Douro in the North of Portugal.

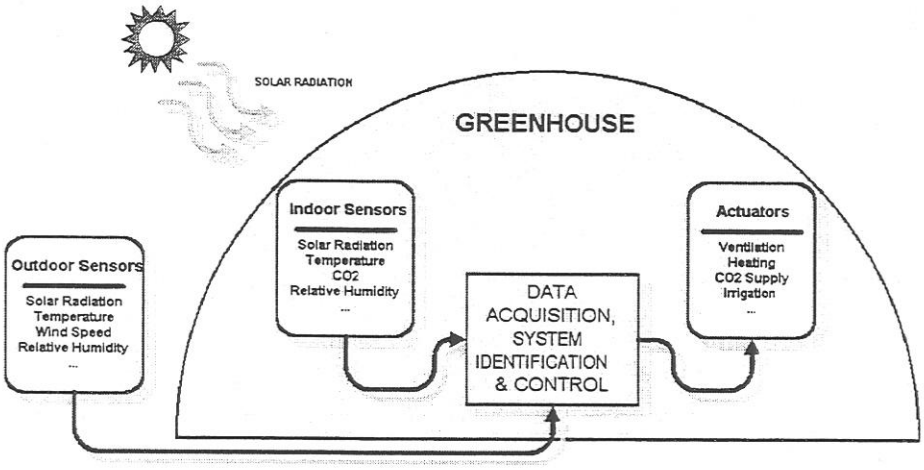


Figure 7: Diagram of the climate control system installed on a greenhouse located in the UTAD campus.

In order to apply optimal control to agricultural buildings it is essential to have a dynamic model that describes the evolution of the variables required for crop development. In an ordinary control system, the aim of the controller is to make efforts in order to maintain controlled variables as close as possible to reference signals. In the case of a greenhouse control system, the reference is a set of optimum values for the variables that influence the plant growing process. The use of *Model Predictive Controllers* for greenhouse indoor environment control has the advantage of providing system with the ability to react before any deviations in the controlled variable take place, avoiding delays in the system's response, [Boaventura, *et al.* (1998)]. This class of control algorithms must employ models to predict solar radiation fluctuations over a specified time horizon. The way to obtain those models using an off-line approach is the subject of the present paper.

5. ILLUSTRATIVE EXAMPLE

In this section, an example of a time-series identification problem is presented using both classical and evolutionary techniques. The aim is to develop a time-series model to describe the evolution of the outside solar radiation in time. This model will be useful to predict heat load fluctuations in the greenhouse caused by high frequency solar radiation changes, and to improve ventilation and heating computation requirements.

Despite of the large estimation and validation period (usually a month in different seasons), on the present paper, for clarity sake, parameters estimation and validation was done using only a day period in different months. The data used for model estimation was acquired for the period of a day with one minute sampling time in May of 1997. For validation purpose the data respects the one day period in September on the same year (Figure 8).

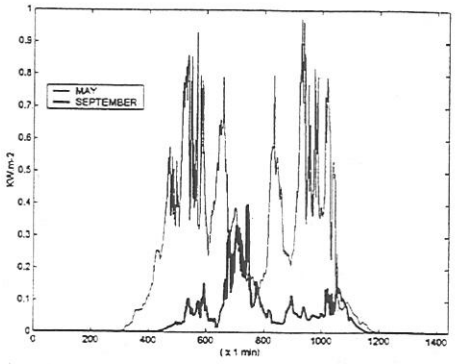


Figure 8: One day outside solar radiation in May and September.

It is assumed that a fourth order autoregressive parametric model is suitable to describe the time-series dynamics (equation 24),

$$y(k) = a_1 \cdot y(k-1) + a_2 \cdot y(k-2) + a_3 \cdot y(k-3) + a_4 \cdot y(k-4) + C \tag{24}$$

in which C is a constant to be estimated, and $y(k-i)$ with $0 \le i \le 4$, denotes the values of solar radiation at time $t = (k-i) \cdot T_s$ with $4 \le k \le 1440$ and T_s=1min. Before feeding the data into the estimator, it was pre-processed in order to remove high frequency noise. This was done using a second order low-pass filter with cut frequency equal to half of the Nyquist frequency was applied.

The identification accuracy is measured by using the Root-Mean Square (*RMS*) error given by

$$J = \sqrt{\frac{e^T \cdot e}{N}} \tag{25}$$

in which N is the window size or the number of data samples. The estimation is complemented with the calculation of the parameters standard deviation and Akaike's *FPE* expressed by equations (26) and (27) respectively,

$$\sigma = \sqrt{diag\left\{\left(\frac{e^T \cdot e}{N-n}\right) \cdot \left(\Phi^T \cdot \Phi\right)^{-1}\right\}} \tag{26}$$

$$FPE = \left(\frac{e^T \cdot e}{N-n}\right) \cdot \frac{\left(1+\frac{n}{N}\right)}{\left(1-\frac{n}{N}\right)} \tag{27}$$

N represents the window size and n is the number of model coefficients, in this case 1440 and 5 respectively.

Akaike's *FPE* was described for the first time by Akaike as a final prediction criterion. It reflects the prediction-error variance achieved when the model is applied as a predictor for a data set different from that used for identification purpose.

The estimated coefficients and respective standard deviation using the off-line *Least Squares* are presented in Table 1. The *RMS* error between the measured output and the estimated one is equal to *0.0158175*. The *FPE* calculated is *0.00025282*.

	Estimated Parameters	*Standard Deviation*
a_1	2.209944	0.0242464
a_2	-2.187333	0.0518663
a_3	1.370280	0.0518663
a_4	-0.399151	0.0242464
C	0.001228	0.0005450

Table 1: Least Squares estimated parameters and respective standards deviation

For validation purposes, a set of data recorded in a September day is applied to the estimated model. The *RMS* validation error is equal to *0.005284* and the simulated *versus* measured plots are showed in figures 9a and 9b. In figure 9a, the simulated response using the estimated model is super-imposed with the real solar radiation data. Figure 9b illustrates the fitting between model and data for a minor interval.

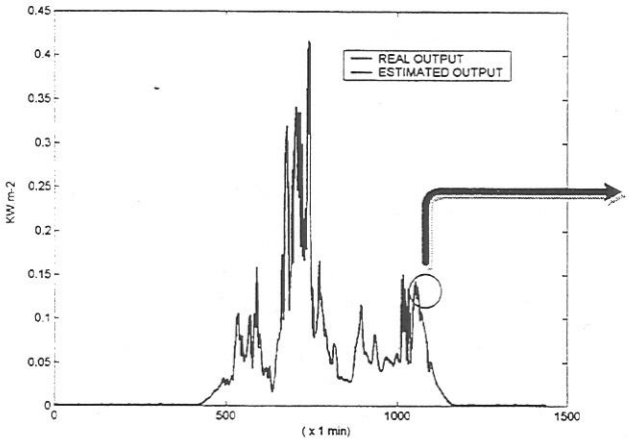


Figure 9a: Simulated and measured Solar radiation over a day in September.

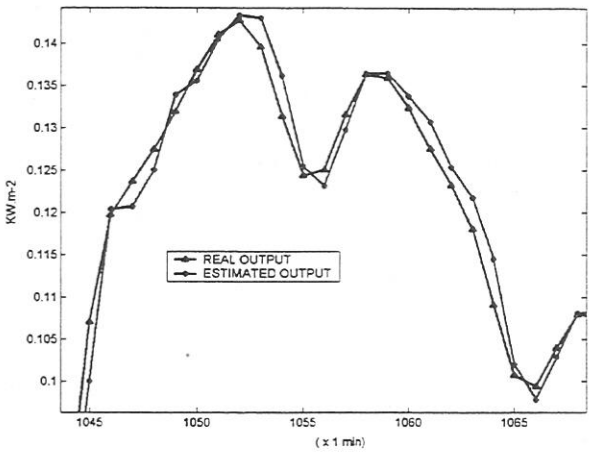


Figure 9b: Zoom view of the simulated and measured data over a period of 27 minutes.

In order to apply *GAs* to perform solar radiation identification, it is necessary to encode them in accordance with a concatenated, mapped, fixed-point coding scheme [Goldberg (1989)]. Thus each set of model parameters, represented in equation (24), is characterized by a string of binary digits.

Table 2 exhibit the Genetic Algorithm specifications used for parameter estimation, and Table 3 presents the results achieved using such specification. The estimation *RMS* error is equal to *0.0158182* and the Akaike's *FPE* equal to *0.000252845*.

Population Size	100
Number of Bits / Parameter	15
Number of Generations	1000
Prob. of Mutation	4%
Prob. of Crossover	70%

Table 2: Genetic Algorithm specification.

	Estimated Parameters	*Standard Deviation*
a_1	2.203742	0.0242474
a_2	-2.171056	0.0518685
a_3	1.353465	0.0518685
a_4	-0.392499	0.0242474
C	0.001282	0.0005450

Table 3: GA estimated parameters and respective standards deviation.

Figure 10 illustrates the normalized convergence rate of population mean and population best fitness. The validation *RMS* error achieved with this algorithm was *0.005296*.

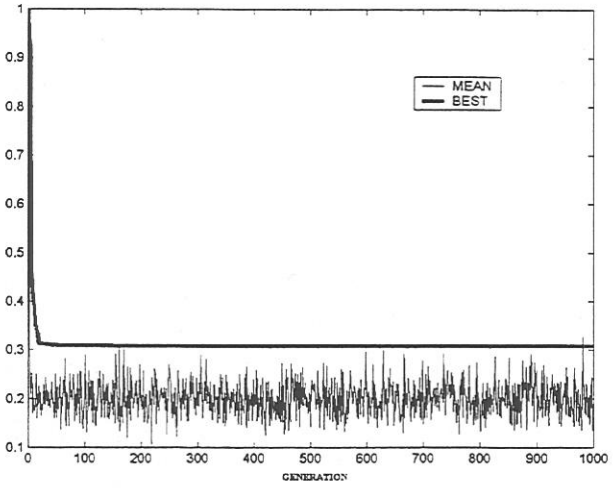


Figure 10: Normalised convergence rate of the mean and best fitness.

Parameter identification with *PBIL* is ruled by the same principle as with *GAs*, concerning the form in which the parameters are coded. Table 4 exhibits the *PBIL* characteristics used for this particular problem of parameter estimation, and *Table 5* present the results achieved using such specification. The estimation *RMS* error is equal to *0.0158178* and the Akaike's *FPE* equal to *0.00025283*.

Population Size	100
Number of Bits / Parameter	15
Number of Generations	1000
Prob. of Mutation	8%
Learning Rate	10%
Negative Learning Rate	8.5%

Table 4: *PBIL* specification.

	Estimated Parameters	*Standard Deviation*
a_1	2.205298	0.0242469
a_2	-2.176275	0.0518673
a_3	1.36024	0.0518673
a_4	-0.395611	0.0242468
C	0.001282	0.0005450

Table 5: *PBIL* estimated parameters and respective standards deviation.

Figure 11 illustrates the normalized convergence rate of population mean and population best fitness. The validation *RMS* error achieved with this algorithm was *0.005294*.

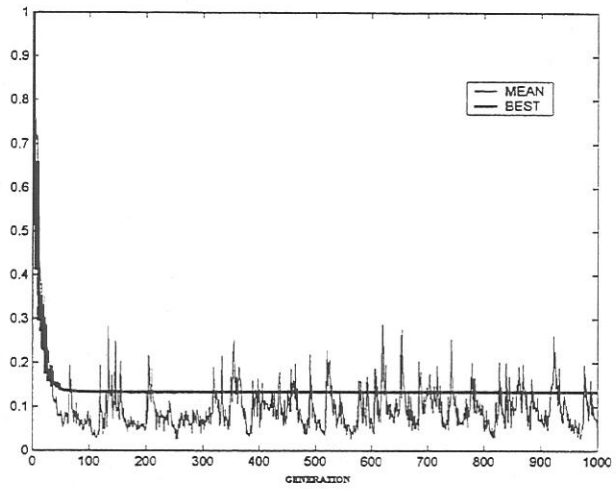
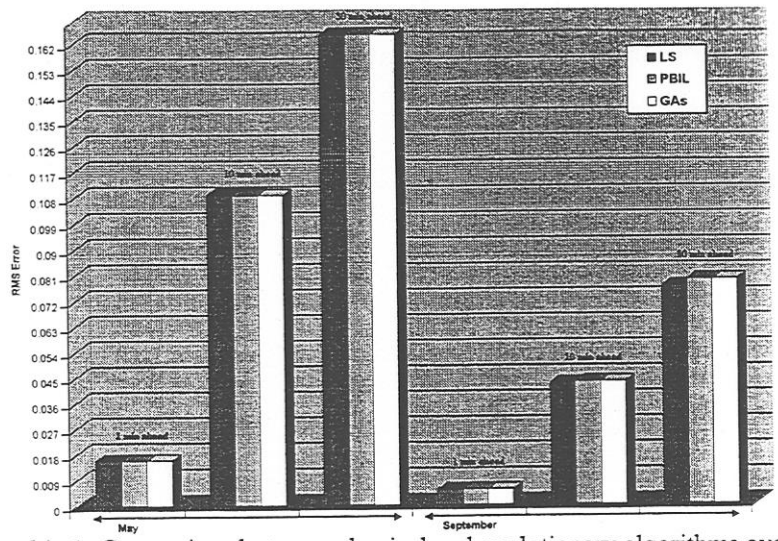


Figure 11: Normalised convergences rate of mean and best fitness.

The model, whose parameters were estimated by the above methods, was applied to solar radiation prediction. The results concerning the prediction of the time-series over a horizon of 10 and 30 minutes are presented in Graphic 1.



Graphic 1: Comparison between classical and evolutionary algorithms over a prediction horizon of 1, 10 and 30 minutes.

The results obtained so far with evolutionary based techniques, (with the available solar radiation data) do not improve the minimum *RMS* error value, when compared with Least Squares technique. However, the differences are not significant. *PBIL* identification converges faster than GAs reducing slightly the *RMS* value.

6. CONCLUSION

Least Squares techniques, Genetic Algorithm and Population Based Incremental Learning Algorithm were applied to identify the solar radiation model within a greenhouse located in UTAD campus. Experimental results reported indicate that there is no significant advantages by using evolutionary based techniques compared with least squares techniques in this particular time-series parameter estimation. However, on-going research indicates that for the identification of other greenhouse variables the evolutionary based techniques can be advantageous over least squares methods, namely when used within a auto-tuning control framework. Indeed, future work will be directed on using the same evolutionary based techniques to perform both on-line system identification and controller design within the greenhouse.

7. REFERENCES

Akaike, H., *A new look at the statistical model identification*, IEEE Trans. Autom. Control, AC-19, 716-723.

Baluja, S., *Population-Based incremental learning: A method for integrating genetic search based function optimization competitive learning*. Carnegie Mellon University (1994). Technical Report. CMU-CS-94-163

Boaventura Cunha, J., Couto, C., Ruano, A.E.B., *A greenhouse climate multivariable predictive controller*, Acta Horticulture N.534, ISHS, 2000, pp: 269-276, (1998)

De Moura Oliveira, P.B., Jones, A.H., *Cooperative co-evolutionary multivariable system identification using structured genetic algorithms*, AMST'98, Bradford, Edited by Whalley R. and Ebrahimi M., pp 149-158, (1998).

Fernado, K.V., Nicholson, H., *Identification of linear systems with input and output noise: the Koopmans-Levin method*, IEE Proc. Pt. D., 132(1), 30-36, (1985).

Goldberg, D.E., *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, (1989).

Holland, J.H., *Adaptation in natural artificial systems*, Ann Arbor, MI: University of Michigan Press, (1975).

Iba, H., Kurita, T., *System Identification using structured Genetic Algorithms*, Proc. on Fifth International Conf. on Genetic Algorithm, pp.279-286, (1993).

Kristinn, K., Guy, A.D., System identification and control using Genetic Algorithms, IEEE Transactions on Systems, Man and Cybernetics, Vol. 22, No. 5, (1992).

Ljung, L., *System Identification – Theory for the user*, Prentice-Hall, Englewood Cliffs, N.J, (1987).

Miller, A.J., *Subset Selection in Regression*, Chapman & Hall, London, (1990).

Parkum, J. E., *Recursive Identification of Time-Varying Systems*, Lyngby (1992)

Rissanen, J., *Modeling by shortest data description. Automatica*, 14:465-471, (1978).

Schwarz, G., *Estimating Dimension of a Model. Ann. Stat.*, 6:461-464, (1978).

Söderström, T. and Stoica, P., *System Identification*, Prentice-Hall International, London (1989).

Tan, K.C., Li, Y., *Evolutionary system identification in the time domain*, In proceedings of IMechE, Vol 211, Part I (1997)