A CASE BASED COMPARISON OF IDENTIFICATION WITH NEURAL NETWORK AND GAUSSIAN PROCESS MODELS

Juš Kocijan *,** **Blaž Banko** * **Bojan Likar** *
Agathe Girard *** **Roderick Murray-Smith** ***,****
Carl Edward Rasmussen †

* *Jozef Stefan Institute, Ljubljana*
** *Nova Gorica Polytechnic, Nova Gorica*
*** *University of Glasgow, Glasgow*
**** *Hamilton Institute, National University of Ireland, Maynooth*
† *Max Planck Institute for Biological Cybernetics, Tübingen*

Abstract: In this paper an alternative approach to black-box identification of non-linear dynamic systems is compared with the more established approach of using artificial neural networks. The Gaussian process prior approach is a representative of non-parametric modelling approaches. It was compared on a pH process modelling case study. The purpose of modelling was to use the model for control design. The comparison revealed that even though Gaussian process models can be effectively used for modelling dynamic systems caution has to be exercised when signals are selected. *Copyright © 2003 IFAC*

Keywords: Systems identification, Gaussian process models, artificial neural networks, pH process

1. INTRODUCTION

Gaussian processes (GPs) are being increasingly used to tackle many of the standard applications usually addressed by artificial neural networks (ANN), see e.g. (Williams, 1998). In fact, the two models are closely related, and in the limit of an infinite number of neurons in the hidden layer in the Bayesian treatment the two are equivalent (Neal, 1996). Nevertheless, the majority of work on GPs shown up to now considers modelling of static non-linearities. The GP prior approach for modelling dynamic systems has been the scope of much recent work (Murray-Smith et. al., 1999; Murray-Smith and Girard, 2001; Girard et al., 2002; Gregorčič and Lightbody, 2002; Kocijan et al., 2003).

While the relationship with neural networks has been established, a comparison in the field of dynamic systems identification has not yet been fully revealed. Whenever a new (control directed) modelling approach for dynamic systems is introduced it is important to compare it with some already established method that is related to the evaluated one and GPs are no exception. The purpose of this contribution is to compare GP dynamic models to a multi-layer perceptron ANN model using the pH neutralisation process benchmark, where the goal is control design.

[1] This work was made possible by EC funded Research Training Network HPRN-CT-1999-00107

Despite being well established and popular, ANNs still lack some properties that would further increase their acceptance. A large number of data points is necessary in order to identify the model properly. Furthermore, a relatively large number of parameters needs to be optimized when an ANN is used for identification.

Unlike ANNs, GPs naturally provide the variance associated with the estimated output for each time sample. Advantages of this information are presented in other works (e.g. (Kocijan et al., 2003)) and will not be the topic of this paper. The main point of investigation here is to evaluate whether dynamic features of the process to be modelled can be captured with the GPs mean value as they are with ANNs response.

In the next section, the pH neutralisation process used is briefly described. This is a frequently used benchmark for comparison of different approaches. Identification with neural networks and with GPs is presented in Section 3 and 4 respectively. The last section gives some concluding remarks.

2. PROCESS MODEL

A pH neutralization process model taken from (Henson and Seborg, 1994) was used for the study. The process consists of an acid stream, buffer stream and base stream that are mixed in a tank T_1. Prior to mixing, the acid stream enters the tank T_2 which introduces additional flow dynamics. The acid and base flow rates are controlled with flow control valves, while the buffer flow rate is controlled manually with a rotameter. The effluent pH is the measured variable. Since the pH probe is located downstream from the tank T_1, a time delay is introduced in the pH measurement. In this study, the pH is controlled by manipulating the base flow rate. The model includes valve and transmitter dynamics as well as hydraulic relationships for the tank outlet flows. Modelling assumptions include perfect mixing, constant density, and complete solubility of the ions involved. The simulation model of pH process which was used for necessary data generation therefore contains various non-linear elements as well as implicitly calculated function (value of highly nonlinear titration curve). A more detailed description of the process with mathematical model and necessary parameters is presented in (Henson and Seborg, 1994).

2.1 Selection of input signals and model simulation

To get a vague idea about our system dynamics, necessary for sampling time and input signal selection, some preliminary tests were pursued. The process model was excited with a combination of step-like signals for estimation of the dominant time constant and settling time. The dominant time constant was estimated in range between 65 and 185 s and settling time between 135 and 325 s (Fig. 1).

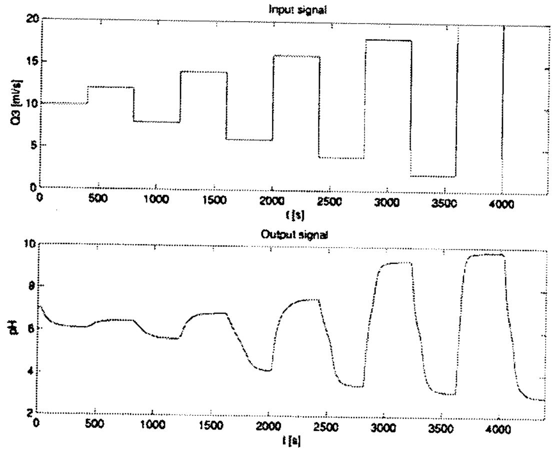


Fig. 1. Input step-like signal (upper figure) and process model response (lower figure)

This 'provisional' dynamics is necessary for the estimation of appropriate sampling time. Based on responses and iterative cut-and-try procedure a sampling time of 25 seconds was selected for these tests. The sampling time was so large that the dead-time mentioned in the previous section disappeared.

Based on these preliminary tests the chosen identification signal (400 samples) was generated from a uniform random distribution and sampling time of 50 seconds.

The validation signal was obtained using a generator of random noise with uniform distribution and sampling time of 500 seconds, so it has lower magnitude and frequency components than the identification signal. The rationale behind this is that if the model was identified using a rich signal, then it should respond well to a signal with less components.

3. DYNAMIC MODEL IDENTIFICATION WITH MULTILAYER PERCEPTRON

As ANNs are a well established approach to system identification, a number of computer tools exist to facilitate the identification. In our case, a Matlab programme package, in particular a Neural Network Based System Identification Toolbox (Nørgaard et al., 2000), was used as the computation tool. This toolbox was chosen because it contains functions for learning, validation, simulation and optimization of multilayer perceptrons with special emphasis on identification of dynamic systems.

3.1 Neural network structure and optimization

First all the data were scaled to have a mean value of 0 and variance of 1. For the neural network structure a nonlinear autoregression model with exogeneous input (NARX, for neural networks NNARX) and Levenberg-Marquardt optimization method for parameter optimization was chosen. The hidden layer contained sigmoid activation functions and the output layer contained a linear activation function. The activation functions are determined by use of the toolbox and could not be changed. However, selection of other non-linear functions for hidden layer would not improve the quality of identified model much.

The reason for chosing the NNARX structure was the relative simplicity of structure (only input and output delayed signals for regressors). The choice of regressors is a difficult one and is common to all black-box modelling approaches. The number of regressors (delayed inputs and outputs) was determined by a toolbox function that determines the lag space. For more detailes see (Nørgaard et al., 2000). The obtained plot did not have distinctive knee-points at lags of 2,3 and 4 which means that it would not be unreasonable to assume that the system can be modelled by the model of form

$$\begin{aligned} y(k) = f(&y(k\quad 1), y(k\quad 2), y(k\quad 3), y(k\quad 4),\\ &u(k\quad 1), u(k\quad 2), u(k\quad 3), u(k\quad 4)) \end{aligned} \tag{1}$$

where k denotes consecutive number of data sample. The optimal number of neurons in the hidden layer was determined by optimization. The network was optimized for each possible number of hidden neurons in a certain range, starting from 2 and the model order from 4 to 6 (which corresponds to the number of delayed outputs contained in the vector of regressors). Levenberg-Marquardt method is the standard method (not necessarily the most efficient) for minimization of mean-square error criteria, due to its rapid convergence properties and robustness. An important factor of choice was also its availability in the software used.

At the end of this lengthy procedure, the optimal parameters were obtained for the model given by equation (1), with the regressor vector composed of four delayed inputs and outputs and with ten neurons in the hidden layer. To avoid redundant connections between neurons, pruning of ANN was pursued. Only one connection was determined as redundant.

3.2 Model validation

Visual inspection of the plot comparing predictions to data used for validation is probably the most important validation tool (Nørgaard et al., 2000). Responses of ANN and comparison with process model response to the identification and validation input signal is given in Fig. 2.

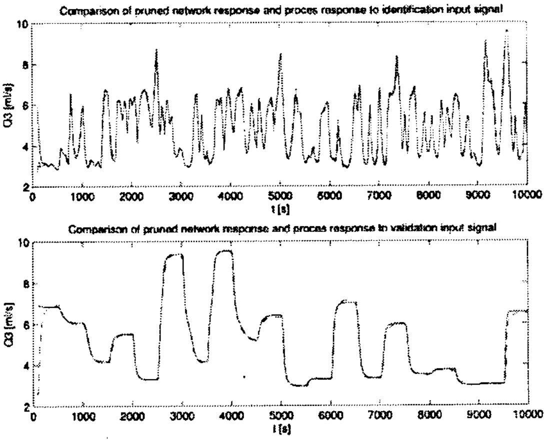


Fig. 2. Simulated responses of neural network (dashed line) and process model (full line) on identification input signal (upper figure) and responses of neural network (dashed line) and process model (full line) on validation input signal (lower figure)

A satisfactory fit can be observed for identification input signal which is understandable, since this signal was used for optimization. The response to the validation input signal is obtained by simulation (not by one-step ahead prediction).

The goodness of the fit of the model response was assessed by calculating the average absolute test (validation) error, AE, and the average squared test error, SE.

$$AE = \frac{1}{N}\sum |\hat{\mathbf{y}} \quad \mathbf{y}| = 0.0672 \tag{2}$$

and

$$SE = \frac{1}{N}\sum (\hat{\mathbf{y}} \quad \mathbf{y})^2 = 0.0142 \tag{3}$$

where N is the number of data in the validation set, $\mathbf{y}$ the process response (target) and $\hat{\mathbf{y}}$ is the model output. The first 20 samples were not used in for this evaluation due to undefined delayed values at the beginning of simulation and a consequent transient response.

The autocorrelation function (Fig. 3) of the error between network response and process model response on the validation signal and the cross correlation between error and input validation signal also indicated a satisfactory identification. From these results it can be concluded that the ANN model captures the dynamics of the pH neuralization process model relatively well. There

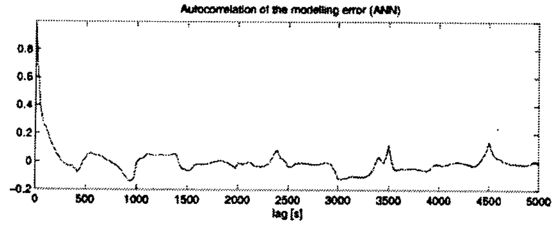


Fig. 3. Autocorrelation function of the error between network response and process model response on the validation signal

may be scope to improve the ANN model, but it is sufficiently good for numerous ANN model-based control designs. The resulting ANN is not too large to be handled and was relatively routinely obtained with the selected software tool. Nevertheless, it has to be stressed that input signals and number of samples were selected suited to identification with ANNs.

4. DYNAMIC MODEL IDENTIFICATION WITH GAUSSIAN PROCESSES

4.1 A quick review of modelling with Gaussian Processes

A Gaussian Process is a collection of random variables which have a joint multivariate Gaussian distribution: $f(x^1),\ldots,f(x^n) \sim \mathcal{N}(0,\Sigma)$, where Σ_{pq} gives the covariance between outputs $f(x^p)$ and $f(x^q)$ corresponding to inputs x^p and x^q.

We have $\mathrm{Cov}(f(x^p),f(x^q)) = C(x^p,x^q)$, where $C(.,.)$ is some function with the property that it generates a positive definite covariance matrix. This means that the covariance between the variables that represent the outputs for cases number p and q is a function of the inputs corresponding to the same cases p and q.

A stationary (depends only on the distance between points) Gaussian covariance function (such as (4)) is used in this work:

$$C(x^p,x^q) = v_1 \exp\left[\ \ \frac{1}{2}\sum_{d=1}^{D}\frac{(x_d^p\ \ x_d^q)^2}{w_d^2}\right] \qquad (4)$$

where D is the input dimension. This choice corresponds to points close together being more correlated than points far apart – a smoothness assumption.

Learning. It is assumed that the data are noisy versions of the function outputs. That is, if it is assumed $y = f(x) + \epsilon$ where ϵ is an uncorrelated (white) noise with variance v_0. Then, the covariance between the training cases y^p and y^q is given by

$$K_{pq} = C(x^p,x^q) + v_0\delta_{pq} \qquad (5)$$

in which the noise contribution is non zero only when $p = q$. Given a set of training cases $\{y^i,x^i\}_{i=1}^N$, where x^i is (possibly) a D-dimensional input vector, the hyperparameters of the covariance function, $\Theta = [w_1 \ldots w_D\ v_1\ v_0]^T$ are to be learned.

The learning is done by maximizing the marginal log-likelihood

$$\mathcal{L}(\theta) = \ \ \frac{1}{2}\log|K|\ \ \ \frac{1}{2}\mathbf{y}^{\mathrm{T}}K^{\ \ 1}\mathbf{y}\ \ \ \frac{N}{2}\log(2\pi) \qquad (6)$$

where $\mathbf{y}$ is the $N \times 1$ vector of training targets and K is the $N \times N$ training covariance matrix .

Predicting. For a new test input x^*, the predictive distribution of the corresponding output is $y|x^* \sim \mathcal{N}(\mu_y(x^*), \sigma_y^2(x^*))$ with

$$\mu_y(x^*) = \mathbf{k}(x^*)^T K^{\ \ 1}\mathbf{y} \qquad (7)$$

$$\sigma_y^2(x^*) = \mathbf{k}(x^*)\ \ \ \mathbf{k}(x^*)^T K^{\ \ 1}\mathbf{k}(x^*) + v_0 \qquad (8)$$

where $\mathbf{k} = [C(x^*,x^1),\ldots,C(x^*,x^N)]^T$ is the $N \times 1$ vector of covariances between the test and training cases and $k(x^*) = C(x^*,x^*)$ is the variance of the new test case.

Validation of the model is obtained by simulation, that we can view as *infinite*-step ahead prediction (where *infinite* is, in practice, the length of the validation set). Currently, k-step ahead prediction can be achieved by either training the model to learn *how* to make k-step ahead predictions (*direct method*) or by doing repeated one-step ahead predictions up to k - *iterative method.* For a model of the form

$$y(k) = f(y(k\ \ 1), y(k\ \ 2),\ldots,u(k\ \ 1),u(k\ \ 2),\ldots)\ , \qquad (9)$$

it corresponds to feeding back the model output at each time step

$$y(k) = f(\hat{y}(k\ \ 1),\hat{y}(k\ \ 2),\ldots,u(k\ \ 1),\ldots)\ . \qquad (10)$$

In our experiment, the iterative approach as presented here is used, although (Girard et al., 2002) proposed a principled approach taking account of the uncertainty of the model output at each time step. An example of modelling a dynamic system with a GP can be found in (Kocijan et al., 2003).

4.2 Regressors and parameter optimization

The same data and the same model representation (1) were used as for the ANN. The counterpart of ANN's choice of number of layers and nodes, is, in the GP modelling framework, the choice of a particular covariance function

Given our choice of covariance function (4), the learning consists in the identification of the covariance hyperparameters and the noise variance v_0. There is a hyperparameter corresponding to each regressor 'component' so that, after the learning, if a hyperparameter is very large it means that the corresponding regressor 'component' has little impact so could potentially be removed.

The optimization method used for identification of GP model was a conjugent gradient with linesearches. The Polack - Ribiere conjugate gradients is used to compute search directions, and a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria is used together with the slope ratio method for guessing initial step sizes. Additionally a number of checks are made to make sure that exploration is taking place and that extrapolation will not be unboundedly large (Rasmussen, 1996).

Obtained hyperparameters after optimization were [2]:

- $w_1 = 1.23\ (y(k))$; $w_2 = 3.07\ (y(k \quad 1))$; $w_3 = 20.4\ (y(k \quad 2))$; $w_4 = 37.8\ (y(k \quad 3))$ which correspond to previous outputs;
- $w_5 = 70.7\ (u(k))$; $w_6 = 6.03\ (u(k \quad 1))$; $w_7 = 14.7\ (u(k \quad 2))$; $w_8 > 1000\ (u(k \quad 3))$ which correspond to previous inputs (in brackets);
- $v_0 = 0.0045$ which is estimated noise variance and
- $v_1 = 2.339$ which is the estimate of the vertical variance.

Results of simulation of the obtained 4th order GP model and its assessment is given in the following section.

4.3 Validation of results

Responses (estimates or predicted mean) of GP model and comparison with process model response to the same identification and validation input signal as in the case of ANNs are given in Fig. 4.

As in the case with the ANN a very good fit can be observed for identification input signal which was used for optimization. The response on validation signal is not as good as in the case of ANN model. The autocorrelation function of the modelling error is given in Fig. 5.

Fitting of the response for validation signal:

- average absolute test error $AE = 0.1494$
- average squared test error $SE = 0.0512$

When the number of points in the training set is reduced then the ANN performance deteriorates. On the other hand, the advantage of GPs is that it performs well given a small number of data (Fig. 6).

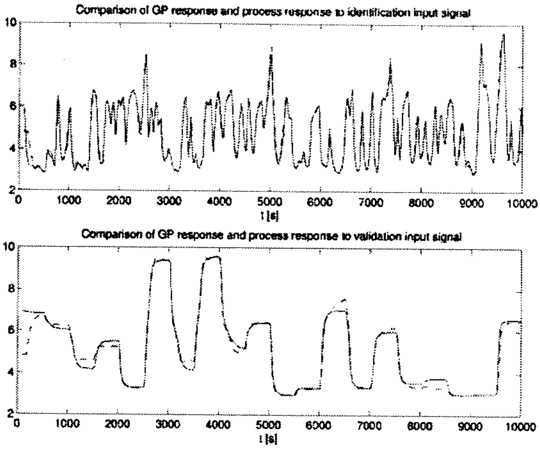


Fig. 4. Responses of Gaussian process model (dashed line) and process model (full line) on identification input signal (upper figure) and responses of Gaussian process model (dashed line) and process model (full line) on validation input signal (lower figure)

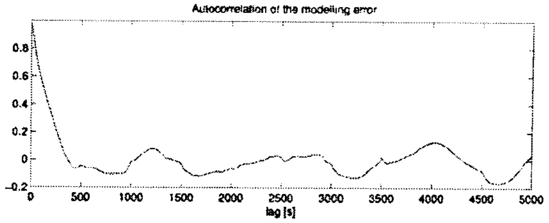


Fig. 5. Autocorrelation function of the error between Gaussian processes response and process model response on the validation signal

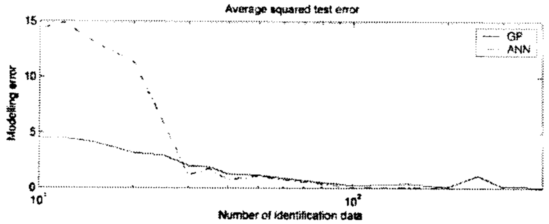


Fig. 6. Graph representing averaged squared test error versus number of identification data for both models

The second point that needs to be addressed is the choice of covariance function and its link to GP predictions. Selecting covariance functions suitable for robust generalisation in typical dynamic systems applications should be a research priority. The validation data was sufficiently different from the training data in this case that the GP with a smoothing covariance function performed poorly, having optimised its parameters to fit the higher frequency components present in the identification data, but then being asked to make predictions in areas of the input space unpopulated by training data. The GP can, however, highlight such areas of the input space where prediction

[2] The code implementing the GP modelling is available at http://www.gatsby.ucl.ac.uk/~edward/code/gp/, for the Matlab programme package.

quality is poor, due to the lack of data or its complexity, by indicating the higher variance of the predicted mean (Girard et al., 2002; Kocijan et al., 2003).

From the presented results it can be seen that GP models can be used for identification of dynamic models, in our case the dynamics of the pH neuralization process model, though under some special considerations. Despite a poorer fit to the validation data, the model could still be useful for control design. The number of parameters for the GP is much smaller than the number of weights in ANN, and it could be used as well to select the model structure (e.g. number of regressors) via the hyperparameters. This means that the procedure could be to start off with many delayed inputs and outputs and decide to keep them or not, depending on the value of the corresponding hyperparameter after learning.

5. CONCLUSIONS

Two models of dynamical system were presented in the paper. The first one is the well established ANN model and the second one is GP model. The purpose of this paper was to evaluate the GP models use for dynamic models identification under the same conditions as ANNs via comparison on a case study, namely pH process model identification. The models were developed for use in system control.

The following conclusions can be drawn from obtained results:

- GP models can be used to model dynamic systems.
- There were no problems encountered with the GP, when dealing with a rather small number of data, but this model is difficult to apply to large data sets because of computational reasons (inversion of the $N \times N$ training covariance matrix). However, efforts are continuing to solve this problem.
- Obtained a GP model is relatively simple to implement and contains a smaller number of structure parameters than a corresponding ANN.
- ANNs and GP models could have more complementary role in systems identification. A role of GPs especially where small number of measurements is available and signals used for identification and validation do not differ very much.
- Beside envisaged GP based control design (see e.g. (Murray-Smith and Sbarbaro, 2002), there are still possibilities for further development of GP model identification, especially in the direction of more efficient algorithm development.

REFERENCES

A. Girard, C.E. Rasmussen, R. Murray-Smith (2002): Multi-step ahead prediction for non linear dynamic sytems - A Gaussian Process treatment with propagation of the uncertainty, Advances in Neural Information processing Systems 16, S. Becker and S. Thrun and K. Obermayer (Eds.), 2002.

G. Gregorčič, G. Lightbody (2002): Gaussian Processes for Modelling of Dynamic Non–linear Systems,Proceedings of the Irish Signals and Systems Conference, 141-147.

M.A. Henson, D.E. Seborg (1994): Adaptive Nonlinear Control of a pH Neutralization Process, IEEE Trans. Control System Technology, **2**, No. 3, 169-183.

J. Kocijan, A. Girard, B. Banko, R. Murray-Smith (2003) Dynamic systems identification with Gaussian processes, 4th Mathmod Conference, Vienna, 776-784.

R. Murray-Smith and A. Girard (2001), Gaussian Process priors with ARMA noise models, Irish Signals and Systems Conference, Maynooth, 147-152.

R. Murray-Smith, T. A. Johansen and R. Shorten (1999), On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures, European Control Conference, Karlsruhe, 1999, BA-14.

R. Murray-Smith and D. Sbarbaro - Hofer 2002, Nonlinear adaptive control using non-parametric Gaussian process prior models, 15th IFAC World Congress on Automatic Control, Barcelona.

Neal R.M., Bayesian learning for neural networks, Lecture notes in statistics, Springer Verlag, New York, 1996.

M. Nørgaard (2000), Neural Network Based System Identification Toolbox, Version 2, Department of Automation, Technical Report 00-E-891, Technical University of Denmark, Lyngby.

C.E. Rasmussen (1996), Evaluation of Gaussian Processes and other Methods for Non-linear Regression, Ph.D. Disertation, Graduate department of Computer Science, University of Toronto, Toronto.

C. Williams (1998), Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond, Learning in Graphical Models, M. I. Jordan,Kluwer, 599–621.

IFAC

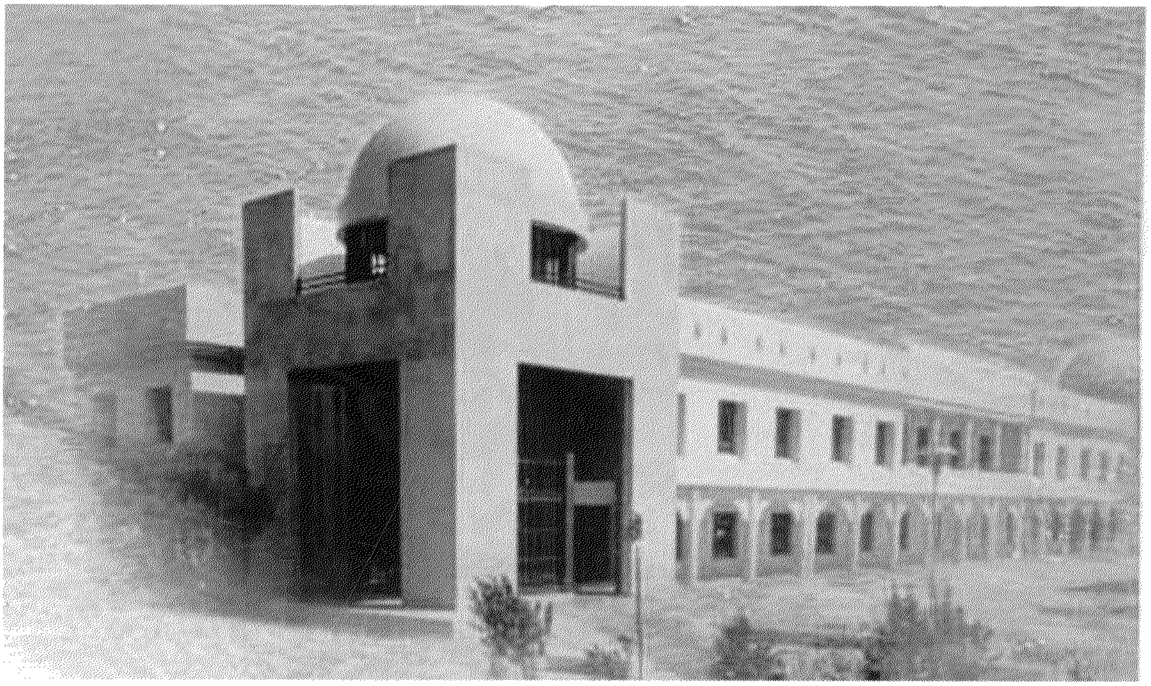
INTERNATIONAL FEDERATION
of AUTOMATIC CONTROL

Preprints

INTELLIGENT CONTROL SYSTEMS and SIGNAL PROCESSING

Edited by A. E. Ruano

ICONS 2003



University of Algarve, Portugal, 8-11 April 2003

Co-sponsored by

IEEE - The Institute of Electrical and Electronic Engineers

IFSA - International Fuzzy Systems Association

and with the support of

EVONET - European Network of Excellence in Evolutionary Computing

Portuguese Society of
Automatic Control

CSI

Centre for Intelligent
Systems