ISSN 0005–1144
ATKAAF 42(3–4), 159–167 (2001)

José Vicente Salcedo, Miguel Martínez, Javier Sanchis, Xavier Blasco

Design of GPC's in State Space*

UDK 681.514
IFAC IA 2.5

Original scientific paper

This paper introduces a methodology for the original design of generalised predictive controllers (GPC's) based on the use of a state space CARIMA model to carry out those predictions. The CARIMA model presented is equivalent to the CARIMA model commonly used in the input/output (I/O) formulation of the GPC's. A connection is settled among the stochastic part of this model and the filter polynomials $\boldsymbol{T}_i(z^{-1})$, making possible the design of the controller once any one of them is known. It is also remarkable that for the estimation of non-measurable states, a full rank observer is proposed, and the fact that its poles are equal to the roots of the filter polynomials $\boldsymbol{T}_i(z^{-1})$ can also be appreciated.

Key words: CARIMA model, optimization, predictive control, state space design

1 INTRODUCTION

In order to formulate the MIMO input/output (I/O) generalised predictive controller (GPC), a model of the process described through the CARIMA model in transfer matrices [1, 2, 3] is assumed:

$$\begin{pmatrix} A_1(z^{-1}) & 0 & \cdots & 0 \\ 0 & A_2(z^{-1}) & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & A_n(z^{-1}) \end{pmatrix} \begin{pmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_n(k) \end{pmatrix} =$$

$$= \begin{pmatrix} B_{11}(z^{-1}) & B_{12}(z^{-1}) & \cdots & B_{1m}(z^{-1}) \\ B_{21}(z^{-1}) & B_{22}(z^{-1}) & \cdots & B_{2m}(z^{-1}) \\ \vdots & \vdots & \ddots & \vdots \\ B_{n1}(z^{-1}) & \cdots & \cdots & B_{nm}(z^{-1}) \end{pmatrix} \begin{pmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_m(k) \end{pmatrix} +$$

$$+ \begin{pmatrix} T_1(z^{-1})/\Delta & 0 & \cdots & 0 \\ 0 & T_2(z^{-1})/\Delta & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & T_n(z^{-1})/\Delta \end{pmatrix} \begin{pmatrix} \xi_1(k) \\ \xi_2(k) \\ \vdots \\ \xi_n(k) \end{pmatrix}. \tag{1}$$

Such model contains n outputs and m inputs. The ξ_i variables represent the uncertainties of the model and are called noise inputs. Two separate parts form this CARIMA model:

– A deterministic associated to the relationship between inputs and outputs given by the polynomials $A_i(z^{-1})$ and $B_{ij}(z^{-1})$.

– A stochastic associated to the relationship between noise variables ξ_i and the outputs given by the polynomials $A_i(z^{-1})$ and $T_i(z^{-1})$(1). This part is called noise model.

On the GPC MIMO design, a quadratic cost index is used [3]:

$$J_k(\hat{u}(k)) = E\left[\sum_{q=1}^{n}\left[\sum_{i=N_1^q}^{N_2^q} \alpha_{qi}\left(y_q(k+i) - w_q(k+i)\right)^2\right] + \right.$$
$$\left. + \sum_{j=1}^{m}\left[\sum_{i=1}^{N_u^j} \lambda_{ji}\Delta u_j^2(k+i-1)\right]\right]$$
$$\hat{u}(k) = \left[\Delta u_1(k) \cdots \Delta u_1\left(k+N_u^1-1\right) \cdots \Delta u_m\left(k+N_u^m-1\right)\right]^T \tag{2}$$

Where:

– N_1^q, N_2^q represent the limits of the prediction horizon for the q-th output.

– N_u^j is the control horizon for the j-th input.

* Partially supported by the project 1FD1997-0974-C02-02, FEDER and DPI2001-3106-C02-02, Spain.

(1) The $T_i(z^{-1})$ polynomials are called filter polynomials

– α_{qi} is the pondering coefficient of the error for output q on instant i inside the prediction horizon.
– λ_{ji} is the pondering coefficient of control action increment for input j on instant i inside the control horizon.

The methodology for the design of the generalised predictive control MIMO [4, 5, 6] is as follows: on each sampling instant k, index (2) has to be optimised in order to determine the control actions that are going to be applied to the process. In order to optimize such index it is necessary to predict the n outputs of process (1) inside their corresponding prediction horizon, and according to:
– The values of the m input variables inside their control horizons (unknown). These are precisely the independent variables from which the quadratic index depends ((k)).
– The ξ_i variables considered as white noises.
– The past values (known) of inputs, noises and outputs.

From of the control actions values obtained after optimising index (2), only the control actions corresponding to the first instant of each control horizon $u_1(k)$, $u_2(k)$,… $u_m(k)$ are applied to the process. This technique is known as receding horizon. After this, the process is repeated for the following sampling period $k+1$.

2 STATE SPACE CARIMA MODEL

2.1 Preliminaries

Plant models are multivariable in most real applications. The literature related to I/O MIMO GPC design presents a common aspect: the extension from SISO case to the MIMO case is conceptually easy, although the required matrix and signal manipulations make it a complex process. However, with state space techniques the extension is easier.

The development of a state space strategy for MIMO GPC designing is furthermore supported by the need for solving certain questions concerning MIMO GPC: stability, robustness, specifications selection, etc., very important in industrial applications. These could be easily analysed with state space techniques.

2.2 Model definition

In order to be able to design state space GPC following the methodology used in the I/O case, a CARIMA model with the same characteristics as the one used for the I/O (1) although for state space, is needed.

The deterministic part of the CARIMA model can be represented according to the following state space model:

$$\bar{\boldsymbol{x}}(k+1)=\boldsymbol{A}\bar{\boldsymbol{x}}(k)+\boldsymbol{B}\bar{\boldsymbol{u}}(k)\,;\quad \bar{\boldsymbol{y}}(k)=\boldsymbol{C}\bar{\boldsymbol{x}}(k) \tag{3}$$

$$\bar{\boldsymbol{y}}(k)=\begin{pmatrix} y_1(k)\\ \vdots \\ y_n(k)\end{pmatrix};\quad \bar{\boldsymbol{x}}(k)=\begin{pmatrix} x_1(k)\\ \vdots \\ x_r(k)\end{pmatrix};\quad \bar{\boldsymbol{u}}(k)=\begin{pmatrix} u_1(k)\\ \vdots \\ u_m(k)\end{pmatrix}. \tag{4}$$

It is a model consisting of n outputs, m inputs and r states.

To obtain a complete CARIMA model, it is necessary to add to the former deterministic model the $\xi_i(k)$ noise variables and their associate states which are called noise states $x_i^*(k)$. These states are nothing more than the accumulation of such inputs:

$$x_i^*(k+1)=x_i^*(k)+\pi_{2i}\xi_i(k)\qquad i=1,2,\ldots,n. \tag{5}$$

When these additional states and inputs are incorporated into the deterministic model given by (3), the following model is obtained:

$$\begin{aligned}\boldsymbol{x}(k+1)&=\bar{\boldsymbol{A}}\boldsymbol{x}(k)+\bar{\boldsymbol{B}}\bar{\boldsymbol{u}}(k)+\boldsymbol{\Pi}\bar{\boldsymbol{\xi}}(k)\\ \bar{\boldsymbol{y}}(k)&=\bar{\boldsymbol{C}}\boldsymbol{x}(k)+\boldsymbol{\Lambda}\bar{\boldsymbol{\xi}}(k)\,.\end{aligned} \tag{6}$$

Being:

$$\boldsymbol{x}(k)=\begin{pmatrix}\bar{\boldsymbol{x}}(k)\\ \boldsymbol{x}^*(k)\end{pmatrix};\quad \bar{\boldsymbol{\xi}}(k)=\begin{pmatrix}\xi_1(k)\\ \vdots\\ \xi_n(k)\end{pmatrix}; \tag{7}$$

$$\bar{\boldsymbol{A}}=\begin{bmatrix}\boldsymbol{A} & \boldsymbol{\Sigma}_{rxn}\\ \boldsymbol{0} & \boldsymbol{I}_n\end{bmatrix};\quad \bar{\boldsymbol{B}}=\begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{0}\end{bmatrix}; \tag{8}$$

$$\boldsymbol{\Pi}=\begin{bmatrix}\boldsymbol{\Pi}_1\\ \boldsymbol{\Pi}_2\end{bmatrix};\quad \bar{\boldsymbol{C}}=\begin{bmatrix}\boldsymbol{C} & \boldsymbol{\Omega}\end{bmatrix}. \tag{9}$$

This model is a state space CARIMA model equivalent in construction to the one used in the I/O case. Matrices $\boldsymbol{\Sigma}$, $\boldsymbol{\Pi}_1$, $\boldsymbol{\Pi}_2$, $\boldsymbol{\Omega}$ and $\boldsymbol{\Lambda}$ can be freely chosen to establish different noise models for the process. This means an increase in the complexity of the choice of noise model parameters with respect to the I/O formulation, in which only the $T_i(z^{-1})$ filter polynomials had to be chosen.

3 EQUIVALENCY BETWEEN CARIMA MODELS

It is possible to prove [2] that I/O CARIMA model (1) and state space CARIMA model (6) are equivalent. The following expression relating the $T_i(z^{-1})$ filter polynomials and the matrices associated to the state space noise model $\boldsymbol{\Sigma}$, $\boldsymbol{\Pi}_1$, $\boldsymbol{\Pi}_2$, $\boldsymbol{\Omega}$ and $\boldsymbol{\Lambda}$:

$$T_j\left(z^{-1}\right)=z^{-(n_j+1)}\left(\Pi'_{jj}(z)(z-1)+\Sigma'_{jj}(z)+\right.$$
$$\left.+\Omega_{jj}A_j(z)+\Lambda_{jj}(z-1)A_j(z)\right) \tag{10}$$

Where:

$$\mathbf{\Pi}_1=\begin{bmatrix}\mathbf{\Pi}_{11}\\ \mathbf{\Pi}_{12}\\ \vdots\\ \mathbf{\Pi}_{1n}\end{bmatrix};\quad \mathbf{\Pi}_{1j}\in\mathbb{R}^{n_j\times n}$$

$$\mathbf{\Pi}_{1j}=\begin{bmatrix}0&\cdots&0&\Pi_{1j}(1,j)&0&\cdots&0\\ 0&\cdots&0&\Pi_{1j}(2,j)&0&\cdots&0\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\ 0&\cdots&0&\Pi_{1j}(n_j,j)&0&\cdots&0\end{bmatrix} \tag{11}$$

$$\Pi'_{ij}(z)=\Pi'_{ij,n_i-1}z^{n_i-1}+\cdots+\Pi'_{ij,1}z+\Pi'_{ij,0}$$
$$\Pi'_{ij,k}=\Pi_{1i}(k+1,j)$$

$\Pi_{1i}(k+1,j)$ represents the element $(k+1,j)$ from the $\mathbf{\Pi}_{1i}$ matrix.

$$\mathbf{\Sigma}_1=\begin{bmatrix}\mathbf{\Sigma}_1\\ \mathbf{\Sigma}_2\\ \vdots\\ \mathbf{\Sigma}_n\end{bmatrix};\quad \mathbf{\Sigma}_j\in\mathbb{R}^{n_j\times n}$$

$$\mathbf{\Sigma}_j=\begin{bmatrix}0&\cdots&0&\Sigma_j(1,j)&0&\cdots&0\\ 0&\cdots&0&\Sigma_j(2,j)&0&\cdots&0\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\ 0&\cdots&0&\Sigma_j(n_j,j)&0&\cdots&0\end{bmatrix} \tag{12}$$

$$\Sigma'_{jj}(z)=\Sigma'_{jj,n_j-1}z^{n_j-1}+\cdots+\Sigma'_{jj,1}z+\Sigma'_{jj,0}$$
$$\Sigma'_{jj,k}=\Sigma_j(k+1,j)$$

$$\mathbf{\Omega}=diag(\Omega_{11},\cdots,\Omega_{22}) \tag{13}$$

$$\mathbf{\Lambda}=diag(\Lambda_{11},\cdots,\Lambda_{22}) \tag{14}$$

$$\mathbf{\Pi}_2=\boldsymbol{I}_n{}^{(2)}. \tag{15}$$

A realization of the model I/O's deterministic part (3) was carried out through the fusion of the observable canonical forms of every output independently considered. This was necessary to obtain the equivalency. In such realization n_j represents the number of states associated to the j-th output.

(2) I_n is the identity matrix with of n-th order

Expression (10) allows, once the $T_j(z^{-1})$ filter polynomials are known, obtaining the matrices for the CARIMA model in state space and viceversa. This expression is only valid when the order of filter polynomials is minor or equal than n_j+1, as this is the degree of the second member of (10). If filter polyno-mials of a bigger order were to be used, it would be necessary to include artificial states to the CARI-MA model, assigning them zero poles which would not add any additional dynamic to the system, allo-wing them to increase the degree of the second member's polynomials.

The direct problem, that is to say, obtaining the model's state space matrices given the filter polynomials, presents no single solution as deduced from (10):

– Because the polynomials are of n_j+1 order, a total number of n_j+2 equations are available to solve the problem.
– The coefficients of the $\Pi'_{jj}(z)$ and $\Sigma'_{jj}(z)$ polynomials and the value of constants Λ_{jj} and Ω_{jj}, are unknown. The polynomials aforementioned have a n_j-1 order and consequently, there will be a total of $n_j+n_j+1+1=2n_j+2$ unknowns.

Therefore there will always be more unknowns than equations and consequently, solving the problem will always be possible although it will not have a single solution. However, the inverse problem, finding the filter polynomials from the matrices of the state space model, always has a single and direct solution using (10).

4 PREDICTING THE OUTPUTS

Once the CARIMA model is presented in state space, the following step consists on obtaining an expression for the prediction of process MIMO outputs in their corresponding prediction horizons. To simplify the expression, the same prediction horizon is considered for every output, and the same control horizon is considered for every input:

$$N_1^i=N_1\forall i;\quad N_2^i=N_2\forall i;\quad N_u^i=N_u\forall i. \tag{16}$$

Under these conditions, the resulting expression for the prediction is shown in [2]:

$$\hat{\boldsymbol{y}}(k)=\boldsymbol{Mx}(k)+\boldsymbol{N}\hat{\boldsymbol{u}}(k)+\boldsymbol{O}\overline{\boldsymbol{u}}(k-1)+\boldsymbol{P}\bar{\boldsymbol{\xi}}(k) \tag{17}$$

Where:

$$\hat{\boldsymbol{y}}(k)=\left(\bar{\boldsymbol{y}}(k+N_1|k)\ \cdots\ \bar{\boldsymbol{y}}(k+N_2|k)\right)^T \tag{18}$$

$\bar{\boldsymbol{y}}(k+i|k)$ is the prediction of the outputs vector in instant $k+i$ from the information available in instant k.

$$\hat{\boldsymbol{u}}(k)=\left(\Delta\bar{\boldsymbol{u}}(k)\ \cdots\ \Delta\bar{\boldsymbol{u}}(k+N_u-1)\right)^T \tag{19}$$

$\boldsymbol{M}$, $\boldsymbol{N}$, $\boldsymbol{O}$ and $\boldsymbol{P}$ are real matrices of adequate dimensions.

If this prediction expression is compared to the one obtained in version I/O [1, 2, 3]:

$$\hat{\boldsymbol{y}}(k)=\boldsymbol{G}\hat{\boldsymbol{u}}(k)+\boldsymbol{\Gamma}\Delta\bar{\boldsymbol{u}}^f(k)+\boldsymbol{F}\bar{\boldsymbol{y}}^f(k) \tag{20}$$

Being:

$$\Delta\bar{\boldsymbol{u}}^f(k)=\left(\Delta\bar{\boldsymbol{u}}^{f1}(k)\ \cdots\ \Delta\bar{\boldsymbol{u}}^{fn}(k)\right)^T \tag{21}$$

$$\Delta\bar{\boldsymbol{u}}^{fq}(k)=\begin{pmatrix}\Delta u_1^{fq}(k-1)\\ \vdots\\ \Delta u_1^{fq}\left(k-\gamma_{q1}\right)\\ \vdots\\ \Delta u_m^{fq}(k-1)\\ \vdots\\ \Delta u_m^{fq}\left(k-\gamma_{qm}\right)\end{pmatrix} \tag{22}$$

$$\Delta u_j^{fq}(k)=\frac{\Delta u_j(k)}{T_q\left(z^{-1}\right)} \tag{23}$$

$$\bar{\boldsymbol{y}}^f(k)=\left(\bar{\boldsymbol{y}}_1^f\ \cdots\ \bar{\boldsymbol{y}}_n^f(k)\right)^T \tag{24}$$

$$\bar{\boldsymbol{y}}_q^f(k)=\left(y_q^f(k)\ \cdots\ y_q^f\left(k-F_q\right)\right)^T \tag{25}$$

$$y_q^f(k)=\frac{y_q(k)}{T_q\left(z^{-1}\right)}. \tag{26}$$

γ_{qk} and F_q are constants associated to the dimensions of matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{F}$ respectively.

The following conclusions are reached:

- The expression of I/O prediction is formulated after a series of matrices that recursively (non directly) depend on the coefficients of the polynomials, forming the transfer matrices of the I/O CARIMA model in process [1, 2, 3]. Unlike the matrices taking part in the expression of the CARIMA prediction in space state, which directly depend on its matrices [2].
- All the information that needs the expression of the prediction in state space, is referred to current instant k, with the exception of the need to know the control actions of the instant before the current one. However, the expression of I/O prediction needs to know the values of the increments of the control actions and the outputs in previous instants to the current one, but filtered through the polynomials $T_j(z^{-1})$.
- Unfortunately, the expression of the state space CARIMA prediction needs for the model's states to be observed.
- The amount of information to be stored in both formulations can be analysed also:
  - In the state space formulation, the following variables are needed: control actions vector $\boldsymbol{u}$ with m elements, outputs vector $\boldsymbol{y}$ with n elements and states vector $\boldsymbol{x}$ with $\left(\sum_{j=1}^n n_j\right)+n$ elements. Noise inputs are not required, since they can be estimated from the states vector and the outputs vector using [2]:

    $$\hat{\boldsymbol{\xi}}\left(k|k\right)=\boldsymbol{\Lambda}^{-1}\left(\boldsymbol{y}(k)-\bar{\boldsymbol{C}}\boldsymbol{x}(k)\right) \tag{27}$$

    At each sampling instant the amount of information storing required is:

    $$\text{Storing Info}=m+2n+\sum_{j=1}^n n_j \tag{28}$$

  - In I/O formulation, the following variables are needed: control actions vector $\boldsymbol{u}$ with m elements, outputs vector $\boldsymbol{y}$ with n elements, filtered outputs vector $\boldsymbol{y}^f$ with

    $$\sum_{r=1}^n\sum_{j=1}^m \max\left(1,\left|\text{degree}\left(B_{rj}\right)\right|-1\right)^{(3)}$$

    elements and filtered control action increments vector $\Delta\boldsymbol{u}^f$ with

    $$\sum_{r=1}^n \max\left(\left|\text{degree}\left(A_r\right)\right|+1,\left|\text{degree}\left(T_r\right)\right|\right)^{(4)}$$

    elements. Consequently:

    $$\begin{aligned}&\text{Storing Info}=\\ &=m+n+\sum_{r=1}^n\sum_{j=1}^m \max\left(1,\left|\text{degree}\left(B_{rj}\right)\right|-1\right)+\\ &+\sum_{r=1}^n \max\left(\left|\text{degree}\left(A_r\right)\right|+1,\left|\text{degree}\left(T_r\right)\right|\right)\end{aligned} \tag{29}$$

    As illustrated in the I/O case, the amount of information storing depends on more variables than in the state space case, in particular on filter polynomials degree and on B_{rj} polynomials degree. In order to simplify the analysis, the case of minimum amount is treated. In such situation, all B_{rj} polynomials degrees are minor or equal than two and filter polynomials have zero degree, therefore:

(3) Equal to the $\boldsymbol{\Gamma}$ matrix column number [2]

(4) Equal to the $\boldsymbol{F}$ matrix column number [2]

$$\sum_{r=1}^{n}\sum_{j=1}^{m}\max\left(1,\left|\text{degree}\left(B_{rj}\right)\right|-1\right)\geq\sum_{r=1}^{n}\sum_{j=1}^{m}1=n\cdot m \quad (30)$$

$$\sum_{r=1}^{n}\max\left(\left|\text{degree}(A_r)\right|+1,\left|\text{degree}(T_r)\right|\right)\geq$$
$$\geq\sum_{r=1}^{n}\left(\left|\text{degree}(A_r)\right|+1\right)=\sum_{r=1}^{n}(n_r+1) \quad (31)$$

$$\sum_{r=1}^{n}\max\left(\left|\text{degree}(A_r)\right|+1,\left|\text{degree}(T_r)\right|\right)\geq n+\sum_{r=1}^{n}n_r^{(5)}$$

Finally the amount of information storing verifies:

$$\text{Storing Info}\geq m+n+n\cdot m+n+\sum_{r=1}^{n}n_r=$$
$$=m+2n+n\cdot m+\sum_{r=1}^{n}n_r. \quad (32)$$

Comparing (28) and (32), it is deducted that I/O formulation requires a greater amount of information storing than state space formulation.

5 COST INDEX OF STATE SPACE GPC MIMO

In order to obtain the GPC control law, a quadratic cost index equivalent to the one proposed for I/O formulation (2), similar to the cost index used in [7], is proposed:

$$J_k(\hat{\boldsymbol{u}})=E\left[\sum_{i=N_1}^{N_2}\left(\bar{\boldsymbol{y}}(k+i)-\bar{\boldsymbol{w}}(k+i)\right)^T Q_i\left(\bar{\boldsymbol{y}}(k+i)-\right.\right.$$
$$\left.\left.-\bar{\boldsymbol{w}}(k+i)\right)+\sum_{i=1}^{N_u}\Delta\bar{\boldsymbol{u}}^T(k+i-1)R_i\Delta\bar{\boldsymbol{u}}(k+i-1)\right] \quad (33)$$

$$\boldsymbol{Q}_i\in\mathbb{R}^{n\times n};\quad \boldsymbol{R}_i\in\mathbb{R}^{m\times m}. \quad (34)$$

$\boldsymbol{w}(k+i)$ is the vector of the desired references in instant $k+i$.

To facilitate the obtention of the control law, it would be desirable to express the performance index on a matricial form. According to this, the following terms are defined:

$$\hat{\boldsymbol{w}}(k)=\left(\bar{\boldsymbol{w}}(k+N_1)\ \cdots\ \bar{\boldsymbol{w}}(k+N_2)\right)^T \quad (35)$$

$$\boldsymbol{Q}=diag\left(\boldsymbol{Q}_{N_1}\ \ \boldsymbol{Q}_{N_1+1}\ \cdots\ \boldsymbol{Q}_{N_2}\right) \quad (36)$$

$$\boldsymbol{R}=diag\left(\boldsymbol{R}_1\ \ \boldsymbol{R}_2\ \cdots\ \boldsymbol{R}_{N_u}\right) \quad (37)$$

(5) The A_r polynomial degree is equal to n_r, the number of states related to r-th output in the realisation of the CARIMA model deterministic part (3)

With these new terms, the index adopts the following matricial form:

$$J_k(\hat{\boldsymbol{u}})=\left(\hat{\boldsymbol{y}}(k)-\hat{\boldsymbol{w}}(k)\right)^T\boldsymbol{Q}\left(\hat{\boldsymbol{y}}(k)-\hat{\boldsymbol{w}}(k)\right)+\hat{\boldsymbol{u}}^T(k)\boldsymbol{R}\hat{\boldsymbol{u}}(k). \quad (38)$$

This expression is only valid if the future references are known. If these were unknown, it would have to assume that the future references are equal to the current one.

6 GPC MIMO CONTROL LAW

Unconstrained GPC control law is obtained by optimizing the performance index (38) with respect to (k) subject to the outputs prediction equation given by (17). Such law has the following form [2]:

$$\hat{\boldsymbol{u}}(k)=-\left(\boldsymbol{N}^T\boldsymbol{Q}\boldsymbol{N}+\boldsymbol{R}\right)^{-1}\boldsymbol{N}^T\boldsymbol{Q}^T\left(\boldsymbol{M}x(k)+\right.$$
$$\left.+\boldsymbol{O}\bar{\boldsymbol{u}}(k-1)+\boldsymbol{P}\bar{\boldsymbol{\xi}}(k)-\hat{\boldsymbol{w}}(k)\right). \quad (39)$$

This expression will be accurate if the $\boldsymbol{N}^T\boldsymbol{Q}\boldsymbol{N}+\boldsymbol{R}$ matrix is positive definite (minimum of the performance index). To ensure this fact, $\boldsymbol{Q}_i$ matrices must be positive definite, $\boldsymbol{R}_i$ matrices must be non-negative definite and $\boldsymbol{N}$ matrix should be of full rank.

With the former expression, all the increments of control actions in the control horizon are obtained. But because the methodology applied will be the receding horizon's, only the increments corresponding to the first instant of the control horizon will be applied. In this sense, matrix $\boldsymbol{\sigma}$ is defined as the first m rows of the $(\boldsymbol{N}^T\boldsymbol{Q}\boldsymbol{N}+\boldsymbol{R})^{-1}\boldsymbol{N}^T\boldsymbol{Q}^T$ matrix. With this assignation it is possible to obtain exclusively the increments belonging to the first instant of the control horizon:

$$\Delta\bar{\boldsymbol{u}}(k)=-\boldsymbol{\sigma}\left(\boldsymbol{M}x(k)+\boldsymbol{O}\bar{u}(k-1)+\boldsymbol{P}\bar{\boldsymbol{\xi}}(k)-\bar{\boldsymbol{w}}(k)\right). \quad (40)$$

7 DESIGN OF THE CARIMA OBSERVER

In any state space control technique, it is necessary to develop an observer system able to estimate the values of those states of the model that can not be directly measured with sensors. Evidently, it would be interesting to design observers for each particular case, however, this work does not intend to examine in depth any determined type of process. Accordingly, a full rank observer has been adopted for estimating the CARIMA model states (6):

$$\begin{aligned}\boldsymbol{x}(k+1)&=\bar{\boldsymbol{A}}\boldsymbol{x}(k)+\bar{\boldsymbol{B}}\bar{\boldsymbol{u}}(k)+\boldsymbol{\Pi}\bar{\boldsymbol{\xi}}(k)\\ \bar{\boldsymbol{y}}(k)&=\bar{\boldsymbol{C}}\boldsymbol{x}(k)+\boldsymbol{\Lambda}\bar{\boldsymbol{\xi}}(k)\end{aligned} \quad (41)$$

A full rank observer for this model is:

$$\hat{x}(k+1|k) = \bar{A}\hat{x}(k|k-1) + \bar{B}\bar{u}(k) + \Pi\bar{\xi}(k) + \\ + K_0\left[\bar{y}(k) - \bar{C}\hat{x}(k|k-1) - \Lambda\bar{\xi}(k)\right] = \\ = \left[\bar{A} - K_0\bar{C}\right]\hat{x}(k|k-1) + \bar{B}\bar{u}(k) + K_0\bar{y}(k) + \\ + \left[\Pi - K_0\Lambda\right]\bar{\xi}(k) \quad (42)$$

Because noise variables $\bar{\xi}(k)$ are white noises, its best prediction is always zero, so the observer's definitive expression is:

$$\hat{x}(k+1|k) = \left[\bar{A} - K_0\bar{C}\right]\hat{x}(k|k-1) + \\ + \bar{B}\bar{u}(k) + K_0\bar{y}(k). \quad (43)$$

For the design of the full rank observer, the feedback matrix K_0 must be chosen so that the matrix $\bar{A} - K_0\bar{C}$ has its eigenvalues on the adequate locations of the complex plane. Particularly these eigenvalues must all possess a module inferior to the unity to allow the stability of the observer's dynamic, and a module minor than that ones of the state matrix A of the CARIMA model (6) (observer's poles faster than the model's) to have the estimate values converging the real ones.

The problem generated by the inclusion of such observer (or any observer), is that it constitutes an artificial element in the control and therefore produces dynamics that are foreign both to the control and the process. In this work, the selection of K_0 is proposed from the CARIMA model manipulation (6):

$$x(k+1) = \bar{A}x(k) + \bar{B}\bar{u}(k) + \Pi\bar{\xi}(k) \\ \bar{y}(k) = \bar{C}x(k) + \Lambda\bar{\xi}(k). \quad (44)$$

Working out $\bar{\xi}(k)$ from the output equation and substituting it in the state equation:

$$x(k+1) = \bar{A}x(k) + \bar{B}\bar{u}(k) + \Pi\Lambda^{-1}\left[\bar{y}(k) - \bar{C}x(k)\right] \\ x(k+1) = \left[\bar{A} - \Pi\Lambda^{-1}\bar{C}\right]x(k) + \bar{B}\bar{u}(k) + \Pi\Lambda^{-1}\bar{y}(k). \quad (45)$$

As it can be seen, (45) is nothing more than a particular case of (43) full rank observer, taking $K_0 = \Pi\Lambda^{-1}$. If a CARIMA model is available, the matrices Π and Λ are given and consequently, with this observer's selection the design becomes limited and there is no guaranty of a stable dynamic. Thus, if this observer were chosen, it would be necessary to verify the dynamic's stability before performing its implementation.

The main advantage presented by this observer with respect to the rest is that because it uses matrix $\Pi\Lambda^{-1}$ as feedback matrix, no foreign dynamic is being introduced to that of the process + controller. This observer is called CARIMA observer, due to the way it is obtained (45).

The matrices corresponding to the noise model of the CARIMA model (not only Π and Λ) establish the dynamic of the CARIMA observer, as it can be seen by expanding its expression given by (45):

$$x(k+1) = \left[\begin{pmatrix} A & \Sigma \\ 0 & I \end{pmatrix} - \begin{pmatrix} \Pi_1 \\ \Pi_2 \end{pmatrix}\Lambda^{-1}\begin{pmatrix} C & \Omega \end{pmatrix}\right]x(k) + \\ + \bar{B}\bar{u}(k) + \Pi\Lambda^{-1}\bar{y}(k) \quad (46)$$

$$x(k+1) = \underbrace{\begin{bmatrix} A - \Pi_1\Lambda^{-1}C & \Sigma - \Pi_1\Lambda^{-1}\Omega \\ -\Pi_2\Lambda^{-1}C & I - \Pi_2\Lambda^{-1}\Omega \end{bmatrix}}_{\Gamma} x(k) + \\ + \bar{B}\bar{u}(k) + \Pi\Lambda^{-1}\bar{y}(k). \quad (47)$$

As a consequence it is also possible to design them according to the placement of the observer's poles in the desired places instead of using the methodology used in section 3, which consisted in obtaining them from determined filter polynomials. Using this alternative, there is no further need for a connection with formulation I/O when selecting the noise model's matrices. Furthermore, it is based on objective criteria regarding the position of the observer's poles in the desired locations, and not on a subjective formulation supported by the use of filter polynomials chosen by the designer according to his experience.

In [2] the problem of assigning particular λ_i values of matrix Γ has been studied, resulting on the one hand in:

$$\lambda_i = 1 - \Omega_{ii} \qquad i = 1, 2, \ldots, n \quad (48)$$

And on the other hand the polynomials'roots:

$$\lambda^{n_j} + \left(a_{n_j-1,j} + \Pi_{1j}(n_j, j)\right)\lambda^{n_j-1} + \cdots + \\ + \left(a_{1,j} + \Pi_{1j}(2,j)\right)\lambda + \left(a_{0,j} + \Pi_{1j}(1,j)\right) \\ j = 1, 2, \ldots, n \quad (49)$$

$a_{k,j}$ are the $A_j(z^{-1})$ polynomials' coefficients. It is necessary to go back to (11) and (13) in order to understand the previous expressions.

Setting the observer's poles from expressions (48) and (49), it is possible to design the noise model's matrices Ω and Π_1. The remaining matrices are given by the expressions:

$$\Pi_2 = \Lambda = I_n; \quad \Sigma = \Pi_1\Omega \quad (50)$$

Likewise, [2] establishes that the CARIMA observer has the roots of the $T_i(z^{-1})$ filter polynomials as poles. This conclusion is derived from the expressions obtained in this section and in section 3. This is a very important result, as it allows:

– Performing an objective selection of the filter polynomials in those applications in which an I/O version of the controller's design has been performed; considering its zeros as the poles of the observer that would be used in the state space version of such controller.

– Analysing the robustness of the I/O version controllers that were designed with filter polynomials selected according to different criteria from the one presented here.

8 APPLICATION EXAMPLE: INVERTED PENDULUM

The following application example is an inverted pendulum represented in figure 1. It is a nonlinear process with the following model [2]:

$$(m+M)\ddot{x}+d\frac{1}{2}ml\cos(\alpha)\ddot{\alpha}-\frac{1}{2}ml\mathrm{sen}(\alpha)\dot{\alpha}^2=d\frac{P}{r} \tag{51}$$

$$\frac{1}{3}ml^2\ddot{\alpha}+\frac{1}{2}ml\cos(\alpha)\ddot{x}-\frac{1}{2}mgl\mathrm{sen}(\alpha)=0 \tag{52}$$

$$P=0.0109V-0.0488\dot{x}\,. \tag{53}$$

P is the torque applied to the cart wheel by the dc motor. However, the control action in this system is the voltage applied to the dc motor, V. In order to design the GPC controller, its linearized model around the equilibrium point $\alpha=0$ was obtained:

$$\begin{bmatrix} X(s) \\ \alpha(s) \end{bmatrix}=\begin{bmatrix} \dfrac{3.387s^2-81.7}{s^4+15.14s^3-31.61s^2-365.1s} \\ \dfrac{-8.329s+0.00245}{s^3+15.14s^2-31.61s-365.1} \end{bmatrix}V(s) \tag{54}$$

being: $m=0.21$ Kg, $M=0.455$ Kg, $r=0.0063$ m and $l=0.61$ m.

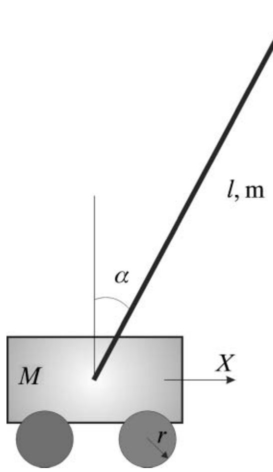

Fig. 1 Inverted pendulum

For this process, a state space GPC MIMO controller is designed after the following parameters:

– Sampling period: $T=0.01$ minutes. Prediction horizon: $N_1=1$ and $N_2=100$. Control horizon: $N_u=2$. Error pondering matrices: $\boldsymbol{Q}_i=\boldsymbol{I}$ $i=N_1$, ..., N_2. Control action increment pondering matrices: $\boldsymbol{R}_i=\boldsymbol{0}$ $i=1$, ..., N_u. Future references are unknown.

– CARIMA observer poles: four poles are located in 0.9 and the remaining in 0. This design is equivalent to use the following filter polynomials en the I/O version:

$$T_1\left(z^{-1}\right)=T_2\left(z^{-1}\right)=\left(1-0.9z^{-1}\right)\left(1-0.9z^{-1}\right). \tag{55}$$

According to these parameters the CARIMA model obtained is:

$$\begin{pmatrix} \bar{\boldsymbol{x}}(k+1) \\ \boldsymbol{x}^*(k+1) \end{pmatrix}=\begin{bmatrix} \boldsymbol{A} & \boldsymbol{\Sigma} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\begin{pmatrix} \bar{\boldsymbol{x}}(k) \\ \boldsymbol{x}^*(k) \end{pmatrix}+\begin{bmatrix} \boldsymbol{B} & \boldsymbol{\Pi}_1 \\ \boldsymbol{0} & \boldsymbol{\Pi}_2 \end{bmatrix}\begin{pmatrix} \bar{\boldsymbol{u}}(k) \\ \bar{\boldsymbol{\xi}}(k) \end{pmatrix}$$

$$\bar{\boldsymbol{y}}(k)=\begin{bmatrix} C & \Omega \end{bmatrix}\begin{pmatrix} \bar{\boldsymbol{x}}(k) \\ \boldsymbol{x}^*(k) \end{pmatrix}+\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Lambda} \end{bmatrix}\begin{pmatrix} \bar{\boldsymbol{u}}(k) \\ \bar{\boldsymbol{\xi}}(k) \end{pmatrix}$$

$$A=\begin{bmatrix} 0 & 0 & 0 & -0.8595 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 3.5814 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & -5.5845 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 3.8626 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.85295 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 & -2.7218 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 & 2.8626 \end{bmatrix} \tag{56}$$

$$\boldsymbol{\Sigma}=\boldsymbol{\Pi}_1=\begin{bmatrix} -0.8595 & 0 \\ 3.5814 & 0 \\ -4.7745 & 0 \\ 2.0626 & 0 \\ 0 & 0.8595 \\ 0 & -1.9118 \\ 0 & 1.0626 \end{bmatrix} \tag{57}$$

$$\boldsymbol{B}=\begin{pmatrix} 0.1532 \\ -0.1456 \\ -0.1694 \\ 0.1611 \\ 0.3768 \\ 0.0195 \\ -0.3963 \end{pmatrix}\cdot 10^{-3} \tag{58}$$

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{59}$$

$$\boldsymbol{\Pi}_2 = \boldsymbol{\Omega} = \boldsymbol{\Lambda} = \boldsymbol{I}_2 \,. \tag{60}$$

The remaining matrices associated to the controller's design are omitted due to their dimensions.

The results obtained from the simulation of the inverted pendulum control with the state space GPC are shown on Figure 2. These results are identical to those obtained designing the I/O version of the controller. The references imposed are 0.1 for output 1 and 0 for output 2. As it can be seen, the answer of both variables is smooth and without oscillations (except the non minimum-phase behaviour). Control action also show a smooth and oscillation-free evolution.

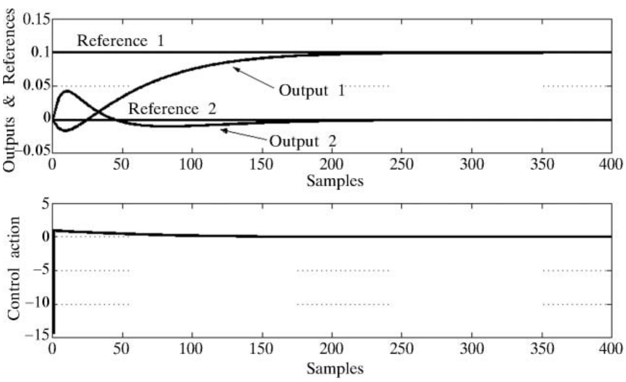

Fig. 2 Control of an inverted pendulum using state space GPC

In order to be able to establish a quantitative comparison between the state space controller designed and the I/O one designed according to the same parameters, figures 3 and 4 are used. These figures represent the control actions calculation time for the typologies of both controllers in the case of the stirred tank reactor's control.

As it can be appreciated, the calculation time in almost every sampling instant is always lower for the state space GPC.

These results are obtained using the following Matlab commands:

```
t1=clock;
% Here the control action algorithm is
developed
t2=clock;
time(i)=etime(t2,t1);
```

If the values `t1` and `t2` are very close the function `etime` returns 0, the reason is that Matlab can not calculate elapsed times lesser than 0.05 seconds.

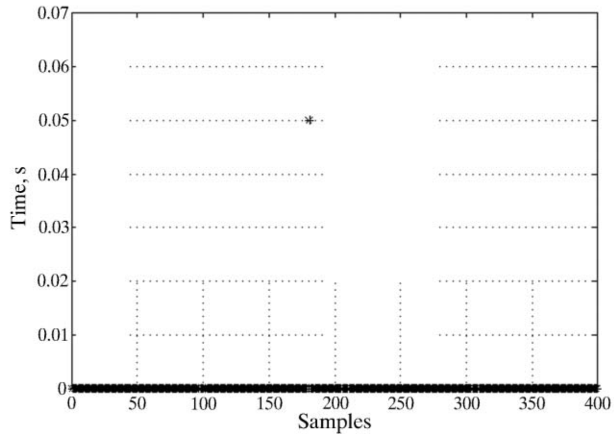

Fig. 3 Calculation time of control action of the state space GPC for

the inverted pendulum

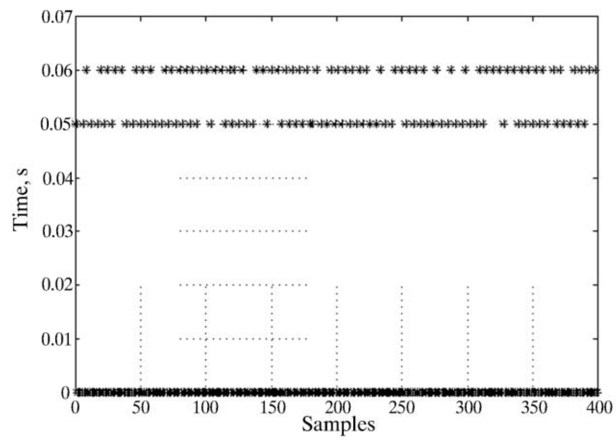

Fig. 4 Calculation time of control action of the I/O GPC for the in-

9 CONCLUSIONS

- An original CARIMA model has been proposed for state space formulation, together with its equivalency to the I/O model used in the literature.
- The matrices obtained in the prediction model directly depend on the CARIMA model matrices, instead of depending recursively as in the I/O version.
- The disadvantage with respect to the I/O methodology is resolved with the use of the CARIMA observer, as this observer introduces a dynamic, which is not foreign to the process.
- The design of such observer allows establishing a determinate CARIMA model, which on the beginning had to be done according to the knowledge of filter polynomials.
- The observer's poles are confirmed to be the filtrate polynomials' roots. From this derives the possibility of designing them according to a control

objective criteria instead of according to the designer's experience, as it was being done up until now.

– State space and I/O methodology reach identical results, although the later generally requires longer calculation time and greater information storing.

ACKNOWLEDGEMENT

The authors would like to thank the Foreign Languages Coordination Area (ACLE) of the Technical University of Valencia (Universidad Politécnica de Valencia) for the translation of this article to the english language.

REFERENCES

[1] J. Sanchis, **GPC Design for MIMO Processes. Constraint Analysis at the Actuators and Application to Real Processes**. Tech. Rep., Dept. of Systems Engineering and Control, Technical Unversity of Valencia, Spain, Sept. 1997, (In Spanish).

[2] J. V. Salcedo, **Comparative Between the Design of GPC's MIMO I/O vs State Space with Constraints Optimized with the Method of Rosen**. Tech. Rep., Dept. of Systems Engineering and Control, Technical University of Valencia, Spain, Sept. 1999, (In Spanish).

[3] E. F. Camacho, C. Bordons, **Model Predictive Control in the Process Industry**. Springer, 1995.

[4] D. W. Clarke, C. Mohtadi, P. S. Tuffs, **Generalized Predictive Control – Part I**. Automatica, vol. 23, no. 2, pp. 137–148, 1987.

[5] D. W. Clarke, C. Mohtadi, P. S. Tuffs, **Generalized Predictive Control – Part II**. Extensions and Interpretations. Automatica, vol. 23, no. 2, pp. 149–160, 1987.

[6] D. W. Clarke, C. Mohtadi, **Properties of Generalized Predictive Control**. Automatica, vol. 25, no. 6, pp. 859–875, 1989.

[7] A. W. Ordys, David W. Clarke, **A State-Space Description for GPC Controllers**. Int. J. Systems Sci., vol. 24, pp. 1727–1744, 1993.

[8] P. Albertos, R. Ortega, **On Generalized Predictive Control: Two alternative formulations**. Automatica, vol. 25, no. 5, pp. 753–755, 1989.

[9] K. V. Ling, K. W. Lim, **A State Space GPC with Extensions to Multirate Control**. Automatica, vol. 32, no. 7, pp. 1067–1071, 1996.

[10] W. H. Kwon, H. Choi, D. G. Byun, S. Noh, **Recursive Solution of Generalized Predictive Control and its Equivalence to Receding Horizon Tracking Control**. Automatica, vol. 28, no. 6, pp. 1235–1238, 1992.

Izvedba poopćenih predikcijskih regulatora u prostoru stanja. U članku se opisuje izvorna metodologija projektiranja i izvedbe poopćenih predikcijskih regulatora (GPC-a) zasnovana na primjeni CARIMA modela u prostoru stanja za predikciju stanja. Opisani CARIMA model ekvivalentan je najčešće korištenom CARIMA modelu koji se koristi pri ulazno/izlaznoj formulacji GPC-a. Uspostavljena je veza između stohastičkog dijela modela i polinoma $\boldsymbol{T}_i(z^{-1})$, čime je omogućena sinteza regulatora kada je poznat bilo koji od njih. Za estimaciju nemjerljivih veličina stanja predložen je estimator punog reda, koji, što je posebno važno, ima polove jednake korijenima polinoma $\boldsymbol{T}_i(z^{-1})$.

Ključne riječi: CARIMA model, optimizacija, prediktivno upravljanje, upravljanje u prostoru stanja

AUTHORS' ADDRESSES:

José Vicente Salcedo. Industrial Engineering. Lecturer.
Miguel Martínez. PhD on Industrial Engineering. Professor.
Javier Sanchis. Bachelor Degree in Computer Science. Lecturer.
Xavier Blasco. PhD on Industrial Engineering. Lecturer.

Department of Systems Engineering and Control
Universidad Polit cnica de Valencia
(Technical University of Valencia) Camino de Vera 14,
P.O. Box 22012 E-46071 Valencia, Spain.
Tel.: +34-963877000 Ext: 5766
Fax +34-963879579
E-mail: jsalcedo@isa.upv.es
web page: ctl-predictivo.upv.es

Received: 2001–07–12