PERGAMON

Annual Reviews in Control 28 (2004) 47–52

Annual Reviews in Control

www.elsevier.com/locate/arcontrol

# A multi-model structure for model predictive control

F. Di Palma[*], L. Magni

*Dipartimento di Informatica e Sistemistica, Università di Pavia, via Ferrata 1, 27100 Pavia, Italy*

Received 20 October 2003; received in revised form 20 January 2004; accepted 24 January 2004

**Abstract**

Model predictive control (*MPC*) is a wide popular control technique that can be applied starting from several model structures. In this paper black-box models are considered. In particular it is analysed the sets of regressors that it is better to use in order to obtain the best model for multi-step prediction. It is observed that for each prediction a different set of real data output and predicted output are available. Based on this observation a multi-model structure is proposed in order to improve the predictions needed in the computation of the *MPC* control law. A comparison with a classical one-model structure is discussed. A simulation experiment is presented.

*Keywords:* Model predictive control; Black-box identification; Multiple models

## 1. Introduction

Model predictive control (*MPC*) has gained wide popularity in industrial process control due to the possibility of reformulating the control problem as an optimization problem in which many physical constraints and nonlinearities can be allowed for (Clarke, 1994; Camacho & Bordons, 1995; Kouvaritakis & Cannon, 2001). Another reason for the success is its ability to obtain good performances starting from rather intuitive design principles and simple models, such as truncated impulse responses or step responses, see e.g. Richalet, Rault, Testud, and Papon (1978) and Cutler and Ramaker (1980) algorithms. In order to reduce the modeling and identification phase black-box model are often used in *MPC* packages. Black-box identification techniques, see e.g. Ljung (1987), Sjoberg et al. (1995) and Nelles (2001), can in fact used to quickly derive models from experimental data by means of an estimation parameters procedure. The general problem is to find a relationship between past observation and future outputs. The choice of this map can be decomposed into two partial problems: how to choose the regressor vector from past inputs and outputs and how to choose the mapping from the regressor space to the output space. The regressors are generally given by past observations of the input signal, past observations of the output, simulated outputs from past input only, prediction errors, simulation errors. Each set of regressors defines a particular class of models that can be suitable for different issues. One of the most common application of a model is the forecast of process behavior. Two cases have to be distinguished: simulation and prediction. If the response of the model to an input sequence has to be calculated while the process outputs are unknown, this is called simulation. If, however, the process outputs are known up to the current time instant and it is asked for the model output at the next step, this is called one-step-ahead prediction. It is well known that a good model for prediction can be not suitable for simulation and vice versa. The aim of this paper is to characterize the most suitable class of models for model predictive control. In particular because, in order to compute the cost function to be minimized, the output must be predicted $l$ step in the future with $l = 1, \ldots, N_p$ where $N_p$ is the prediction horizon, the use of different models to make each prediction is proposed. Each model is obtained with a proper set of regressors that contain the different composition of the past observations of the output and of the simulated outputs. The organization of the paper is as follows. In Section 2 we introduce the main model structure used in system identification. In Section 3 the *MPC* control strategy is proposed and the use of an internal model is discussed. Section 4 describes the new multi-model structure. The identification and prediction algorithms for this structure are discussed. In Section 5 a simulation example is given.

[*] Corresponding author. Tel.: +39-0382-505437; fax: +39-0382-505373.

*E-mail addresses:* federico.dipalma@unipv.it (F. Di Palma), lalo.magni@unipv.it (L. Magni).

1367-5788/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.arcontrol.2004.01.004

## 2. Nonlinear black-box structures

In this section a general black-box identification problem is introduced for nonlinear systems. Given two vectors

$$u^{t,n_u} = [u(t) \quad u(t-1) \quad \cdots \quad u(t+1-n_u)] \tag{1}$$

$$y^{t,n_y} = [y(t) \quad y(t-1) \quad \cdots \quad y(t+1-n_y)] \tag{2}$$

where $t$ is the discrete time index, $u \in R^m$ is the input, $y \in R^m$ is the output while $n_u$ and $n_y$ are two real positive constants, we are looking for a relationship between present and past observations $[u^{t,n_u}, y^{t,n_y}]$ and future output $y(t+1)$

$$y(t+1) = g(u^{t,n_u}, y^{t,n_y}) + v(t+1)$$

The term $v(t+1)$ accounts for the fact that the next output $y(t+1)$ will not an exact function of past data. However, a goal must be that $v(t+1)$ is small, so that we may think of $g(u^{t,n_u}, y^{t,n_y})$ as a good prediction of $y(t+1)$ given past data. In order to find the function $g$ let us parametrize a family of function within we have to search for it:

$$g(u^{t,n_u}, y^{t,n_y}, \theta) \tag{3}$$

The target is to find a suitable parametrization in order to obtain a good model. Because a parametrization of the function $g$ with a finite-dimension vector $\theta$ is an approximation, in order to decide the structure it is crucial to know what the model will be used for. Once we have decided the structure and we have collected $N$ data the parameter $\theta$ can be obtained by means of the fit between the model and the data record:

$$\sum_{t=\max(n_u,n_y)}^{N} \|y(t) - g(u^{t-1,n_u}, y^{t-1,n_y}, \theta)\|$$

The model structure family (3) is quite general, and it turns out to be useful to write $g$ as a concatenation of two mappings: one that takes the increasing number of past observations and maps them into a finite-dimensional vector $\varphi(t)$ of fixed dimension and one that takes vector to the space of outputs:

$$g(u^{t,n_u}, y^{t,n_y}, \theta) = g(\varphi(t), \theta) \tag{4}$$

According to a rather classical terminology, we shall call the vector $\varphi(t)$ the regression vector, and its components will be referred to as regressors. The choice of the nonlinear mapping (3) has thus been decomposed into two partial problems for dynamic systems: how to choose the regression vector $\varphi(t)$ from past inputs and outputs and how to choose the nonlinear mapping $g(\varphi)$ from the regressor space to the output space. We thus work with prediction models of the kind

$$\hat{y}(t+1, \theta) = g(\varphi(t), \theta) \tag{5}$$

The regressors, in the general case, are given by

(i) $u(t-k)$, past inputs;
(ii) $y(t-k)$, past outputs;
(iii) $\hat{y}_u(t-k)$, simulated outputs from past $u$ only;
(iv) $\varepsilon(t-k) := y(t-k) - \hat{y}(t-k)$, prediction errors;
(v) $\varepsilon_u(t-k) := y(t-k) - \hat{y}_u(t-k)$, simulation errors.

The simulated output $\hat{y}_u(t-k, \theta)$ is the output from the model (4) if all measured outputs $y(t-k)$ in the regressors are replaced by the last computed $\hat{y}_u(t-k, \theta)$. The regressors (iii)–(v) depend on the black-box model (3), so we should write $\varphi(t, \theta)$ instead of $\varphi(t)$ in (4).

Based on different combinations of the regressors we could thus distinguish between the following model classes:

- *NFIR* (Nonlinear Finite Impulse Response) models, which use only $u(t-k)$ as regressors;
- *NARX* (Nonlinear AutoRegressive with eXogenous input) models, which use $u(t-k)$ and $y(t-k)$ as regressors;
- *NARXAR* (Nonlinear AutoRegressive with eXogenous AutoRegressive input) models, which use $u(t-k)$, $y(t-k)$ and $\hat{y}_u(t-k, \theta)$ as regressors;
- *NOE* (Nonlinear Output Error) models, which use $u(t-k)$ and $\hat{y}_u(t-k, \theta)$ as regressors;
- *NARMAX* (Nonlinear AutoRegressive Moving Average with eXogenous input) models, which use $u(t-k)$, $y(t-k)$ and $\varepsilon(t-k, \theta)$ as regressors;
- *NBJ* (Nonlinear Box-Jenkins) models, which use $u(t-k)$, $\hat{y}(t-k, \theta)$, $\varepsilon(t-k, \theta)$ and $\varepsilon_u(t-k, \theta)$ as regressors; in this case the simulated output $\hat{y}_u$ is obtained as the output from (3), by using the same structure, replacing $\varepsilon$ and $\varepsilon_u$ by zeros in the regression vectors $\varphi(t, \theta)$.

The aim of this paper is to find the better model structure in order to make prediction for nonlinear model predictive control. In the next section the control problem and its main peculiarity are discussed.

## 3. Nonlinear model predictive control

The system under control is assumed to be described by an unknown state equation of the form

$$x(t+1) = f(x(t), u(t), e(t)), \qquad x(t_0) = \bar{x} \tag{6}$$

$$y(k) = h(x(t)) \tag{7}$$

where $t_0$ is the initial state, $x \in R^n$ represents the system state, $u \in R^m$ is the input vector, $y \in R^m$ is the output and $e(t) \in R^p$ is the noise.

The problem here considered is to design a control algorithm such that the output must track the reference signal $y_{\text{ref}}(\cdot)$.

If the model of the system (6) and (7) is known, $e(t) = 0$ and any equilibrium associated with a constant input is asymptotically stable, the *MPC* control law (Mayne, Rawlings, Rao, & Scokaert, 2000) can be solved by the following

*Finite Horizon Optimal Control Problem* (*FHOCP*): given the state system $\bar{x}$, the last control value $u(t-1)$, the positive integers $N_c$ (*control horizon*) and $N_p$ (*prediction horizon*), $N_c \leq N_p$, the positive definite matrices $Q$, $R$, minimize, with respect to $u^{t+N_c-1,N_c}$, the performance index

$$J(\overline{x}, u(t-1), u^{t+N_c-1,N_c}, N_c, N_p) = \sum_{k=t}^{t+N_p-1} \{(y(k) - y_{\text{ref}}(k))' Q (y(k) - y_{\text{ref}}(k)) + (u(k) - u(k-1))' R (u(k) - u(k-1))\} \tag{8}$$

subject to

(i) the model system dynamics (6) and (7);
(ii) the control signal

$$u(t+k) = \begin{cases} u^{t+N_c-1,N_c}, & k \in [0, N_c - 1] \\ u(t+N_c-1), & k \in [N_c, N_p - 1] \end{cases} \tag{9}$$

□

According to the *Receding Horizon* approach, the state-feedback *MPC* control law is derived by solving the *FHOCP* at every time instant $t$, and applying the control signal $u = u_{OPT}^{t,t}$, where $u_{OPT}^{t+N_c-1,N_c}$ is the solution of the *FHOCP*. In so doing, one implicitly defines the state-feedback control law

$$u(t) = \kappa^{RH}(x(t)) \tag{10}$$

In order to compute the cost function (8) the future output values $y^{t+N_p-1,N_p-1}$ must be predicted according to an internal model of the system. In this paper we assume that the state-space model (6) and (7) is unknown and that a black-box input–output model in the form (5) described in Section 2 has been identified. Then the cost function (8) must be re-written as

$$J(\varphi(t), u(t-1), u^{t+N_c-1,N_c}, N_c, N_p) = \sum_{k=t}^{t+N_p-1} \{(\hat{y}(k|t) - y_{\text{ref}}(k))' Q (\hat{y}(k|t) - y_{\text{ref}}(k)) + (u(k) - u(k-1))' R (u(k) - u(k-1))\} \tag{11}$$

where $\hat{y}(k|t)$ indicates the prediction output values at time $k$ based on the input values $u(i)$, $i \leq k$, on the measured output values up to time $t$ (on other word $t+i$, $i \leq 0$) and on a model in the form (5). Note that in this case the role of the initial state $\bar{x}$ is up to the regression vector at time $t$, $\varphi(t)$.

Usually, (Nelles, 2001) a single model is used to evaluate the whole future output's sequence $\hat{y}^{t+N_p-1,N_p-1}$ associated with the input sequence (9). Employing only one model, the requested sequence is obtained iterating the internal model $N_p$ times with the available data. To generate the first prediction $\hat{y}(t+1|t)$ according to (5) the internal model must be feed with the past regressors values $\varphi(t|t)$. The notation $\cdot|t$, as previously used, indicate that the vector $\varphi(t)$ may be built bounding the old system output only up to time $t$.

Once $\hat{y}(t+1|t)$ has been computed to obtain the next prediction $\hat{y}(t+2|t)$ a new vector $\varphi(t+1|t)$ must be provided. The vector $\varphi(t+1|t)$ may be built from $\varphi(t|t)$ using a *Shift Register* (*SR*) for each kind of regressors. For each *SR* the update is made losing the oldest value and putting the new element. Obviously the *SR* number is due to the kind of the chosen model. The process is repeated $N_p$ times until the prediction $\hat{y}(t+N_p|t)$ is reached.

It is important to notice that the whole procedure must be executed at any time $t$ and for every input sequences (9) used to solve the *FHOCP*. In fact at system time $t+1$ each *SR* must be re-initialized with the proper component of regression vector $\varphi(t+1|t+1)$.

The prediction algorithm can be summarized in the following algorithm:

(1) Initialize the *SR* to the proper component of regression vector $\varphi(t|t)$, set $i = 1$.
(2) Compute $\hat{y}(t+i|t)$ according to (5).
(3) Set $i = i+1$ and update the *SR*:
  (a) for each *SR*, remove the oldest regressor values;
  (b) for each *SR*, insert the new regressor values.
(4) If $i \leq N_p$ then go to point 2.

At point 3(b) the new regressor values are asked in order to compute the next prediction. The problem is that the past real output values are available only up to current time $t$. Then, if a model which has the past outputs among its regressors is used, the new regressors are not available in order to implement point 3(b). This happens, for example, when a general *NARX*, *NARMAX*, *NARXAR* and *NBJ* models are considered (see Section 2). In practice the only thing that one can do in order to avoid this problem without modifying the model structure is to use the prediction at the previous steps instead of the unknown output real values. It is clear that following this procedure the model is identified considering a particular set of regressors and used for prediction with a different set of regressors (i.e. predictions are used instead of real outputs).

It is well know that a model identified in order to minimize the one-step-ahead prediction error is not in general a good model for simulation. For linear systems, for example, it is possible that the model of a stable system that minimize the one-step-ahead prediction error is even unstable (Nelles, 2001).

Another possibility is the use of *NOE* or *NFIR* models that can be correctly used because they do not count among its regressors past outputs as remarked in Section 2. Otherwise, using a *NOE* or *FIR* models the information of the real output value up to time $t$ is not used.

In the next section we propose a Multi-Model (*MM*) structure that fully used the available information with a correct chosen of the regressors.

## 4. Multi-model structure

To avoid the difficulty explained in the precedent section, we propose the use of different models for each requested prediction. The basic observation that motivates the proposed structure is that the set of the regressors that is possible to use in order to make the prediction is different for each $k$. In fact increasing $k$ the number of past known outputs decreases, while one can use only the output simulated with the identified model. This means that for $k = t + 1$, the regressors we can use are $u(i)$, $i < k$ and $y(i)$, $i < k$ so that we are considering a *NARX* model. On the contrary for $k > t + 1$ the regressors we can use are $u(i)$, $i < k$, $y(i)$, $i \leq t$ and $\hat{y}_u(i|t)$, $t < i < k$ so that we are considering a *NARXAR* model with a particular set of regressors. In view of this consideration, in the *MM* scheme $N_p$ models are identified and each model is characterized by a different regressor vector $\varphi_j$ composed by a different composition of real outputs and simulated outputs.

If only a finite number of regressors is considered, when $k$ is strictly greater than $t+n_y$, where $n_y$ is the maximum delay in the output regressors (2), only input and simulated output can be considered in the regressors so that a *NOE* model must be considered. In this case, if $n_y < N_P$ only $n_y + 1$ models can be used to compute the performance index (11).

Remarkably each model must be initialized with its own regression vector $\varphi_j(t|t - j + 1)$, and then run up for $j$ steps in order to obtain the predicted value at time $t + j$. This means that the first model in one simulation step reaches the requested predicted output $\hat{y}(t + 1|t)$ while $\hat{y}(t + 2|t)$ is obtained in two simulation steps of the second model, and so on. Moreover you can note that, in order to obtain a correct use of the model, the vector $\varphi_j(t|t - j + 1)$ must be formed by the old simulation even if some real output is known.

Each model is then defined by the regression vector $\varphi_j(t|t - j + 1)$, the function $g_j(\cdot, \cdot)$ (5) and the parameter vector $\theta(j)$. The predicted output will be called $\hat{y}_j$ and its Shift Register $SR_j$.

### 4.1. Multi-model prediction algorithms

Once understood the *MM* scheme aim, the prediction algorithm is naturally obtained extending the one reported in Section 3.1. In fact in the *MM* scheme $n_y + 1$ models are present, and the prediction algorithm is the parallel of $n_y+1$ prediction algorithm with progressive arrest time.

(1) Set $i = 1$ and for each model $j$ with $1 \leq j \leq n_y + 1$, initialize the $SR_j$ to the proper component of regression vector $\varphi_j(t|t - j + 1)$.
(2) For $j \geq i$ calculate $\hat{y}_j(t + i|t - j + i)$ according to (5).
(3) Set $\hat{y}(t + i|t) = \hat{y}_i(t + i|t)$.
(4) Set $i = i + 1$ and update the $SR_j$, with $j \geq i$:
   (a) for each $SR_j$, remove the oldest regressor value;
   (b) for each $SR_j$, insert the new regressor value.
(5) If $i \leq n_y$ then go to pint 2.
(6) Calculate $\hat{y}_{n_y+1}(t + i)$ according to (5).
(7) Set $\hat{y}(t + i|t) = \hat{y}_{n_y+1}(t + i)$.
(8) Set $i = i + 1$ and update the $SR_{n_y+1}$:
   (a) for each $SR_{n_y+1}$, remove the oldest regressor value;
   (b) for each $SR_{n_y+1}$, insert the new regressor value.
(9) if $i \leq N_p$ then go to point 6.

### 4.2. Multi-model identification algorithm

In order to reduce the computational burden while maintain the main properties of the multi-model structure (i.e. the use of a different mix of output data and simulated data among the regressors) we assume that all the functions $g_j(\cdot, \cdot)$ are equal to $g(\cdot, \cdot)$. On the other hand, each model will be characterized by a possible different set of parameter $\hat{\theta}(j)$ for each $j$. In this way, the problem to find a proper structure is solved only one time.

Then, given the nonlinear mapping $g(\cdot, \cdot)$ and the integer constants $n_u$ end $n_y$, the *MM* identification algorithm is based on the following steps:

(1) Given the regression vector $\varphi_1(t|t)$, function of $y(t + i)$, $-n_y \leq i < 0$, $u(t + i)$, $- n_u \leq i < 0$, find the optimal value of $\hat{\theta}(1)$ for the *NARX* model with a prediction criterion.
(2) For each $1 < j \leq n_y$, find the optimal value of $\hat{\theta}(j)$ using the regression vector $\varphi_j(t|t - j + 1)$ given by $\varphi_{j-1}(t|t - j + 2)$with $y(t - j)$ substituted by $\hat{y}_{u_j}(t - j|t - j + 1)$ for the *NARXAR* model with a prediction criterion.
(3) Find the optimal value of $\hat{\theta}(n_y + 1)$ using the regression vector $\varphi_{n_y+1}(t)$ given by $\varphi_1(t|t)$ with $y(t - i)$ substitute by $\hat{y}_{u_{n_y+1}}(t - i)$ $\forall i$ for the *NOE* model with a prediction criterion.

Note that in step (3) $\varphi_{n_y+1}(t)$ does not contain past output but only simulated output from past input.

The real importance of these two algorithms is that the different models are identified and used with the same sets of regressors differently from the standard approach described in Section 3. It is obvious that in this way the obtained cost function (11) will be more carefully computed because each prediction is based on the “best” model within the considered class of function determined by $g(\cdot, \cdot)$.

## 5. Simulation example

In this section a simulation example based on the single-input, single-output linear time invariant system

$$A(d)y(t) = C(d)e(t) + \frac{B(d)}{F(d)}u(t) \tag{12}$$

where $d$ is the backward shift operator (Ljung, 1987), $A(d) = 0.3d^2 - 1.1d + 1$, $B(z) = -d^2 + d$, $C(d) = -1.5d + 1$, $F(d) = 0.08d^2 - 0.4d + 1$ is given.

It is shown that the proposed *MM* structure guarantees significant advantages even for a very simple model. In particular the *MM* structure makes the prediction at each step with the best model so that is gained more accuracy than

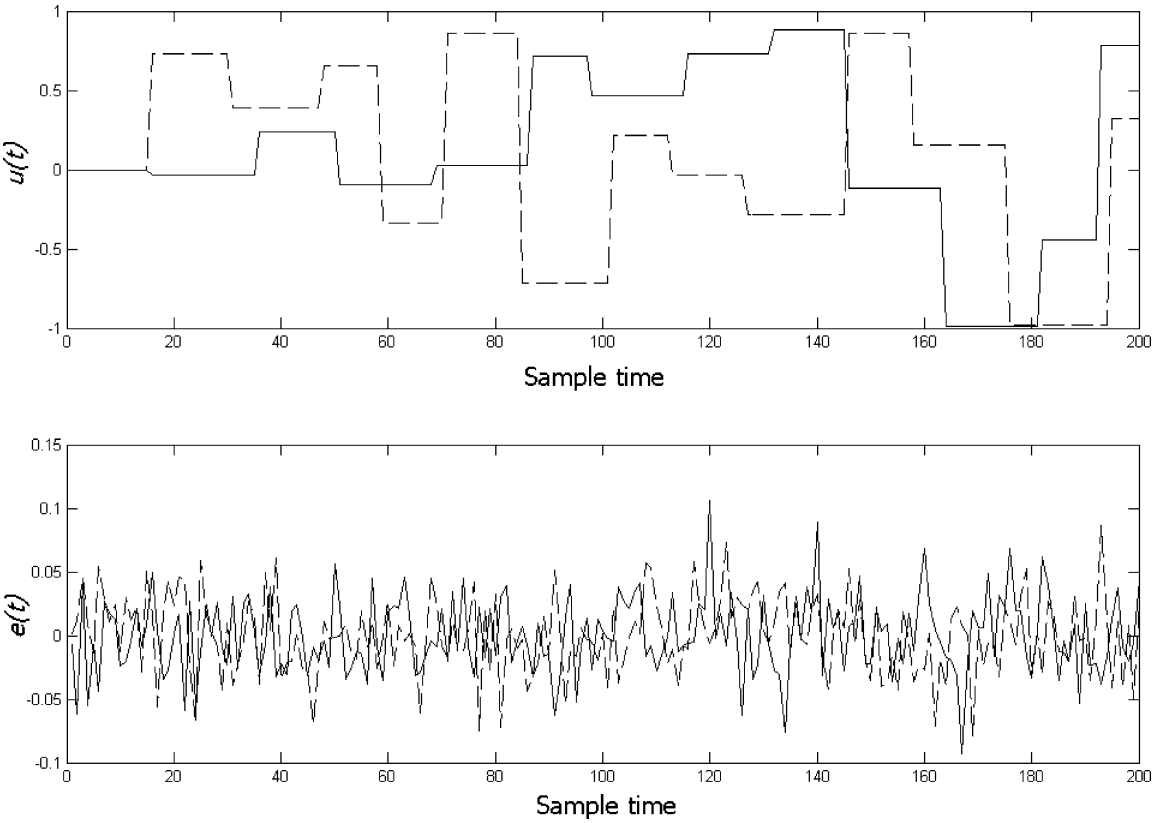

Fig. 1. Identification (solid line) and validation (dashed line) input signals.

with a single model structure such as *NARX* or *NOE* that are usually used in literature.

### 5.1. MM structure identification

First of all the identification data (1), (2) was obtain feeding the system (12) with a *Multi-level Pseudo-Random Signal* (*MPRS*) for the input $u(t)$ (Braun, Rivera, Stenman, Foslien, & Hrenya, 1999) and a *Random Gaussian Signal* (*RGS*) for the error $e(t)$ (Ljung, 1987). The signals are reported in Fig. 1 with solid line.

Then the mapping $g_j(\cdot,\cdot)$ has been chosen as a linear map and the constants $n_u$ and $n_y$ has been respectively fixed equal to 2 and 4. Then the identification algorithm of Section 4.2 was applied and the next five models are obtained:

$$\begin{aligned}\hat{y}_1(t+1|t) = {} & +0.6766u(t) - 0.6739u(t-1) - 0.8269y(t) \\ & - 0.1674y(t-1) + 0.296y(t-2) \\ & - 0.08282y(t-3)\end{aligned}$$

$$\begin{aligned}\hat{y}_2(t+2|t) = {} & +1.957u(t+1) - 1.951u(t) \\ & + 0.2193\hat{y}_2(t+1) - 1.134y(t) \\ & + 0.62y(t-1) - 0.09467y(t-2)\end{aligned}$$

$$\begin{aligned}\hat{y}_3(t+3|t) = {} & +1.523u(t+2) - 1.522u(t+1) \\ & - 0.04702\hat{y}_3(t+2) - 0.4564\hat{y}_3(t+1) \\ & - 0.5582y(t) + 0.2463y(t-1)\end{aligned}$$

$$\begin{aligned}\hat{y}_4(t+4|t) = {} & +0.7386u(t+3) - 0.7347u(t+2) \\ & - 1.122\hat{y}_4(t+3) + 0.1193\hat{y}_4(t+2) \\ & + 0.1717\hat{y}_4(t+1) - 0.03102y(t)\end{aligned}$$

$$\begin{aligned}\hat{y}_5(t+5) = {} & 0.7168u(t+4) - 0.7141u(t+3) \\ & - 1.298\hat{y}_5(t+4) + 0.2712\hat{y}_5(t+3) \\ & + 0.2308\hat{y}_5(t+2) - 0.068\hat{y}_5(t+1)\end{aligned}$$

### 5.2. MM prediction analysis

Once identified the *MM* structure, we test them on the validation data set reported in Fig. 1 with dashed line. As quality index we consider the *Sum of Prediction Error* (*SPE*) that, given the prediction step $i$ and the model $j$, is defined as

$$SPE(i, j) = \sum_{t=n_m}^{N} \|\hat{y}_j(t+i|t) - y(t+i)\|$$

where $n_m = \max(n_u, n_y)$. In order to appraise the different model's quality the values of $SPE(i, j)$ are reported in Fig. 2.

To globally evaluate the model effectiveness in computing the *MPC* cost function (11), where we fixed $N_p = 10$, the *SPE* must be extended along the prediction horizon. Then for a single model based structure the total cost is given by

$$SPE_j = \sum_{i=1}^{N_p}(SPE(i, j))$$

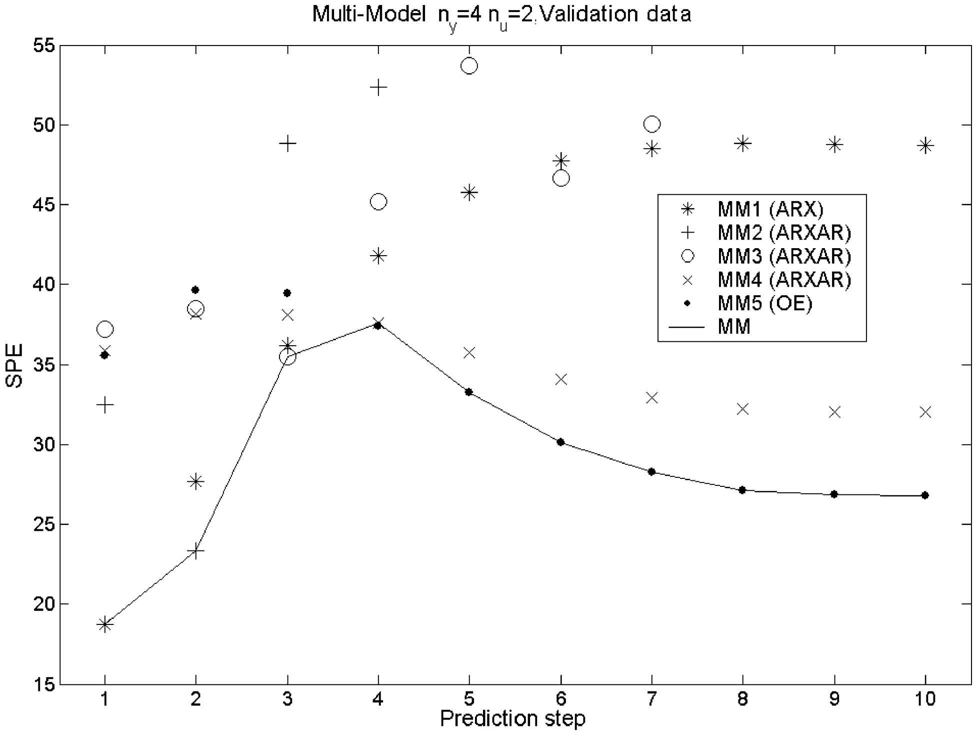

Fig. 2. $SPE(i, j)$ for singol models and $MM$ scheme.

Table 1
Confront single-model/*MM* structure

| $SPE_2/SPE_1$ | $SPE_3/SPE_1$ | $SPE_4/SPE_1$ | $SPE_5/SPE_1$ | $SPE_{MM}/SPE_1$ |
|---|---|---|---|---|
| 262.5% | 116.3% | 84.5% | 78.6% | 69.7% |

while for the *MM* structure is given by

$$SPE_{MM} = \sum_{j=1}^{N_p} (SPE(j, j))$$

In Table 1 the *SPE* variation of the different model related to the *SPE* of the *ARX* model are reported. It is apparent that the *MM* structure obtains better performance that single models, in view of the possibility to use different model at each step.

## 6. Conclusion

In this paper a new structure for system identification of models for *MPC* is presented. This structure, called *MM* structure, is based on the identification of different models in order to optimize the prediction on each step involved in the computation of the cost function. An example is presented to show the potentiality of this approach.

## Acknowledgements

The author acknowledge the partial financial support by *MURST* Project “New techniques for the identification and adaptive control of industrial systems”.

## References

Braun, M. W., Rivera, D. E., Stenman, A., Foslien, W., & Hrenya, C. (1999). Multi-level pseudo-random signal design and “model-on-demand” estimation applied to nonlinear identification of a rtp wafer reactor. *American Control Conference*. San Diego, CA.

Camacho, E., & Bordons, C. (1995). *Model predictive control in the process industry*. Springer.

Clarke, D. W. (1994). *Advances in model-based predictive control*. Oxford University Press.

Cutler, C. R., & Ramaker, B. C. (1980). Dynamic matrix control—a computer control algorithm. *Automatic Control Conference*.

Kouvaritakis, B., & Cannon, M. (2001). *Nonlinear predictive control: Theory and practice*. IEE.

Ljung, L. (1987). *System identification, theory for the user*. Englewood Cliffs, NJ: Prentice-Hall.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*, 789–814.

Nelles, O. (2001). *Nonlinear system identification: From classical approaches to neural networks and Fuzzy models*. Springer.

Richalet, J., Rault, A., Testud, J. L., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, *14*, 413–428.

Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P., et al. (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, *31*, 1691–1724.