Proceedings of the 39th IEEE
Conference on Decision and Control
Sydney, Australia • December, 2000

The benefits of implicit modelling for predictive control

J.A. Rossiter
Department of Mathematical Sciences,
Loughborough University
Leicestershire.
LE11 3TU, UK
Email: J.A.Rossiter@lboro.ac.uk
Tel: 44 (1509) 222860

B. Kouvaritakis L. Huaccho Huatuco
Dept. Engineering Science
Parks Road
Oxford.
OX1 3PJ, UK
Email: Basil.Kouvaritakis@eng.ox.ac.uk
Tel: 44 (1865) 273105

Abstract

Traditionally, predictive control is understood to rely on accurate predictions in order to give good control. Here it is shown that in fact one can bypass the prediction model altogther and go straight to an implicit model which represents the parameters of the objective function to be minimised. This is demonstrated to have significant advantages.

1 Introduction

It is well known that the success of predictive control (MPC) is closely related to the quality of the model. That is, unless one is able to find an accurate model of a process, it is unlikely that the associated predictive controller will give good performance. This fact has encouraged many authors to go beyond conventional linear control. For instance many authors are prepared to consider physical nonlinear models and neural networks (to name just two) in a bid to get ever better modelling accuracy. This is despite the fact that such models can significantly increase the complexity of the resulting MPC as well as introducing a large on-line computational burden. In this paper the intent is to look at the modelling requirement of MPC more closely than indicated in most papers and hence to propose a modelling strategy directly linked to the ultimate objective of control design. The focus will be on linear models.

During the main development of popular MPC strategies (e.g. GPC [2], DMC [1]) it was common practice to use one-step ahead prediction models, that is difference equations which can be used to predict the next value of the output given knowledge of past data and the current input. These models were used recursively (e.g. [7]) to estimate predictions for the output over a future horizon, usually referred to as the output horizon. It was realised later on [10], [11] that such models were usually identified using algorithms which focused on minimising solely the error in the one-step ahead predictions and hence may not be ideal candidates for computing predictions over longer horizons. It was proposed that the identification data should be suitably prefiltered to place more emphasis on those frequencies associated to longer range prediction. In essence these filters produce a trade-off between the accuracy of short range and long range prediction. However, computation of the optimal filter was not straightforward, even for the SISO (Single-input-single-output) case and no algorithm was given to cater for the MIMO (Multi-input-multi-output) case.

A better solution was recently considered independently by two groups ([12], [8]); the former of these papers introduces the concept without much comparison whereas the latter attempts to give a comprehensive comparision with earlier approaches. In fact the concept had already been utilised in the early eighties [6], [4] but its significance had not been recognised, perhaps because the focus of the papers in which the concept was used was adaptive control and global stability. The idea was to identify directly independent models for each prediction horizon. This avoids the need for filtering to emphasise different frequency ranges and allows the objective of the identification to be precisely stated, e.g. find a model that best estimates the output predictions n samples into the future. In the case of an exact linear model and no uncertainty in the data, independently identified models would lead to predictions that match those derived from a recursion on the one-step ahead model, however for the more general case of uncertain (noisy) signals and mild non-linearity, this is not the case. Here we label this approach the multi-model approach. The multi-model approach is always

0-7803-6638-7/00$10.00 © 2000 IEEE

able to outperform a prediction based on a recursion as one has far more parameters available and this allows a more precisely stated objective. We should also note at this point that there are strong parallels with sub-space identification. The similarities are evident by a comparison of papers such as [3, 5] with [8] and are discussed in more detail in a full paper [9].

The question that remains to be considered is, *to what use does MPC put the model and does this have repercussions on what is the best model for MPC design ?* The multi-model approach improved significantly on the one-step model with regard to this because one could argue the following. MPC is based on system predictions, therefore the best model is the one with the smallest prediction errors over the whole horizon. However, this is not the complete story. In MPC the prediction errors are used to formulate an objective function, which function is minimised to compute the new control action. It could be argued therefore that in fact the true goal of the modelling phase is to estimate the parameters of the objective function, in which case the multi-model only achieves this indirectly and not directly. In this paper it will be shown how the objective function can be identified directly and moreover that this improves on the multi-model approach as well as of course it improves on the conventional approach using recursive prediction. The proposed approach will be referred to as implicit modelling.

The paper is structured as follows. Section 2 contains background on the role of the model in MPC and an overview of the multi-model approach. Section 3 gives algorithms for computing an improved multi-model and the proposed implicit model. Section 4 contains examples and section 5 draws conclusions.

2 Background

2.1 Multi-models

In this paper, the intent is to consider MIMO and SISO systems. State-space models do not lend themselves to the multi-model approach [8] and hence the model type used will be MFD (matrix fraction description) e.g.

$$D(z)\mathbf{y}(z) = N(z)\mathbf{u}(z) \qquad (1)$$

where $N(z)$, $D(z)$ are matrix polynomials in the delay operator containing model parameters and $\mathbf{u}, \mathbf{y}$ are the system input/output respectively. For convenience here the superscript $(.)^{[n]}$ is used to denote model parameters for n-step ahead prediction.

Remark 2.1 *MPC usually takes an incremental model, that is A MODEL based on input increments $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$. From hereon it is assumed that the models above refer to such incremental models, that is the input to the identification algorithms will be $\Delta\mathbf{u}, \mathbf{y}$.*

Let the 1-step ahead prediction model of appropriate order be

$$\begin{aligned}\mathbf{y}_{k+1|k} \;=\;& N_1^{[1]}\Delta\mathbf{u}_k + N_2^{[1]}\Delta\mathbf{u}_{k-1} + \ldots + N_l^{[1]}\Delta\mathbf{u}_{k-l} \\ & - D_1^{[1]}\mathbf{y}_k - D_2^{[1]}\mathbf{y}_{k-1} - \ldots - D_m^{[1]}\mathbf{y}_{k-m}\end{aligned} \qquad (2)$$

where, at sampling instant k, the system has inputs and outputs $\mathbf{u}_k, \mathbf{y}_k$ respectively. Then the corresponding n-step ahead prediction model (e.g. see [2], [7] for standard recursions) is given as

$$\begin{aligned}\mathbf{y}_{k+n|k} \;=\;& M_{n-1}^{[n]}\Delta\mathbf{u}_{k+1|k+n-2} + \ldots \\ & + M_1^{[n]}\Delta\mathbf{u}_{k+1|k} + N_1^{[n]}\Delta\mathbf{u}_k \\ & + N_2^{[n]}\Delta\mathbf{u}_{k-1} + \ldots + N_l^{[n]}\Delta\mathbf{u}_{k-l} \\ & - D_1^{[n]}\mathbf{y}_k - D_2^{[n]}\mathbf{y}_{k-1} - \ldots - D_m^{[n]}\mathbf{y}_{k-m}\end{aligned} \qquad (3)$$

A general prediction model can be given as

$$\mathbf{Y} = P_1\Delta\mathbf{U}_{fut} + P_2\Delta\mathbf{U}_{past} + P_3\mathbf{Y}_{past} \qquad (4)$$

where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{k+1|k} \\ \mathbf{y}_{k+2|k} \\ \vdots \\ \mathbf{y}_{k+n_y|k} \end{bmatrix}; \quad \Delta\mathbf{U}_{fut} = \begin{bmatrix} \Delta\mathbf{u}_k \\ \Delta\mathbf{u}_{k+1|k} \\ \vdots \\ \Delta\mathbf{u}_{k+n_u-1|k} \end{bmatrix}; \qquad (5)$$

$$\Delta\mathbf{U}_{past} = \begin{bmatrix} \Delta\mathbf{u}_{k-1} \\ \Delta\mathbf{u}_{k-2} \\ \vdots \\ \Delta\mathbf{u}_{k-l} \end{bmatrix}; \quad \mathbf{Y}_{past} = \begin{bmatrix} \mathbf{y}_k \\ \mathbf{y}_{k-1} \\ \vdots \\ \mathbf{y}_{k-m} \end{bmatrix} \qquad (6)$$

$$P_1 = \begin{bmatrix} N_1^{[1]} & 0 & 0 & \ldots \\ N_1^{[2]} & M_1^{[2]} & 0 & \ldots \\ N_1^{[3]} & M_2^{[3]} & M_1^{[3]} & \ldots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix};$$

$$P_2 = \begin{bmatrix} N_2^{[1]} & \ldots & N_l^{[1]} \\ N_2^{[2]} & \ldots & N_l^{[2]} \\ N_2^{[3]} & \ldots & N_l^{[3]} \\ \vdots & \vdots & \vdots \end{bmatrix}; \quad P_3 = \begin{bmatrix} D_1^{[1]} & \ldots & D_m^{[1]} \\ D_1^{[2]} & \ldots & D_m^{[2]} \\ D_1^{[3]} & \ldots & D_m^{[3]} \\ \vdots & \vdots & \vdots \end{bmatrix} \qquad (7)$$

Remark 2.2 *MFD descriptions can have pole/zero cancellations which are problematic and the identified denominator can have unstable roots (even for a stable* *process). In [8] however, a separate model is identified for each prediction horizon. Hence any cancelling pole/zero pairs will appear in the model for just one prediction horizon and not be in all the models. As a conseqeunce such pairs will not appear in the controller and so cannot cause instability. The advantage of such an observation is that over parameterisation is no longer risky wheras wih conventional recusive prediction cancelling unstable pole/zero pairs (often caused by over parameterisation) cannot be allowed.*

2.2 MPC performance objective

For convenience of illustration, a conventional MPC regulation[1] objective function is taken here, for instance that popularised in GPC [2]:

$$\begin{aligned} J_k &= \sum_{i=1}^{n_y} \mathbf{y}_{k+i|k}^T \mathbf{y}_{k+i|k} \\ &\quad +\lambda \sum_{i=0}^{n_u-1} \Delta \mathbf{u}_{k+i|k}^T \Delta \mathbf{u}_{k+i|k} \end{aligned} \tag{8}$$

where n_u, n_y are the input and output prediction horizons.

Objective: In order for an MPC control law to propose sensible control actions, the cost function J_k must be known accuractly. Hence the modelling objective should be to identify J_k rather than for instance the one-step ahead prediction errors or even those of the prediction model (4).

Remark 2.3 *The part of the objective $\sum_{i=0}^{n_u-1} \Delta \mathbf{u}_{k+i|k}^T \Delta \mathbf{u}_{k+i|k}$ can be ignored in the following as this value is known exactly.*

The cost J_k (minus the control terms) comprises the sum of prediction errors over each loop and over each horizon and hence, for a system with m outputs (e.g. $\mathbf{y}_k^T = [y_k^{(1)}, \ldots, y_k^{(m)}]$) can be expanded as

$$J_k = \sum_{j=1}^{m} \sum_{i=1}^{n_y} (y_{k+i|k}^{(j)})^T y_{k+i|k}^{(j)} \tag{9}$$

From predictions (4), let

$$y_{k+i|k}^{(j)} = \mathbf{p}^{[j,k]} \mathbf{Z}_k; \quad \mathbf{Z}_k = [\Delta \mathbf{U}^T, \Delta \mathbf{U}_{past}^T, \mathbf{Y}_{past}^T]^T \tag{10}$$

where $\mathbf{p}^{[j,i]}$ comprises the relevant rows of P_1, P_2, P_3 respectively stacked together (j the loop and i the horizon). Substituting (10) into (9) gives

$$J_k = \sum_{j=1}^{m} \sum_{i=1}^{n_y} \mathbf{Z}_k^T S^{[j,i]} \mathbf{Z}_k; \quad S^{[j,i]} = (\mathbf{p}^{[j,i]})^T \mathbf{p}^{[j,i]} \tag{11}$$

The identification objective can now be summarised as the determination of either the vectors $\mathbf{p}^{[j,k]}$ or the symetric matrices $S^{[j,i]}$ which best represent the prediction cost.

3 Modelling for MPC cost functions

An identification strategy that has synergy with the MPC objectives is one which gives accurate parameter estimates for either $\mathbf{p}^{[j,i]}$ or $S^{[j,i]}$. The former of these strategies is summarised in the earlier works on multi-model identification, however there the focus was on minimising prediction errors directly and not errors in the cost function J_k. Here an algorithm is developed for identifying the prediction models $\mathbf{p}^{[j,k]}$ which minimise errors in the cost J_k directly. If however one is prepared to loosen the structure of $S^{[j,k]}$ so that is merely required to be symetric, then one has more d.o.f. for minimising errors in J_k than are available with $\mathbf{p}^{[j,i]}$. An algorithm for identifying the matrices $S^{[j,i]}$ direct is also proposed.

3.1 Identifying multi-models to minimise errors in J_k

The aim is to identify the vector $\mathbf{p}^{[j,k]}$ which minimises the following cost

$$\sum_{k=1}^{r} [V_k^{[j,i]} - \mathbf{Z}_k^T (\mathbf{p}^{[j,i]})^T \mathbf{p}^{[j,i]} \mathbf{Z}_k]^2; \quad V_k^{[j,i]} = (y_{k+i|k}^{(j)})^T y_{k+i|k}^{(j)} \tag{12}$$

where r is the number of pieces of data available for identification. For convenience hereafter, let $\mathbf{p} = \mathbf{p} = \mathbf{p}^{[j,i]}$ and $V_k = V_k^{[j,i]}$ where it is understood that the computations refer to a given loop and horizon. As $\mathbf{p}$ is a vector, efficient algorithms can be used to minimise (12) even though it is quartic in $\mathbf{p}$. It is noted once again that we do not want to solve $\min \sum_k \|(y_{k+i|k}^{(j)} - \mathbf{p}\mathbf{Z}_k\|_2^2$, as this minimises prediction errors, not the errors in the cost function itself. The minimisations required (for each loop and each prediction horizon) are of the form

$$\min_{\mathbf{p}} \quad W = \sum_k (V_k - \mathbf{Z}_k^T \mathbf{p} \mathbf{p}^T \mathbf{Z}_k)^2 \tag{13}$$

Now, noting that $\mathbf{p}$ is a vector, W has a well defined derivative, that is

$$\frac{dW}{d\mathbf{p}} = \sum_k 4(\mathbf{p}^T \mathbf{Z}_k)^3 \mathbf{Z}_k - 4V_k (\mathbf{p}^T \mathbf{Z}_k) \mathbf{Z}_k \tag{14}$$

Lemma 3.1 *A simple iterative gradient search can be used to locate the optimal* $\mathbf{p}$. *For any iterate one can*

[1] The treatment of cases inclusion of tracking problems and more involved weighting matrices is straightforward but there is an increase in notational complexity which reduces clarity.

find the point of minimum cost by solving for the least positive root of a cubic.

Proof: For an initial guess $\mathbf{p} = \mathbf{p}^0$ and $\frac{dW}{d\mathbf{p}} = \mathbf{g}$ let the search vector be $\mathbf{p} = \mathbf{p}^0 - \mu\mathbf{g}$. Then

$$W = \sum_k ([\mathbf{p}^0 - \mu\mathbf{g}]^T\mathbf{Z}_k)^4 - 2V_k([\mathbf{p}^0 - \mu\mathbf{g}]^T\mathbf{Z}_k)^2 \quad (15)$$

Minimising wrt to μ gives

$$0 = \sum_k -(4([\mathbf{p}^0 - \mu\mathbf{g}]^T\mathbf{Z}_k)^3 - 4V_k([\mathbf{p}^0 - \mu\mathbf{g}]^T\mathbf{Z}_k))\mathbf{g}^T\mathbf{Z}_k \quad (16)$$

Expanding this out gives a simple cubic in μ.

$$0 = \sum_k [(a_k + \mu b_k)^3 - (c_k + \mu d_k)]b_k \quad (17)$$

where $a_k = [\mathbf{p}^0]^T\mathbf{Z}_k$, $b_k = -\mathbf{g}^T\mathbf{Z}_k$, $c_k = V_k([\mathbf{p}^0]^T\mathbf{Z}_k))$, $d_k = -V_k\mathbf{g}^T\mathbf{Z}_k$. □

Clearly as (17) is a simple cubic in μ, it can be solved very efficiently.

Remark 3.1 *We have deliberately avoided many technical details which do not aid the understanding of the philosophy in this short paper. However the reader should note that the vector* $\mathbf{Z}$ *will vary with prediction horizon as* $\mathbf{U}_{fut}$ *(eqn. 5) varies with prediction horizon.*

3.2 Identifying J_k directly - implicit models

It maybe that constraining matrix $S^{[j,i]}$ (eqn. 11) to have a structure PP^T does not allow one to model the cost function as accurately as possible, especially where the model (4) does not represent reality exactly. Hence it maybe advantageous to identify the elements of $S^{[j,k]}$ directly with the only constraint that the matrix be symmetric. This section outlines how such a procedure can be set up. As before, to avoid cumbersome notation the following development will use the notation S, V_k without the arguments $[j,i]$ which are implicit.

The optimisation implied for each loop and horizon is the following:

$$\min_S \quad W = \sum_k (V_k - \mathbf{Z}_k^T S \mathbf{Z}_k)^2 \quad (18)$$

Let the elements of $\mathbf{Z}$ be z_i, $i = 1, 2, ..., h$ and the elements of S be s_{ij}. Then it is easy to rearrange minimisation (18) into the following form

$$\begin{array}{c} \min_\theta \quad W = \sum_k (V_k - \mathbf{m}_k^T\theta)^2 \\ \mathbf{m} = [z_1^2, 2z_1z_2, ..., 2z_1z_h, z_2^2, ..., 2z_2z_h, z_3^2, ..., 2z_{h-1}z_h, z_h^2] \\ \theta = [s_{11}, s_{12}, ..., s_{1h}, s_{22}, ..., s_{2h}, s_33, ..., s_{h-1,h}, s_{hh}] \end{array} \quad (19)$$

Minimisation (19) is a standard linear least squares problem in the parameters of S and hence can be solved easily.

Remark 3.2 *The difficulty with optimisation (19) is the rapid growth in the dimension of the unknown vector* θ *for MIMO problems and high prediction horizons. As this optimisation is offline this need not be a major issue and furthermore recursive least squares algorithms offer some computational advantages.*

4 Examples

In this section some examples are used to demonstrate the benefits of the proposed implicit model. The values $V^{[j,k]}$ (12) are computed and contrasted with the corresponding estimates to give the approximation errors W, e.g. eqn.(13). Four values are computed and plotted using the following notation:

Model	Notation	Graph
Recursive (e.g. conventional [2])	W_r	solid line
Multi-model (see [8])	W_m	asterix
Multi-model (section 3.1)	W_p	circles
Implicit model (section 3.2)	W_s	dashed

Table 1. Notation for errors of eqn.(13)

4.1 Example 1

This is a SISO example. The true model is

$$\begin{aligned} D(z) &= 1 - 1.2241z^{-1} + 0.8721z^{-2} \\ &\quad -0.6223z^{-3} + 0.1357z^{-4} \\ N(z) &= -0.7643z^{-1} + 0.3059z^{-2} + 0.0514z^{-3} \end{aligned} \quad (20)$$

and this model is estimated using a model where P_3 has 4 columns and P_2 has one column, that is it is under-parameterised by one order in both the numerator and denominator. The input is selected as a suitable known random signal whereas the output measurement is disturbed by an unknown random signal. 4000 samples of data are taken; this is referred to as identification data. In order to assess the accuracy of the models, a separate set of 4000 samples of data is also available - referred to as test data. Clearly for the identification data, one must find that $W_s \leq W_p \leq W_m \leq W_r$ and this transpires to be the case (Figure 1a). However, what is of more interest is that this is also true for the test data (Figure 1b and Table 3). It is interesting that the conventional (popular) one-step ahead model and recursion gives relatively poor results.

Horizon	1	3	6	10	20
W_s	1.3181	4.1448	5.4439	6.2240	6.0849
W_p	1.3328	4.3470	6.0135	7.2164	7.8244
W_m	1.3326	4.3557	6.0447	7.2586	7.9061
W_r	1.3326	4.3839	6.1473	7.4513	8.4482

Table 2. Values of W for different horizons for test data

4.2 Example 2

Example 2 is based on the MIMO SIMULINK model [13] of a boiler turbine unit. This model has significant non-linearities but can be adequately controlled by a linear control law. The system is very high order, but here the model order selected gave 6 columns for P_2 and 10 columns to P_3 which is equivalent to a 4th order MFD as the system has 2 inputs and 2 outputs. Conventional models and recursions failed totally to capture the predictions of this model (see W_r in table 2 and [8]) and so multi models or implicit models were essential to the succesful design of MPC. Again, as for example 1 it is clear that $W_s \leq W_p \leq W_m \leq W_r$ for the identification data, so only the test data is tabulated (Table 3) but both sets are illustrated in Figure 2a-d. Once more the implicit model has significantly outperformed the other modelling approaches.

Loop 1 (scaled by 10^{-3})					
Horizon	1	3	6	10	15
W_s	0.2909	0.2820	0.2557	0.2513	0.3444
W_p	0.4157	0.7947	1.7378	3.6863	7.6619
W_m	0.4177	0.7982	1.7409	3.6944	7.6751
W_r	0.4177	4.50	52.10	224.36	458.71
Loop 2 (scaled by 10^{-6})					
W_s	3.2958	3.2439	3.3728	3.9566	4.3733
W_p	4.3780	4.5138	4.5416	4.6014	4.6403
W_m	5.7460	5.7167	5.2592	5.0544	4.9658
W_r	5.746	21.108	35.806	60.691	59.937

Table 3. Values of W for different horizons for test data

5 Conclusion

A novel approach to modelling for MPC has been presented. The technique is based on snyergy between the identification procedure and the MPC control design. The consequent model is in fact an implicit model in that it is not allied to the normal one step ahead difference equations or state-space models, but rather summarises the effect of past and future data on the performance index to be minimised. The improvements in modelling accuracy afforded by the implicit model are illustrated by way of example.

References

[1] C.R. Cutler and B.L. Ramaker, 1980, Dynamic matrix control - a computer control algorithm, Proc. ACC, San Fransisco

[2] D.W. Clarke, C. Mohtadi and P.S. Tuffs, 1987, Generalised predictive control, Parts 1 and 2, Automatica, 23, 137-160.

[3] W. Favoreel, B. De Moor, B. and M. Gevers, SPC: Subspace predictive control, 1999,, *Proc. IFAC World Congress*, Beijing, pp235-240

[4] C. Greco, G.Menga, E. Mosca and G. Zappa, Performance improvements of self-tuning controllers by multistep horizons: The MUSMAR approach, 1984, Autoamtica, 20, pp681-699

[5] H. Hjalmarsson, M. Gevers and S. Gunnarsson, Iterative feedback tuning: theory and applications, *IEEE control systems magazine*, 1998, 16, pp26-41

[6] E. Mosca, G.Zappa and J. M. Lemos, Robustness of a multipredictor adaptive regulator: Musmar, 1989, Automatica, 25, pp521-529

[7] J.A. Rossiter, Notes on multi-step ahead prediction based on the principle of concatenation, Proc. IMechE, 207, pp261-264

[8] J.A. Rossiter, L. Yao and B. Kouvaritakis, Identification of prediction models for a non-lienar power generation model, IEE Control 2000

[9] J.A. Rossiter and B. Kouvaritakis, Modelling and implicit modelling for predictive control, submitted to IJC

[10] D.S. Shook, C. Mohtahdi and S.L. Shah, 1991, Identification for long range predictive control, Proc. IEE Pt-D, 138, 1, pp75-84

[11] D.S. Shook, C. Mohtahdi and S.L. Shah, 1992, A control relevant identification strategy for GPC, IEEE Trans on AC, 37, 7, p975-980

[12] D. Liu, S.L. Shah and D.G. Fisher, 1999, Multiple prediction models for long range predictive control, Proc. IFAC World Congress, 1999

[13] Simulink model kindly supplied by Prismtech Ld., Gateshead, UK.

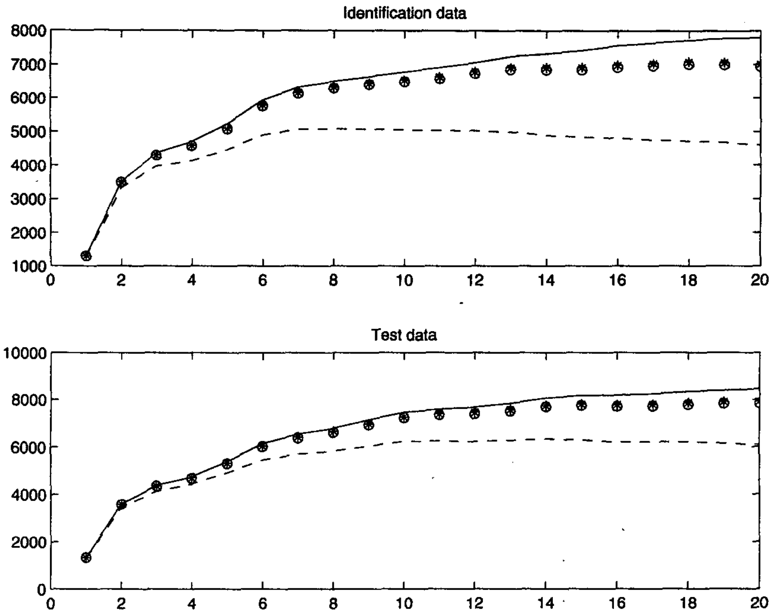

Figure 1. Plot of errors W (y-axis) vs prediction horizon (x-axis)

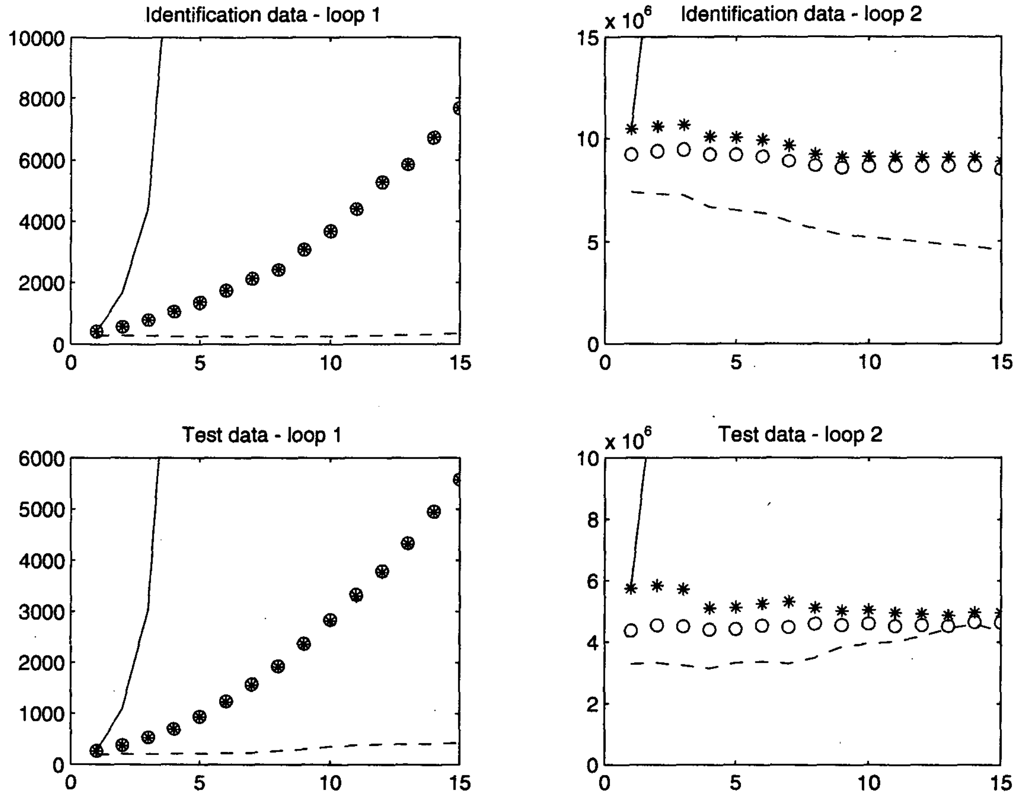

Figure 2. Plot of errors W (y-axis) vs prediction horizon (x-axis)