ELSEVIER

Signal Processing 52 (1996) 209–215

SIGNAL PROCESSING

Subspace identification from closed loop data

Lennart Ljung*, Tomas McKelvey

Department of Electrical Engineering, Linköping University, Linköping, Sweden

Received 3 April 1995; revised 27 February 1996

Abstract

The so-called subspace methods for direct identification of linear models in state space form have drawn considerable interest recently. They have been found to work well in many cases but have one drawback – they do not yield consistent estimates for data collected under output feedback. The present paper points to the reasons for this. We stress how the basic idea is to focus on the estimation of the state-variable candidates – the k-step ahead output predictors. By recomputing these from a 'non-parametric' (or, rather, high order ARX) one-step ahead predictor model, closed loop data can be handled.

Zusammenfassung

In letzter Zeit haben sogenannte Unterraumtechniken zur direkten Identifikationen linearer Modelle im Zustandsraum betrachtliches Interesse auf sich gezogen. Man hat festgestellt, daß sie in vielen Fällen gut funktionieren, aber einen Nachteil aufweisen – sie liefern keine konsistenten Schätzungen für Daten, die bei Ausgangsrückführung gewonnen werden. Im folgenden Beitrag zeigen wir die Gründe hierfür auf. Wir betonen, daß die Grundidee darin besteht, sich auf die Schätzung der Zustandsvariablen zu konzentrieren – durch die Prädiktion von k Schritten voraus. Indem man diese Werte aus einem 'nichtparametrischen' (besser: einem hochgardigen ARX-) Einschritt-Prädiktor neu berechnet, kann man Daten aus rückgekoppelten Schleifen handhaben.

Résumé

Les méthodes par sous-espace pour l'identification directe de modèles linéaires dans l'espace d'état ont suscité beaucoup d'intérêt récemment. Elles fonctionnent bien dans beaucoup de cas, mais ont un défaut – elles ne donnent pas d'estimées consistantes pour des données recueillies lorsqui'il y a retour de la sortie vers l'entrée. Cette contribution montre la raison de ce fait. Nous soulignons que l'idée de base est de se concentrer sur l'estimation des candidats pour les variables d'état – les prédicteurs de sortie pour k pas d'avance. En recalculant ces derniers à partir d'un modéle de prédiction pour un pas d'avance 'non paramétrique' (ou, plus exactement, ARX d'ordre élevé), les données en boucle fermée peuvent être traitées.

Keywords: Subspace methods, State-space modeling, Closed-loop identification

* Corresponding author. E-mail: ljung@isy.liu.se.

PII S0165-1684(96)00054-0

1. Introduction

We consider state space models of the standard type

$$x(t+1) = Ax(t) + Bu(t) + w(t),$$
$$y(t) = Cx(t) + Du(t) + e(t), \tag{1}$$

or in innovation form

$$x(t+1) = Ax(t) + Bu(t) + K\varepsilon(t),$$
$$y(t) = Cx(t) + Du(t) + \varepsilon(t). \tag{2}$$

The problem is to estimate the matrices A, B, C, D and possibly K from input–output data $\{u(t), y(t),\ t = 1, 2, \ldots, N\}$ without any prior knowledge of the structure. Several techniques for this have been developed. The 'classical' maximum likelihood (ML) approach would imply that the matrices are parameterized in some canonical way, and then these parameters are being estimated, see e.g., [4]. This has the advantage of being statistically efficient but the disadvantage that an iterative search – that might end up in a local minimum – has to be employed, and that the canonical parameterization might be numerically ill-conditioned.

So-called subspace methods [9–12] form an interesting alternative to the ML approach. The idea behind these methods can be explained as first estimating the state vector $x(t)$, and then finding the state space matrices by a linear least squares procedure. The state vector in innovations type parameterizations like (2) is always found as linear combinations of k-step ahead output predictors. See [1, 5] and Appendix 4.A in [4].

In the subspace methods these k-step ahead predictors are found by the so-called oblique projections [10]. While this constitutes algorithms that are very efficient, numerically, it has the drawback that closed loop data cannot be handled in this way.

The problem of identifying a system operating in closed loop has been analyzed extensively in the past, with the general conclusion that a direct use of the prediction error methods could be seen as the basic approach [3]. This however requires a complete noise model. Some recent results describe alternative approaches which convert the closed loop problem into an open loop problem [6, 7].

The purpose of this contribution is to show why the existing subspace methods fail for the case of closed loop data and show how to circumvent this drawback. We do that by stressing the outlined perspective on subspace methods that gives a direct clue how to move on to closed loop data.

2. The basic steps of subspace identification

As clearly pointed out by [8] the subspace approach focuses on the state vector: first, find the state vector $x(t)$ from data, then the state space matrices A, B, C, D and K can be defined from linear regressions. There are two possibilities:

1a. Treat, for given $\{x(t)\}$ the equation

$$y(t) = Cx(t) + Du(t) + e(t)$$

as a linear regression, giving C and D by the least squares method, and leaving $e(t)$ as the residuals.

1b. Then for given $x(t), u(t)$ and $e(t)$ (computed as above) estimate A, B and K from the linear regression

$$x(t+1) = Ax(t) + Bu(t) + Ke(t) + \hat{e}(t)$$

with $\hat{e}$ as the residuals.

An alternative is

2a. Treat

$$\begin{bmatrix} x(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + \begin{bmatrix} w(t) \\ e(t) \end{bmatrix}$$

for given x, y and u as a linear regression, with residuals $v^{\mathrm{T}}(t) = [w^{\mathrm{T}}(t), e^{\mathrm{T}}(t)]^{\mathrm{T}}$

2b. The Kalman gain K can then, if desired, be computed from A, C and the covariance matrix of $v(t)$.

So, once the state sequence $x(t)$ has been determined, we may easily find the corresponding state space model. Therefore, let us discuss how to find these states. The literature on state space identification has very elegantly shown how the state can be estimated directly from the data by certain projections. We shall not use that language here but describe the process (equivalently) along the classical realization path, as developed by [1, 2, 5]. (See Appendix 4.A in [4] for an account.)

The essence is as follows. Let $\{e(t)\}$ be the innovations and let the impulse response matrix from

$$\xi(t) = \begin{bmatrix} u(t) \\ e(t) \end{bmatrix} \tag{3}$$

to $y(t)$ be H_k so that

$$y(t) = \sum_{k=0}^{\infty} H_k \xi(t-k). \tag{4}$$

H_k is thus a $p \times (p+m)$ matrix where p is the number of outputs and m the number of inputs. Define

$$\hat{y}(t+j|t) = \sum_{k=j}^{\infty} H_k \xi(t+j-k). \tag{5}$$

If $\{\xi(t)\}$ were white noise, this would be the correct j step ahead predictor for $y(t)$, hence the notation. Introduce the vector of j-step ahead predictors:

$$Y_m(t) = \begin{bmatrix} \hat{y}(t+1|t) \\ \vdots \\ \hat{y}(t+m|t) \end{bmatrix}. \tag{6}$$

The basic points of the realization theory are

- The system (4) has a minimal state space realization of order n if and only if

$$\operatorname{rank}\{Y_m(t)\} = n \quad \text{for } m \geqslant n. \tag{7}$$

(The rank refers to the vector sequence $Y_m(t)$, $t = 1,\ldots.$)

- All 'Kalman states', i.e., state vectors that can be reconstructed from past inputs and outputs are obtained by picking a basis from $\{Y_m(t)\}$.

$$x(t) = LY_m(t) \tag{8}$$

for some matrix L.

Now the identification strategy is clear:

1. Find $\hat{y}(t+j|t)$ as defined by (5), for $j = 1,\ldots,m$.
2. Form $Y_m(t)$ for $t = 1,\ldots,N$ as in (6) and estimate its rank n.
3. Select L in (8) to obtain a well conditioned basis and $x(t) = LY_m(t)$, $t = 1,\ldots,N$.
4. Find the matrices A,B,C,D and K in (7) by either of the procedures 1 or 2 described in the beginning of this section.

This can be seen as the basic recipe for direct state space identification. Depending on the details in the different types, we arrive at various implementations of the recipe. The most crucial step is number 1. We shall discuss how to achieve this in the next section.

3. Estimating the j-step ahead predictor

Let us rewrite (4) as

$$y(t+j) = \sum_{k=0}^{\infty} H_k^e e(t+j-k) + \sum_{k=0}^{\infty} H_k^u u(t+j-k).$$

By the innovations definition, there is a one-to-one correspondence between $e(s)$ and $y(s)$, so past $e(s)$ can be replaced by $y(s)$ at no loss of information. We do so for $s \leqslant t$:

$$y(t+j) = \sum_{k=0}^{j-1} H_k^e e(t+j-k) + \sum_{k=0}^{\infty} \tilde{H}_{j,k}^{y} y(t-k) + \sum_{k=0}^{j-1} H_k^u u(t+j-k) + \sum_{k=0}^{\infty} H_{k+j}^u u(t-k). \tag{9}$$

The four sums have different interpretations:

- The first one will be denoted by $\varepsilon_j(t+j)$ and is made up from innovations between $t+1$ and $t+j$.
- The second and fourth ones are the past input–output data from time t and beyond. We will collect these data in the (infinite-dimensional) vector $\varphi^{(j)}(t)$. The corresponding weights $\tilde{H}^y$ and H^u are collected in the (infinite-dimensional) parameter vector $\theta_0^{(j)}$. The second and fourth terms of (9) can thus be written as $[\theta_0^{(j)}]^{\mathrm{T}} \varphi^{(j)}(t)$.
- The third sum consists of inputs between time $t+j$ and $t+1$. Analogously to the above expression we shall write this as $[\theta_{u,0}^{(j)}]^{\mathrm{T}} \varphi_u^{(j)}(t+j)$.

The expression (9) can thus be rewritten as

$$y(t+j) = \varepsilon_j(t+j) + \begin{bmatrix} \theta_0^{(j)} \\ \theta_{u,0}^{(j)} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \varphi^{(j)}(t) \\ \varphi_u^{(j)}(t+j) \end{bmatrix} \triangleq \varepsilon_j(t+j) + [\bar{\theta}_0^{(j)}]^{\mathrm{T}} \bar{\varphi}^{(j)}(t+j). \tag{10}$$

The sought predictor is then

$$\hat{y}(t+j|t) = [\theta_0^{(j)}]^{\mathrm{T}} \varphi^{(j)}(t). \tag{11}$$

How to find $\theta_0^{(j)}$ from (10)? The natural way will be to estimate it using the least squares method. This estimate will converge to the theoretical estimate (the

projection)

$$\hat{\bar{\theta}}^{(j)} = [E\bar{\varphi}^{(j)}(t)\bar{\varphi}^{(j)}(t)^{\mathrm{T}}]^{-1}E\bar{\varphi}^{(j)}(t)y(t+j)$$

$$= \bar{\theta}_0^{(j)} + [E\bar{\varphi}^{(j)}(t)\bar{\varphi}^{(j)}(t)^{\mathrm{T}}]^{-1} \times \begin{bmatrix} 0 \\ E\varphi_u^{(j)}(t+j)\varepsilon_j(t+j) \end{bmatrix}$$

$$= \bar{\theta}_0^{(j)} + \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} 0 \\ E\varphi_u^{(j)}(t+j)\varepsilon_j(t+j) \end{bmatrix}. \tag{12}$$

In the second last expression we used the fact that $\varphi^{(j)}(t)$ and $\varepsilon_j(t+j)$ are independent.

The upper part of this expression, i.e.

$$\hat{\theta}^{(j)} = [I\ 0]\hat{\bar{\theta}}^{(j)} \tag{13}$$

corresponds to the terms associated with inputs and outputs beyond time t. Inserting this estimate in (11) gives an expression for the prediction just in terms of inputs and outputs, i.e. $\bar{\varphi}^{(j)}$ and $y(t+j)$. This is a projection expression, that is *oblique* (see [10]). This refers to the fact that the part $\hat{\theta}_u$, which corresponds to 'future inputs' – i.e. from the time interval $t+1$ to $t+j$ – has been thrown away from $\hat{\bar{\theta}}^{(j)}$.

In practice the expectation operator E will be replaced by sample sums, and the infinite vector $\varphi^{(j)}(t)$ will have to be truncated to contain only inputs and outputs up to $t-n_b$ and $t-n_a$, say. This is the same as estimating the coefficients of a linear regression

$$y(t+j) = \sum_{k=0}^{na} \tilde{H}_{j,k}^{y} y(t-k) + \sum_{k=0}^{j-1} H_{j,k}^{u} u(t+j-k) + \sum_{k=0}^{nb} H_{k+j}^{u} u(t-k) + \varepsilon(t+j) \tag{14}$$

(and then throwing away the part corresponding to the second sum).

Having now formed an estimate of the predictor $\hat{y}(t+j|t)$, we can stack the estimates on top of each other to form the corresponding estimate of $Y_m(t)$ in (8). This gives an expression of the type

$$\hat{Y}_m(t) = \text{projection type expression in terms of } \varphi^{(j)}(l),\ y(l+j), \text{ etc.} \tag{15}$$

This is the predictor estimate employed in one or other form in all the subspace identification variants.

Now, does this lead to a consistent estimate of $Y_m(t)$? The key to this question is the expression (12), and we can formulate the result as follows.

Lemma 1. *The outlined oblique projection procedure for estimating $Y_m(t)$ is consistent if and only if*

$$R_{12}E\varphi_u^{(j)}(t+j)\varepsilon_j(t+j) = 0, \tag{16}$$

where R_{12} is defined in (12).

Typically (16) holds only when $e(t+j),\ldots,e(t+1)$ and $u(t+j),\ldots,u(t+1)$ are independent, i.e. *when there is no feedback in the input.*

It is now clear why closed loop data will cause problems for subspace methods. If the input is generated by output feedback, the estimates of the predictors (i.e. the states) will be biased. This bias will not go away when the number of observed data tends to infinity nor when the horizons (*na* and *nb*) tend to infinity.

Are there other alternatives to estimate $\hat{y}(t+j|t)$ that do not have this problem with closed loop data? Well, if there is a delay in the system, $H_0^u = 0$, – as is typically the case for systems under feedback control – then (14) will work without problems for $j = 1$. The one step ahead predictor is then easily formed.

Now, there is a simple recursive way to compute further step ahead predictors such as (5) from the one step ahead predictor. The recursive step is a follows:

- Given that we have determined $\hat{y}(t+i|t)$, $i = 1,\ldots,j-1$.
- Then replace in the expression for $\hat{y}(t+j|t+j-1)$ all occurrences of $y(t+k)$, $k = 1,\ldots,j-1$, by $\hat{y}(t+k|t)$ and all occurrences of $u(t+k)$, $k = 1,\ldots,j-1$ by zero.

This process gives a way to determine all the predictors, not suffering from problems with feedback. It is functionally equivalent to defining an (na, nb)-order ARX-model from the data, and then computing this model's j-step ahead prediction and replacing 'future' $u(t)$ in this prediction by zero.

4. A test subspace algorithm for feedback data

4.1. A simple algorithm

The algorithm can be summarized as follows:

1. Estimate an (na, nb)-order ARX model from input and output data.
2. Use the ARX model to calculate the j-step ahead predictions $\hat{y}(t+j|t)$ from data by replacing 'future' $u(t)$ in this prediction by zero as described above.
3. Form $Y_m(t)$ as in (8) and estimate its rank n.
4. Select a state-space basis L and take $x(t) = LY_m(t)$.
5. Find A, B, C, D, K by procedure 1 in the beginning of Section 2.

We would like to stress that the outlined algorithm is not to be taken as the 'best' way of identifying systems operating in closed loop but should merely be seen as an illustration of a feasible method, as well as to illustrate that our lemma has put its finger on the cause of problems with closed loop data.

In particular, it should also be remarked that this approach lacks a considerable advantage of the N4SID algorithm [10] (and its analogs). By forming the vector Y_m explicitly, we work in a provisional basis that could be ill-conditioned, while N4SID directly forms $x(t)$ from the data.

4.2. Simulation example

We have performed a number of tests with the algorithm on simple data sets, and its performance is compatible to that of N4SID, except for closed loop data, where N4SID might fail. We illustrate this by applying these algorithms to two identification setups; one when the system is operating in open loop and the other when the system is operating under negative unity feedback. The system considered is described by

$$y(t) = \frac{0.21q^{-1} + 0.07q^{-2}}{1 - 0.6q^{-1} + 0.8q^{-2}}u(t) + \frac{1}{1 - 0.98q^{-1}}e(t),$$

where $e(t)$ is zero mean white noise with variance 4. In open loop operation, the input $u(t)$ is white noise with unit variance. In closed loop operation, the input is given by

$$u(t) = r(t) - y(t),$$

where the reference signal $r(t)$ is white noise with unit variance.

For both setups a Monte Carlo simulation is performed consisting of 50 identifications of models of order 3 using the method described in Section 4.1 and N4SID. Each data set has a length of 400.

The results are shown in Fig. 1. The top graph shows the case of open loop operation and the bottom graph closed loop operation. Both graphs show the magnitudes of the resulting transfer function error averaged over the 50 identified models. As predicted, both identification methods give comparable results for the open loop case while the new method of estimating the predictors significantly improves the performance for the case of closed loop operation. For some frequencies the average transfer function error for the N4SID method is more than eight times higher.

5. Some additional insights

The subspace algorithms involve a number of choices. We have pointed to the numbers na, nb and m in (14) and (6). These correspond to the split into past and future in the subspace algorithms. Our discussion has pointed to some insights about these numbers, that can be summarized as follows:

1. na, nb and m need not and probably should not be the same.
2. na and nb reflect the length of the tails of the true one-step ahead predictor of the system. They should be chosen to minimize the mean-square error, by balancing bias and variance in an ARX-model.
3. m is just the largest model order we wish to investigate. Once the desired order has been selected as n, it may suffice to work with Y_n.

We should also mention that in order to get consitent estimates of the predictors $\hat{y}(t + j|t)$ (and hence of the states) we need the horizons na and nb to tend to infinity. This will then also give consistent estimates of the noise properties. However, as demonstrated in, e.g., [10], the matrices A, B and C can be consistently estimated, even for finite values of these horizon parameters.

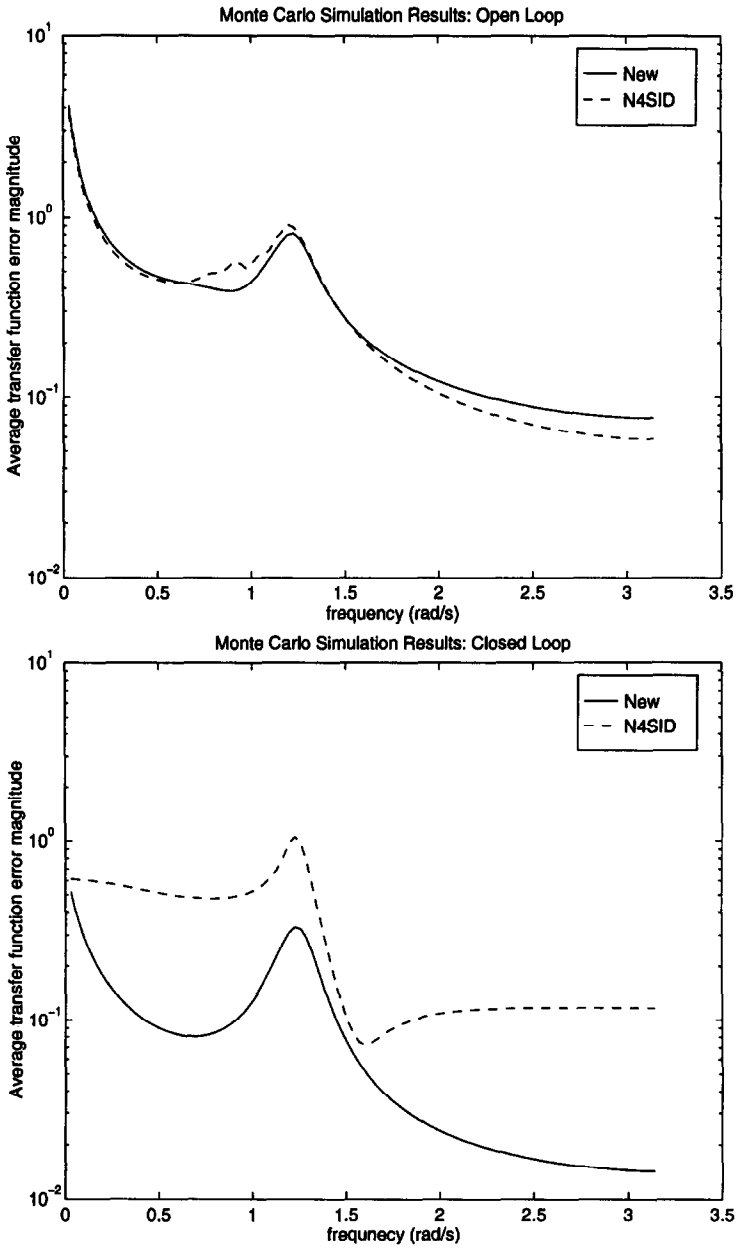

Fig. 1. Bode plots of identification error averaged over 50 Monte Carlo simulations. A third order model obtained by the algorithm given in this paper (solid line) and a third order model obtained by N4SID, a standard sub-space method (dashed line).

6. Conclusions

We have, in this contribution, stressed how the subspace identification algorithms are primarily based on estimates of the k-step ahead output predictors. This gives some insight into the essence of the algorithms, it explains why closed loop data may cause difficulties and how to avoid these difficulties.

The main contribution might be these insights. We have also given one concrete algorithm, based on our insights that can handle closed loop data. This algorithm performs about as well on a simple test case as the N4SID algorithm, but we have demonstrated its distinct advantage when it comes to handling closed loop data.

References

[1] H. Akaike, "Markovian representation of stochastic processes by canonical variables", *SIAM J. Control Optim.*, Vol. 13, 1975, pp. 162–173.

[2] P. Faurre, "Stochastic realization algorithms", in: R. Mehra and D. Lainiotis, eds., *System Identification: Advances and Case Studies*, Academic Press, New York, 1976.

[3] I. Gustavsson, L. Ljung and T. Soderström, "Identification of processes in closed loop – Identifiability and accuracy aspects, *Automatica*, Vol. 13, 1977, pp. 59–75.

[4] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[5] J. Rissanen, "Basis of invariants and canonical forms for linear dynamic systems", *Automatica*, Vol. 10, 1974, pp. 175–182.

[6] P.M.J. Van Den Hof and R.J.P. Scrama, "An indirect method for transfer function estimation from closed loop data", *Automatica*, Vol. 29, 1993, pp. 1523–1527.

[7] P.M.J. Van Den Hof and R.J.P. Scrama, "Identification and control – Closed loop issues, *Proc. 10th IFAC Symp on System Identification, Copenhagen, Denmark*, Vol. 2, 1994, pp. 1–14.

[8] P. Van Overschee, Subspace Identification, Theory – Implementation – Application, PhD thesis, Katholieke Universiteit Leuven, Kard. Mercierlaan 94, 3001 Leuven (Heverlee), Belgium, February 1995.

[9] P. Van Overschee and B. De Moor, "Subspace algorithms for the stochastic identification problem", *Automatica*, Vol. 29, No. 3, 1993, pp. 649–660.

[10] P. Van Overschee and B. De Moor, "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems", *Automatica*, Vol. 30, No. 1, 1994, pp. 75–93.

[11] M. Verhaegen, "Identification of the deterministic part of MIMO state space models given in innovations form from input–output data", *Automatica*, Vol. 30, No. 1, 1994, pp. 61–74.

[12] M. Viberg, B. Ottersten, B. Wahlberg and L. Ljung, "Performance of subspace based state-space system identification methods", *Proc. 12th World Congress Internat. Federation of Automatic Control*, Sydney, Australia, Vol. 7, 1993, pp. 369–372.