An iterative method for identification of ARX models from incomplete data

Ragnar Wallin and Alf J. Isaksson
Division of Process Control
Department of Signals, Sensors and Systems
Royal Institute of Technology (KTH)
SE–100 44 Stockholm, Sweden
ragnarw@s3.kth.se, alf@s3.kth.se

Lennart Ljung
Division of Automatic Control
Department of Electrical Engineering
Linköpings University
SE–581 83 Linköping, Sweden
ljung@isy.liu.se

Abstract

This paper describes a very simple and intuitive algorithm to estimate parameters of ARX models from incomplete data sets. An iterative scheme involving two least squares steps and a bias correction is all that is needed.

1 Introduction

Identification experiments are costly for the process industry. Time when saleable products could have been manufactured is "wasted" on experiments. Having to discard an incomplete data set collected for identification and do a new experiment is not necessarily an acceptable option. Algorithms that can take care of the missing data problem in an efficient way are therefore very welcome.

The problem with missing data has been studied extensively in statistics, but less so in engineering literature. A survey of the research in statistics is given in the book by Little and Rubin [5]. In engineering literature parameter estimation of ARX models has been studied in [4], [3], [1] and [7]. Methods in the frequency domain are studied in [6]. The reference list in [4] has further references.

A very simple iterative off-line algorithm for estimating the parameters of an ARX model is:

1. Guess the parameters.
2. Do a least squares estimate of the missing data assuming the parameters to be the true ones.
3. Do a least squares estimate of the parameters assuming the estimated data to be the correct data.
4. Iterate from 2 until convergence.

This procedure will however give a biased parameter estimate except for some special cases, as will be shown in section 4. A minor modification of the algorithm will however remove the bias.

2 Two ways of writing the ARX model

An ARX model is described by

$$y_k = -a_1 y_{k-1} - \ldots - a_{na} y_{k-na} + b_1 u_{k-nk} + \ldots + b_{nb} u_{k-nk-nb+1} + e_k, \quad (1)$$

where y_k and u_k are the output and input at time k respectively, and nk is the time delay between input and output. The term e_k is a zero mean white noise with variance λ. The number of parameters (n_a and n_b) are here assumed to be known a priori. Thus, estimation of model order - albeit a very interesting problem - is considered beyond the scope of this paper. The equation is linear in the parameters and can be written in matrix form as

$$Y = \Phi\theta + E, \quad (2)$$

where Y is a vector with the outputs of the left hand side of the equation (1), Φ is built up by the old inputs and outputs of the right hand side of equation (1), θ is a vector containing the a and b parameters and E is a vector with the noise terms. This way of writing the equation is preferred when the parameters are to be estimated.

Another nice feature of equation (1) is that it is also linear in the data and hence another matrix form is

$$\Omega Z + E = 0, \quad (3)$$

where Ω is a matrix with the a and b parameters as elements and

$$Z = \begin{bmatrix} y_N & \ldots & y_1 & u_N & \ldots & u_1 \end{bmatrix}^T.$$

This way of writing the equation is preferred when the missing data are to be estimated.

Example: The ARX model

$$y_k = -a_1 y_{k-1} + b_1 u_{k-1} + e_k,$$

with measurements from time 1 to time 3 can be written with the representation of equation (2)

$$\underbrace{\begin{bmatrix} y_3 \\ y_2 \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} -y_2 & u_2 \\ -y_1 & u_1 \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} a_1 \\ b_1 \end{bmatrix}}_{\theta} + \underbrace{\begin{bmatrix} e_3 \\ e_2 \end{bmatrix}}_{E}.$$

Notice that the rows are in the reverse order from what is common in identification literature. This choice is only one of convenience and will, of course, not affect the results. The model can also be written with the representation of equation (3)

$$\underbrace{\begin{bmatrix} -1 & -a_1 & 0 & 0 & b_1 & 0 \\ 0 & -1 & -a_1 & 0 & 0 & b_1 \end{bmatrix}}_{\Omega} \underbrace{\begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ u_3 \\ u_2 \\ u_1 \end{bmatrix}}_{Z} + \underbrace{\begin{bmatrix} e_3 \\ e_2 \end{bmatrix}}_{E} = 0.$$

□

The alternative descriptions are related in the following way

$$Y = -C_0 Z, \tag{4}$$

the columns of Φ look like

$$\Phi_k = C_k Z \qquad k = 1, 2, \ldots, na + nb \tag{5}$$

and Ω is a linear combination of the C_k matrices

$$\Omega = C_0 + \sum_{k=1}^{na} a_k C_k + \sum_{k=1}^{nb} b_k C_{na+k}, \tag{6}$$

where

$$C_k = \begin{cases} -R & k = 0 \\ -RS & k = 1 \\ \vdots & \vdots \\ -RS^{na} & k = na \\ RS^{N+nk} & k = na+1 \\ \vdots & \vdots \\ RS^{N+nb+nk-1} & k = na+nb \end{cases}$$

The matrix R picks out the first elements of a vector and the matrix S shifts the data in the data vector. The matrices look like

$$R = \begin{bmatrix} I & 0 \end{bmatrix}$$

$$S = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \ldots & \ldots & \ldots & 1 \\ 0 & \ldots & \ldots & \ldots & 0 \end{bmatrix}.$$

Define the estimation error as the estimated value minus the true value.

$$\tilde{Y} = \hat{Y} - Y \tag{7}$$

$$\tilde{\Phi} = \hat{\Phi} - \Phi \tag{8}$$

$$\tilde{Z} = \hat{Z} - Z \tag{9}$$

It follows from equations (2)–(3) and (7)–(9) that

$$\hat{Y} - \hat{\Phi} = -\Omega \hat{Z}$$

$$\tilde{Y} - \tilde{\Phi}\theta = -\Omega \tilde{Z}.$$

This result will be used frequently in the following sections.

3 Missing data estimation

If we assume that we have the true model we can estimate the missing data with the least squares method. Introduce the selection matrix Q_1 to pick out the missing data in a vector of minimal length, Z_m, and the selection matrix Q_2 to pick out the observed data in a vector of minimal length, Z_o. If a data point is missing the corresponding column in the identity matrix should be included in Q_1 and analogously for Q_2 if a data point is observed.

Example: Assume that we have input and output data from time 1 to time 3. If the output at time 3 and the input at time 2 are missing we get

$$Z_m = Q_1^T Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ u_3 \\ u_2 \\ u_1 \end{bmatrix} = \begin{bmatrix} y_3 \\ u_2 \end{bmatrix}$$

$$Z_o = Q_2^T Z = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ u_3 \\ u_2 \\ u_1 \end{bmatrix} = \begin{bmatrix} y_2 \\ y_1 \\ u_3 \\ u_1 \end{bmatrix}$$

□

Since $Q_1 Q_1^T + Q_2 Q_2^T = I$ equation (3) can be rewritten as

$$\Omega Q_1 \underbrace{Q_1^T Z}_{Z_m} + \Omega Q_2 \underbrace{Q_2^T Z}_{Z_o} + E = 0.$$

The least squares estimate of the missing data becomes

$$\hat{Z}_m = -(\Omega Q_1)^\dagger \Omega Q_2 Z_o$$

and the corresponding estimation error is

$$\tilde{Z}_m = (\Omega Q_1)^\dagger E.$$

Here we have used the notation $A^\dagger$ for the (Moore–Penrose) pseudoinverse, i.e.

$$A^\dagger = (A^T A)^{-1} A^T$$

To get the estimate of the whole Z we first premultiply $\hat{Z}_m$ by Q_1. This results in a vector with the estimated data points in their correct places and zeros where the observed data points are located. Then the observed data points are filled in by adding $Q_2 Q_2^T Z$, i.e.

$$\hat{Z} = (I - Q_1(\Omega Q_1)^\dagger \Omega) Q_2 Q_2^T Z$$

In the positions where data are observed we make no error in the estimate. Thus, the estimation error of the whole Z is $\tilde{Z}_m$ premultiplied by Q_1.

$$\tilde{Z} = Q_1(\Omega Q_1)^\dagger E$$

4 Condition for unbiased parameter estimate

Using equations (7) and (8) we can rearrange equation (2). We get

$$\hat{Y} = \hat{\Phi}\theta + \underbrace{(\tilde{Y} - \tilde{\Phi}\theta + E)}_{V}. \tag{10}$$

Doing a least squares estimate, assuming the estimated data are correct (the algorithm described in the introduction) yields

$$\theta = \underbrace{(\hat{\Phi}^T\hat{\Phi})^{-1}\hat{\Phi}^T\hat{Y}}_{\hat{\theta}} - \underbrace{(\hat{\Phi}^T\hat{\Phi})^{-1}\hat{\Phi}^T V}_{\bar{\theta}} \tag{11}$$

As a consequence, a least squares estimate of the parameters using *any* estimate of the missing data is unbiased if the following holds

$$\mathbf{E}[\hat{\Phi}^T V] = \mathbf{E}[\hat{\Phi}^T(\tilde{Y} - \tilde{\Phi}\theta + E)] = 0. \tag{12}$$

Hence, every column in $\hat{\Phi}$ must be orthogonal to the equation error. Unfortunately, this is usually not the case for the missing data estimation approach described above, as will be demonstrated next. This means that the true parameter vector is not a stationary point of the iterative method.

The criterion (12) can be divided into three terms

$$\Delta = \mathbf{E}[\hat{\Phi}^T(\tilde{Y} - \tilde{\Phi}\theta + E)] =$$
$$\mathbf{E}[\underbrace{\tilde{\Phi}^T(-\Omega\tilde{Z} + E)}_{\text{Term I}} + \underbrace{\Phi^T E}_{\text{Term II}} - \underbrace{\Phi^T\Omega\tilde{Z}}_{\text{Term III}}].$$

As we saw in section 2 the columns in Φ and $\tilde{\Phi}$ look like

$$\Phi_k = C_k Z$$

and

$$\tilde{\Phi}_k = C_k\tilde{Z} = C_k Q_1(\Omega Q_1)^\dagger E.$$

Thus, for the term I we get

$$\mathbf{E}[\tilde{\Phi}_k^T(-\Omega\tilde{Z} + E)] =$$
$$\mathbf{E}[E^T(\Omega Q_1)^{\dagger T} Q_1^T C_k^T (I - (\Omega Q_1)^{\dagger T} Q_1^T \Omega^T) E] =$$
$$\lambda \cdot \text{trace}[(I - (\Omega Q_1)^{\dagger T} Q_1^T \Omega^T)(\Omega Q_1)^{\dagger T} Q_1^T C_k^T] =$$
$$\lambda \cdot \text{trace}([(\Omega Q_1)^{\dagger T} - (\Omega Q_1)^{\dagger T}] Q_1^T C_k^T) = 0$$

and for term II

$$\mathbf{E}[\Phi^T E] = 0$$

because of the model structure.

The third term is usually not zero and is the one causing the biased parameter estimate. However, for an FIR model with the input uncorrelated with the noise it is in fact zero (regardless whether inputs or outputs are missing). This follows from the fact that Φ only contains inputs and $\tilde{Z}$ is a linear combination of elements in the noise vector E. The third term can, given the parameters, be evaluated element by element as

$$\mathbf{E}[-\Phi_k^T\Omega\tilde{Z}] = -\mathbf{E}[Z^T C_k^T \Omega Q_1(\Omega Q_1)^\dagger E] =$$
$$-\text{trace}(\Omega Q_1(\Omega Q_1)^\dagger \mathbf{E}[EZ^T] C_k^T)$$

In summary, only the third term of Δ is non zero and the elements of Δ are

$$\Delta_k = -\text{trace}(\Omega Q_1(\Omega Q_1)^\dagger \mathbf{E}[EZ^T] C_k^T).$$

Since the input is assumed to be uncorrelated with the noise, we simply have to solve a linear banded system of equations to get the elements of $\mathbf{E}[EZ^T]$.

Example: For the ARX model

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} + b_1 u_{k-1} + e_k,$$

with data from time 1 to time 8 we have to calculate the correlations $r_{ey}(\tau) = \mathbf{E}[e_{k+\tau} y_k]$ for τ = -5, -4, -3, -2, -1, 0. We have to solve the linear system of equations

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ a_1 & 1 & 0 & 0 & 0 & 0 \\ a_2 & a_1 & 1 & 0 & 0 & 0 \\ 0 & a_2 & a_1 & 1 & 0 & 0 \\ 0 & 0 & a_2 & a_1 & 1 & 0 \\ 0 & 0 & 0 & a_2 & a_1 & 1 \end{bmatrix} \begin{bmatrix} r_{ey}(0) \\ r_{ey}(-1) \\ r_{ey}(-2) \\ r_{ey}(-3) \\ r_{ey}(-4) \\ r_{ey}(-5) \end{bmatrix} = \begin{bmatrix} \lambda \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

□

Obviously, the noise variance λ has to be known. If we had the true parameters we could calculate the variance

of the equation error, V, of (10) as

$$\begin{aligned}\mathrm{Var}(V) =& \mathbf{E}[(\hat{Y}-\hat{\Phi}\theta)^T(\hat{Y}-\hat{\Phi}\theta)] = \\ & \mathbf{E}[(\tilde{Y}-\tilde{\Phi}\theta+E)^T(\tilde{Y}-\tilde{\Phi}\theta+E)] = \\ & \mathbf{E}[(-\Omega\tilde{Z}+E)^T(-\Omega\tilde{Z}+E)] = \\ & \lambda\cdot\mathrm{trace}(I-\Omega Q_1(\Omega Q_1)^\dagger).\end{aligned}$$

Hence an estimate of the noise variance λ is

$$\hat{\lambda} = \frac{(\hat{Y}-\hat{\Phi}\hat{\theta})^T(\hat{Y}-\hat{\Phi}\hat{\theta})}{\mathrm{trace}(I-\Omega Q_1(\Omega Q_1)^\dagger)}.$$

5 The Algorithm

The final algorithm is based on a bias correction of the parameter estimates.

1. Guess parameters and noise variance.
2. Do a least squares estimate of the missing data assuming the model is correct and form $\hat{Y}$ and $\hat{\Phi}$.
3. Compute the bias causing term $\Delta = -\mathbf{E}[\Phi^T\Omega\tilde{Z}]$ using the guessed parameters.
4. Compute the parameter estimate

   $$\hat{\theta} = (\hat{\Phi}^T\hat{\Phi})^{-1}\hat{\Phi}^T\hat{Y} - (\hat{\Phi}^T\hat{\Phi})^{-1}\Delta,$$

   where the last term will compensate for the bias, and the noise variance estimate

   $$\hat{\lambda} = \frac{(\hat{Y}-\hat{\Phi}\hat{\theta})^T(\hat{Y}-\hat{\Phi}\hat{\theta})}{\mathrm{trace}[I-\Omega Q_1(Q_1^T\Omega)^\dagger]}.$$

5. Iterate from 2 until convergence

The only differences between this algorithm and the algorithm described in the introduction is the term $(\hat{\Phi}^T\hat{\Phi})^{-1}\Delta$ which makes the parameter estimate unbiased and the way we estimate the noise variance.

Modifying (11) by adding and subtracting our bias compensation yields

$$\theta = \underbrace{[(\hat{\Phi}^T\hat{\Phi})^{-1}\hat{\Phi}^T\hat{Y} - (\hat{\Phi}^T\hat{\Phi})^{-1}\Delta]}_{\hat{\theta}} - \underbrace{[(\hat{\Phi}^T\hat{\Phi})^{-1}\hat{\Phi}^TV - (\hat{\Phi}^T\hat{\Phi})^{-1}\Delta]}_{\tilde{\theta}}$$

Therefore the parameter variance is evaluated as

$$\mathbf{E}(\tilde{\theta}\tilde{\theta}^T) = \mathbf{E}[(\hat{\Phi}^T\hat{\Phi})^{-1}(\hat{\Phi}^TV-\Delta)(\hat{\Phi}^TV-\Delta)^T(\hat{\Phi}^T\hat{\Phi})^{-1}].$$

This expression involves higher order moments, but should be computable.

6 Simulations

Experiment 1: The first example is an AR model (ARX without input, 500 samples) with 66 % of the outputs ($\approx$ 333 samples) missing at random. We made 800 Monte-Carlo simulations of the system

$$y_k = 1.5y_{k-1} - 0.7y_{k-2} + e_k$$

with noise variance 1. There is a significant bias if we do not make the bias corrections of the algorithm (Figure 1). However, if the corrections are made an unbiased estimate of the parameters results (Figure 2).

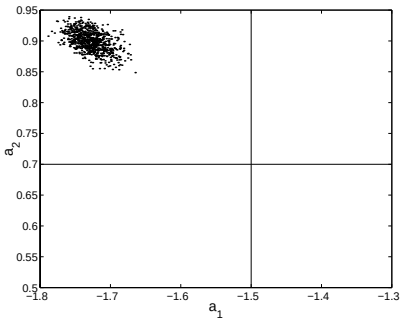

Figure 1: Results from 800 Monte Carlo simulations of experiment 1 without bias compensation. True parameters are marked by the cross.

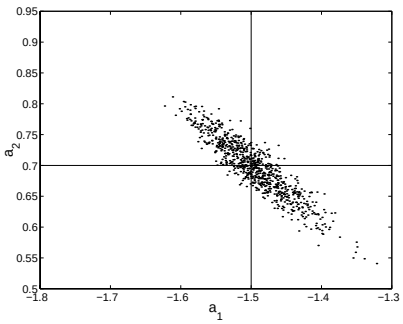

Figure 2: Results from 800 Monte Carlo simulations of experiment 1 with bias compensation. True parameters are marked by the cross.

Table 1 gives the mean and variance of the Monte Carlo simulations.

Parameter	Mean	Variance
biased a_1	-1.7285	$4.1173\cdot10^{-4}$
biased a_2	0.8994	$2.6577\cdot10^{-4}$
biased λ	0.2639	$1.3178\cdot10^{-3}$
unbiased a_1	-1.4956	$2.2551\cdot10^{-3}$
unbiased a_2	0.6967	$2.1167\cdot10^{-3}$
unbiased λ	0.9955	$1.8954\cdot10^{-2}$

Table 1: Mean and variance of the parameter estimates of experiment 1.

□

Experiment 2: The second example is 800 Monte-Carlo simulations of the ARX model (500 samples of the input, 500 samples of the output and noise variance 1)

$$y_k = 0.8y_{k-1} + 0.3u_{k-1} + e_k.$$

Data are missing periodically with pattern $< 0011 >$ in the output and $< 1101 >$ in the input, i.e. 50 % of the output data (250 samples) and 25 % of the input data (125 samples) are missing (zero means that a data point is missing). The input was generated by the AR process of experiment 1. The results with and without corrections are shown in Figures 3 and 4 as well as in Table 2.

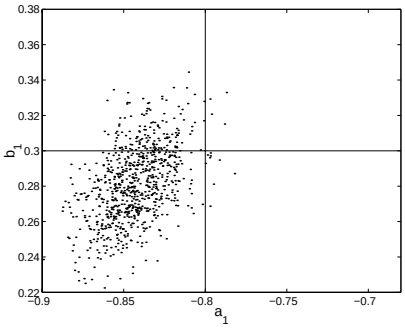

Figure 3: Results from 800 Monte Carlo simulations of experiment 2 without bias compensation. True parameters are marked by the cross.

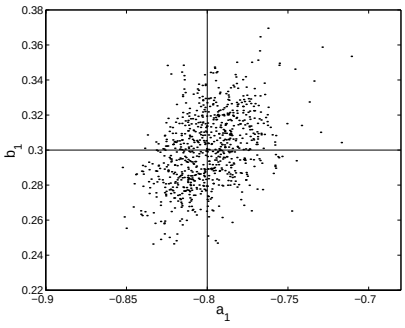

Figure 4: Results from 800 Monte Carlo simulations of experiment 2 with bias compensation. True parameters are marked by the cross.

Parameter	Mean	Variance
biased a_1	-0.8431	$3.2308 \cdot 10^{-4}$
biased b_1	0.2812	$4.4716 \cdot 10^{-4}$
biased λ	0.2357	$9.0030 \cdot 10^{-4}$
unbiased a_1	-0.7991	$4.1930 \cdot 10^{-4}$
unbiased b_1	0.3006	$4.4353 \cdot 10^{-4}$
unbiased λ	0.9879	$1.7082 \cdot 10^{-2}$

Table 2: Mean and variance of the parameter estimates of experiment 2.

□

7 Relation to the EM method

An interesting question is what the relation is to the method described in [4], which is based on the EM algorithm. In that paper the missing data are computed with a Kalman filter smoother and a bias correction is done based on the Kalman filter covariances.

In [2] it is shown that the least squares method is equivalent to Kalman filtering under certain conditions. Our conjecture, although left for future research, is that if the correction is the same the methods may also be equivalent.

What is then the point of implementing the method the way we have done? There are at least four reasons:

- There is no need for the input model of the EM method.
- The EM method does not give an estimate of the parameter variance. Here we have a potential to compute an estimate of the variance.
- The pedagogic value of the new algorithm is high. It is relatively easy to understand what is done and why.
- By utilizing the fact that many of the matrices in the algorithm are sparse, a faster implementation than the Kalman filter can be made.

8 Modelling the input

The algorithm works fine for massive losses of data in either output or input. One advantage of the algorithm compared to the one described in [4] using the EM algorithm is that it does not require a model for the input. However, if data are missing in both output and input in an unfortunate pattern the algorithm may fail as ΩQ_1 will not have full column rank. One way to solve this problem is to introduce a model for the input signal and in this way get more equations to use for the least squares estimate of the missing data.

The difference from the original algorithm is that more parameters have to be estimated. However, the size of the matrix that has to be inverted in the data estimation step is still number of missing data. We have done a few experiments and it seems that you should not introduce an input model if you do not have to, as the parameter variance increases.

9 Conclusions

We have in this paper studied identification from incomplete data sets. A criterion for when a least squares estimate of the model parameters from estimated missing data is unbiased was given. Usually such an estimate is biased. Therefore a bias correction was derived.

The performance of the proposed algorithm was illustrated by Monte-Carlo simulations. Implementation can be rather efficient if the fact that many matrices are sparse is taken into consideration.

The algorithm usually works fine for large amounts of missing data in either input or output. However, when much data are missing in both input and output and the missing data pattern is unfavourable, the algorithm will not work. This problem can however be solved by introducing a model for the input.

References

[1] Adams, G. J., Albertos P., Goodwin G. C. and Isaksson A. J., "Parameter estimation for ARX models with missing data", *Postprint volume from the IFAC Symposium on System Identification (SYSID '94)*, Volume 1, 1995, pages 163-168.

[2] Brown R. G. and Hwang Y. C., *Introduction to random signals and applied Kalman filtering*, 3rd editon, Wiley, 1997.

[3] Isaksson, A. J., "A recursive EM algorithm for identification subject to missing data", *Postprint volume from the IFAC Symposium on System Identification* (SYSID '94), Volume 2, 1995, pages 953-958.

[4] Isaksson, A. J.,"Identification of ARX-models subject to missing data", *IEEE Transactions on utomatic Control*, Volume 38, No 5, 1993, pages 813 - 819.

[5] Little R. J. A., Rubin D. B., *Statistical Analysis with missing data*, Wiley, New York, (1987).

[6] Pintelon R., Schoukens J., "Identification of continuous-time systems with missing data", *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference*, Volume 2 ,1999 , pages 1081 -1085.

[7] Rosen, Y. and Porat, B., "Optimal ARMA parameter estimation based on the sample covariances for data with missing observations", *IEEE Transactions on Information Theory*, Volume 35, No 2, 1989, pages 342 - 349.