A Study of the Choice of Model Orders in Arxstruc-Type Methods for Open-Loop Time-Delay Estimation in Linear Systems

Svante Björklund

Control & Communication
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
E-mail: svabj@isy.liu.se

31st August 2003

Report no.: LiTH-ISY-R-2536

Technical reports from the Control & Communication group in Linköping are available at http://www.control.isy.liu.se/publications.
# A Study of the Choice of Model Orders in Arxstruc-Type Methods for Open-Loop Time-Delay Estimation in Linear Systems

In this report we study estimation of time-delays in linear dynamical systems with additive noise. Estimating time-delays is a common engineering problem, e.g. in automatic control, system identification and signal processing.

The purpose with this work is to test and evaluate a certain class of methods for time-delay estimation, especially with automatic control applications in mind. The principle of the methods in the class is to estimate several discrete-time models of a certain model structure with different explicit time-delays. The estimated time-delay is the time-delay whose model has the lowest mean square difference between the true and estimated output signal. The methods are evaluated experimentally with the aid of simulations and plots of RMS error and confidence intervals for different cases.

The results are: The output error (OE) model structure has the lowest RMS error but is very slow. Low model orders give the best result. The ARX model structure has a higher RMS error but is very fast. High model orders give the best result. An ARX model structure with prefiltered input and output signals was also tested. It has an RMS error that is nearly as good as for the OE model structure and is fast but not as fast as the unfiltered ARX. The best model orders are high for the denominator polynomial and low for the numerator polynomial.

**Keywords**
- time-delay, dead-time, estimation, system identification, linear dynamic systems, ARX, OE, PEM, ANOVA, confidence intervals, simulations, open loop
## Contents

1 Introduction 1

2 The methods 2

2.1 Arxstruc 2
2.2 Oestruc 2
2.3 Met1struc 2
  2.3.1 Principles 3
  2.3.2 Implementation 5

3 Simulation setup 7

4 Results 10

4.1 Choice of arxstruc parameters 10
4.2 Choice of oestruc parameters 11
4.3 Choice of met1struc parameters 11
4.4 Comparison of methods 13

5 Discussion and conclusions 14

5.1 Discussion 14
5.2 Conclusions 14
5.3 Future work 15

References 17

A Analysis by confidence intervals 19

A.1 Choice of arxstruc parameters 19
A.2 Choice of oestruc parameters 19
A.3 Choice of met1struc parameters 21
1 Introduction

The problem we address in this report is estimating time-delays in linear dynamical systems with additive noise. A synonym for time delay is dead-time. Estimating time-delays is a common engineering problem, e.g. in control performance monitoring of industrial processes [Hor00, Swa99], in design and tuning of controllers, in range estimation in radar [KQ92] and in direction estimation by time-delay of arrival in signal intelligence [HR97, FHJ02, Wik02]. Dead-time estimation is also a necessary part in all system identification [Lju99].

The purpose with this work is to test and evaluate a certain class of methods for time-delay estimation, especially with automatic control applications in mind. The principle of the methods in the class is to estimate several discrete-time models of a certain model structure with different explicit integer time-delays. The estimated time-delay is the time-delay whose model has the lowest mean square difference between the true and estimated output signal. The methods are tested and evaluated experimentally with the aid of simulations.

In the next chapter, the time-delay estimation methods are briefly described. Then, Chapter 3 is about the simulation setup. After that, in Chapter 4 the analysis of the simulations is conducted. Following, Chapter 5 contains discussion, conclusions and suggestions for further work. Appendix A contains validation of required prerequisites for the analysis.
Algorithm 1 MATLAB code for arxstructd using the Matlab System Identification Toolbox.

```matlab
function dtEst = arxstructd(zIn);
zIn = [outSig, inSig];
na = 10;
rb = 5;
nkVec = 1:20;
nkMax = length(nkVec);
n = na*ones(nkMax,1), nb*ones(nkMax,1), nkVec';
V = arxstruc(zIn,zIn,nn);
modelStruc = selstruc(V,0);
dtEst = modelStruc(3);
```

2 The methods

2.1 Arxstruc

In the method arxstruc several ARX models [Lju99]  

\[ A(q)y(t) = B(q)u(t - n_k) + e(t) \]

are estimated with PEM (Prediction Error Method) [Lju99] for different time-delays \( n_k \). The delay whose model has the lowest loss function is chosen. The estimation is quick since the ARX model can be written as a linear regression and can be estimated by solving a linear equation system [Lju99]. See Algorithm 1 for MATLAB code using the Matlab System Identification Toolbox.

2.2 Oestruc

The principle of the methods oestruc is the same as for arxstruc but OE (output error) models [Lju99]  

\[ y(t) = \frac{B(q)}{F(q)} u(t) + e(t) \]

are estimated instead of ARX models. To estimate OE models is much more computationally demanding than ARX models since a multidimensional optimization with a numerical search must be carried out [Lju99]. See Algorithm 2 for MATLAB code using the Matlab System Identification Toolbox.

2.3 Met1struc

The method oestruc gives a better result than arxstruc in the simulations presented in this report (see Sections 4.1 and 4.2). On the other hand, arxstruc has a much lower computation time than oestruc. In this section we suggest a new time-delay estimation method with the aim to imitate oestruc but with much lower computational demands.
Algorithm 2 MATLAB code for oestructd using the Matlab System Identification Toolbox.

```matlab
function dtEst = oestructd(zIn);
zIn = [outSig, inSig];
f = 2;
b = 1;
nkVec = 1:20;
for nnn = 1:length(nkVec),
    model = oe(z,[nb nf nkVec(nnn)], 'Covariance','None')
    lossFunc(nnn) = model.NoiseVariance;
end
[minVal,nnnmin] = min(lossFunc);
dtEst = nkVec(nnnmin);
```

2.3.1 Principles

Assume the true system is given by

\[ y = G_0 u + H_0 e, \]

where \( G_0 \) and \( H_0 \) are rational functions in the delay operator \( q^{-1} \). The noise \( e \) is white. The model structure used to estimate the true system is

\[ y = Gu + He, \]

where \( G \) and \( H \) also are rational functions in the delay operator \( q^{-1} \).

The reason for the difference between OE and ARX model structures can be two-fold:

1. If the model structure \( G \) can not exactly describe the true system \( G_0 \), the estimated model \( G \) will have a bias, even if the number of data \( N \rightarrow \infty \). The OE and ARX model structures will behave differently ;[Lju99, Ex. 8.5, p. 268-269];

   - The ARX model structure will give models with a good fit to the true system at high frequencies.
   - The OE model structure will give models with a good fit to the true system at low frequencies.

2. If the model structure \( G \) can exactly describe the true system \( G_0 \), again the OE and ARX model structures behave differently:

   - The ARX model structure will give models \( G \) with a bias if the noise model structure \( 1/A \) cannot describe the true noise system \( H_0 \). See Equations 8.63 and 8.69 and page 267 in [Lju99].
   - The OE model structure will give estimates without bias. The bias of the OE model structure is thus not dependent on the noise model. See [Lju99, Eq. 8.71, p. 266].
Algorithm 3 Proposed time-delay estimation method.

1. Estimate an ARMAX model \( A_1(q)y(k) = B_1(q)u(k) + C_1(q)e(k) \).

2. Prefilter \( u \) and \( y \) through \( 1/C_1(q) \).

3. Arxstruc gives an estimate of \( n_k \).

We propose the time-delay estimation method in Algorithm 3. The motivation for this method is the following. Assume the true system is given by

\[
y = G_0u + H_0e = \frac{B_0}{F_0}u + \frac{C_0}{D_0}e,
\]

where \( G_0 \) and \( H_0 \) are rational functions and \( B_0 \), \( F_0 \), \( C_0 \) and \( D_0 \) are polynomials in the delay operator \( q^{-1} \). The noise \( e \) is white.

Estimate a first model

\[
y = \frac{B_1}{F_1}u + \frac{C_1}{D_1}e \tag{2.1}
\]

where \( F_1 \), \( B_1 \), \( C_1 \) and \( D_1 \) are polynomials in \( q^{-1} \), i.e. a Box-Jenkins model [Lju99].

The model \( C_1/D_1 \) will be an approximation of \( H_0 = C_0/D_0 \). Then, filter the output signal \( y \) through \( D_1/C_1 \) giving \( y_F \):

\[
y_F = \frac{D_1}{C_1}y = \frac{D_1}{C_1} \left( \frac{B_0}{F_0}u + H_0e \right) = \frac{B_0}{F_0} \frac{D_1}{C_1}u + H_0 \frac{D_1}{C_1}e.
\]

We notice that \( u_F = uD_1/C_1 \) is the input signal \( u \) filtered through \( D_1/C_1 \). We rewrite the equation for \( y_F \) as

\[
y_F = \frac{B_0}{F_0}u_F + H_0 \frac{D_1}{C_1}e \tag{2.2}
\]

Now, if the polynomials \( F_1 \), \( B_1 \), \( C_1 \) and \( D_1 \) in the model structure 2.1 have high enough orders so that they can exactly describe the true system and the input-output data is informative enough, then \( F_1 \), \( B_1 \), \( C_1 \) and \( D_1 \) will converge to the true values \( F_0 \), \( B_0 \), \( C_0 \) and \( D_0 \) as the number of data \( N \to \infty \) [Lju99, p. 273]. When this happens, \( D_1/C_1 = 1/H_0 \) and Equation 2.2 simplifies to

\[
F_0y_F = B_0u_F + e \tag{2.3}
\]

where \( e \) as before is white noise. This true system is obviously an ARX system. This means that if we estimate an ARX model \( A_1y_F = Bu_F + e \), there will be no bias due to an incorrect noise model. If an ARX model has high enough orders, we can get a model for the system \( G \) without bias (as in the case with an OE model of high enough orders).
1. Estimating a state space model by state space method.
2. Converting to $A_1(q)y(k) = B_1(q)u(k) + C_1(q)e(k)$.
3. Prefiltering $u$ and $y$ through $1/C_1(q)$.
4. Arxstruc gives an estimate of $n_k$.

Let us now assume that the true system instead has OE structure:

$$y = G_0u + H_0e = \frac{B_0}{F_0}u + e \Rightarrow F_0y = B_0u + F_0e.$$  \hfill (2.4)

Since this has ARMAX structure [Lju99] we estimate a first model

$$A_1y = B_1u + C_1e,$$  \hfill (2.5)

where $A_1$, $B_1$ and $C_1$ are polynomials in $q^{-1}$. $A_1$ and $C_1$ will be approximations of $F_0$. If we filter the input and output signal through $1/C_1$ we will get the following true system

$$F_0yF = B_0uF + e,$$  \hfill (2.6)

which is also of ARX structure and can be approximated by an ARX model of enough model order without bias.

Since the systems simulated in this report (Chapter 3) have OE structure (equation (2.4)), we will filter the input and output signals through $1/C_1$ where $C_1$ is from equation (2.5).

2.3.2 Implementation

In Algorithm 3 the first step is to estimate an ARMAX model. Unfortunately, the standard way to estimate a model of this model structure also requires a numerical search as with the OE model which we tried to avoid. Another way is to first estimate a state space model and then convert it to an ARMAX model. This conversion will be possible if the order of the state space model and the orders of the polynomials $A_1$, $B_1$ and $C_1$ are high enough to describe the true system. The state space model can be quickly estimated by a subspace method [Lju99]. See Algorithm 4 for the resulting method, which we call met1struc. Matlab code for met1struc is given in Algorithm 5. The order of the state space model is 10. This will hopefully be enough for most systems. The orders of $A_1$, $B_1$ and $C_1$ will depend on the order of the state space model.
Algorithm 5 MATLAB code for met1struct using the Matlab System Identification Toolbox. The function arxstructd is described in Section 2.1.

```matlab
function dtEst = met1structd(inSig, outSig, Ts)
    order = 10;
    modelSs = n4sid(iddata(outSig, inSig, Ts),order,'cov','none');
    [A,B,C,D,F] = polydata(idpoly(modelSs));
    BFilt = 1;
    AFilt = C;
    uFilt = filter(BFilt,AFilt,inSig);
    yFilt = filter(BFilt,AFilt,outSig);
    zIn = [yFilt, uFilt];
    na = 10;
    nb = 1;
    nkVec = 1:20;
    dtEst = arxstructd(zIn,nkVec,na,nb);
```
Figure 3.1: Impulse response of system $G_1-G_2$ and $G_5-G_6$. True time-delay after sampling $T_d = 10$.

3 Simulation setup

The setup for the simulations is the same as in [Bjo03a] with the following exceptions:

- The number of trials was 2048 for arxstruc, 192 for oestruc and 512 for met1struc. The reason for the different number of trials is the very different execution times for the methods. The trials were split into four groups and each group was used to compute an estimate of the RMS error (our response variable) of the time-delay estimate. This gave 4 estimates of the RMS error that was used in the calculation of the confidence intervals. See [Bjo03a].

- The methods were obviously different. See Chapter 2.

Three environment factors were varied during the simulations: The system, the input signal type and the SNR [Bjo03a]. The signal-to-noise ratio (SNR) was either 1 or 100. See [Bjo03a] for the definition of the SNR. The impulse responses of the four used systems are depicted in Figure 3.1. Note that for all the systems the time delay will be 10 after the sampling. More information about the systems can be found in [Bjo03a].

Figures 3.2-3.4 show the used input signals in the time and frequency domains. Figure 3.2 depicts the signal RBS 10-30% which is a bandpass random signal with frequency contents
between 10% and 30% of the Nyquist frequency. Figure 3.2 depicts the signal RBS 0-100% which is a wideband random signal with frequency contents between 0% and 100% of the Nyquist frequency. It is thus white noise. Figure 3.4 depicts the signal Steps which is a signal with three steps. It has a frequency contents between 0% and about 5% of the Nyquist frequency. More information about the input signals can be found in [Bjo03a].

In addition to the environment factors, three method factors were varied during the simulations: The model orders $n_a$ (or $n_f$) and $n_b$ and using prewhitening (see [Bjo03a]) or not.
Figure 3.4: Time signal (left) and frequency spectrum (right) for a realization of the input signal type Steps.
4 Results

This chapter presents plots of the RMS error of the time-delay estimation on simulated signals. The RMS values have the unit sampling interval.

4.1 Choice of arxstruc parameters

For the method arxstruc we see in Figure 4.1 that there are a lot of values of $n_a$, $n_b$ and prewhitening that give approximately the same average RMS error. The lowest RMS error has $n_a = 10$, $n_b = 3$ and without prewhitening in this simulation. However, we prefer $n_a = 10$, $n_b = 5$ and without prewhitening because these values gave the best result in a similar simulation (t102b1.m and t102b2.m). We call this combination arxstruc3.

The ANOVA of this simulation required much computing power, both execution time and memory. It is questionable if the ANOVA and confidence intervals are useful because the residuals are very non-Gaussian and the variance is not constant, see Figure A.1-A.2 in Appendix A.
4.2 Choice of oestruc parameters

For the method oestruc it is seen in Figure 4.2 that the best model orders with respect to time-delay estimation are the lowest \((n_f = 2, n_b = 1)\) of the tested. If \(n_b > 1\) this would enable more models that give a low optimization criterion value but with different time-delays. It is also seen in the same figure that without prewhitening is the best.

The ANOVA of this simulation required much computing power, both execution time (2 h) and memory (590 MB) on a SunBlade 100 computer. If we study Figure A.3 in Appendix A, we will doubt that the ANOVA and the confidence intervals are useful because the residuals are non-Gaussian. On the other hand the result in Figure A.5 is very clear. The confidence interval with the model orders \(n_f\) and \(n_b\) and prewhitening (or not) with the lowest RMS error is clearly separated from the other confidence intervals. This makes the confidence interval analysis robust and the confidence intervals confirm that \(n_f = 2, n_b = 1\) and no prewhitening is the best choice. We call this combination oestruc3.

4.3 Choice of met1struc parameters

In Figure 4.3 it is seen that the best model parameters for the method met1struc with respect to time-delay estimation are \(n_a = 10, n_b = 1\) and without prewhitening. We call this combination met1struc3.
Figure 4.3: RMS error for met1struc as a function of the model orders $n_a$ and $n_b$ and prewhitening or not. (t163b2.m)
It is questionable if the ANOVA and confidence intervals are useful because the residuals are non-Gaussian and the variance is not constant, see Figure A.6-A.7 in Appendix A.

4.4 Comparison of methods

When we look at Figures 4.1-4.3 and measure the execution time we see that

- **oestruc** has the lowest RMS error (1.8 sampling intervals) but is very slow (One estimation took 13.0 s on a SunBlade 100 computer).
- **met1struc** has a RMS error that is nearly as good as for **oestruc** (met1struc: 2.0 sampling intervals) and is fast (One estimation took 0.741 s).
- **arxstruc** has a higher RMS (2.7 sampling intervals) but is very fast (One estimation took 0.143 s).

The RMS errors are much lower than for thresholding methods [Bjö03c].
5 Discussion and conclusions

5.1 Discussion

We find in this report that oestruc is the best method in the tested cases. Also Swanda in [Swa99] consider that oestruc is better than arxstruc. It is not surprising that oestruc is better than arxstruc since the tested systems have OE structure. This also helps met1struc to give good results.

When estimating a discrete-time state space model (zoh sampling) of a system with a long time-delay (longer than the sampling interval) the order of the model will increase with one for each sampling interval of the time-delay [AW84, p. 42]. This could indicate that the used order 10 of the state space model in met1struc could be too low for long time-delays. If the continuous-time time-delay is 9, a model order of 10 seems to be on the limit to be too low. Another way for the state space model to handle the time-delay is to approximate it with non-minimum phase zero(s). In this way a lower model order can be sufficient. This is also what happens in met1struc. However, for longer time-delays than used in this report, it would be advisable to use a higher fixed model order or to chose the model order automatically to give a good model. This can be done by giving 'best' as the input parameter order to the function n4sid (Algorithm 5) in the Matlab Identification Toolbox.

The advantage of met1struc over oestruc is the higher execution speed. A disadvantage is that it is more complicated. In applications where the time-delay is changing and the noise does not change it should be possible to estimate the noise model once off-line and use it in many subsequent time-delay estimations with a modified met1struc method. It is not necessary to estimate this noise model with a subspace state space method as in Section 2.3 but can be done by a less complicated method.

In Section 4.1 the best choice of model orders for arxstruc was the highest of the tested $n_a$ and $n_b$. The reason for this is probably that high orders are needed to approximate the noise system well by $1/A$ since the true systems are not of ARX structure. An all-pole system $1/A$ of enough high order should be able to approximate the noise system enough well. Such an approximation is used in [FMS91, p. 655]. See also the discussion in Section 2.3.1 about bias in the model $G$ for different cases of system/model structures and orders.

In Section 4.2 the best choice of model orders for oestruc was the lowest of the tested ($n_a = 2$ and $n_b = 1$). These orders are enough to accurately model the true system. The true systems are either of second or fourth order. This would mean that $n_a = 2$ or $n_a = 4$ would be appropriate. In average $n_a = 2$ is apparently better. If the order $n_b$ were higher than 1 it would introduce an ambiguity for the time-delay. For example, both the true time-delay and one minus the true time-delay (with the the first $B$-parameter equal to zero) would give good fits to the data. Therefore it is understandable that $n_b = 1$.

The theoretical explanation for the choice of model orders for met1struc in Section 4.3 is not clear.

5.2 Conclusions

We draw the following conclusions from the work in this report:
• The method \textit{oestruc} give the best estimates but it is very slow. The lowest model orders \((n_f = 2 \text{ and } n_b = 1)\) give the best result since they can describe the input-output dynamics well.

• The method \textit{arxstruc} give estimates that is not as good as \textit{oestruc} in the tested cases (true OE systems) but it is very fast. High model orders (e.g. \(n_a = 10, n_b = 5\)) give the best result.

• The method \textit{met1struc} give nearly as good estimates as \textit{oestruc} in the tested cases but is much faster. However, it is slower than \textit{arxstruc}. The best model orders are high \(n_a (n_a = 10)\) and low \(n_b (n_b = 1)\).

5.3 Future work

Possible future work is:

• Test with other true model structures than output error.

• Test with closed-loop.

• Try also filter the input and output signals with \(D_1/C_1\) to handle arbitrary noise systems. See Section 2.3.

• Test with other orders of the state space model in \textit{met1struc}.
References


This appendix contains an attempt to analyze with ANOVA and confidence intervals for pair-wise comparisons. The Analysis was performed in the same way as in [Bjo03b, Bjo03c]. Since we consider the prerequisites (see [Mon97, Bjo03a]) only to be fulfilled for the method oestruc, only for this methods confidence intervals are presented. Common validation graphs [Mon97, Bjo03a] are shown for all tested methods.

A.1 Choice of arxstruc parameters

Figure A.1-A.2 shows plots for testing whether the prerequisites for ANOVA and confidence intervals are fulfilled when using arxstruc. The positive transformation $x^{0.505127}$ was used. This means ”The lower the better” in a confidence interval plot.

A.2 Choice of oestruc parameters

Figure A.3-A.4 shows plots for testing whether the prerequisites for ANOVA and confidence intervals are fulfilled when using oestruc. Figure A.5 contains confidence intervals for pair-wise comparisons between different model orders and with/without prewhitening. The
Figure A.2: Standard deviation of residuals vs. factor levels for arxstruc.
positive transformation $x^{0.4296}$ was used. This means "The lower the better" in the confidence interval plot.

### A.3 Choice of met1struc parameters

Figure A.6-A.7 shows plots for testing whether the prerequisites for ANOVA and confidence intervals are fulfilled when using met1struc. The positive transformation $x^{0.4028}$ was used. This means "The lower the better" in a confidence interval plot.
Figure A.4: Standard deviation of residuals vs. factor levels for oestruc.
49 groups have population marginal means significantly different from Group 1

Figure A.5: oestruc: Confidence intervals (the lines in the circles) for pair-wise comparisons (95% simultaneous confidence level) for different thresholding methods and input signals. Positive transformation: (RMS error)^(0.4296) = > "The lower the better". (t156b2.m)
Figure A.6: Residual analysis for ANOVA of met1struc.
Figure A.7: Standard deviation of residuals vs. factor levels for met1struc.