

INITIALISATION ASPECTS FOR SUBSPACE AND OUTPUT-ERROR IDENTIFICATION METHODS

Lennart Ljung

Division of Automatic Control, Linköping University, SE-581 83, Linköping, Sweden, email: ljung@isy.liu.se

Keywords: system identification, subspace techniques, ARX models, continuous-time model estimation

Abstract

This paper is inspired by a recent contribution by Rao and Garnier about identification of continuous time models. They show examples where methods that directly estimate continuous time models, based on smoothed differentiated input-output data outperform methods that are based on discrete time model estimation. The reasons for that situation are investigated in this contribution. It turns out that the key problem is that ARX-type models are very biased for the example in that study, which leads to problems for initializations for output error models both based on ARX/IV techniques and on subspace (CVA estimation techniques). The remedy is to decrease the ARX-bias via low pass data filtering, which in turn also explains why the direct continuous-time estimation techniques (with inherent data smoothing) do not suffer from this problem.

1 Introduction

A recent paper, [5] shows comparisons between two ways of estimating continuous time models:

1. Directly fitting smoothed derivative approximations of input and outputs to continuous-time models, e.g. [1], [9]
2. Estimating discrete time models from the data, which are then transformed to continuous time.

The results show, for the chosen example, that approach (1) is much better than approach (2). This is intriguing, since theoretically the route via discrete time models cannot be inferior to direct fitting. In this paper we confirm that the selected system in [5] indeed gives severe problems for the basic discrete time identification techniques, including both prediction-error, output error and subspace (CVA/MOESP/N4SID) techniques.

We will investigate the reasons for these problems, and show that they can be traced to extreme sensitivity of the ARX-model for data from this system. ARX-models are behind many initialization techniques for methods based on minimization of model fits. They can also be seen as a basis for the subspace-techniques. This will explain the difficulties associated with the selected test system.

It also turns out that the remedy is to move the focus in the model fit to lower frequencies by proper pre-filtering. Since

pre-filtering is inherent in the direct continuous-time techniques this also explains why such initialization problems do not occur for those techniques.

The calculations in MATLAB given in teletype font below are all done in [4]. This version has some features (frequency domain data and easy focus definition) that are not present in previous versions.

2 The Rao-Garnier test system

In [5] a continuous time system is tested with various inputs and noise levels. The system is

$$G_0 = \frac{-4s + 1}{(s^2/400 + 0.01s + 1)(s^2/4 + 0.25s + 1)} \quad (1)$$

It has two resonance frequencies, 20 and 2 rad/sec, with damping ratios 0.01 and 0.25 respectively. The settling time of the system's impulse response is about 10 seconds.

The system is sampled with a sampling interval $T_s=0.01$ sec, and the input is chosen as

$$u(t) = \sin(t) + \sin(1.9t) + \sin(2.1t) + \sin(18t) + \sin(22t) \quad (2)$$

This sampling time is not unrealistic, in view of the resonance frequency of 22 rad/sec. The sampling frequency will be about 10 times the bandwidth of the system, an often given rule-of-thumb. On the other hand, the settling time of system will be about 1000 samples, quite a high figure.

The system is simulated (with a zero order hold input) with this input and sampling interval, and white measurement noise corresponding to a SNR of 10 dB was added. All this corresponds to "trial2" in [5], except that in that paper, the input is not subjected to a zero-order-hold block. A portion of the data set is shown in Figure 1. With the relatively fast sampling used here, there is no major difference between sampled models and continuous time models, so we can check out discrete time models estimated from the data to start with.

2.1 Time-domain data

A sampled version of (1) will have 4 poles, of course, an in general 3 zeros (4 parameters in the numerator). Possible prior knowledge that there is only one zero in the continuous time model, is not so easy to accommodate in discrete time modeling.

A direct estimation of an output error model (in input/output polynomial form) gives

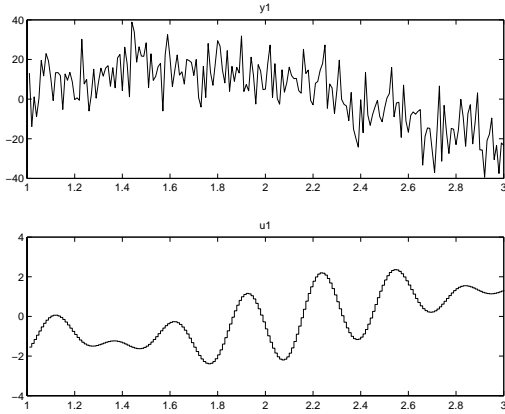


Figure 1: A portion of the generated data set.

```
m1 = oe(data,[4 4 1]);
```

and a general 4th order state space model is obtained by

```
m2 = pem(data,4);
```

(this model is obtained by initialization of a CVA-type subspace method, and then minimizing the prediction error of the state-space model in a free parameterization for the measured data.) Fifty data sets with different realizations of the additive noise were tested. The bode plots of these models are shown together with the true system in Figures 2 and 3. These models

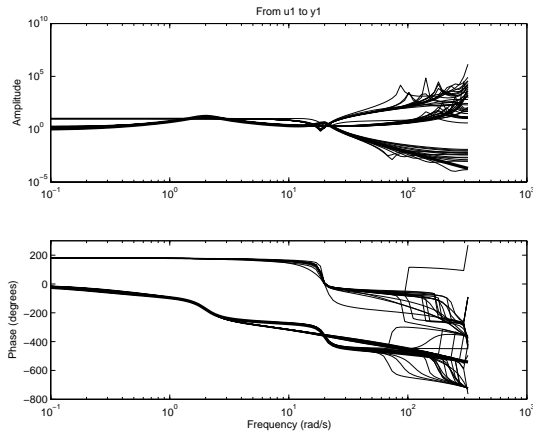


Figure 2: The Bode plots of the identified OE-models together with the true system. The true system is the thicker line.

are not good, a fact that was noticed also in [5].

2.2 Frequency-domain data

One may convert the data to the frequency domain and estimate data using the Fourier transforms of the sampled data; still building discrete-time models:

```
df = fft(data);
m1f = oe(df,[4 4 1]);
```

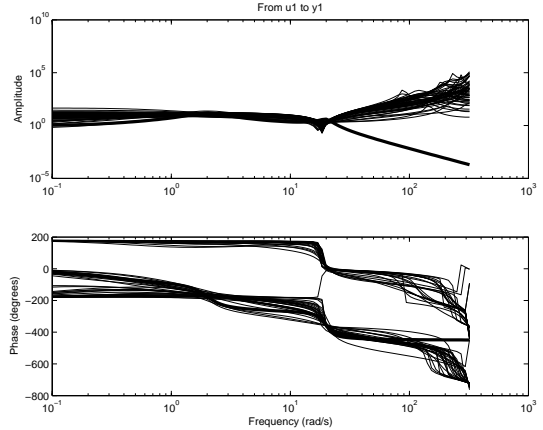


Figure 3: The Bode plots of the identified PEM-models together with the true system. The true system is the thicker line.

```
m2f = pem(df,4);
```

This gives models of basically the same quality as those in Figure 2.

Viewing the data sequence as a vector, the transformation to the Fourier domain is just an orthonormal change of basis. Therefore no major changes in the resulting models should be expected. This was consequently confirmed in the estimation above.

3 An initialization problem?

One may ask if the bad results are consequences of some inherent problem in the output error and state-space-prediction error methods. Trying to initialize the iterative search at models closer to the true system, however gives good models. The problem is thus one of poor initialization of the iterative search method. We can confirm that also by looking at the models, where the search is initialized:

```
m10 = oe(data,[4 4 1], 'maxiter', 0)
m20 = n4sid(data,4)
```

These models show the same behavior as those in Figure 2. The models from the PEM initialization are shown in Figure 4. We see that also the initial models are bad, and obviously lie in the domain of attraction of the bad models shown in Figure 2.

Both initializations have as a root ARX-models of the type

$$y(t) + a_1 y(t-1) + \dots + a_n y(t-n) = b_1 u(t-1) + \dots + b_m u(t-m) \quad (3)$$

The output error method `oe` uses $n = m = 4$ to get a first model, which is then used to generate instruments for an instrumental variable model. The subspace method `n4sid`, can be seen, somewhat simplified, ([3], Section 8.4) as model reduction of an ARX model, where n and m are the “horizons”

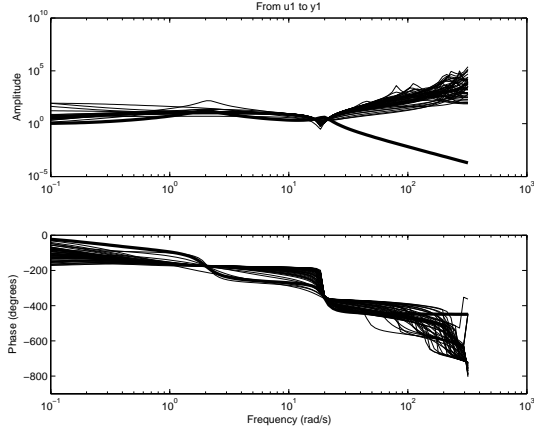


Figure 4: The Bode plots of the initial models for PEM (obtained by a CVA subspace method N4SID) together with the true system. The true system is the thicker line.

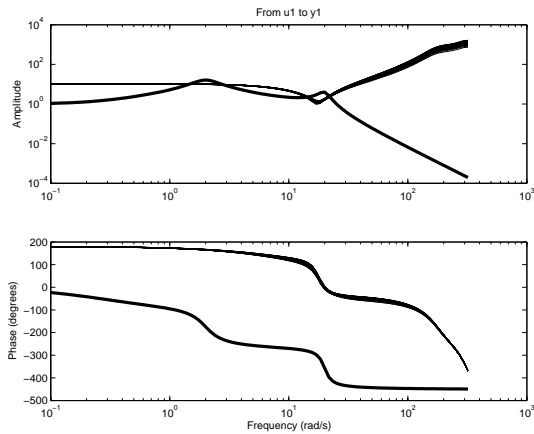


Figure 5: The Bode plots of the 50 identified fourth order ARX models, together with the true system (the thicker line).

or auxiliary orders. Figure 5 also shows the ARX-estimate ($n = m = 4$) for the the 50 different data sets.

It confirms that the characteristics of the first ARX-model looms over both the initial estimate and the resulting output error and PEM estimates.

The fact the the subspace/CVA estimate is so bad for this particular system should deserve an analysis of its own, since CVA is known to be very reliable in general. The basic reason in this particular case is probably that only 5 sinusoids are exciting the system, so the higher order ARX-models employed by CVA/subspace are not reliable.

4 A bias analysis of ARX-models

Let us therefore more closely analyze the ARX-method for this system. Suppose that the true system can be represented by

$$y(t) + a_1y(t-1) + \dots + a_4y(t-4) = b_1u(t-1) + \dots + b_4u(t-4) + v(t)$$

The limiting model is obtained as

$$\theta^* = R^{-1}f = \theta_0 + R^{-1}g \quad (4a)$$

$$R = E\varphi(t)\varphi^T(t); \quad f = E\varphi(t)y(t) \quad (4b)$$

$$g = E\varphi(t)v(t) \quad (4c)$$

$$\varphi^T(t) = [-y(t-1) \quad \dots \quad -y(t-n) \quad \dots \quad u(t-m)] \quad (4d)$$

Let us first consider noise free data ($v = 0$). If the input is white noise with unit variance, the matrix R can be calculated exactly using the Lyapunov-equation. The upper 4-by-4 block is

$$\begin{bmatrix} 1.6304 & 1.6238 & 1.6040 & 1.5720 \\ 1.6238 & 1.6304 & 1.6238 & 1.6040 \\ 1.6040 & 1.6238 & 1.6304 & 1.6238 \\ 1.5720 & 1.6040 & 1.6238 & 1.6304 \end{bmatrix} \quad (5)$$

The condition number of the full matrix is $1.4 \cdot 10^8$. The noise free-data thus has a very ill-conditioned regression matrix. Even if we have a small disturbance we may still have a non-trivial bias. Suppose that there is white measurement noise in the data with variance δ , that is $v(t) = A(q)e(t)$, $Ee^2(t) = \delta$. Then the bias can be expressed exactly as

$$\theta_{BIAS} = \delta(R + \delta D)^{-1} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (6)$$

$$D = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \quad (7)$$

It is interesting to investigate this expression as a function of δ . Figure 6 shows $\text{norm}(\theta_{BIAS})$ as a function of δ . Note that the

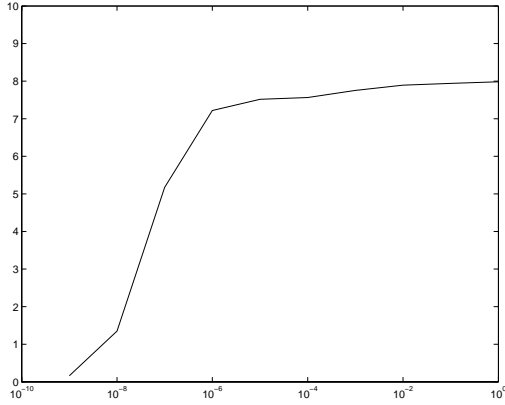


Figure 6: $\text{Norm}(\theta_{BIAS})$ as a function of δ for the system (1), with a white noise, unit variance input and white measurement noise with variance δ .

bias is quite substantial even for very small noise levels. Contrary to what one may think, it does not grow very much with the noise level. The reason is that the regression matrix $R + \delta D$ becomes better conditioned for larger δ . Note from Figure 5 that the bias dominates in the obtained ARX-models. Despite the bad conditioning of the regression matrix, and the associated noise sensitivity, there is not so much variation between the different realizations. It is important to realize that the bias effect illustrated in Figure 5 is an *input-output property*, that does not depend in the internal model parameterization. It is thus not a case of numerical errors.

Another way to see this bias problem is as follows. It is known, e.g. [8], that ARX-model fitting uses a frequency weighting that is $|A_0(e^{i\omega})|^2$. This function is shown in Figure 7. It can be seen that high frequencies have a weight that is 10^{12} times larger than low frequencies. Note the relationship to the condition number of the regression matrix: For high orders of the estimated A -polynomial, the “ A -part” of the regression matrix has a condition number which is the ratio of the largest to the smallest value of the curve in Figure 7.

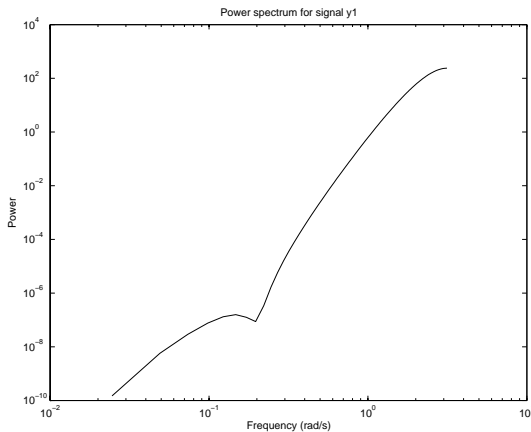


Figure 7: The Bode plot of $|A_0(e^{i\omega})|^2$.

4.1 A remedy: prefiltering

It is clear that if the equation error was white, there would be no bias problems, even if the regression matrix is ill-conditioned. That should mean that prefiltering the data by $A_0(q)$ which transforms white measurement error into white equation error noise, should be helpful.

```
md = c2d(m0,0.01)
maff = arx(data,[4 4 1], 'focus', {1,md.f});
mnff = n4sid(data,4, 'focus', {1,md.f})
```

The resulting Bode plots are shown in Figures 8 and 9.

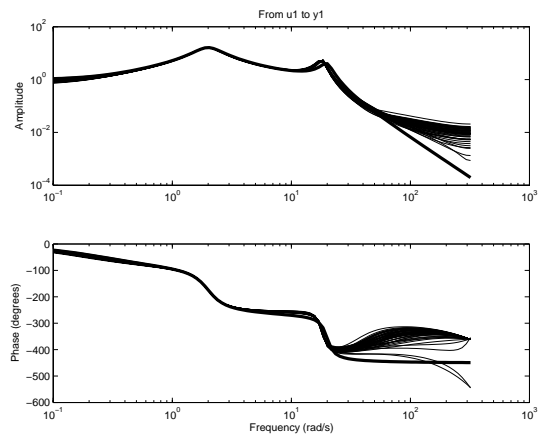


Figure 8: The Bode plots of the identified models using filtered ARX together with the true system.

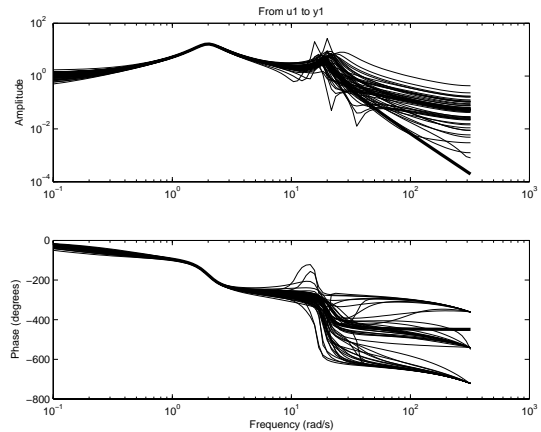


Figure 9: The Bode plots of the identified models using filtered N4SID together with the true system.

For the ARX-case, the potential benefit of the filtering is clear: It corresponds to prewhitening of the (equation error) noise, and is also the motivation for the Steiglitz-McBride method, [7]. This should also mean that it is beneficial for subspace identification methods, since they can be interpreted as model reduction of higher order ARX-models. This insight is also behind the suggested preweighting of [2], which is analyzed there from another starting point.

5 Filtering data

It seems that keeping an eye on the high frequency contents of the signals will be important to achieve good results. The key problems apparently is that the ARX-models give a very large weight to high frequencies, and thus gives starting models for the output-error type model fits that are not in the domain of attraction of a good model.

Therefore, let us prefilter the data by a low pass filter. The following models were calculated for the 50 data sets

```

mof = oe(data,[4 4 1], 'focus', [0,50]);
mpf = pem(data,4, 'focus', [0,50]);

```

The 50 models obtained by this filtering and the OE method are all shown in Figure 10, together with the true system. Performing the same test with frequency domain data (`daf = fft(data)`) gave essentially the same result. Notice that the

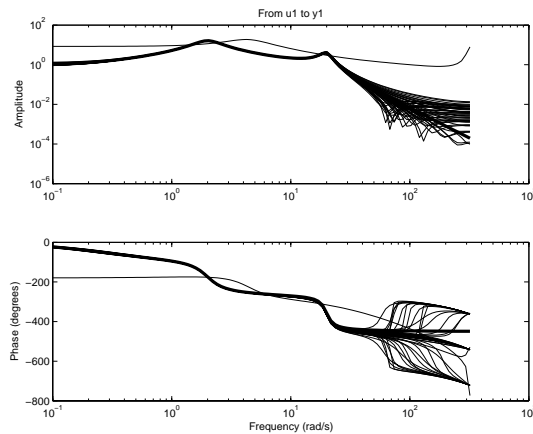


Figure 10: The Bode plots of the identified models using pre-filtered data and OE, together with the true system. There is one “outlier”, but the other 49 curves follow the true system quite well up to ca 40 rad/s.

behavior at frequencies above 30 rad/sec is quite reliable. Recall that there is no excitation if the system above the sinusoid at 22 rad/s. It is important to realize that the model is “aware” of its uncertainty at high frequencies. Figure 11 shows the uncertainty region corresponding to 3 standard deviations for one of the models.

It is worth stressing again, that the problem is just one of initialization. If we use the filtered data for just computing the initial model, and estimate the model from the original non-filtered data,

```

m_i = oe(data,[4 4 1], 'fo', [0 50], 'maxi', 0)
mo = oe(data,m_i, 'foc', 'pre', 'maxiter', 20)

```

a figure just like Figure 10 is obtained.

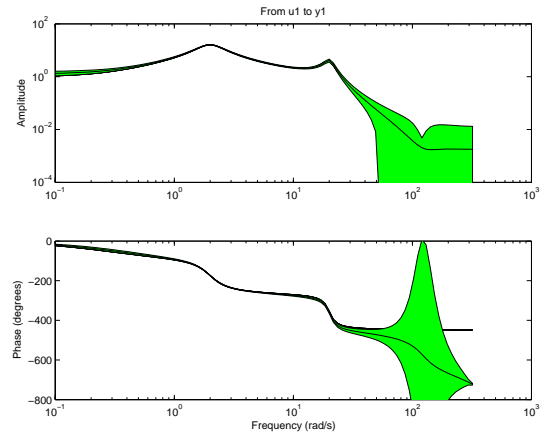


Figure 11: The Bode plot of one of the identified models using pre-filtered data, together with the true system and a 3 standard deviation uncertainty region.

6 Decimation of Data

An idea that is a variant of focus-filtering is to decimate the data after proper antialiasing filtering.

```

dd = decimate(data,10)
    % 10 times slower sampling
mod = oe(dd,[4 4 1]);
mpd = pem(dd,4);

```

Running 50 realizations of the data and plotting all 100 models in the same Bode diagram gives Figure 12. As expected, the result as good as prefiltering alone. This confirms that the fast sampling is part of the problems encountered in this example.

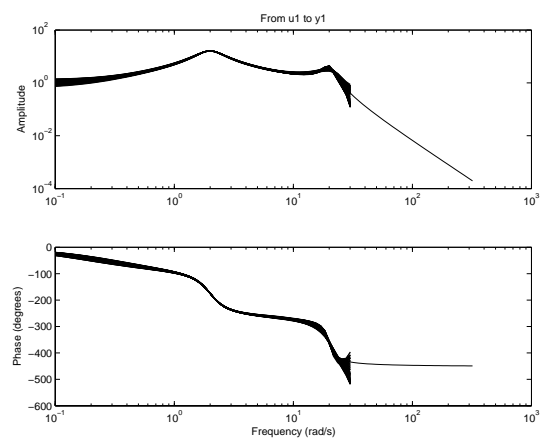


Figure 12: Bode plot of the 100 identified models using decimated data, together with the true system and.

7 Continuous-time Frequency-domain Data

Let us convert one of the estimated models to continuous time. With one standard deviation uncertainty regions it gives:

$$G(s) = B(s)/F(s)$$

$$B(s) = -(1.271 \pm 0.7081)s^3 - (3.368 \pm 13.69)s^2 - (6294 \pm 152.2)s + (1337 \pm 139.8)$$

$$F(s) = s^4 + (4.773 \pm 0.2372)s^3 + (402.6 \pm 4.886)s^2 + (410.1 \pm 9.697)s + (1596 \pm 33.39)$$

The true values are

$$B_0(s) = -6400s + 1600$$

$$F_0(s) = s^4 + 5s^3 + 408s^2 + 416s + 1600$$

We see that the relative uncertainty in the two leading coefficients in the B -polynomial (whose true values are zero) will contribute to uncertain frequency response at higher frequencies. It can be expressed saying that the data do not contain much information about these coefficients. This no doubt is related to the poor accuracy and high standard deviation at frequencies above 20 rad/s.

The true continuous system has only one zero. If we know that for a fact, this obviously has great importance for the model accuracy at high frequencies. However, this is a difficult constraint to handle in the sampled models. To use it, we could fit directly a continuous time model. Since the data are sampled fast we can treat their Fourier transforms as if they were computed from continuous time data. This gives the result according to Figure 13.

```
datafc = fft(data)
datafc.ts=0;
mof = oe(datafc,[2 4 1], 'focus', [0 25]);
```

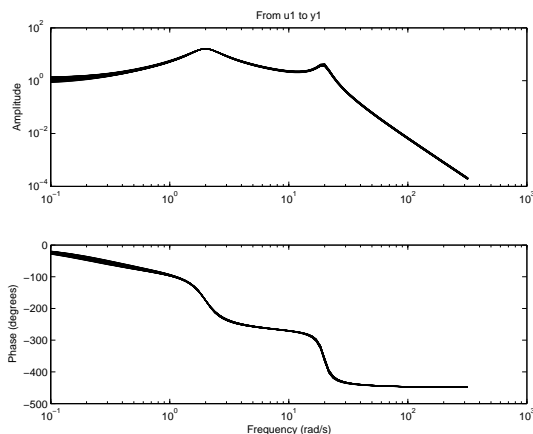


Figure 13: The Bode plots of the directly identified continuous time model together with the true system (51 curves). The Fourier transforms of the inputs and output are treated as obtained by continuous time signals.

8 Conclusions

The system used by [5] deserves special attention by people who develop discrete-time identification methods. Techniques such as CVA/subspace methods and prediction error methods may give quite bad results if not proper data prefiltering is applied. We have studied some reasons for these difficulties. We have found that ARX-models are very bias-sensitive to the system (especially with sinusoidal inputs). Figure 5 stresses that the bias is substantial, despite the good signal to noise ratio. Even though the bad conditioning of the regression matrix is part of the reason for the bias, it is not a question of numerical errors. This means that typical initialization routines based on ARX models will have problems. One should discuss various remedies, in addition to pre-filtering, for this initialization problem. See, e.g. [6].

Acknowledgments

This work was supported by the Swedish Research Council (VR). Useful comments by H. Garnier and G.P. Rao are also gratefully acknowledged.

References

- [1] H. Garnier and M. Mensler. The CONTSID toolbox: A Matlab toolbox for CONTinuous-Time System IDENTification. In *Proc. 12th IFAC Symposium on Identification*, Santa Barbara (USA), July 2000.
- [2] T. Gustafsson. Subspace-based system identification: weighting and pre-filtering of instruments. *Automatica*, 38:433–443, 2002.
- [3] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.
- [4] L. Ljung. *System Identification Toolbox for use with MATLAB. Version 6*. The MathWorks, Inc, Natick, MA, 6th edition, 2003.
- [5] G.P. Rao and H. Garnier. Numerical illustration of the relevance of direct continuous-time model identification. In *Proc. 15th IFAC Triennial World Congress*, Barcelona, July 2002.
- [6] Y. Rolain and R. Pintelon. Generating robust strating values for frequency-domain transfer function estimation. *Automatica*, 35:965–972, 1999.
- [7] K. Steiglitz and L. E. McBride. A technique for the identification of linear systems. *IEEE Trans. Autom. Control*, AC-10:461–464, 1965.
- [8] B. Wahlberg and L. Ljung. Design variables for bias distribution in transfer function estimation. *IEEE Trans. Autom. Control*, AC-31:134–144, 1986.
- [9] P. C. Young. Parameter estimation for continuous-time models - a survey. *Automatica*, 17(1):23–39, 1981.