Technical report from Automatic Control at Linköpings universitet

Experiments with Identification of Continuous Time Models

Lennart Ljung

Division of Automatic Control

E-mail: ljung@isy.liu.se

13th May 2009

Report no.: LiTH-ISY-R-2907

Accepted for publication in 15th IFAC Symposium on System Identification, Saint-Malo, France

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Abstract

Identification of time-continuous models from sampled data is a long standing topic of discussion, and many approaches have been suggested. The Maximum Likelihood method is asymptotically and theoretically superior to other methods. However, it may suffer from numerical inaccuracies at fast sampling and it also requires reliable initial parameter values. A number of efficient and useful alternatives to the maximum-likelihood method have been developed over the years. The most important of these are State-Variable filters, combined with Instrumental Variable methods, including the simplified refined IV method. In this contribution we perform unpretentious numerical experiments to comment on these methods, and their mutual benefits.

Keywords: System Identification

Experiments with Identification of Continuous Time Models

Lennart Ljung

Linköping University, Sweden, `ljung@isy.liu.se`

Abstract: Identification of time-continuous models from sampled data is a long standing topic of discussion, and many approaches have been suggested. The Maximum Likelihood method is asymptotically and theoretically superior to other methods. However, it may suffer from numerical inaccuracies at fast sampling and it also requires reliable initial parameter values. A number of efficient and useful alternatives to the maximum-likelihood method have been developed over the years. The most important of these are State-Variable filters, combined with Instrumental Variable methods, including the simplified refined IV method. In this contribution we perform unpretentious numerical experiments to comment on these methods, and their mutual benefits.

1. INTRODUCTION

The problem considered is to estimate continuous time (CT) transfer functions

$$y(t) = G(p)u(t)$$
$$G(s) = \frac{b_0 s^m + b_1 s^{m-1} + \ldots + b_m}{s^n + a_1 s^{n-1} + \ldots + a_n} \tag{1}$$

from discrete time (DT) input output data $\{u(t_k), y(t_k)\}$. In general, G will be a multiple-input-multiple-output (MIMO) transfer function.

It may be noted that multiple outputs pose no problems: each output channel can be treated as a separate problem. Multiple inputs, though, mean conceptual and algorithmic problems: It is a matter of distinguishing the contributions of each input to the output.

The input and output signals are CT functions, but sampled at discrete time instants t_k. For the problem to be well posed it is formally necessary to know the intersample behavior of the input signal, so that the continuous time input can be inferred from the sampled values. There are three typical intersample behaviors:

- **zoh** (zero order hold): The input is piecewise constant between the samples.
- **foh** (first order hold): The input is piecewise linear between the samples
- **bl** (bandlimited): The CT input is bandlimited, contains no frequencies above the Nyquist frequency, and can hence be reconstructed by the sampling theorem.

One can distinguish between two approaches to the problem:

(1) The formal approach with Maximum Likelihood calculations including sampling taking adequate account of the inter-sample behavior
(2) Various approaches based on state-variable filtering and Instrumental Variable methods.

This contribution discusses pros and cons of these approaches, and how they can complement each other. We first describe the two approaches in somewhat more detail.

2. THE MAXIMUM LIKELIHOOD METHOD

The theoretically optimal solution is to apply the maximum likelihood (ML) method. It has long been known how to do this, e.g. Mehra and Tyler (1973), Ljung (1999) and a recent discussion is given in Ljung and Wills (2008). If the disturbances on the system are Gaussian, the ML method coincides with the prediction error method (PEM).We describe it in case the additive disturbances at the output are white: Collect the transfer function parameters in the parameter vector θ. Let

$$\hat{y}(t|\theta) = G(p, \theta)u(t) \tag{2}$$

be the simulated output for a particular set of parameters, where the simulation is done taking the intersample input properties into account. Then the ML estimate is

$$\hat{\theta} = \arg\min_{\theta} \sum \|y(t_k) - \hat{y}(t_k)\|^2_{L^{-1}} \tag{3}$$

where the matrix L is the assumed covariance matrix of the additive disturbance.

Normally the minimization has to be carried out by iterative numerical search, and then a reasonable initial parameter value $\hat{\theta}_0$ is required.

If the additive output noise is not white, Kalman filter techniques should be applied. See the aforementioned references.

3. METHODS BASED ON STATE VARIABLE FILTERS

At the same time, a large number of alternative techniques have been developed, most notedly (simplified refined) Instrumental Variable methods in conjunction with state variable filters (SVF), e.g. Young and Jakeman (1980), Young (1981), Garnier et al. (2003), Young (2008), and Garnier and Wang (2008). This is not the place for a survey of such methods, but we refer to the excellent papers, just mentioned.

To fix the ideas, it is necessary with a brief description of the main issues. For simplicity we focus on SISO systems.

Let the model be given by

$$y(t) = \frac{B(s)}{A(s)}u(t) \tag{4}$$

(allowing ourselves to mix time functions with Laplace variables.)

$$\begin{aligned} &y^{(n)}(t) + a_1 y^{(n-1)}(t) + \ldots + a_n y(t) = \\ &\qquad b_1 u^{(m-1)}(t) + b_2 u^{(m-2)}(t) + \ldots + b_m u(t) \end{aligned} \tag{5}$$

We disregard for the moment any influence from a noise source. Here, $y^{(k)}(t)$ denotes the k:th derivative of $y(t)$ with respect to time. Assume $n \geq m$. If these derivatives were all reliably accessible, it would be a simple task to phrase (5) as a linear regression and compute/estimate the parameters a_k and b_k accordingly. To handle this, we low pass the whole equation by a continuous time filter $L(s)$. Let the pole excess of this filter be at least n. Then the variables

$$\begin{aligned} z_k(t) &= L(s)y^{(k)}(t) \\ w_k(t) &= L(s)u^{(k)}(t) \end{aligned} \tag{6}$$

are all well defined signals, that can be computed by proper filters.

These variables obey exactly

$$\begin{aligned} z_n(t) + a_1 z_{n-1}(t) + \ldots + a_n z_0(t) = \\ b_1 w_{m-1}(t) + \ldots + b_m w_0(t) \end{aligned} \tag{7}$$

(except for a possible transient.) Note also, that if equation error noise $e(t)$ is present in (5), it will simply appear as $L(s)e(t)$ in (7). With w and z known, the equation (7) is a perfect linear regression.

Now, we have only sampled measurements of y and u available and the question is whether we can compute w and z from those. As mentioned in the introduction, it is necessary to know the inter-sample behavior of the input, for example that is *zoh*. Then it is perfectly straightforward to compute exactly the variable $w_k(t)$ by standard software, regardless of the sampling interval. It is more cumbersome to compute $z_k(t)$ exactly, though, since the inter-sample behavior of $y(t)$ is unknown and depends on the system. If the sampling interval is small compare to the system's time constants, it may be reasonable to treat $y(t)$ as *foh*, though.

The question is how to choose $L(s)$. Common ideas are

- **SVF**: Basic State Variable Filter
$$L(s) = \left(\frac{\lambda}{s+\lambda}\right)^n \tag{8}$$
- **GPMF**: Generalized Poisson Moment Function
$$L(s) = \left(\frac{\lambda}{s+\lambda}\right)^{n+1} \tag{9}$$
- **Refined**: Refined choice of filter (the denominator of the system)
$$L(s) = \frac{1}{A(s)} \tag{10}$$

For the first two methods λ reflects the dynamics of the system, often taken as somewhat larger than the guessed bandwidth. In the third case, the denominator polynomial is of course unknown, and must be replaced by estimates, typically iteratively improved.

With any of these choices, (7) will have well defined quantities $z_k(t)$ and $w_k(t)$ at the sampling instants $t = t_j$, and there is a linear relationship between them and the parameters. If there is no noise present, the parameters can easily be solved for by the least squares method:

$$\hat{\theta} = \left(\sum_j \varphi(t_j)\varphi^T(t_j)\right)^{-1} \sum_j \varphi(t_j) z_n(t_j) \tag{11}$$

with the usual notation of φ built up from z_k and w_k. In the normal case that a disturbance is present in (7) that is not a white noise sequence, the LS method will lead to biased results. A common solution to this problem is to use the *Instrumental Variable* (IV) method, e.g. Section 7.6 in Ljung (1999). Then the noise effected outputs z_k are replaced by "instruments" $\hat{z}_k$ to form an instrument vector $\zeta(t_j)$ of the same format as $\varphi(t_j)$ giving the IV estimate:

$$\hat{\theta} = \left(\sum_j \zeta(t_j)\varphi^T(t_j)\right)^{-1} \sum_j \zeta(t_j) z_n(t_j) \tag{12}$$

Typically the instruments are formed from u analogously to z, often based on a model of the system, sometimes complemented with appropriate pre-filtering. This is not the place to go into detail with such choices, but we refer to the earlier mentioned references.

4. METHODS IN EXISTING TOOLBOXES

4.1 CONTSID and CAPTAIN

The CONTSID Toolbox, Garnier et al. (2008), and the CAPTAIN toolbox, Young (2009) are probably the best known toolboxes devoted primarily to CT model estimation from DT data. They estimate multiple input, single output, (MISO) transfer function models of the kind (1). They offer many routines, primarily of the type mentioned in Section 3.

CONTSID have among many examples the following commands, relevant for the current discussion:

- SRIVC: A routine based on the Simplified Refined IV method (10). Could be initialized by the result of IVGPMF or by a DT model obtained by SRIV.
- IVSVF: A routine based on the basic SVF method (8) with auxiliary model-based instrumental variables.
- IVGPMF: A routine based on the GPMF method (9) with auxiliary model-based instrumental variables.
- COE: A routine which appears to be a variant of the PEM/ML method (3). See Mensler (1999). Typically initialized by the result of IVGPMF.

CAPTAIN has related commands, like `rivc`, that estimates continuous time MISO models based on the refined IV approach (10).

4.2 The System Identification Toolbox: SITB

Continuous time models are supported in the MATLAB's system identification toolbox Ljung (2007) in various ways:

- Estimate a DT model and transform to continuous time by `d2c`. This is generally available, but has two disadvantages:
  - · All DT systems cannot be transformed to CT, and the mapping may be ill conditioned
  - · The number of poles and zeros cannot be individually assigned to the CT system. Typically it leads to a pole excess of 1.
- Estimate Process models of the type (`idproc`)
$$\frac{K}{1+sT}e^{-sD} \tag{13}$$
  - · This is limited to models of at most order 3.
- Define and estimate CT grey box models using `idgrey`, or structured or canonical `idss` models, like in
`m = pem(data,4,'ss','can','ts',0)`
- Directly estimate CT output error models (transfer functions) from CT frequency domain data.
  - · This is limited to frequency domain data, but a route via transfer function estimation has been possible, as in

```
g = spa(data);
g.ts=0;
m = oe(g,[nb nf])
```

The methods in the SITB are based on maximum likelihood/prediction error techniques.

4.3 Estimating CT MIMO Transfer Functions with Arbitrary Orders in the SITB

Suppose we would like to identify a MIMO transfer function

$$y(t) = G(p)u(t)$$
$$G_{ij}(s) = \frac{b_0 s^m + \ldots + b_m}{s^n + a_1 s^{n-1} + \ldots + a_n} \tag{14}$$

where G_{ij} is the transfer function from input j to output i. (We have suppressed those indices in the coefficients.) We can then define a structured state-space model for these channels (illustrated for $m = 1$, $n = 3$):

$$A = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ b_0 \\ b_1 \end{bmatrix} \quad C = [1\ 0\ 0]$$
$$As = \begin{bmatrix} \mathrm{NaN} & 1 & 0 \\ \mathrm{NaN} & 0 & 1 \\ \mathrm{NaN} & 0 & 0 \end{bmatrix} \quad Bs = \begin{bmatrix} 0 \\ \mathrm{NaN} \\ \mathrm{NaN} \end{bmatrix} \quad Cs = [1\ 0\ 0]$$

The CT state space model for this block of the transfer function is then

```
ms = idss(A,B,C,0,zeros(3,1),'As',As,'Bs',Bs,...
          'Cs',Cs,'ts',0);
```

The MIMO model is built of from these submodels by block matrices in an obvious way. With `ms` thus defined, CT MIMO models can be estimated using the ML/PEM method by

```
m = pem(data,ms);
```

They only question is how to find the initial parameter values a_i, b_i. They could be obtained from IV/SVF methods, described in Section 3. Within the SITB, they could be found by applying `d2c` to discrete time models obtained by e.g. `oe` or `n4sid`.

A very simple way is to estimate a DT `oe` model, apply `d2c` and trow away numerator parameters that exceed the desired numerator orders:

```
m = oe(data,[n n 1]);
mc = d2c(m);
mc.b(1:n-m) = zeros(1,n-m)
```

and use `mc` when constructing the state-space model as above.

The possible down side of this is that the `d2c` transformation may be ill-conditioned, and/or that throwing away numerator coefficients may not be the best way of taking care of the information.

5. SOME ASPECTS OF D2C

The transformation from discrete time to continuous time models (the Matlab command `d2c`) hovers over the problem of constructing CT models from DT data. Such estimation, as we have seen, could involve explicit use of this command, but even if it does not, the properties of this transformation are relevant: Estimating CT models from DT data is a `d2c` operation. It is well known that `d2c` need not be a unique transformation, in the sense that several different CT models may have the same DT counterpart for a certain sampling interval. Conversely, there exist DT models (e.g. with poles on the negative real axis) that do not have a CT counterpart (of the same order). All of this is reflected, mathematically, by the fact that `d2c` involves matrix logarithms which may pose several numerical problems. In this section we shall illustrate some such properties.

5.1 Sensitivity of c2d/d2c

Let us consider a simple example

```
>> m0=tf([1 2],[1 1 1])

Transfer function:
   s + 2
-----------
s^2 + s + 1
```

This system has a bandwidth of around 1.4 rad/s, and following a simple rule of thumb (Ljung (1999), Section 13.8) a suitable sampling frequency for identification is ten times the bandwidth. So good sampling rates should be around 0.5 sec. To transform to discrete time and then back with a sampling interval up to around the time constant should be safe, but not necessary with longer sampling intervals:

```
>> d2c(c2d(m0,3))

Transfer function:
   s + 2
-----------
s^2 + s + 1

>> d2c(c2d(m0,4))

Transfer function:
   s + 1.493
----------------
s^2 + s + 0.7467

>> d2c(c2d(m0,10))

Transfer function:
   s + 0.613
----------------
s^2 + s + 0.3065
```

We show the step responses of the original system and of this transformed system in Figure 1. Clearly, the true nature of the step response can not be revealed from a sampling interval of 10.

Of course, this is confirmed when CT model is estimated from data with different sampling intervals:

```
>> u=iddata([],randn(1000,1),'ts',0.1)

Time domain data set with 1000 samples.
Sampling interval: 0.1

Inputs       Unit (if specified)
   u1

>> z1=[sim(m0,u),u];;
>> tf(d2c(pe(z1,[2 2 1])))

   s + 2
-----------
 s^2 + s + 1
```

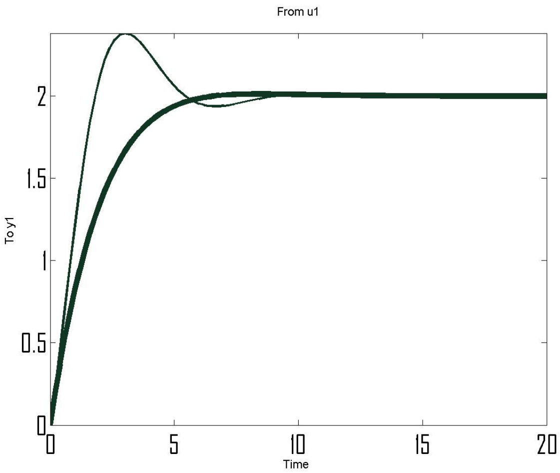

Fig. 1. Step responses of `m0` (thin line) and of `d2c(d2c(m0,10))` (thick line).

```
>> u.ts = 1;
>> z2=[sim(m0,u),u];
>> tf(d2c(oe(z2,[2 2 1])))
   s + 2
-----------
 s^2 + s + 1


>> u.ts=10;
>> z3=[sim(m0,u),u];
>> tf(d2c(oe(z3,[2 2 1]))

   s + 0.613
----------------
s^2 + s + 0.3065
```

5.2 Sensitivity of State Variable Filter (LS)

So, let's see what happens with the equation based methods in Section 3. Two simple MATLAB routines have been written for this: `lssvf(data,[nb na],lambda)` and `lsref(data,[nb na],Apol)` that implement (8) and (10), respectively, both for LS solutions.

```
>> mls = lssvf(z1,[2 2],0.1/z1.ts)
 0.9992 s + 2.002
-----------------
 s^2 + s + 1.001


>> mls = lssvf(z2,[2 2],0.1/z2.ts)
  1.063 s + 2.017
------------------
 s^2 + 1.079 s + 1.009


>> mls = lssvf(z3,[2 2],0.1/z3.ts);
   0.3065 s + 0.06406
------------------------
 s^2 + 0.3117 s + 0.03203
```

As the sampling interval increases from 0.1 to 1 to 10, the estimates deteriorate, despite the fact that the relation (7) is exact for all sampling rates. The reason must be that the filtering to z_k in (6) does not use the correct intersample behavior. Let us now try the refined svf, with the correct numerator polynomial:

```
mls = lsref(z3,[2 2],[1 1 1]);

0.9994 s + 2
--------------
 s^2 + s + 1
```

Surprisingly, this gives an (almost) correct estimate despite the long sampling interval. The reason must be that w_k are computed with the correct intersample behavior, so actually $Bw_k = y$, and $y = Az_k$, so the incorrect intersample behavior does not enter the picture. Now try the numerator from `d2c(c2d(m0,10))`:

```
>> mls = lsref(z,[2 2],[1 1 0.3065]);
   s + 0.613
----------------
s^2 + s + 0.3065
```

and we see that we obtain the "simplistic" transfer function `d2c(c2d(m0,10))`.

6. SOME CASE STUDIES

6.1 Experimental Setup

Some tests of routines for estimating CT models were set up as follows

(1) 10 randomly selected 2-input 1-output continuous time, stable systems were generated. The number of poles for each of the two transfer functions varied from 1 to 10. The number of zeros was randomly selected from 0 to the number of poles - 1 for each of the transfer functions. The generated systems are listed in the Appendix. Note that the two transfer functions have different denominators, as is usually the case for `oe` model. For the SRIVC method this is discussed, e.g. in Garnier et al. (2007).
(2) Each system was simulated with two Random Binary Signal inputs, with a mutual correlation of 0.5 between the inputs . For each system, a "natural identification sampling interval" T found as follows. The bandwidth of each of the two transfer functions were determined. (Could be quite widely apart). The sampling frequency $2\pi/T$ was determined as ten times the maximum of these. In three separate sets of experiments, it was tested to use T as sampling interval (normal sampling), $T/10$ (fast sampling) and $10T$ (slow sampling). (The same systems were used in the three sets of experiments). 1000 data were generated.
(3) To the simulated outputs were added white noise corresponding to (amplitude) signal-to-noise ratio of about 10. For each system ten data set were generated, corresponding to ten different random disturbances.
(4) For each of the 300 (10 systems, 3 sampling intervals and 10 noise realizations) data sets a CT model was estimated using the "correct" model orders. Four methods were tried: (1) SRIVC (Contsid) with default initialization (SRIV). (2) COE (Contsid) initialized with IVGPMF (with $\lambda = 0.5$/sampling interval.) (3) PEM (SITB) initialized from DT OE as described in Section 4.3 and (4) PEM initialized in the estimate from SRIVC (Method called S/PEM below). The calculations were performed in MATLAB (with the

system identification toolbox) Version R2008 and CONTSID5-0.

(5) All commands were used in their default form. The bandwidth parameter λ required fo the startup COE was chosen as half of the sampling frequency (five times the bandwidth for normal sampling). No doubt each model could have been improved by individual attention to tuning the optional parameters in the method, but that was not done.

(6) Each model was evaluated by the fit to the corresponding noise-free data (the simulated data without the additive measurement noise). The following measure was calculated:

$$\text{fit} = 100 * \left[1 - \frac{\text{norm}(\hat{y}(t) - y(t))}{\text{norm}(y(t) - \text{mean}(y(t)))}\right] \quad (15)$$

where $\hat{y}$ is the model's simulated output for the input in question. A fit of 100% thus means that the model's output coincides with the measured output. A fit of 0% means that the model does no better than guessing the output to be its mean. A fit of less than -100% is considered to be a "failure", and was replaced by -100%. The medians of the fits over the ten different noise realizations were formed and these are shown in the tables below.

6.2 Fast Sampling

Medians of the fits over the ten runs, for each system and method.(Ts = sampling interval)

#	Ts	SRIVC	COE	PEM	S/PEM
1	0.0045	99.2903	99.4246	99.4724	99.4242
2	0.0209	99.1587	99.0884	84.3475	99.1587
3	0.0045	98.2369	63.0385	90.2109	99.1446
4	0.0208	97.1395	87.7728	82.5816	98.1756
5	0.0058	97.8705	36.6399	1.3654	98.4043
6	0.1291	84.7125	90.7202	72.8593	98.4951
7	0.0607	90.9173	66.2241	41.3407	91.1278
8	0.0826	97.5910	80.0920	-63.5990	98.1465
9	0.1144	67.4782	71.5483	17.1539	67.7454
10	0.0066	77.7634	-26.4090	-100.0000	19.7961

6.3 Normal Sampling

#	Ts	SRIVC	COE	PEM	S/PEM
1	0.0451	99.3128	99.3128	99.4466	99.3128
2	0.2086	99.0851	99.0853	99.2109	99.1981
3	0.0451	99.0016	96.1341	98.9337	99.0023
4	0.2081	98.7129	98.3360	97.1051	98.7129
5	0.0576	85.8352	43.9570	81.1964	87.9092
6	1.2914	53.0971	56.4459	98.0792	97.7056
7	0.6068	97.4537	82.7733	89.5411	98.0286
8	0.8261	90.8069	87.7111	98.1496	90.9662
9	1.1444	30.8549	28.3197	80.6335	30.6140
10	0.0659	92.3024	60.6496	-37.9357	96.7195

6.4 Slow Sampling

#	Ts	SRIVC	COE	PEM	S/PEM
1	0.451	99.4058	99.4587	99.4176	99.4058
2	2.086	2.4571	3.1343	99.1803	4.3494
3	0.451	98.9098	98.3843	98.9250	98.9528
4	2.081	20.3322	15.8173	97.4486	98.3417
5	0.576	69.2093	97.5522	98.5972	98.6086
6	12.914	69.9063	61.0590	51.6331	70.2159
7	6.068	79.6647	79.7018	80.1545	79.9483
8	8.261	24.0774	41.6909	-100.0000	24.0332
9	11.444	20.9608	20.9828	-100.0000	21.3846
10	0.659	97.4926	97.7061	94.2959	96.8638

6.5 Variability

What is striking in the runs is that they show considerable variations between the different noise realizations, even though the SNR is as high as 10. We show here the ten different runs for model number 5, in the case of "normal sampling".

run #	SRIVC	COE	PEM	S/PEM
1	88.1053	20.3460	96.2996	88.1058
2	88.6402	45.8884	24.1037	98.6481
3	86.4322	57.4648	73.7396	89.0804
4	86.4986	40.9107	-100.0000	87.1612
5	52.9565	93.4966	98.2055	86.9474
6	79.0969	78.0857	-100.0000	87.6296
7	85.2383	44.6478	96.5217	96.4723
8	75.7232	35.1760	83.9547	84.9565
9	97.7383	34.7705	78.4381	98.6895
10	84.2291	43.2662	84.3672	87.7127
median	85.8352	43.9570	81.1964	87.9092
mean	82.4659	49.4053	43.5630	90.5403
std	11.8957	21.6987	78.5948	5.2426

6.6 Comments on the Results

The cases of bad performance of the PEM method are all preceded by warnings from `d2c` that it is ill-conditioned, has poles on the negative real axis or other problems. So the failure is announced in advance so to speak. (There are also cases with such warnings which eventually lead to good results). It is clear that it is quite necessary to offer to provide the PEM method with initial value estimates that are not dependent on `d2c`. The S/PEM is such a good complement to PEM.

7. DISCUSSION

The numerical experimentation shows a number of things to think about. One is that the larger examples are somewhat fragile. We had quite small amount of noise. Still, the estimates do not show a smooth behavior of "mean $\pm$ variance" character over the different realizations, (and over the different systems). Instead we see rather non-smooth effects of occasional outlier-like performance. No doubt, one reason for this are nonlinear threshold effects, like stability. It is also likely that the non-smooth transformation `d2c` plays a role here. As a result, it is not easy to compare different methods.

One might discuss what is the natural and best way to evaluate the quality of CT models. We have here chosen the common measure of the ability to reproduce output signals for the same input as used in the estimation. This is also, via Parseval, a measure for how close the model frequency function is to the true one in a quadratic norm given by the input spectrum. (Actually, since the simulation by necessity is DT, the closeness of the sampled frequency functions). The reason for building CT models could be a particular interest in certain parameters of physical interest. We have not at all considered that aspect here, but only evaluated the input-output behavior. If that is the focus, it is worth while to comment on how discrete time modeling performs. We have repeated the experiments with the same data ("normal sampling") for two DT model structures: (1) An OE model with two independent transfer functions (command `oe(data,[nb na 1 1])` in the SITB) (2) An OE State-space model (command `pem(data,2*na,'disturbancemodel,'none')`) giving the following result (no tuning of any optional variables):

#	OE	OE-SS
1	99.3905	99.1740
2	99.0295	96.5609
3	97.1118	93.6404
4	97.7562	96.6918
5	65.3220	37.2948
6	98.2538	97.9743
7	84.0135	97.6200
8	98.1353	97.5071
9	97.6030	97.5198
10	50.8909	63.9668

It is interesting to note that for some systems, the best CT-model outperforms the corresponding DT-model.

A final word of caution: the randomly generated test systems may very well have artifacts that are not so relevant for data from real-life systems.

ACKNOWLEDGMENTS

This work was supported by the Swedish Research Council VR in the Linnaeus Center CADICS, and by Swedish Foundation for Strategic Research in the center MOVIII. I am very grateful to Hugues Garnier who (1) encouraged me to write this contribution, (2) provided me with a copy of CONTSID-5.0 before its official release, (3) advised me how to best use its routines, (4) pushed me to finish the paper and (5) provided very useful comments on the text.

REFERENCES

H. Garnier and L. Wang, editors. *Identification of Continuous-time Models from Sampled Data*. Springer-Verlag, London, 2008.

H. Garnier, M. Mensler, and A. Richard. Continuous-time model identification from sampled data: Implementation issues and performance evaluation. *Int. J. Control*, 76 (13):1337–1357, 2003.

H. Garnier, M. Gilson, P. C. Young, and E. Huselstein. An optimal IV technique for identifying continuous-time transfer function model of multiple input systems. *Control Engineering Practice*, 15(4):471–486, April 2007.

H. Garnier, M. Gilson, T. Bastogne, and M. Mensler. The CONTSID toolbox: a software support for continuous-time data-based modelling. In H. Garnier and L. Wang, editors, *Identification of Continuous-time Models from Sampled Data*, pages 249 – 290. Springer-Verlag, London, 2008. (`www.iris.cran.uhp-nancy.fr/contsid`.

L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.

L. Ljung. *The System Identification Toolbox: The Manual*. The MathWorks Inc. 1st edition 1986, 7th edition 2007, Natick, MA, USA, 2007.

L. Ljung and A. Wills. Issues in sampling and estimating continuous-time models with stochastic disturbances. In *Proc IFAC World Congress*, Seoul, Korea, July 2008.

R. Mehra and J. S. Tyler. Case studies in aircraft parameter identification. In P. Eykhoff, editor, *Proc. 3rd IFAC Symposium on System Identification*, pages 117–144, The Hague, 1973. North Holland.

M. Mensler. *Analye et étude comperative de methodes d'identification de sytstèmes à représentation continue. Développment d'une boîte à outils logicielle*. PhD thesis, Université de Henri Poincaré, Nancy, France, 1999. In French.

P. C. Young. The refined instrumental variable method: unified estimation of discrete and continuous-time transfer function models. *Journal Europeen des Systemes Automatises*, 42:149–179, 2008.

P. C. Young. Parameter estimation for continuous-time models - a survey. *Automatica*, 17(1):23–39, 1981.

P. C. Young and A. J. Jakeman. Refined instrumental variable methods of time-seies analysis: Part III, extensions. *Int. J. Control*, 31:741–764, 1980.

P.C. Young. The CAPTAIN toolbox for Matlab. In *Proc. 15th IFAC Symposium on System Identification, SYSID09*, Saint-Malo, France, 2009. `www.es.lancs.ac.uk/cres/captain/tf.html`.

8. APPENDIX: THE TEST SYSTEMS

All code and data for the tests in Section 6 can be downloaded from `www.control.isy.liu.se\~ljung\cttf`. The k:th randomly generated test systems has the form

$$\begin{aligned} y(t) &= G_1^{(k)}(s)u_1(t) + G_2^{(k)}(s)u_2(t) \\ &= \frac{B_1^{(k)}(s)}{A_1^{(k)}(s)}u_1(t) + \frac{B_2^{(k)}(s)}{A_2^{(k)}(s)}u_2(t) \end{aligned}$$

with the transfer functions/polynomials given below:

$$\begin{aligned}
G_1^{(1)} &= \frac{0.3928}{s+13.95} \quad G_2^{(2)} = \frac{9.528}{s+1.2} \\
G_1^{(2)} &= \frac{-1.688s-2.764}{s^2+2.761s+1.279} \quad G_2^{(2)} = \frac{3.213s-4.325}{s^2+0.02755s+2.033} \\
G_1^{(3)} &= \frac{-3.934s^2-0.2401s-0.5596}{s^3+3.09s^2+3.113s+1.738} \quad G_2^{(3)} = \frac{-7.847s^2+1.502s+6.023}{s^3+8.894s^2+16.32s+8.404} \\
G_1^{(4)} &= \frac{-0.8471s^3+0.9081s^2-5.972s+5.519}{s^4+3.115s^3+5.036s^2+4.577s+2.254} \\
G_2^{(4)} &= \frac{-6.253s^3-4.68s^2+3.012s-3.671}{s^4+2.066s^3+2.909s^2+2.858s+1.054} \\
G_1^{(5)} &= \frac{-2.188s^4+5.463s^3-11.21s^2-5.098s+5.282}{s^5+4.158s^4+6.742s^3+6.222s^2+3.534s+0.9117} \\
G_2^{(5)} &= \frac{-2.826s^4-11.35s^3-2.465s^2-2.23s-1.233}{s^5+3.397s^4+5.907s^3+7.269s^2+4.613s+3.176} \\
G_1^{(6)} &= \frac{-4.711s^5+3.297s^4+3.752s^3-2.579s^2+6.975s-6.927}{s^6+3.31s^5+5.409s^4+6.404s^3+5.541s^2+2.845s+0.6121} \\
G_2^{(6)} &= \frac{9.201s^2-3.21s+3.712}{s^6+4.103s^5+7.976s^4+11.33s^3+11.41s^2+6.669s+1.727} \\
G_1^{(7)} &= \frac{-1.274s^2-3.452s-3.306}{s^7+3.38s^6+7.185s^5+11.67s^4+12.93s^3+8.976s^2+3.668s+0.7617} \\
G_2^{(7)} &= \frac{1.693s^5-0.9844s^4-0.9852s^3+8.129s^2+1.814s-9.339}{s^7+4.862s^6+11.22s^5+16.14s^4+15.64s^3+10.23s^2+4.217s+0.8853} \\
B_1^{(8)} &= 1.634s^5-1.508s^4-3.528s^3-5.012s^2-5.166s-5.616 \\
A_1^{(8)} &= s^8+5.42s^7+14.49s^6+25s^5+30.11s^4+25.74s^3+15.41s^2 \\
&\quad +6.101s+1.226 \\
B_2^{(8)} &= -2.568 \\
A_2^{(8)} &= s^8+22.35s^7+118.8s^6+255.9s^5+306s^4+227s^3+105.8s^2 \\
&\quad +31.66s+4.4 \\
B_1^{(9)} &= -7.318 \\
A_1^{(9)} &= s^9+5.757s^8+17.05s^7+32.96s^6+44.94s^5+44.16s^4+30.88s^3 \\
&\quad +14.42s^2+3.867s+0.3816 \\
B_2^{(9)} &= -0.5505s^8+7.26s^7-6.155s^6+1.286s^5+1.597s^4 \\
&\quad +1.638s^3+6.024s^2-9.465s+12.35 \\
A_2^{(9)} &= s^9+7.78s^8+32.61s^7+82.9s^6+138.6s^5+162.6s^4 \\
&\quad +137.5s^3+81.6s^2+30.12s+5.031 \\
B_1^{(10)}) &= -10.85s^9-0.6696s^8+0.3589s^7+4.749s^6-5.438s^5 \\
&\quad -0.889s^4+5.853s^3+1.001s^2+3.558s+4.975 \\
A_1^{(10)} &= s^{10}+6.953s^9+25.92s^8+64.27s^7+113.1s^6+145.5s^5 \\
&\quad +137.3s^4+93.31s^3+43.61s^2+12.63s+1.718 \\
B_2^{(10)} &= 2.335s^8-2.546s^7+4.388s^6-7.821s^5-3.515s^4-0.2918s^3 \\
&\quad +4.979s^2-13s+0.03923 \\
A_2^{(10)} &= s^10+6.43s^9+21.13s^8+46s^7+72.56s^6+86.15s^5 \\
&\quad +77.47s^4+51.8s^3+24.48s^2+7.384s+1.091
\end{aligned}$$

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2009-05-13

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________________

Rapporttyp
Report category

☐ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☒ Övrig rapport
☐ ________________

ISBN
—

ISRN
—

Serietitel och serienummer
Title of series, numbering

ISSN
1400-3902

LiTH-ISY-R-2907

URL för elektronisk version

`http://www.control.isy.liu.se`

Titel
Title

Experiments with Identification of Continuous Time Models

Författare
Author

Lennart Ljung

Sammanfattning
Abstract

Identification of time-continuous models from sampled data is a long standing topic of discussion, and many approaches have been suggested. The Maximum Likelihood method is asymptotically and theoretically superior to other methods. However, it may suffer from numerical inaccuracies at fast sampling and it also requires reliable initial parameter values. A number of efficient and useful alternatives to the maximum-likelihood method have been developed over the years. The most important of these are State-Variable filters, combined with Instrumental Variable methods, including the simplified refined IV method. In this contribution we perform unpretentious numerical experiments to comment on these methods, and their mutual benefits.

Nyckelord
Keywords

System Identification