Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

State of the Art in Linear System Identification: Time and Frequency Domain Methods

Lennart Ljung

Abstract— **Identification of time-invariant linear dynamic systems is a mature subject. In this contribution we focus on the interplay between methods that use time and frequency domain data, respectively. The frequency domain data could be either input/output Fourier transforms or frequency functions. We explain how these different kinds of data types are used to fit models, and how closely related the methods are. Of special interest is how transients (initial conditions and deviations from periodic signals) are handled. Direct estimation of time-continuous models is also discussed, as well as software aspects.**

I. Introduction

A linear dynamic system in discrete or continuous time can be described by

$$y(t) = G(\sigma)u(t) + v(t) \tag{1}$$

where σ is the differentiation operator p in continuous time and the shift operator q in discrete time. The identification problem is to find the transfer operator G and possibly also the spectrum of the additive noise v. There is an extensive literature on this problem, see among many books, e.g. [4], [15], [10], and [14].

Basically, the approach to find such transfer functions contains four steps:

1) Collect observed data from the actual system.
2) Select a class of models.
3) Select that member in the model class that describes the observations "best" (according to some criterion).
4) Validate the resulting model to see if it will be "good enough" for its purpose. If necessary, go back to earlier steps and revise some choices.

For linear systems and models the interplay between time and frequency domain methods is well known and much utilised in, for example, control design. Indeed, this duality is a cornerstone in many design and analysis methods.

For linear system identification the links between time- and frequency domain methods are equally important. However, the tools have traditionally not been quite integrated. Of course, methods to directly estimate frequency responses from time domain data, through various spectral analysis techniques are classical. They belong to the standard kit of tools since the 1960's. At the same time one can distinguish one "community" (mostly control people) that basically works with data in the time domain and primarily estimates parametric time domain models (state-space and denominator/numerator transfer function models) and occasionally complements that with spectral analysis. Another community (mostly "instrumentation and measurement" people) uses frequency domain data, periodic inputs and well controlled experiments to build models of similar kind (transfer functions) as well as refined frequency function estimates. Frequency analysers are often used to collect and compress data. Vibration and modal analysis are common applications of this type.

Over a period of time, there was not so much contact between these communities. For example, the fact that the input not necessarily is periodic was perceived as an obstacle to use frequency domain techniques.

Recently, the true duality between time- and frequency domain methods have become clear. Estimating "initial conditions" in the frequency domain can fully compensate for non-periodic data, and so called subspace methods originally developed for time domain data can also be applied to frequency domain data. The importance and implications of various inter-sample properties (like zero order hold or band-limited) of the input has also been clarified. The relative merits of periodic and non-periodic data have been studied carefully. See for example [4] and [10] for comprehensive treatments.

In the latest version of MATLAB's SYSTEM IDENTIFICATION TOOLBOX, [6], this clarified duality has been used to create an estimation and validation framework that is fully symmetric with respect to the time and frequency domains.

II. A Basic Problem: Curve Fitting

The topic of model and parameter estimations is a very rich subject with many approaches, theories and techniques. It may be a good idea to go back to the roots and consider for a while "Gauss's perspective" of the problem in terms of curve fitting.

To bring out the basic features, let us study a simple example. Suppose the problem is to estimate an unknown function $g_0(x), -1 \le x \le 1$. The observations we have are noisy measurements $y(k)$ at points x_k which we may or may not choose ourselves:

$$y(k) = g_0(x_k) + e(k) \tag{2}$$

The disturbance e is typically described as a zero-mean stochastic variable with variance

$$Ee^2(k) = \lambda_k \tag{3}$$

How to approach this problem?

This work was supported by the Swedish Research Council (VR)

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

0-7803-8335-4/04/$17.00 ©2004 AACC

A. Parametric Methods

Global Parameterisations: One way or another, we must decide "where to look for" g. We could, for example, have the information that g is a third order polynomial. This would lead to the – in this case – *grey box* model structure

$$g(x,\theta) = \theta_1 + \theta_2 x + \theta_3 x^2 + \ldots + \theta_n x^{n-1} \tag{4}$$

with $n = 4$, and we would estimate the parameter vector θ from the observations y, using for example the classical least squares method.

Now suppose that we have no structural information at all about g. We would then still have to assume something about it, e.g. it is an analytical function, or that it is piecewise constant or something like that. In this situation, we could still use (4), but now as *black-box* model: if we assume g to be analytic we know that it can be approximated arbitrarily well by a polynomial. The necessary order n would not be known, and we would have to find a good value of it using some suitable scheme.

Note that there are several alternatives in this black-box situation: We could use rational approximations:

$$g(x,\theta) = \frac{\theta_1 + \theta_2 x + \theta_3 x^2 + \ldots + \theta_n x^{n-1}}{1 + \theta_{n+1} x + \theta_{n+2} x^2 + \ldots + \theta_{n+m-1} x^{m-1}} \tag{5}$$

or Fourier series expansions

$$g(x,\theta) = \theta_0 + \sum_{\ell=1}^{n} \theta_{2\ell-1}\cos(\ell\pi x) + \theta_{2\ell}\sin(\ell\pi x) \tag{6}$$

These are only a few examples of how the model class can be parametrised in a global way.

Local Parameterisations: Alternatively, we could approximate the function by expanding it into function expansions with basis functions with local support:

$$g(x) = \sum_{n=1}^{d} \theta_k g_k(x) \tag{7a}$$

Here, $g_k(x)$ could be a local function around a certain center point γ_k so that

$$g_k(x) = 0 \text{ if } |x - \gamma_k| > \beta_k \tag{7b}$$

Paramaterizations like these are the basis for neural network models, e.g. [4], Chapter 5.

B. Estimation Techniques and Basic Properties

It suggests itself that the basic least-squares method is a natural approach for curve fitting:

$$\begin{aligned}\hat{\theta}_N &= \arg\min_{\theta} V_N(\theta) \\ V_N(\theta) &= \sum_{k=1}^{N} \mu_k (y(k) - g(x_k,\theta))^2\end{aligned} \tag{8}$$

Here μ_k is a weight that in a suitable way reflects the

- "reliability" of the measurement k. This is typically evaluated as the variance of $e(k)$, so we would have $\mu_k \sim 1/\lambda_k$, where λ_k is the variance of $e(k)$.
- "relevance" of the measurement k. It could be that we do not fully believe that the underlying model $g(x,\theta)$ is capable of describing the data for all x. We could then down-weigh a measurement at a point x_k outside a region of prime relevance for the model.

In case y and g_0 are vector-valued (column vectors), the criterion takes the form

$$V_N(\theta) = \sum_{k=1}^{N} (y(k) - g(x_k,\theta))^T \Lambda_k^{-1} (y(k) - g(x_k,\theta)) \tag{9}$$

where the matrix Λ_k takes care of the weightings. For the reliability aspect, Λ_k would be the covariance matrix of $e(k)$.

It may be that a suitable weighting is not known off-hand. An idea would be to let also the weights μ_k depend on the parameter θ:

$$V_N(\theta) = \sum_{k=1}^{N} \mu_k(\theta)(y(k) - g(x_k,\theta))^2$$

A moment's reflection shows that this will not work. Minimizing this criterion would be achieved by making the μ_k "very small", without necessarily giving priority to fit $g(x_k)$ to $y(k)$. The size of the weight must be counterbalanced in some way. A clue to how this could be done is obtained by the *Maximum Likelihood* method. Assume that the data are generated according to

$$y(k) = g(x_k,\theta) + e(k) \tag{10}$$

where $e(k)$ in (2) is Gaussian with zero mean and (unknown) variance $\lambda_k(\theta)$. Then $y(k)$ has the distribution (probability density function, pdf)

$$\frac{1}{2\pi\sqrt{\lambda_k(\theta)}} e^{-(y(k)-g(x_k,\theta))^2/(2\lambda_k(\theta))} \tag{11}$$

If the noise terms $e(k)$ are independent at different k, the joint pdf for the observation vector $Y^N = \{y(1),\ldots,y(N)\}$ is obtained by taking the product of the terms (11). The *Likelihood function* is the value of this pdf with the observed values $y(k)$ inserted. It is thus, for given observations, a function of θ. The maximum likelihood estimate of θ is the value that maximizes this function. Alternatively we can take the logarithm of the likelihood function, change sign, multiply by 2 and minimise the result:

$$\sum_{k=1}^{N} \left[(y(k) - g(x_k,\theta))^2/\lambda_k(\theta) + \log\lambda_k(\theta)\right] + 2N\log 2\pi \tag{12}$$

The suggestion is that in case a weighted least squares criterion is used with adjustable weights, it is a good idea to balance the criterion by

$$V_N(\theta) = \sum_{k=1}^{N} \mu_k(\theta)(y(k) - g(x_k, \theta))^2 - \log \mu_k(\theta) \quad (13)$$

A further comment that will be useful later: Suppose that the measurements are scaled as in

$$y(k) = \alpha_k g_0(x_k) + e(k), \quad Ee^2(k) = \lambda_k \quad (14)$$

with a known scaling α_k. We could then interpret $z(k) = y(k)/\alpha_k$ as a measurement of the "curve-values"

$$z(k) = g_0(x_k) + v(k), \quad Ev^2(k) = \lambda_k/\alpha_k^2 \quad (15)$$

Applying the Least squares criterion to (15) gives

$$V_N(\theta) = \sum_{k=1}^{N} \mu_k(z(k) - g(x_k, \theta))^2$$

If the weights are chosen as the inverse of the noise variances, we get

$$\begin{aligned} V_N(\theta) = & \sum_{k=1}^{N} \frac{\alpha_k^2}{\lambda_k}(z(k) - g(x_k, \theta)^2 = \\ & \sum_{k=1}^{N}(\alpha_k z(k) - \alpha_k g(x_k, \theta))^2/\lambda_k = \\ & \sum_{k=1}^{N}(y(k) - \alpha_k g((x_k, \theta))^2/\lambda_k \quad (16) \end{aligned}$$

which, of course, is the corresponding weighted least squares criterion applied to (14). This is very natural, but it captures a principle, that we will apply later on.

C. Local Methods and Local Smoothing

A simple idea to form an estimate of the function value $g(x)$ at a point x is to form some kind of average of the observations $y(k)$ corresponding to x_k in the neighbourhood of x:

$$\hat{g}(x) = \sum_{k=1}^{N} C(x, x_k)y(k) \quad (17)$$

where the weights C are chosen appropriately, and typically being zero when the distance between x and x_k is larger than a certain value ("the bandwidth"). The choice of such weights is the subject of an extensive literature in statistics. See, among many references, e.g. [2], [16], [1], and [9].

The aspects of the weights are the same as on the weights in (8), relevance and reliability. The relevance is typically related to the distance $x - x_k$ as mentioned above, while the reliability should reflect the variance of the additive noise in (2). This suggests weights of the kind

$$C(x, x_k) = \tilde{c}(x - x_k, x)/\lambda_k \quad (18)$$

normalised so that

$$\sum_{k=1}^{N} C(x, x_k) = 1 \quad \forall x \quad (19)$$

Here λ_k is the variance of the measurement $y(k)$. The function $\tilde{c}(\xi, x)$ is such that it is zero for $|\xi| \leq B(x)$, where the bandwidth $B(x)$ very well could depend on the "target point" x. See also [12] for a recent discussion on the choice of weights. One many also note that smoothing the observations in this way has a close relationship with adjusting locally parameterised models as in (7).

III. Linear Dynamic Models

A linear time-invariant system with input u and output y can always be described by

$$y(t) = G_0(p)u(t) + v(t) \quad (20)$$

where G is the transfer function, p the differentiation operator and $v(t)$ additive noise. Let the spectrum of v be denoted by $\Phi_v(\omega)$. All the following discussion is applicable to systems with multiple inputs and multiple outputs (MIMO), but in the notation we mostly will assume scalar inputs and outputs (SISO).

A discrete time system can be described by the same expression, except that the differentiation operator p is replaced by the shift operator q.

A. Model Parameterisations

A model of a linear dynamic system is, one way or another, a parameterisation of its transfer function:

$G(s, \theta)$, s is the Laplace variable in continuous time (21a)

$G(z, \theta)$, z is the Z-transform variable in discrete time (21b)

Alternatively, we can consider parameterisations of the frequency function by replacing s by $i\omega$ and z by $e^{i\omega T}$ (T being the sampling interval.)

The model can possibly come together with a parameterisation of the additive noise spectrum

$$\Phi_v(\omega, \theta) = \lambda |H(i\omega, \theta)|^2 \quad (22)$$

where the second step shows the spectrum factorised using a monic, stable and inversely stable transfer function H.

Comparing to the simple curve fitting perspective of Section II, the parameterised frequency function $G(i\omega, \theta)$ plays the role of the function $g(x, \theta)$. The spectrum $\Phi_v(\omega, \theta)$ will relate to the weighting coefficients, μ_k, λ_k, as will become clear later.

The actual parameterisation can be done in many different ways. The underlying description could be a discrete time ARMAX model

$$A(q)y(t) = B(q)u(t) + C(q)e(t)$$

with the coefficients of the polynomials (in q^{-1}) A, B and C comprising θ. This gives

$$G(e^{i\omega}, \theta) = \frac{B(e^{i\omega})}{A(e^{i\omega})}$$
$$H(e^{i\omega}, \theta) = \frac{C(e^{i\omega})}{A(e^{i\omega})}$$

Note the similarity with the basic forms (4) and (5) for $x = e^{-i\omega}$.

A physically parameterised, continuous time state space model

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) + w(t); \\ &\quad \mathrm{E}w(t)w^T(s) = Q(\theta)\delta(t-s) \\ y(t) &= C(\theta)x(t) + D(\theta)u(t) + e(t); \\ &\quad \mathrm{E}e(t)e^T(s) = R(\theta)\delta(t-s) \end{aligned}$$

corresponds to

$$\begin{aligned} G(i\omega, \theta) &= C(\theta)(i\omega I - A(\theta))^{-1}B(\theta) + D(\theta) \\ H(i\omega, \theta) &= C(\theta)(i\omega I - A(\theta))^{-1}K(\theta) + I \\ \Phi_v(\omega.\theta) &= H^*(i\omega, \theta)\Lambda(\theta)H(i\omega, \theta) \end{aligned}$$

where $K(\theta)$ and $\Lambda(\theta)$ are computed from A, C, Q and R as the steady state Kalman filter's gain and innovations variance.

Simple process models are obtained by parameterisations of the kind

$$G(i\omega, \theta) = \frac{K}{1 + sT_1} e^{i\omega\tau}, \quad \theta = \{K, T_1, \tau\} \tag{23}$$

with static gain, dominating time constant and time delay as the parameters.

Many other types of parameterisations are of course possible.

Note that these model parameterisation issues are entirely independent of the domain in which the "curve-fitting" will be made.

IV. The Data

A. Data Formats

Identification of linear time invariant models can be seen as fitting the parameterised frequency function to data.

What are then the data? Several cases can be distinguished:

a) Sampled time domain input-output data: The data are obtained by sampling the input and output with the (constant) sampling interval T:

$$Z^N = \{u(T), y(T), u(2T), y(2T), \dots, u(NT), y(NT)\} \tag{24}$$

This is no doubt the most common situation, and also the information source for several of the other data formats, as we shall discuss in Section V. While the data collection in practice always happens in discrete time, it may in general not necessarily be based on a constant sampling rate.

b) Frequency domain data from continuous measurements:

$$Z^N = \{U_N(i\omega_1), Y_N(i\omega_1), \dots, U_N(i\omega_N), Y_N(i\omega_N)\} \tag{25}$$

where $U(i\omega)$ and $Y(i\omega)$ are the (continuous time) Fourier transforms of the input and the output:

$$U_N(i\omega) = \int_0^\infty u(t)e^{-i\omega t}dt \tag{26}$$

Clearly, in practice we neither have infinite nor continuous time data records. Still, the Fourier transform above could be evaluated under some circumstances. See Section V-A below.

c) Frequency domain data from sampled measurements:

$$\begin{aligned} Z^N = \{&U_N(e^{i\omega_1 T}), Y_N(e^{i\omega_1 T}), \dots, \\ &\dots, U_N(e^{i\omega_N T}), Y_N(e^{i\omega_N T})\} \end{aligned} \tag{27}$$

where U_N and Y_N are the discrete time Fourier transforms of sampled inputs and outputs:

$$U_N(e^{i\omega T}) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} u(kT)e^{-i\omega kT} \tag{28}$$

If these transforms are computed on the "DFT-grid"

$$\omega_k = 2\pi k/(NT), k = 0, 1, \dots, N-1 \tag{29}$$

the data (27) will become the DFT (Discrete Fourier Transform) of the time domain data (24) (with a special normalisation).

It is interesting to note (see, e.g. Theorem 14.25 in [10]) that under weak assumptions $U_N(\omega)$ and $Y_N(\omega)$ will have an asymptotically (as $N \to \infty$) normal distribution. The values will also be (asymptotically) independent at different values of ω_k on the DFT grid.

d) Measurements of the continuous time frequency response function: The observations from the system could also be delivered as

$$Z^N = \{G_m(i\omega_1), \dots, G_m(i\omega_N)\} \tag{30a}$$

where $G_m(i\omega)$ is some measurement of the system's continuous time frequency function $G_0(i\omega)$. Possibly, these data could be complemented with some uncertainty estimate W of the measurements:

$$Z_U^N = \{W(i\omega_1), \dots, W(i\omega_N)\} \tag{30b}$$

e) Measurements of the sampled frequency response function: This is quite analogous to the continuous time case, but we deal with measurements of a sampled-data frequency function $G(e^{i\omega T})$:

$$Z^N = \{G_m(e^{i\omega_1 T}), \dots, G_m(e^{i\omega_N T})\} \tag{31a}$$
$$Z_U^N = \{W(i\omega_1), \dots, W(i\omega_N)\} \tag{31b}$$

It is worth while commenting on the difference between (30) and (31): A physical interpretation of the frequency functions $G(i\omega)$ and $G(e^{i\omega T})$ is in both cases that they describe the amplitude change and the phase shift of an input sinusoid $\cos(\omega t)$ or $\cos(\omega kT)$. The actual response of the system to this input will depend on the intersample behaviour of the cosine. If it indeed is applied as a continuous time signal, but only recorded at the sampling instants, it is still $G(i\omega)$ that describes the output signal. If, on the other hand, the input follows a cosine at the sampling instants, but is piecewise constant or piecewise linear in between, the output will still be a sinusoid of the same frequency at the sampling instants. However, it will not be $G(i\omega)$ that describes the amplitude change and the phase shifts, but another frequency function, which we denote by $G(e^{i\omega T})$. The exact relationship between $G(i\omega)$ and $G(e^{i\omega T})$ will depend on the intersample behaviour of the input. Clearly, if T is small compared to the time constants of the system, the difference will be small for interesting frequencies ω.

V. Where do the Data Come From?

Each of the data formats (24), (25), (27), (30) and (31) could be the starting point for model estimation. In this section we shall discuss how these data sets may be obtained from primary measurements.

The time domain data (24) is quite straightforward to obtain. The same is true for the Fourier transforms (27) - (28).

A. Frequency domain data from continuous measurement:

To get correct values for (25)-(26) in practice, it is required that the whole continuous signal $u(t),\ -\infty < t < \infty$ can be reconstructed from the sampled, observed data $\{u(T), \ldots, u(NT)\}$. This is the case if u is *periodic* and *bandlimited.* (In the periodic case, the Fourier transform values $U_N(i\omega_k)$ will have to be interpreted as the Fourier coefficients corresponding to the fundamental frequency and its harmonics, resulting from applying (26) to one period of data only.) That is to say, that the results of the simple DFT transformation (28)+(29) can be interpreted as the Fourier coefficients of the infinite signal obtained by making $u(t)$ periodic with period N, and making it time continuous by trigonometric interpolation (no power above the Nyquist frequency).

Even if the input u is periodic and bandlimited, the output need not be that. However, y is the sum of Gu and v and *the component of y that originates from u is periodic and bandlimited.* Since the estimate of G only should depend on this component, frequencies in y higher than the highest frequency in u could be removed. Then the output could be averaged over the periods to get accurate values of the relevant components of $Y_N(i\omega_k)$.

Another, more pragmatic way is when the sampling interval T is small compared to the interesting dynamics of the system and compared to the rate of change in u. Then the input could be considered to bandlimited in practice, and the values from (28) (after suitable normalisation) will be good estimates of $U_N(i\omega)$.

B. Frequency response function estimates.

The frequency function estimates G_m in (30a) or (31) can be directly measured by certain hardware equipment, *frequency analysers*. Such an equipment could implement Fourier analysis as in (32) below, or could rely upon the definition of frequency responses by directly measuring phase and amplitude shifts for a number of different sinusioldal inputs.

The frequency responses can also be estimated/constructed from measured data either in the time or the frequency domain. This is the topic of *Spectral Analysis*, which is further dealt with in Section VII. Let us comment on the simplest case of spectral analysis, vis. the *Empirical Transfer Function Estimate, ETFE.* It is formed as the ratio of the output and input Fourier transforms

$$\hat{\hat{G}}_N(i\omega) = \frac{Y_N(i\omega)}{U_N(i\omega)} \tag{32}$$

and correspondingly in the discrete time case

$$\hat{\hat{G}}_N(e^{i\omega T}) = \frac{Y_N(e^{i\omega T})}{U_N(e^{i\omega T})} \tag{33}$$

In case the observations y and u have been obtained from a noise-corrupted linear system with frequency function $G_0(i\omega)$ it can be shown that the ETFE has the following statistical properties: (Lemma 6.1 in [4].)

$$\mathrm{E}\hat{\hat{G}}_N(i\omega) = G_0(i\omega) + \frac{\rho_1}{\sqrt{N}U_N(i\omega)} \tag{34a}$$

$$\mathrm{E}|\hat{\hat{G}}_N(i\omega) - G_0(i\omega)|^2 = \frac{\Phi_v(\omega)}{|U_N(i\omega)|^2} + \frac{\rho_2}{N|U_N(i\omega)|^2} \tag{34b}$$

Here $\Phi_v(\omega)$ is the spectrum of the additive noise (at the output of the system) and ρ_i are constant bounds that depend on the impulse response of the system, the bound on the input, and the covariance function of the noise.

All this means that we can think of the ETFE as a "noisy measurement" of the frequency function:

$$\hat{\hat{G}}_N(i\omega_k) = G_0(i\omega_k) + v_k \tag{35}$$

with v_k being a zero mean random variable with variance $\Phi_v(\omega_k)/|U_N(\omega_k)|^2$. We have then ignored the terms with ρ in the expressions above. Note that the variance of v_k would correspond to the uncertainty estimate $W(i\omega_k)$ in (30b).

Something must also be said about the frequency grid in (35): If the Fourier transforms are obtained by DFT of equidistantly sampled data, the natural frequencies to use in (35) are the DFT grid:

$$\omega_k = 2k\pi/(NT); \qquad k = 0, \ldots, N-1 \tag{36}$$

This gives two advantages:

- Frequencies in between these grid points carry no extra information: they are merely (trigonometric) interpolations of the values on the DFT grid. This also determines the maximum frequency resolution of the frequency function.

- v_k are (asymptotically) uncorrelated on this grid.

In the case of p outputs, v_k is a column vector and Φ_v is a $p \times p$ matrix.

The expressions (32)-(33) assume that u is a scalar (single input system). Formulas for multi-input systems are given in [5].

C. Data Compression

An advantage with frequency domain data is that different frequency grids can be chosen, and the selection of frequencies can be matched to regions of prime interest.

In this context one should note that there is a fundamental difference between frequency domain data as input and output Fourier transforms on the one hand, and frequency response data on the other.

Frequency response function data could be rather "noisy" estimates of a function that is known to be smooth. It is therefore possible to compress the information in frequency function data by smoothing the function and retain it in fewer points. The techniques for that follow equation (49) below. Often a logarithmic frequency grid is chosen in this context, since that makes the frequency points equidistant in a Bode plot.

However, a similar data compression cannot be applied to the input and output Fourier transform values in (27) without serious loss of information. Simply put, the phase information in $Y(i\omega_k)$ and $U(i\omega_k)$ (which carry information about the system's phase at ω_k) would be lost by such smoothing.

VI. Fitting the Model to Data

A. Time Domain Data

The discrete time domain version of the model (1), (22) is

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) \tag{37}$$

For equidistantly sampled data we can form the prediction errors for this model as

$$\varepsilon(t,\theta) = H^{-1}(q,\theta)(y(t) - G(q,\theta)u(t)) \tag{38}$$

Remark: Normally $G(q)$ and $H^{-1}(q,\theta)$ have infinite impulse responses, so the above expression will require all past inputs and outputs $y(\tau),\ u(\tau),\ \tau < t$ to compute $\varepsilon(t)$. Normally, the values prior to time $t = 1$ are not known. Often these values are then set to zero, at the expense of creating a *transient* or *initial state* error. We shall discuss in Section VIII how to deal with this problem.

A natural approach to estimate θ will be to minimise the size of these prediction errors:

$$V_N(\theta, Z^N) = \sum_{k=1}^{N} \varepsilon^2(t,\theta) \tag{39}$$

In the case of multiple outputs, where the prediction errors have covariance matrix Λ, a natural criterion to minimise is

$$V(\theta, Z^N) = \sum_{k=1}^{N} \varepsilon^T(t,\theta)\Lambda^{-1}\varepsilon(t,\theta) \tag{40}$$

which equals the maximum likelihood criterion if Λ is known and e is Gaussian. If Λ is unknown we can still use the same criterion, but balance it with the additional term like in (12):

$$V(\theta, Z^N) = \sum_{k=1}^{N} \varepsilon^T(t,\theta)\Lambda^{-1}\varepsilon(t,\theta) + \log\det\Lambda \tag{41}$$

B. Frequency Domain Data

Consider now the case that we are given Fourier transform values of the inputs and the outputs as in (25) and (27). The relationship between these values is obtained from (20):

$$\begin{aligned} Y_N(e^{i\omega T}) &= G_0(e^{i\omega T})U_N(e^{i\omega T}) + V_N(e^{i\omega T}) \\ E|V_N(e^{i\omega T})|^2 &= \Phi_v(\omega) \end{aligned} \tag{42}$$

where V is the transform of the noise, corresponding to (28). The relationship is not exact, only approximate, since there are transients and deviations due to the fact that the data may not be periodic. For the moment we disregard the transient term, which is dealt with in detail in Section VIII.

The relation (42) is like (14) (with $\alpha = U_N$) and a direct application of the weighted least squares criterion (16) gives

$$V_N(\theta, Z^N) = \sum_{k=1}^{N} |Y_N(e^{i\omega_k T}) - G(e^{i\omega_k T},\theta)U_N(e^{i\omega_k T})|^2/\Phi_v(\omega_k) \tag{43}$$

In case the spectrum for v is not given, but parameterised as in (22), we get a parameterised weighting, and should balance that as in (13):

$$\begin{aligned} V_N(\theta, Z^N) = {} & \sum_{k=1}^{N} |Y_N^k - G(e^{i\omega_k T},\theta)U_N^k|^2/(\lambda|H(e^{i\omega_k T},\theta)|^2) + \\ & \sum_{k=1}^{N} \log\lambda|H(e^{i\omega_k T},\theta)|^2 \end{aligned} \tag{44}$$

where, for short, $Y_N^k = Y(e^{i\omega_k T})$. Indeed, as in (12), this will be the true log likelihood function in case $V_N(e^{i\omega_k T})$ are Gaussian distributed and independent for different k.

Remark: It may be noted that

$$\int_{-\pi}^{\pi} \log|H(e^{i\omega})| = 0 \tag{45}$$

for any monic, stable and inversely stable transfer function H. This means that the last sum in (44) is almost θ-idependent for large N and equidistant frequency points.

Analogously, if we have continuous time Fourier transforms of input and output given, as in (25), the continuous time frequency function could be fit directly:

$$V_N(\theta, Z^N) = \sum_{k=1}^{N} |Y_N(i\omega_k) - G(i\omega_k, \theta) U_N(i\omega_k)|^2 / \Phi_v(\omega_k) \qquad (46)$$

In this expression we assumed $\Phi_v(\omega)$ to be known (or estimated separately), but the extension the a parameterised noise spectrum would be analogous to (44).

C. Frequency Response Data

Suppose now that the data is given in terms of measured frequency response function values, (30a) or (31). A clear cut curve fitting approach to estimating the model would be to form the analog of (8):

$$V_N(\theta, Z^N) = \sum_{k=1}^{N} |G_m(i\omega_k) - G(i\omega_k, \theta)|^2 / W(i\omega_k) \quad (47)$$

where we used the uncertainty measure in (30b) for the weights. The corresponding expression holds for the discrete time case.

In the case that the frequency function estimate is an ETFE as in (32) we would use the uncertainty measure (34b). This gives

$$\begin{aligned} &|G_m(i\omega) - G(i\omega, \theta)|^2 / W(i\omega) = \\ &|\frac{Y_N(i\omega)}{U_N(i\omega)} - G(i\omega, \theta)|^2 \frac{|U_N(i\omega)|^2}{\Phi_v(\omega)} = \\ &|Y_N(i\omega) - G(i\omega) U_N(i\omega)|^2 / \Phi_v(\omega) \end{aligned}$$

This means that for these frequency function estimates, the criterion (47) *exactly coincides with (46).*

D. Connections between time and frequency domains

Let us show the relationship between frequency domain fit (43) and the time domain fit (39) using Parseval's relationship. The Fourier transform (28) of the prediction error (38) is (neglecting transients or assuming periodic data):

$$\begin{aligned} E_N(e^{i\omega T}, \theta) = &H^{-1}(e^{i\omega T}, \theta)(Y_N(e^{i\omega T}) - \\ &G(e^{i\omega T}, \theta) U_N(e^{i\omega T})) \end{aligned}$$

Applying Parseval's relationship to (39), (38) and ignoring transient effects (or assuming periodic data) now gives for this criterion

$$\begin{aligned} V(\theta, Z^N) = &\sum_{k=1}^{N} |Y_N(e^{i\omega_k T}) - G(e^{i\omega_k T}, \theta) U_N(e^{i\omega_k T})|^2 \\ &/ |H(e^{i\omega_k T}, \theta)|^2 \end{aligned} \qquad (48)$$

Dividing this expression by λ and using (22) we see that this expression is exactly equal to (43). Consequently, also the time domain expression (39) can be interpreted as curve fitting the parameterised model to the ETFE. We have also displayed the nature of the noise model in (37): It just provides the weighting in this fit.

VII. Estimation of Frequency Functions

The criteria of the kind (47) where a parameterised set of frequency functions is fitted to a measured frequency response G_m can be seen as a way to smooth the measured response. Another way would be to locally smooth the measured response as in (17) – (19).

If measurements of the frequency function along with uncertainty information is given as in (31) and we follow the basic idea (18), the following weighting is obtained:

$$C(\omega, \omega_k) = \tilde{c}(\omega - \omega_k, \omega_k) / W(i\omega_k)$$

where $\tilde{c}(\xi, \omega_k)$ defines the bandwidth – or *frequency resolution* – around the frequency ω_k. If the measured frequency response is the ETFE (32), the associated uncertainty measure is given by (34b)

$$W(i\omega) = \frac{\Phi_v(\omega)}{|U_N(i\omega)|^2}$$

which gives the smoothed estimate corresponding to (17):

$$\hat{G}(i\omega) = \sum_{k=1}^{N} \hat{\hat{G}}_N(i\omega_k) \tilde{c}(\omega - \omega_k, \omega) |U_N(i\omega)|^2 / \Phi_v(\omega) \qquad (49)$$

It is shown in [4], Section 6.4, that for a function $\tilde{c}(\xi, \omega) = W_\gamma(\xi)$ (that does not depend on ω in a second argument), this is equivalent to the familiar Blackman-Tukey spectral analysis method. W_γ then corresponds to the "frequency window" and its inverse Fourier transform is the "Lag window" (e.g. of Hanning, Hamming, Parzen or Bartlett type).

The simple and natural curve-fitting smoothing approach (17) – (19), thus corresponds to well-known spectral analysis when applied to the ETFE. The resolution of the method is determined by the "width" of the smoothing window W_γ. It follows from our discussion that this window, $\tilde{c}(\xi, \omega)$ very well may depend on the "target frequency" ω, thus allowing for *Frequency dependent resolution.*

VIII. Transients and Initial States

Let us go back to the basic relationship (1) and focus on discrete time. The arguments below are applicable to multi-input-multi-output system, even though the notation suggests a SISO system.

Let us consider the noise-free part of the response

$$y_u(t) = G(q) u(t) \qquad (50)$$

Generally speaking, we assume only a finite number of samples of inputs and outputs (24) be known:

$$\begin{aligned} &y(1), y(2), \ldots, y(N) \\ &u(1), u(2), \ldots, u(N) \end{aligned}$$

The inputs prior to $t = 1$ are thus not known. Lets us denote by $\check{y}_u(t)$ what would be the output corresponding to a particular assumption about $u(t), t = -\infty, \ldots, -2, -1, 0$. Two typical cases would be

$$\begin{aligned}\check{y}_u(t) &= y_u^0(t) \text{ outputs of (50) if } u(t) = 0, t \leq 0 \\ \check{y}_u(t) &= y_u^p(t) \text{ outputs of (50) if } u(t) \text{ is periodic} \\ &\quad \text{with period } N \text{ from } t = -\infty \text{ to } t = N\end{aligned}$$

Let now (50) be realized in state space form:

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

Whatever assumption about prior values of $u(t)$ we have made, it would have left us in a certain state $x(0) = \check{x}$ at time t. (For example, all prior u:s being zero would give $x(0) = 0$.) Let the actual, typically unknown, initial state be $x(0) = x^*$. Then

$$\begin{aligned}y_u(t) &= \check{y}_u(t) + \tilde{y}_u(t) \\ \tilde{y}_u(t) &= C(qI - A)^{-1}A(x^* - \check{x})\delta(t) \\ \delta(t) &= \begin{cases} 0 & \text{if } t \neq 0 \\ 1 & \text{if } t = 0 \end{cases}\end{aligned}$$

The term $\tilde{y}_u(t)$ is thus the response from the initial conditions. Alternatively it can be seen as the impulse response *from an additional input, which is an impulse*:

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) + R\delta(t) \\ x(0) &= \check{x} \text{ (the assumed input behaviour prior to } t = 0) \\ y(t) &= Cx(t) \\ R &= A(x^* - \check{x})\end{aligned}$$

The consequence is that *any (possibly erroneous) guess of input behaviour prior to time $t = 0$ can always be made up for by adding an extra input which is an impulse at time 0. The dynamics from this input has the same poles as the system but unknown zeros.* Note that *one* extra input is sufficient, even if there are several regular inputs.

The typical two cases for assumed prior behaviour of the inputs are

1) In the time domain: Assume that all prior values of $u(t)$ are zero. This will give the simple predictor (38) with all values of u and y prior to $t = 1$ being zero.
2) In the frequency domain: Assume that all prior values of u are obtained by periodic continuation of u backwards in time. This will make the Fourier transformed relation in (42) exact for the u-influence at the DFT-gridpoints (29).

Now, for general data sets, these two assumptions are not correct, but the point is that an extra input signal which is an impulse will make them correct, if this input is passed through a system with the same poles as the model, and the zeros are adjusted to data (to match the assumption.) This extra input can be neglected, only if we know that the input is periodic in the frequency domain case, or past values are zero in the time domain. For long data records, it may be of less importance, since the effects of this impulse response may decay quickly compared to the data length.

This way to compensate for non-periodic frequency domain data was described in [11]. See also [13] for an instructive discussion.

Example: The system

$$y(t) = \frac{q^{-1} + 0.5q{-}2}{1 - 1.5q^{-1} + 0.99q^{-2}}$$

was simulated noise-free over 150 data points with a white noise input. Samples 101 to 150 were selected for identification. Models were fit to these data both in the time and frequency domains and both with and without adding an extra input being an impulse. This gave the following estimates:

Time domain and frequency domain with extra input:

$$y(t) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.99q^{-2}}$$

The "initial state" or numerator polynomial for the input were $B_2(q) = -14.06q^{-1} - 1.079q^{-2}$ and $B_2(q) = -18.99 - 1.63q^{-1}$ respectively.

Time domain without extra input:

$$y(t) = \frac{1.32q^{-1} - 0.26q^{-2}}{1 - 1.55q^{-1} + 0.94q^{-2}}$$

Frequency domain without extra input:

$$y(t) = \frac{1.04q^{-1} + 0.68q^{-2}}{1 - 1.47q^{-1} + 0.87q^{-2}}$$

This illustrates clearly that erroneous assumptions about past inputs (they are neither zero nor periodic in this case) can give bad results even for noise-free data. However, they can be handled with proper use of initial conditions (an extra input impulse) both in the time and the frequency domains.

(The four models could be reproduced in [6] by the code

```
m0 = idpoly(1,[0 1 0.5],1,1,...
          [1 -1.5 0.99]);
u = randn(150,1):
y = sim(m0,u);
z = iddata(y,u);
ze = z(101:150);
udel = [1;zeros(49,1)];
zed = ze;
zed.u = [ze.u udel];
m1 = oe(zed,[2 2 2 2 1 1],'ini','z');
m2 = oe(fft(zed),[2 2 2 2 1 0],'ini','z');
m3 = oe(ze,[2 2 1],'ini','z');
m4 = oe(fft(ze),[2 2 1],'ini','z');
```

(The pair `'ini','z'` is there to inhibit the default estimation of initial states.))

IX. Continuous and Discrete-Time Models

An interesting advantage with frequency domain data is that continuous time signals can be represented as the Fourier transforms at a finite number of frequencies. This opens up the possibility to estimate directly continuous time models as in (46) or (47). The crux is not really the estimation method, but how to find the Fourier transforms of the continuous time signals. This was discussed in Section V-A. See also [3].

X. Software Aspects

We have in this contribution showed the duality between time and frequency domain data and methods for identifying linear models. It is natural and desirable that the duality is handled in a transparent way in software packages for identification. In this section we shall describe how this is done in the System Identification Toolbox (SITB) [6]). The goal of the syntax is to handle time domain data, frequency domain input-output data, and frequency response data in entirely analogous fashions both for estimating and validating models.

A. Input-Output Fourier Data

The `iddata` object contains input – output data. In the time domain case, (24), the definition of the object from data vectors or matrices is straightforward:

```
dat = iddata(Y,U,Ts)
```

where `Ts` is the sampling interval. The object also allows definition of input/output data in the frequency domain over arbitrary frequencies as in

```
dat = iddata(Y,U,'Frequencies',W,...
         'Domain','Freq','Ts',0);
```

This corresponds to (25) and (27). Note that the sampling interval, T, (`'Ts'`) is still relevant, since it has information of how the signal Fourier transforms `Y` and `U` have been computed from time domain data. Discrete time Fourier Transforms conceptually have the frequency argument $e^{i\omega T}$. Note however, that frequency domain data, unlike time domain data allow continuous time signals ($T = 0$). Compare the discussion in Section V-A.

With frequency domain data objects, several MATLAB commands are naturally overloaded:

```
DF = fft(dat)
dat = ifft(DF)
da = abs(dat)
df = phase(dat)
```

etc.

B. Frequency Response Data

Frequency response data as in (30) and (31) can be stored in the `idfrd` object in the SITB. It corresponds to the frequency response data object `frd` in the CONTROL SYSTEMS TOOLBOX. The object is formed by

```
dat = idfrd(G,fre,Ts,'cov',W)
```

Comparing with (31), `G` contains the response data G_m, and `fre` the frequencies ω_k. `Ts` is the sampling interval T (`Ts = 0` denotes continuous time) and `W` is the uncertainty measure (31b), which can be omitted if not known.

C. Estimation and Validation

The point now is that whatever the format of `dat`, estimation and validation of models follow the same syntax:

```
m1 = oe(dat,[2 2 1])
m2 = n4sid(dat,3)
m3 = pem(dat)
compare(dat,m1,m2,m3)
resid(dat,m1)
```

etc. The prediction error approaches (`oe, pem` etc) implement the routines of Section VI, while the subspace estimation command `n4sid` is described in [8] for frequency domain data. (See also [7].)

Arbitrary weighings c_k in the frequency domain fits, replacing $1/\Phi_v(\omega_k)$ in (43) and $1/W(i\omega_k)$ in (47), can be obtained by

```
m  = oe(dat,[2 2 1],'focus',c)
```

By default, initial states are always estimated, as described in Section VIII. This estimation can be inhibited by

```
m = oe(dat,[2 2 1],'InitialState','zero')
```

If the frequency domain data is denoted as continuous time, a continuous time model is estimated directly (without `d2c` transformations). Compare also [3].

D. Direct Frequency Function Estimation by Local Smoothing

The local smoothing technique, described in Section VII – which is an extension of traditional spectral analysis methods – is implemented in a new function that estimates `idfrd` objects (frequency functions and disturbance spectra) from time or frequency `iddata` objects:

```
 g = spafdr(data)
```

This allows Frequency Dependent Resolution, with a logarithmic frequency grid as default along with a resolution that as adopted to the grid. This could be an efficient way of compressing measured data. It is often the case that a courser resolution (in rad/s) can be used at higher frequencies, and that a constant *relative resolution* is to be preferred. Figure 1 illustrates the effect of the frequency dependent resolution.

E. Some Further Features

Frequency domain data offer useful potentials also for other problems:

- A *focus filter* can be implemented as specific frequencies for which the fit should be made. For example,

  ```
  m = oe(dat,[2 2 1],'focus',[0.2 1])
  ```

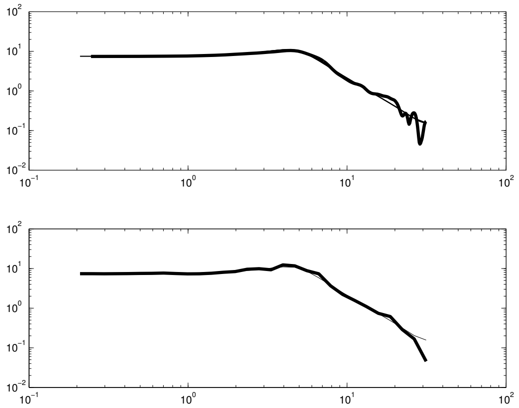


Fig. 1. Estimate of the frequency function for the data set IDDATA1, using SPA (above) and SPAFDR (below) with default arguments for frequency vector and resolution. Thin line: the true frequency response.

will concentrate the fit to the pass band from 0.2 to 1 rad/s. The desired frequency bands may not necessarily be known *a priori*, but could be selected from a preliminary model, like using frequencies that correspond to the Nyquist curve being in the third quadrant, or being close to the critical point -1. Example:

```
m = n4sid(data,5);
f = idfrd(m);
ph = phase(squeeze(f.resp));
fs = fselect(f,find(ph>-pi & ph<-
pi/2));
mp = pem(fs,'p1d');
```

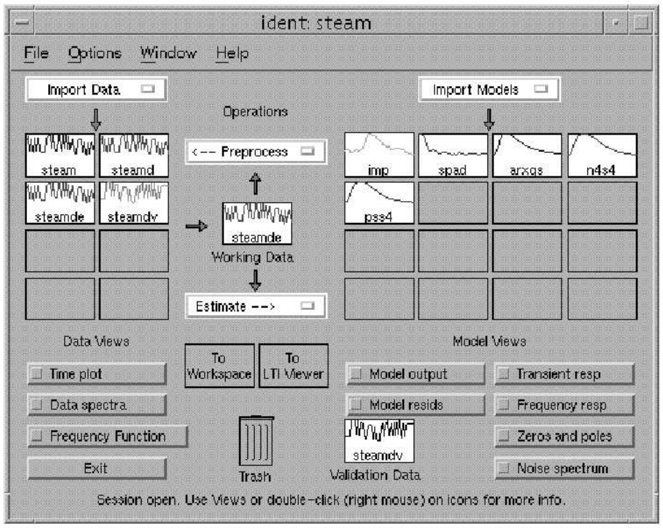
- If the inter-sample input behaviour is band-limited, moving to the frequency domain will be the easiest way to handle the sampling. The FFT (discrete Fourier transform) of the input will then be equal to the Fourier transform of the underlying continuous time input signal. (Cf Section V-A.) The FFT of the output will similarly correctly describe the continuous time Fourier transform of that part of the output that originates from the input, and we can directly fit a continuous time model – in this case of the kind

$$G(s) = \frac{b}{s^2 + f_1 s + f_2}$$

```
z = iddata(y,u,0.5);
zf = fft(z);
zf.Ts = 0;
mp = oe(zf,[1 2])
```

F. GUI support

The graphical user interface (GUI) has been extended to be transparent wrt the data domain. Frequency domain `iddata` and frequency response data as `frd` or `idfrd` objects can be imported into the GUI in the same way as time domain data. See Figure 2. The icons for the different types of data sets are marked by different background colours. The `data preprocessing` menus allow the transformation between the various representations. Also the use of data objects of different types for estimation and validation is entirely transparent. For example, if an `idfrd` object is chosen as validation data, the `Model output view` shows the frequency responses of the models, together with the data.



Session open. Use Views or double-click (right mouse) on icons for more info.

Fig. 2. The GUI

XI. Industrial Application Example

Paulstra CRC, Grand Rapids, MI manufactures among other things rubber dampers for the automotive industry. Frequency response data were collected from a vehicle model with four inputs and two outputs. The inputs are locations of the body relative to the tires (at two points) and to contact points with the damper (two other points) and the outputs are the locations relative to the damper at two relevant points. The frequency function data $G(i\omega)$ is thus a 2-by-4 matrix for each ω, and it was given at 152 frequency values.

Four of these 8 transfer functions are shown in Figures 3 and 4.

The frequency function data, which had no uncertainty information, were fitted to a 6th order linear state-space model using the criterion (47) with $W \equiv 1$. Since the frequency functions are matrices, the norm in (47) is interpreted as the Frobenius norm (sum of squares of all the matrix elements). The fit is accomplished in [6] as

```
for ky = 1:2
  for ku = 1:4
    resp(ky,ku,:)=frf_ky_ku
  end
end
g = idfrd(resp,freq)
m = pem(g,6,'nk',[0 0 0 0])
bode(g,m)
```

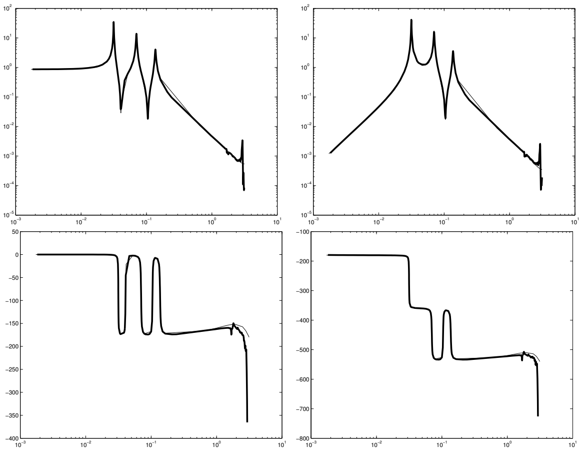


Fig. 3. The measured frequency functions from input 1 (thick line) and the frequency function of the 6th order model (thin line). The lines almost overlap. The upper plots are amplitude and the lower phase. The left figures are to output 1 and the right ones to output 2. (The frequency scale has been normalised.)

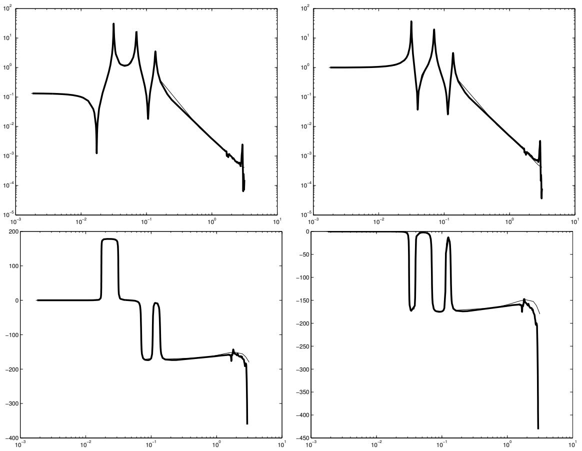


Fig. 4. As figure 3, but from input 2.)

Here `frf-ky-ku` is the measured frequency response from input `ku` to output `ky` and `freq` is the vector of frequencies. Moreover, `'nk',[0 0 0 0]` indicates zero delay from each of the inputs.

The last command generated the figures 3 and 4. They show that a 6th order linear model gives a very good fit to all 8 measured frequency functions.

XII. Conclusions

The duality between time and frequency domain data and methods is a powerful asset in linear system identification. Indeed, as stressed in [13], the question is not *Time* **or** *Frequency Domain?* but the best attitude is the statement *Time* **and** *Frequency Domain!*.

We have in this contribution outlined the basic approached and pointed out the equivalences between the methods.

Phrasing standard methods for linear system identification as *curve fitting* brings out several common features and gives some additional insight. It also shows that the bottom line in identification is quite simple and relies upon early work in statistics. In particular the kinship between time and frequency domain method, both parametric and nonparametric are brought forward. All techniques can actually be seen as different ways of smoothing the empirical transfer function estimate (32).

XIII. Acknowledgments

I would like to thank Borz Fariborzi and Bertrand Auboiron at Paulstra CRC for letting me use their data and for interesting discussions. The work has been supported by the Swedish Research Council (VR).

References

[1] W.S. Cleveland and E. Grosse. Computational methods for local regression. *Statistics and Computing*, 1:47–62, 1991.

[2] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1996.

[3] H. Garnier and P. C. Young. Time-domain approaches to continuous-time model identification of dynamical systems from sampled data. In *Proc. American Control Conference*, page This conference, Boston, MA, July 2004.

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

[5] L. Ljung. Linear system identification as curve fitting. 286:203 – 215, Dec 2003.

[6] L. Ljung. *System Identification Toolbox for use with* MATLAB. *Version 6*. The MathWorks, Inc, Natick, MA, 6th edition, 2003.

[7] T. McKelvey. Subspace methods for frequency domain data. In *Proc. American Control Conference*, page This conference, Boston, MA, July 2004.

[8] T. McKelvey, H. Akçay, and L. Ljung. Subspace-based multivariable system identification from frequency response data. *IEEE Trans. on Automatic Control*, 41(7):960–979, Jul 1996.

[9] E. Nadaraya. On estimating regression. *Theory of Prob. and Applic.*, 9:141–142, 1964.

[10] R. Pintelon and J. Schoukens. *System Identification – A Frequency Domain Approach*. IEEE Press, New York, 2001.

[11] R. Pintelon, J Schoukens, and G Vandersteen. Frequency domain system identification using arbitrary signals. *IEEE Trans. Automatic Control*, 42:17171–1720, 1997.

[12] J. Roll, A. Nazin, and L. Ljung. A non-asymptotic approach to local modelling. In *Proc. IEEE Conf. on Decision and Control*, Las Vegas, NV, Dec 2002.

[13] J. Schoukens, R. Ointelin, and Y. Rolain. Time domain identification, frequency domain identification. equivalences! differences? In *Proc. American Control Conference*, page This conference, Boston, MA, July 2004.

[14] J. Schoukens and R. Pintelon. *Identification of Linear Systems: A Practical Guideline to Accurate Modeling*. Pergamon Press, London (U.K.), 1991.

[15] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall Int., London, 1989.

[16] C.J. Stone. Consistent non-parametric regression (with discussion). *Ann. Statist.*, 5:595–645, 1977.