

Integrated Frequency-Time Domain Tools for System Identification

Lennart Ljung

Abstract—The duality between time and frequency domain methods for linear systems is well known. It plays a crucial role for example in control systems design, and the domains are thought of complementing rather than competing. Quite recently, the full interplay and duality between the two domains have been clear also in system identification applications. In this contribution, this duality will be discussed. The emphasis is on how it can be used to create a software environment for linear system identification that is as transparent as possible with respect to the data domains.

I. INTRODUCTION

The latest version of MATLAB'S SYSTEM IDENTIFICATION TOOLBOX (SITB), [3] supports time and frequency domain data and methods in a symmetric fashion. This contribution will describe how this is done, as well as the underlying theories and techniques. A related paper at this conference, [4], contains more details.

For linear system identification, that is, methods to estimate linear models from measured input-output data, the links between time- and frequency domain methods are 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 analyzers 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 intersample 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 [2] and [7] for comprehensive treatments.

II. THE BASIC TIME-DOMAIN APPROACH

The basic setup for system identification can be described briefly in the following familiar way:

The starting point for parametric methods is a parameterized set of transfer functions $G(q, \theta)$, where θ is a finite-dimensional parameter vector, and q is the shift operator. In the notation here, we thus work with discrete time models. The transfer function from input u to output y can possibly be complemented with an assumption of the spectrum of an additive disturbance v :

$$\begin{aligned} y(t) &= G(q, \theta)u(t) + v(t) \\ v \text{ has spectrum } &\Phi_v(\omega) \end{aligned} \quad (1)$$

It is often useful to think of v as generated from a white noise source e :

$$\begin{aligned} y(t) &= G(q, \theta)u(t) + H(q, \theta)e(t) \\ \Phi_v(\omega) &= \lambda |H(e^{i\omega}, \theta)|^2 \end{aligned} \quad (2)$$

From this model, the natural one-step ahead prediction can be computed:

$$\hat{y}(t|\theta) = (1 - H^{-1}(q, \theta))y(t) + H^{-1}(q, \theta)G(q, \theta)u(t) \quad (3)$$

which gives the following prediction errors (or residuals);

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta) = H^{-1}(q, \theta)(y(t) - G(q, \theta)u(t)) \quad (4)$$

A basic method for estimating the parameter vector θ is to minimize the size of the prediction errors:

$$\hat{\theta}_N = \arg \min_{\theta} V_N(Z^N, \theta) \quad (5a)$$

$$V_N(Z^N, \theta) = \sum_{t=1}^N \varepsilon^2(t, \theta) \quad (5b)$$

Here Z^N denotes the available data record:

$$Z^N = \{y(1), u(1), \dots, y(N), u(N)\} \quad (6)$$

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

III. A COMMON DENOMINATOR: CURVE FITTING OF THE EMPIRICAL TRANSFER FUNCTION ESTIMATE

Estimation of a linear system can be interpreted as finding the curve that is the frequency response function of the system:

$$G(e^{i\omega T}) \quad \text{Discrete time, Sampling interval } T \quad (7a)$$

$$G(i\omega) \quad \text{Continuous time} \quad (7b)$$

This is a complex-valued function of a real variable ω . This function is the Fourier transform of the impulse response of the system. If $y(t) = G(q)u(t)$, there is the well-known and simple relationship between the frequency function and the (Discrete) Fourier transforms of the inputs and outputs:

$$Y_N(e^{i\omega T}) = \frac{1}{\sqrt{N}} \sum_{k=1}^N y(kT) e^{-i\omega k T} \quad (8a)$$

$$U_N(e^{i\omega T}) = \frac{1}{\sqrt{N}} \sum_{k=1}^N u(kT) e^{-i\omega k T} \quad (8b)$$

$$Y_N(e^{i\omega T}) = G(e^{i\omega T}) U_N(e^{i\omega T}) + R_N(e^{i\omega T}) \quad (8c)$$

Here $R_N(e^{i\omega T})$ is a transient, that adjusts for the circular convolution that is inherent in the discrete Fourier transform (DFT). Note that if u is periodic with period N

$$R_N(e^{2\pi i k/N}) = 0, k = 0, \dots, N-1. \quad (9)$$

The corresponding frequencies are

$$\omega_k = 2\pi k/N T, k = 0, \dots, N-1 \quad (10)$$

These frequencies will be referred to as *The DFT grid*. The character of R_N in the general case is further discussed in Section V.

The relationship (8c) motivates the use of the *Empirical Transfer Function Estimate* (ETFE):

$$\hat{G}_N(e^{i\omega T}) = \frac{Y_N(e^{i\omega T})}{U_N(e^{i\omega T})} \quad (11)$$

In case the observations y and u have been obtained from a noise-corrupted linear system with frequency function $G_0(e^{i\omega})$ it can be shown that the ETFE has the following statistical properties: (Lemma 6.1 in [2].)

$$E\hat{G}_N(e^{i\omega T}) = G_0(e^{i\omega T}) + \frac{\rho_1}{\sqrt{N}U_N(e^{i\omega T})} \quad (12a)$$

$$E|\hat{G}_N(e^{i\omega T}) - G_0(e^{i\omega T})|^2 = \frac{\Phi_v(\omega)}{|U_N(e^{i\omega T})|^2} + \frac{\rho_2}{N|U_N(e^{i\omega T})|^2} \quad (12b)$$

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. Moreover, it can be shown that the values of the ETFE are asymptotically uncorrelated at frequencies on the DFT grid.

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

$$\hat{G}_N(e^{i\omega_k T}) = G_0(e^{i\omega_k T}) + v_k \quad (13)$$

with v_k being a zero mean random variable with variance $\Phi_v(\omega_k)/|U_N(e^{i\omega_k T})|^2$. We have then ignored the terms with ρ in the expressions above.

If we indeed treat (13) as measurement from an unknown curve that we want to estimate, two approaches suggest themselves: (Set $T = 1$ in the sequel for simplicity)

- Adjust the parameterized function family $G(e^{i\omega}, \theta)$ to these measurement. Use a weighted least squares criterion, with weights μ_k that reflect the reliability of the measurements:

$$\hat{\theta}_N = \arg \min_{\theta} V_N \quad (14a)$$

$$V_N = \sum_{k=1}^N \mu_k |\hat{G}(e^{i\omega_k}) - G(e^{i\omega_k}, \theta)|^2 \quad (14b)$$

$$\mu_k = |U_N(e^{i\omega_k})|^2 / \Phi_v(\omega_k) \quad (14c)$$

- Smooth the observations locally around a target frequency ω , again paying attention to the reliability of the measurements:

$$\hat{G}(e^{i\omega}) = \sum_{k=1}^N \tilde{c}(\omega - \omega_k, \omega) \mu_k \hat{G}(e^{i\omega_k}) \quad (15a)$$

$$\mu_k \text{ as in (14c)} \quad (15b)$$

$$\tilde{c}(\xi, \omega) = 0 \text{ if } |\xi| > B(\omega) \quad (15c)$$

Here $B(\omega)$ is the “bandwidth” or *frequency resolution* around frequency ω

Let us for a moment return to (14). Inserting the weighting μ_k , the criterion will be

$$V_N = \sum_{k=1}^N (Y_N(e^{i\omega_k}) - G(e^{i\omega_k}, \theta) U_N(e^{i\omega_k})) / \Phi_v(\omega_k) \quad (16)$$

This shows that fitting the ETFE to a parametric model is the same as fitting the model to *Frequency Domain Input-output data*. The conceptual relationship in the frequency domain is

$$Y_N(e^{i\omega}) = G_0(e^{i\omega}), U_N(e^{i\omega}) + V_N(e^{i\omega})$$

$$E|V_N(e^{i\omega})|^2 = \Phi_v(\omega)$$

Moreover, if we apply Parseval’s relationship to the time domain criterion (5), we note that the Fourier transform of $\varepsilon(t, \theta)$ is

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

Applying Parseval’s relationship to (5) and ignoring transient effects (or assuming periodic data) now gives exactly the criterion (14) for the model assumptions (2)

IV. MODEL PARAMETERIZATIONS

The actual parameterization can be chosen 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 that this corresponds to rational function approximations in the variable $x = e^{i\omega}$.

A physically parameterized, continuous time state space model

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) + w(t);$$

$$Ew(t)w^T(s) = Q(\theta)\delta(t-s)$$

$$y(t) = C(\theta)x(t) + D(\theta)u(t) + e(t);$$

$$Ee(t)e^T(s) = R(\theta)\delta(t-s)$$

corresponds to

$$G(i\omega, \theta) = C(\theta)(i\omega I - A(\theta))^{-1}B(\theta) + D(\theta)$$

$$H(i\omega, \theta) = C(\theta)(i\omega I - A(\theta))^{-1}K(\theta) + I$$

where $K(\theta)$ is computed from A , C , Q and R as the steady state Kalman filter gain.

Simple process models are obtained by parameterizations of the kind

$$G(i\omega, \theta) = \frac{K}{1 + sT_1} e^{i\omega\tau}, \quad \theta = \{K, T_1, \tau\} \quad (17)$$

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

Many other types of parameterizations are of course possible.

V. TRANSIENTS AND INITIAL STATES

It is well known that Fourier transformation of finite data records assume ‘‘circular convolution’’. This means that unless the data is periodic, there will be an error as in (8c) (the term R_N). This was long thought as an obstacle to using frequency domain methods in identification, since any deviation from periodicity in the input would lead to bias errors in the models.

In fact, the deviation from periodicity in frequency domain data is no different from the lack of knowledge of past data in time domain methods. We shall show that fact in the present section.

Let us go back to the basic relationship (1). 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

$$y_u(t) = G(q)u(t) \quad (18)$$

Generally speaking we assume only a finite number of samples of inputs and outputs (6) be known:

$$y(1), y(2), \dots, y(N)$$

$$u(1), u(2), \dots, u(N)$$

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

$$\check{y}_u(t) = y_u^0(t) \text{ outputs of (18) if } u(t) = 0, t \leq 0$$

$$\check{y}_u(t) = y_u^p(t) \text{ outputs of (18) if } u(t) \text{ is periodic}$$

with period N from $t = -\infty$ to $t = N$

Let now (18) be realized in state space form:

$$x(t+1) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$

Whatever assumption about prior values of $u(t)$ 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

$$y_u(t) = \check{y}_u(t) + \tilde{y}_u(t)$$

$$\tilde{y}_u(t) = C(qI - A)^{-1}(x^* - \check{x})\delta(t)$$

$$\delta(t) = \begin{cases} 0 & \text{if } t \neq 0 \\ 1 & \text{if } t = 0 \end{cases}$$

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:

$$x(t+1) = Ax(t) + Bu(t) + R\delta(t)$$

$$x(0) = \check{x} \text{ (the assumed input behavior prior to } t = 0)$$

$$y(t) = Cx(t)$$

$$R = x^* - \check{x}$$

The consequence is that *any (possibly erroneous) guess of input behavior 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 behavior 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 (3) 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 (8c) exact for the u -influence at the DFT-grid-points (10).

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.

How to compensate for non-periodic frequency domain data was described in [8]. See also [9] for an instructive discussion.

VI. SOFTWARE ASPECTS

A. Desired Features

A software package on linear system identification should offer

- Full support of using both time- and frequency domain input-output data. The handling of the frequency grids for frequency domain data should be automatic.
- It should also accept frequency response data (like e.g., the ETFE (11)) as a data type for both parametric and non-parametric estimation.
- Data that relate both to continuous time and discrete time descriptions should be taken care of.
- It should allow both simulation, estimation and validation using any type of data.
- The syntax and, if present, the graphical user interfaces (GUI) should be fully transparent w.r.t. to the data domain.

In this section we shall describe how the latest version of the SYSTEM IDENTIFICATION TOOLBOX (SITB) for MATLAB [3]) deals with these issues. 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.

B. Input-Output Data

Time domain: The basic situation is that vectors of input and output data are given in the time-domain:

$$\begin{aligned} &\{u(T), \dots, u(NT)\} \\ &\{y(T), \dots, y(NT)\} \end{aligned}$$

where T is the sampling interval. The corresponding data object is the `iddata` format, which contains the data as in

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

where `Ts` is the sampling interval. Further properties, like channel names and units, arbitrary sampling instants etc can be contained in this object. Of special interest is to store information about the *inter-sample behavior* of the input between the sampling instants (like piecewise constant, piecewise linear or band-limited.) Moreover one `iddata` object can store the data from several different experiments.

Frequency domain: Fourier transforms for input and output data can be computed as in (8) and stored for selected frequencies $\omega_k, k = 1, \dots, M$

Remark: : A common choice is $\omega_k = 2\pi k/NT, k = 0, \dots, N/2$, which corresponds to the DFT-grid. This takes into account that for real data, frequencies above the Nyquist frequency π/T give complex conjugate values of U_N .

One should also bear in mind that the data can be collected in equipment that directly produce Fourier transforms.

In some cases it may be reasonable to assume that the data are given as Fourier transforms of continuous-time data:

$$U(\omega) = \int u(\tau)e^{-i\tau\omega} d\tau \quad (19)$$

This may be the case for fast sampled data or for band-limited signals.

The `iddata` object can also hold input/output data in the frequency domain over arbitrary frequencies as in

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

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 $e^{i\omega T}$. See (8). Note however, that frequency domain data, unlike time domain data allow continuous time signals ($T = 0$), that is Fourier transforms that (at least conceptually) have been formed by (19).

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

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

etc.

C. Frequency Response Data

The frequency response function of a linear system is the Fourier transform of its impulse response. See (7). In several cases the frequency response can be seen as the primary measured information about the system:

- It is computed from input-output measurements as the ETFE (11) or by spectral analysis (see Section VI-E).
- It is delivered from special hardware equipment, *frequency analyzers*, which use either Fourier Analysis or determine phase and amplitude shift of applied sinusoidal inputs ("swept sinusoid").
- It is computed from a detailed, high order model, for which simpler approximations are sought.

In either case we would have a finite collection of measurements

$$Z^N = \{G_m(i\omega_1), \dots, G_m(i\omega_N)\} \quad (20a)$$

or in discrete time

$$Z^N = \{G_m(e^{i\omega_1 T}), \dots, G_m(e^{i\omega_N T})\} \quad (20b)$$

Possibly, these data could be complemented with some uncertainty estimate W of the measurements:

$$Z_U^N = \{W(i\omega_1), \dots, W(i\omega_N)\} \quad (20c)$$

For example, for an ETFE estimate of the frequency response, the uncertainty measure could be the variance (12b).

Frequency response data can be stored in the `idfrd` object in the SITB. It corresponds to the frequency response data object `frd` in the CONTROL SYSTEM TOOLBOX.

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

Comparing with (20), 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 (20c), which can be omitted if not known.

Measurements of frequency response functions are typically rather rough estimates of a function that is known to be quite smooth. It may therefore be a good idea to compress the data by smoothing the original response measurement (e.g. like (15)) and then store the response at fewer frequency values. The frequency resolution $B(\omega)$ in (15c) is often chosen to be logarithmic, so the frequency values become logarithmically spaced. This gives apparent constant resolution in a Bode plot. See the command `spafdr` below in Section VI-E.

D. 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 minimization of V_N in (5), (14) and (16) while the subspace estimation command `n4sid` is described in [6] for frequency domain data. (See also [5].)

Arbitrary weightings c_k in the frequency domain fits, replacing $1/\Phi_v(\omega_k)$ in (16) and μ_k in (14b), can be obtained by

```
m = oe(dat, [2 2 1], 'focus', [c_1, ..., c_N])
```

By default, initial states are always estimated, as described in Section V). 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 [1].

E. Direct Frequency Function Estimation by Local Smoothing

The local smoothing technique, described in (15), 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 is adopted to the grid. This could be an efficient way of compressing measured data. It is often the case that a coarser 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.

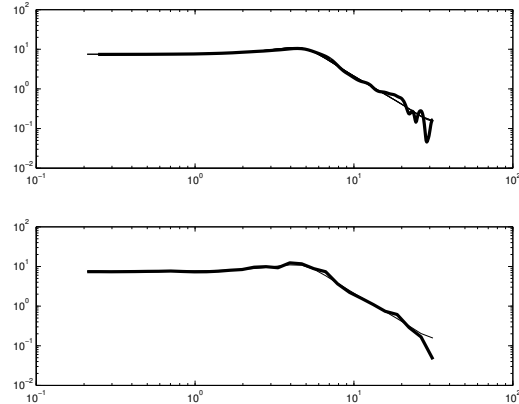


Fig. 1. Estimate of the frequency function for the data set IDDATA1, using SPA (above) and SPAFDR (below) with default arguments for frequency vector and resolution. Thin line: the true frequency response.

F. 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])` 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(resp));
```

```

fs = fselect(f, find(ph > -pi & ph < -
pi/2));
mp = pem(fs, 'pld');

```

- If the inter-sample input behavior 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. 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:

```

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

```

This would estimate a model of the kind

$$\frac{b}{s^2 + f_1 s + f_2}$$

- How are frequency response function dealt with in estimation and validation?
The difficulty is really to deal with multiple inputs. While the frequency response is an $Ny|Nu|Nf$ object (Ny outputs, Nu inputs and Nf frequencies), the input output data is an $(Ny + Nu)|Nf$ object. This alone shows that the problem requires some thought. The solution is to create a multi-experiment data object, with one experiment for each input.

G. 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 colors. 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.

VII. SUMMARY

For identification of linear systems there are both time and frequency domain techniques available to find good models. It is desirable to use both these “worlds” in an effective manner to come up with a good identification result. In this presentation we have pointed to methods and criteria for estimation and validation in both domains. In particular we have stressed the close kinship between the domains, also from an identification perspective. In effect, all methods can be seen as various ways to smooth the empirical transfer function estimate.

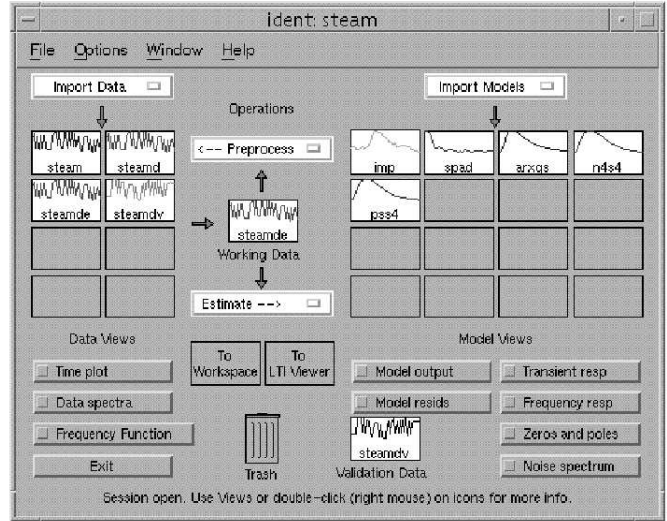


Fig. 2. The GUI

It is also important that the software support handles this duality between the time and frequency domains in a transparent manner. We have discussed how this can be done, by describing some new features in the SYSTEM IDENTIFICATION TOOLBOX.

REFERENCES

- [1] 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.
- [2] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.
- [3] L. Ljung. *System Identification Toolbox for use with MATLAB. Version 6*. The MathWorks, Inc, Natick, MA, 6th edition, 2003.
- [4] L. Ljung. State of the art in linear system identification: Time and frequency domain methods. In *Proc. American Control Conference*, page This conference, Boston, MA, July 2004.
- [5] T. McKelvey. Subspace methods for frequency domain data. In *Proc. American Control Conference*, page This conference, Boston, MA, July 2004.
- [6] 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.
- [7] R. Pintelon and J. Schoukens. *System Identification – A Frequency Domain Approach*. IEEE Press, New York, 2001.
- [8] R. Pintelon, J. Schoukens, and G. Vandersteen. Frequency domain system identification using arbitrary signals. *IEEE Trans. Automatic Control*, 42:1717–1720, 1997.
- [9] 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.