

Using DSPT SigLab with the Frequency Domain System Identification Toolbox

SigLab makes it easy for users of the Frequency Domain System Identification Toolbox¹ to get high quality measurements for analysis. The most direct measurement that can be utilized by the toolbox is the transfer function. SigLab's FFT-based and Swept Sine Network analyzers make it easy to get accurate transfer function estimates. This note describes how transfer function estimates can be analyzed by the FDID toolbox.

Overview

Two primary classes of identification techniques.

There is no one identification technique that is optimal for all problems, hence the large number of approaches available in the MATLAB[®] environment. The MATLAB System Identification Toolbox² and MMLE3 Identification Toolbox are based on time domain analysis techniques. DSPT SigLab™ is optimized to provide alias protected, time-domain measurements for these identification schemes. SigLab also supplies its own GUI-based time domain identification technique.

The Frequency Domain System Identification Toolbox (FDID) has advantages for analyzing certain classes of systems. The FDID techniques are most naturally used with transfer function (frequency domain) data. SigLab can make excellent transfer function measurements by using either a swept-sine technique or a broadband FFT- based technique. These measurements are controlled by easy-to-use GUI-based user interfaces. The results can be supplied directly to the FDID toolbox.

Identification: why a frequency domain technique?

A frequency domain technique has advantages over the time domain techniques under the following conditions.

1. A model is desired over a strictly prescribed frequency range.
2. The device (plant) being modeled is part of an operating control system.
3. The swept-sine measurement technique must be used due to noise and non-linearity.
4. The use of non-uniformly spaced frequency measurements is needed.

Consider the first condition. Often physical devices with distributed energy storage elements have extremely complex dynamic behavior. To completely model the device over a large bandwidth becomes difficult due to the large number of in-band resonances. A simple, but effective, strategy is to decompose the modeling task by making measurements and models in adjacent sub-bands. The resulting models can be then be combined, if desired, to create an overall model. This strategy is most easily accomplished in the frequency domain.

The second condition poses many interesting measurement challenges. Good measurements are more difficult to obtain because measurements often must be made while the control system is actually operating. The measurement of the open-loop dynamics of the control system are important and this measurement can be obtained by a simple frequency-domain mapping.

The swept sine (stepped-sine) measurement approach is often employed for transfer function estimation since it can better cope

with system noise and non-linearity (condition 3). Such measurement problems are common in control systems. A side benefit of using swept sine is that the measurement frequency resolution does not have to be uniform (condition 4). Mechanical systems at low frequencies often have a frequency response which goes as $\frac{1}{f^2}$ or $\frac{1}{f}$

followed by bands containing closely spaced resonances. The swept sine measurement allows for an increase the number of user-defined frequency measurement points in the areas where there are interesting resonances. Often, the lower frequency areas need fewer measurements saving both measurement and analysis time.

This note focuses on using SigLab's transfer function estimates with the FDID toolbox. Techniques for making the transfer function estimates are covered in the *Estimating Transfer Functions with SigLab* application note³ and these details will not be repeated here.

Using SigLab to Make the Transfer Function Measurement.

Using the transfer function

Rudimentary data acquisition systems provide time-domain measurement data which often lacks alias protection. Although the FDID toolbox can be used with time-domain data, there is little advantage to using this method since M-functions in the toolbox are used to estimate frequency domain information from the time histories. SigLab, however, can provide excellent transfer function (frequency domain) measurements directly to the FDID algorithm. SigLab uses extensive on board digital signal processing capabilities for fast, accurate, transfer function estimates.

The steps to using the FDID toolbox with SigLab are as follows:

1. make a transfer function measurement using the swept sine or network analyzer virtual instruments
2. save the results to a file
3. use a simple m-file script to read in the measurement file and invoke the FDID *elis* function

Measurement setup

As explained in *Estimating Transfer Functions with SigLab*, the optimal measurement configuration is shown in Figure 1. The swept sine or network analyzer virtual instruments can then be used to make transfer function estimates of the system $H(\omega)$.

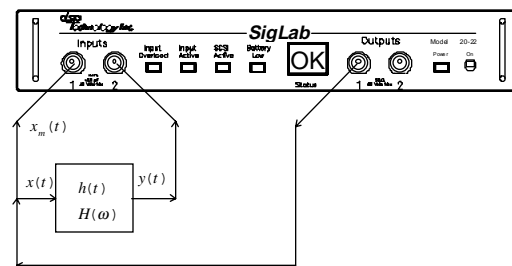


Figure 1 - The optimal measurement configuration.

Making *unbiased* transfer function estimates is crucial to successful identification. If the guidelines in *Estimating Transfer Functions with SigLab* are followed, estimation bias can be reduced to negligible levels.

Supplying Transfer Function Data to the FDID *elis* Function

The parameters required

Using the transfer function estimate from SigLab considerably simplifies using the FDID tools since much of the complexity comes from the toolbox's ability to deal with time domain data. The core routine in the FDID toolbox is the *elis.m* function.

```

% demonstration of supplying elis.m with SigLab transfer function measurement
load measdata.vna -mat % load SigLab measurement file which defines:
% Fvec = vector of frequency points
% XferDat = vector with transfer function estimate
% CohDat = vector of coherence data used as a weighting
%         to discount the importance of potentially poor
%         measurement data in the transfer function
numord = 10 ; % order of numerator
denord = 10; % order of denominator
domain = 's'; % s or z plane model

nf = length(Fvec); % number of frequency points in xfer measurement

results = elis([Fvec,ones(nf,1),XferDat(:,1)], % frequency vector and xfer est
               [zeros(nf,1),1./CohDat(:,1)],... % coherence^-1 for variance
               [domain,numord,denord]); % desired model type and order

[domain,num,den,delay,fs]=imppar(results); % reformat results into a more meaningful
                                           % representation (FDID toolbox function)

```

Listing 1 - Loading SigLab data and calling *elis.m* in MATLAB.

The FDID documentation describes nine major input parameters to the *elis* function with five output parameters. These parameters are, in general, vectors which supply the measurement data to the function as well as a plethora of analysis options (e.g. whether to use the Levenburg-Marquardt iteration algorithm or the Newton-Gauss). Fortunately, the casual user need not trouble over most of the parameters.

The important input parameters, in the following cases, are:

1. frequency vector
2. the measured transfer function data
3. the magnitude squared coherence data (used for weighting the fit)
4. the model domain(s or z)
5. numerator & denominator orders

This information can be passed to the *elis* function in the first three input arguments (note: arrays are often combined to form a single input argument).

An example of calling *elis*

A simple example serves to illustrate the usage of the *elis* function. Listing 1 does the following:

1. loads a SigLab data file.

2. passes measurements and minimal estimation information to the *elis* function.
3. reformats results using the *imppar* function.

A few words of explanation are in order.

First, argument 1 of the *elis* function requires three column vectors. The transfer function measurement is passed as the third column vector while the second is set to all ones. This is tantamount to saying that the excitation to the network was flat with no phase characteristics (ones) and the response was the measured transfer function.

Secondly, argument 2 usually contains the measurement variances. Since it is assumed the input is known, its variance is zero. The output measurement is then assumed to have errors which are related to the inverse of the coherence function: low coherence, high variance.

When the *elis.m* function is executed, a window is created with three graphs as shown in Figure 2.

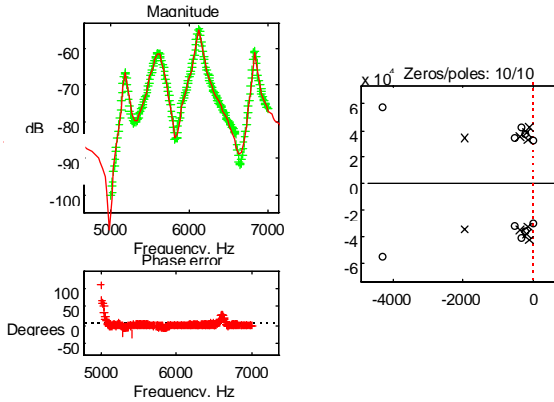


Figure 2 - The FDID toolbox window.

The upper left plot shows the model frequency response as a continuous line, with the measured data plotted as "+"s. The lower left plot shows the phase difference between the model and the measurement. This is a good indicator of modeling error. The right plot shows the model's poles and zeros in the s-plane

When the iterations are complete, the `elis` function returns the model (along with numerous other quantities) in the output arguments. For this example, only one output argument is used (results). To convert the results vector into a more meaningful format, the `imppar` function is used. This FDID Toolbox function produces a numerator/denominator polynomial model format (refer to FDID documentation).

Some Real-World Examples

Example 1: closed to open loop mapping with non-uniform measurement frequency resolution

The flexibility of the SigLab/FDID combination is demonstrated with an analysis of a closed loop control system. The SigLab swept-sine virtual instrument was used for measurement, as described in *Estimating Transfer Functions with SigLab*, pp. 14. This measurement provided an unbiased estimate of the system's closed loop transfer function. The setup for the swept sine analysis is shown in Figure 3.

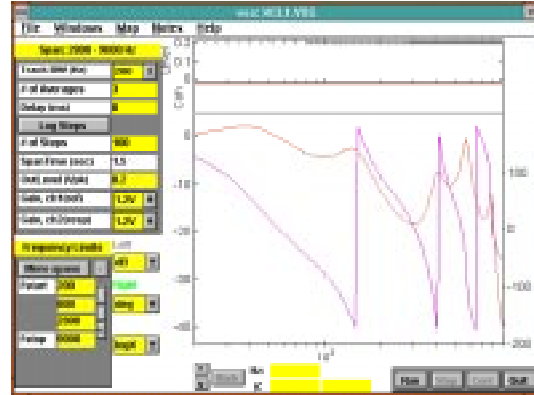


Figure 3 - Swept sine analysis of a control system (closed loop).

The measurement was broken into three frequency bands: 200-800, 800-2000, 2000-9000 Hz. The swept sine application allows the acquisition and excitation parameters to be specified independently for each band. A logarithmic frequency resolution was used in each band and each band had a different number of measurement points. The same excitation amplitude was used for each band as shown on the upper graph (level) of Figure 3.

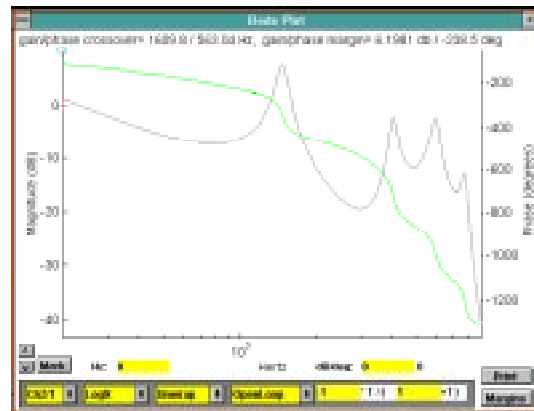


Figure 4 - Open loop response from closed loop measurement.

The measurement results can be transformed into an open loop representation (plotted in Figure 4) by this simple mapping:

$$X_{ol}(j\omega) = \frac{-X_{cl}(j\omega)}{1 + X_{cl}(j\omega)}$$

X_{ol} and X_{cl} represent the open and closed loop transfer functions respectively. This mapping is described in more detail in

Estimating Transfer Functions with SigLab
Equation 6.

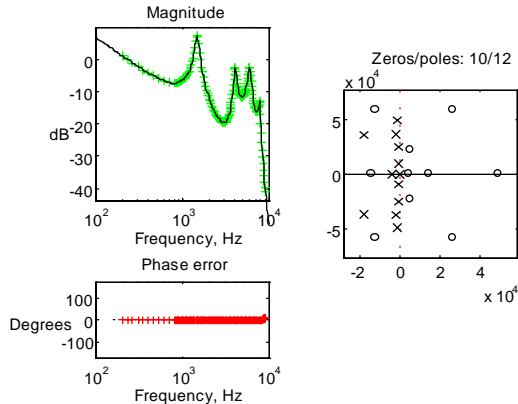


Figure 5 - Identification of open loop control system transfer function from mapped data.

This closed loop measurement data was saved to a file and then used in a MATLAB script similar to Listing 1. The only difference was the addition of the closed to open loop mapping by inserting a line in MATLAB script:

```
XferDat=-XferDat./(1+XferDat);
```

before the call to the *elis* function. Additional parameters were passed to the *elis* function to select a logarithmic x axis for the magnitude and phase plots (Figure 5).

A tenth order numerator and twelfth order denominator provided an excellent fit to the open loop response as shown in Figure 5. The magnitude agreement is virtually perfect, while the phase shows only a small discrepancy up at 9 kHz. The model is stable and clearly shows the pole due to integration in the controller at $s=0$.

Example 2: narrow band modeling.

Often a highly accurate model over a limited frequency region needs to be created. Frequency domain techniques naturally lend themselves to this task. Although given proper hardware support (real-time zoom processing) and proper algorithm design (the ability to handle complex input / output time histories) time domain techniques⁴ can also address this task, the more natural solution is the frequency domain approach.

The SigLab network analyzer virtual instrument was used to make the measurement Figure 6. It shows the transfer function of a complex dynamic system analyzed over the dc-10 kHz frequency range. It is desired to gain a better understanding of the physics associated with the resonance at approximately 6775 Hz.

The SigLab hardware has the ability to do real-time zoom which focuses the excitation and analysis capability of the hardware over a user specified analysis band.

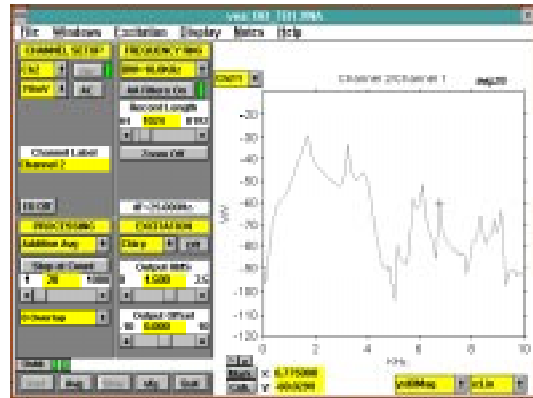


Figure 6 - DC-10 kHz analysis of a complex dynamic system.

The result of this analysis is shown in Figure 7. There is a marked improvement in frequency resolution (2 Hz vs. 25 Hz) around the resonance.

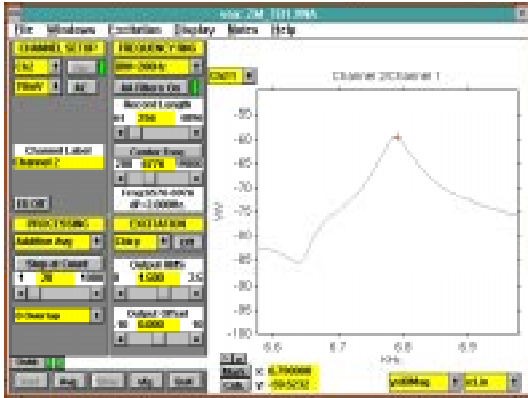


Figure 7 - Narrowband (6576-6976 Hz) analysis of a selected resonance.

This measurement was saved to a file and read in by a script M-file for analysis by the *elis* function. A fourth order numerator and sixth order denominator were found to provide an excellent fit to the data. The results of the identification are shown in Figure 8.

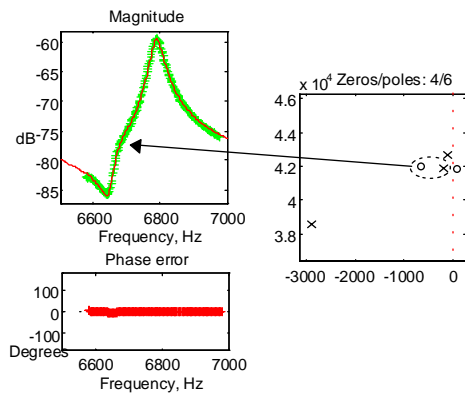


Figure 8 - Frequency Domain ID provides a better understanding of the resonance.

The FDID analysis showed a more complicated resonance than initially expected. It is easy to see how a zero-pole pair could cause the anti-resonance at approximately 6640 Hz and the resonance at 6770. It was a surprise to find *two* zero-pole pairs involved with the resonance. The slight but noticeable inflection (note the arrow in figure 8) is due to the lower Q pole-zero pair encircled by the ellipse.

Time Domain Data

Occasionally, it might be advantageous to send pure time domain data to the ID toolbox in lieu of the transfer function. SigLab can provide (alias free) time domain data records for the FDID toolbox. In this mode, specialized output sequences may also be constructed in MATLAB and generated by SigLab's output subsystem. The time domain data can be transformed to the frequency domain by an M-file function provided with the FDID toolbox and then analyzed by the core FDID algorithm *elis*.

When using the pure time-domain approach, the FDID toolbox also handles the case of correlated noise being present in both the excitation and response data. It should be noted that if the measurement guidelines given in *Estimating Transfer Functions with SigLab* are used, there will be little, if any, advantage in using the time domain option.

Conclusion

A variety of system identification tools are available in the MATLAB environment. Finding the optimal tools for the existing problem is a challenge. The FDID toolbox provides a frequency domain technique that works well in many situations. SigLab is optimized to make transfer function estimates on dynamic systems. and being MATLAB based, these estimates are immediately available for the FDID toolbox.

For more information contact:

DSP Technology Inc.

Signal Analysis Group

48500 Kato Road, Fremont, CA 94538

Phone: (510) 657-7555

Fax: (510) 657-7576

<http://www.dspt.com>

¹ Kollár, István, *Frequency Domain System Identification Toolbox User's Guide*, The MathWorks Inc., January 1994

² Ljung, Lennart, *System Identification Toolbox Users Guide*, The MathWorks Inc., May 1993.

³ , *Estimating Transfer Functions with SigLab*, AN5, DSP Technology Inc., July 1995

⁴ *System Identification: A Practical Tool from a Fiddler's Paradise*, AN4, DSP Technology Inc., June 1995

