FREQUENCY DOMAIN SYSTEM IDENTIFICATION TOOLBOX FOR MATLAB: CHARACTERIZING NONLINEAR ERRORS OF LINEAR MODELS

István Kollár[*], Rik Pintelon[**], Johan Schoukens[**]

*[*]Budapest University of Technology and Economics
Department of Measurement and Information Systems
H-1521 Budapest, Magyar tudósok krt. 2. Hungary
Fax: +36 1 463-4112, email:kollar@mit.bme.hu
[**]Vrije Universiteit Brussel, Dienst ELEC, Belgium
Pleinlaan 2, B-1050 Brussel, Belgium, fax: +32 2 629-2850
Email: [Johan.Schoukens,Rik.Pintelon]@vub.ac.be*

Abstract: System identification often means the determination of linear models from input-output data. The behaviour of many systems can be described by an *s*-domain or *z*-domain transfer function model, at least for a given excitation amplitude range. The quality of the fit can be assessed by the analysis of the residuals, that is, of the difference between the measured data and the model.

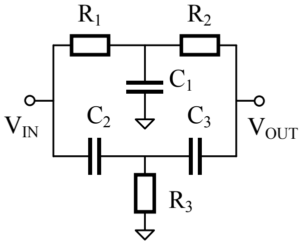
However, even slight nonlinearities can be misleading, by causing part of the residuals non-explicable by the linear model. We cannot simply tell if the excess residual error is due to undermodelling or to nonlinear system behaviour. This can lead to erroneous overmodelling. Therefore, characterisation of the nonlinear system behaviour is essential in the verification of linear models.

The Frequency Domain System Identification Toolbox has been extended with analysis tools of nonlinear system behaviour. Specially designed excitation signals allow the description of nonlinearity levels. By this, model verification becomes possible even if nonlinear error terms excess linear additive noise. *Copyright © 2006 IFAC*

Keywords: system identification, nonlinear error, random phase multisine, parameter estimation, frequency domain, MATLAB.

1 INTRODUCTION

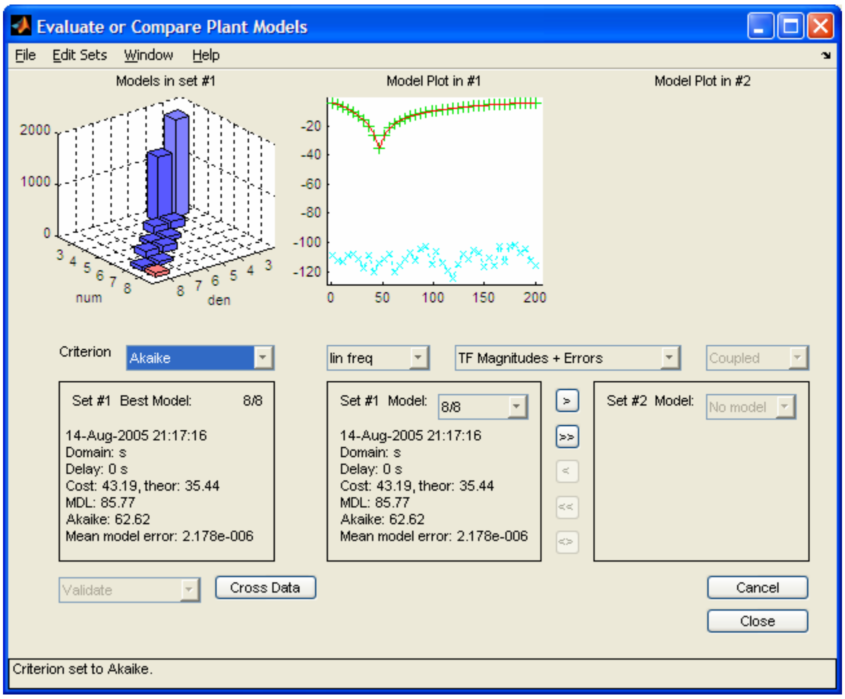
In model verification, there is a strange contradiction between low noise level of measurements and verification. Verification is usually based on analysis of the residuals. The error between measurements and the identified model is compared to the calculated noise levels. When measurements have low noise, even small nonlinearities may dominate the residuals. An example was studied by Németh *et al* (1999). A notch filer as in Fig. 1 was investigated.



R_1= R	C_1=2C
R_2= R	C_2= C
R_3=R/2	C_3= C

Fig. 1 A notch filter

[1] with support of the Flemish-Hungarian bilateral agreement B-15/04, and OTKA TS49743



Criterion Akaike

lin freq

TF Magnitudes + Errors

Coupled

Set #1 Best Model: 8/8

14-Aug-2005 21:17:16
Domain: s
Delay: 0 s
Cost: 43.19, theor: 35.44
MDL: 85.77
Akaike: 62.62
Mean model error: 2.178e-006

Set #1 Model: 8/8

14-Aug-2005 21:17:16
Domain: s
Delay: 0 s
Cost: 43.19, theor: 35.44
MDL: 85.77
Akaike: 62.62
Mean model error: 2.178e-006

Set #2 Model: No model

Criterion set to Akaike.

Fig. 2 Model validation from 3/3 to 8/8.

The measurement was done under laboratory conditions. As the central plot of Fig. 2 illustrates, the model (continuous line) well follows the FRF, the residuals (stars) are small. The signal-to-noise ratio is over 60-80 dB. Even if the equality of certain component values cannot be perfectly achieved due to component imperfections, in theory a 3/3 model should be sufficient to describe the behaviour. However, one can observe that e.g. the Akaike criterion ($cost$*$(1+n_p/F)$), where $cost$ is the value of the cost function, n_p is the number of free parameters, and F is the number of frequencies (Schoukens *et al*, 2002) consistently decreases with increasing orders, at least until 8/8 (left plot in Fig. 2).

Although the decrease is not significant from 4/4 to 8/8, the 3/3 model would certainly not be accepted, since for it the cost function and the Akaike criterion are both relatively large. On the other hand, we know that the MacMillan order of such an electronic circuit is not larger than the number of energy storing elements, that is, 3.

Why is this apparent contradiction? In validation, we compare the residuals to the calculated noise level. When the noise is small, nonlinear errors dominate the residuals. "Blind" criteria, like the Akaike criterion, indicate too large modelling errors, and force the degree of freedom of the model be larger than necessary.

How can we detect this, and keep model orders at a reasonable level? A possibility for this is to include nonlinearity products into the measure of the noise. This includes the design of appropriate excitation signals, a special "nonlinear noise" analysis, and corresponding identification procedure with model validation. This has been incorporated now into the Frequency Domain System Identification Toolbox. We will illustrate this on an example.

Nonlinear analysis can be simply switched on in the graphical user interface by pressing a taskbar button. The basic flowchart remains the same, but the behaviour of individual elements change.

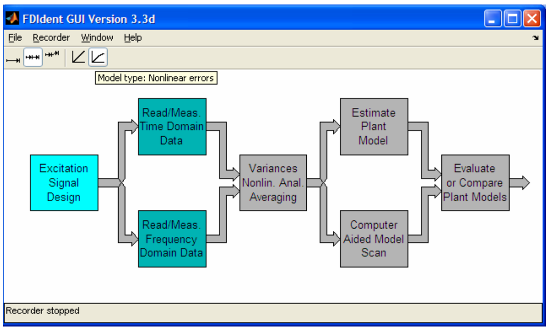

FDIdent GUI Version 3.3d
File
Recorder
Window
Help
Model type: Nonlinear errors
Excitation Signal Design
Read/Meas. Time Domain Data
Read/Meas. Frequency Domain Data
Variances Nonlin. Anal. Averaging
Estimate Plant Model
Computer Aided Model Scan
Evaluate or Compare Plant Models
Recorder stopped

Fig. 3 Switching nonlinear analysis on

The basic idea is as follows. The deviation of the measurements from the model can be caused in general by the following sources:

- observation noise
- nonlinearities
- modelling errors (e.g. undermodelling)
- transients during measurement

During identification, we eliminate transients by executing measurement in steady-state, or modelling transients, and choose a transfer function model which is complex enough to be able to describe the linear behaviour. Therefore, in a good model, the first two items in the above list describe the residuals. If we have a good descriptor of *both* of them, we can verify the model.

The problem is that nonlinear products are bound to the input signal. Therefore, they cannot be analysed by repeating the experiments, since the same input signal produces the same errors.

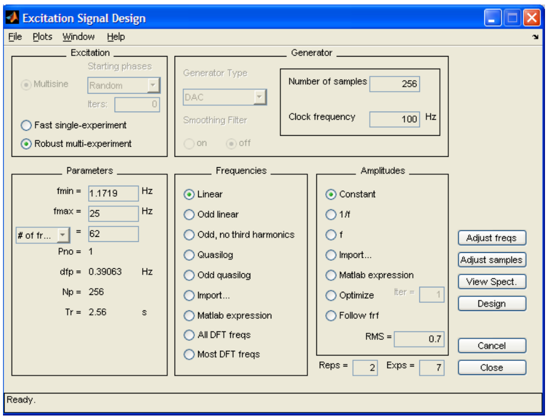
There are two ideas which can help in this (Schoukens *et al*, 2001).

One is to use signals which produce FRF results which are similar, so averaging and variance analysis of them is doable, while the nonlinear errors are different. Random phase multisines fulfil these conditions. The random phase of the components yields a different combination of nonlinear errors for each realisation, while we can keep the excitation level at each frequency constant. The random phase is eliminated by calculating the FRF, or by turning the phases of the Fourier amplitudes to the same direction by using a reference signal (this can be available since we only need the noiseless excitation signal). In such a set of experiments, nonlinear errors behave like Gaussian random noise (Pintelon and Schoukens, 2001). We can execute variance analysis, and use the "nonlinear" variances for identification and validation. This eliminates the problem of nonlinear errors present, since we compare the residuals to the *combined* "variance", describing both the observation noise and nonlinear errors. The only difficulty is that we have to generate a set of different multisines, and execute a series of experiments – the toolbox can help with generating these excitation signals automatically. This method is called "robust" method.

Another possibility is to observe the effect of nonlinearities at certain frequencies where no excitation is present. If a special multisine is generated which contains zero excitation at certain frequencies, here the output signal contains here only noise and nonlinearity products. If the frequency grid is furthermore restricted to odd frequencies, the effect of even nonlinearities is avoided. The calculated "nonlinear variances" can be used to interpolate these "variances" at the excited frequencies, so we obtain FRF values with "nonlinear variances", describing the level of residuals as combined from noise and nonlinear products, so validating the model when no modelling error is present. This method is called "non-robust" or "fast and simple" method.

2 RANDOM PHASE EXCITATION

As we have seen above, we need a few random phase multisines, theoretically at least 7 of them. The graphical user interface generates these, the user needs to export the signals and execute the experiments. In this example, each multisine contains two repeated periods, to enable the analysis of noise alone (see below).



Excitation
Starting phases
● Multisine Random
Iters: 0
○ Fast single-experiment
● Robust multi-experiment

Generator
Generator Type DAC
Smoothing Filter ○ on ● off
Number of samples 256
Clock frequency 100 Hz

Parameters
fmin = 1.1719 Hz
fmax = 25 Hz
# of fr... = 62
Pno = 1
dfp = 0.39063 Hz
Np = 256
Tr = 2.56 s

Frequencies
● Linear
○ Odd linear
○ Odd, no third harmonics
○ Quasilog
○ Odd quasilog
○ Import...
○ Matlab expression
○ All DFT freqs
○ Most DFT freqs

Amplitudes
● Constant
○ 1/f
○ f
○ Import...
○ Matlab expression
○ Optimize Iter = 1
○ Follow frf
RMS = 0.7

Reps = 2 Exps = 7

Ready.

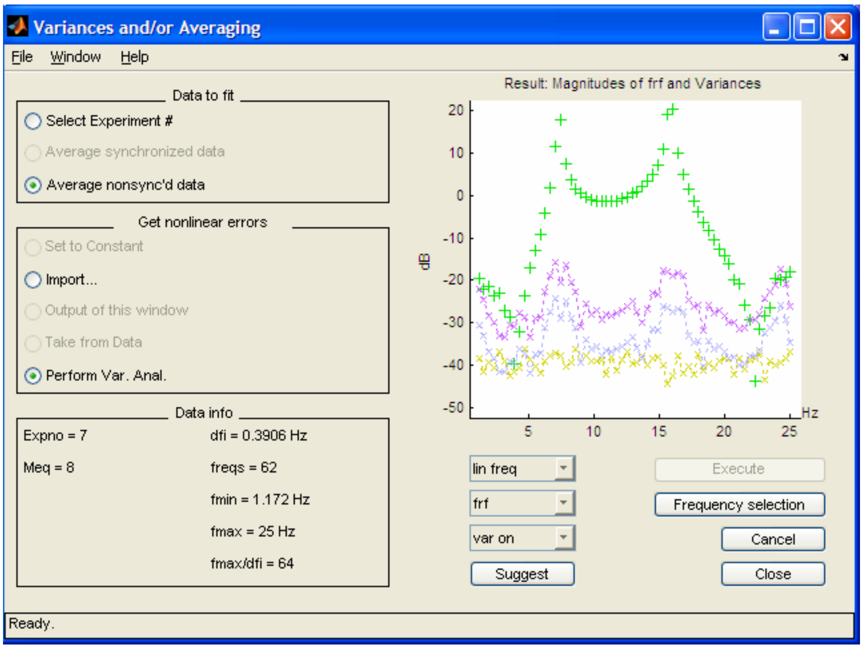
Fig. 4 Excitation signal design for random phase multisines

When the results of all experiments are imported to the graphical user interface, variance analysis can be executed. Fig. 5 illustrates the result.

The green[2] + signs at the top illustrate the FRF. Below it, the blue x marks around approx. -20 dB denote the nonlinearity level of one experiment. This curve is repeated in light blue by $10*\log_{10}(7)$ dB lower (we have used 7 experiments). These are the "variance" values which will be used when fitting the average of the experiments and during validation. Finally, the noise level is shown with yellow x marks at about –40 dB, well illustrating that the noise is significantly smaller than the "nonlinear noise", so model validation using linear noise would be misleading.

Model validation is illustrated in Fig. 6. The nonlinear upper limits bound the residuals from above (left plot), while the linear ones do not (right plot).

[2] The description refers to colors on the screen – for black and white printouts, form and darkness of the markers hopefully gives sufficient clue for the Reader.



Data to fit

○ Select Experiment #
○ Average synchronized data
● Average nonsync'd data

Get nonlinear errors

○ Set to Constant
○ Import...
○ Output of this window
○ Take from Data
● Perform Var. Anal.

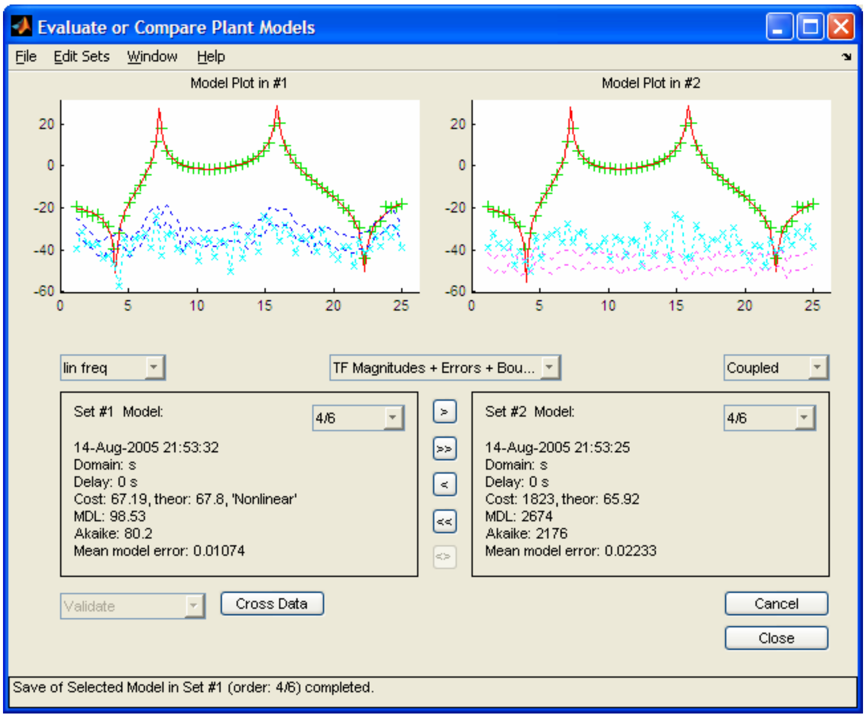
Data info

Expno = 7
Meq = 8
dfi = 0.3906 Hz
freqs = 62
fmin = 1.172 Hz
fmax = 25 Hz
fmax/dfi = 64

lin freq
frf
var on

Ready.

Fig. 5 The result of noise analysis using random phase multisines



lin freq
TF Magnitudes + Errors + Bou...
Coupled

Set #1 Model: 4/6
14-Aug-2005 21:53:32
Domain: s
Delay: 0 s
Cost: 67.19, theor: 67.8, 'Nonlinear'
MDL: 98.53
Akaike: 80.2
Mean model error: 0.01074

Set #2 Model: 4/6
14-Aug-2005 21:53:25
Domain: s
Delay: 0 s
Cost: 1823, theor: 65.92
MDL: 2674
Akaike: 2176
Mean model error: 0.02233

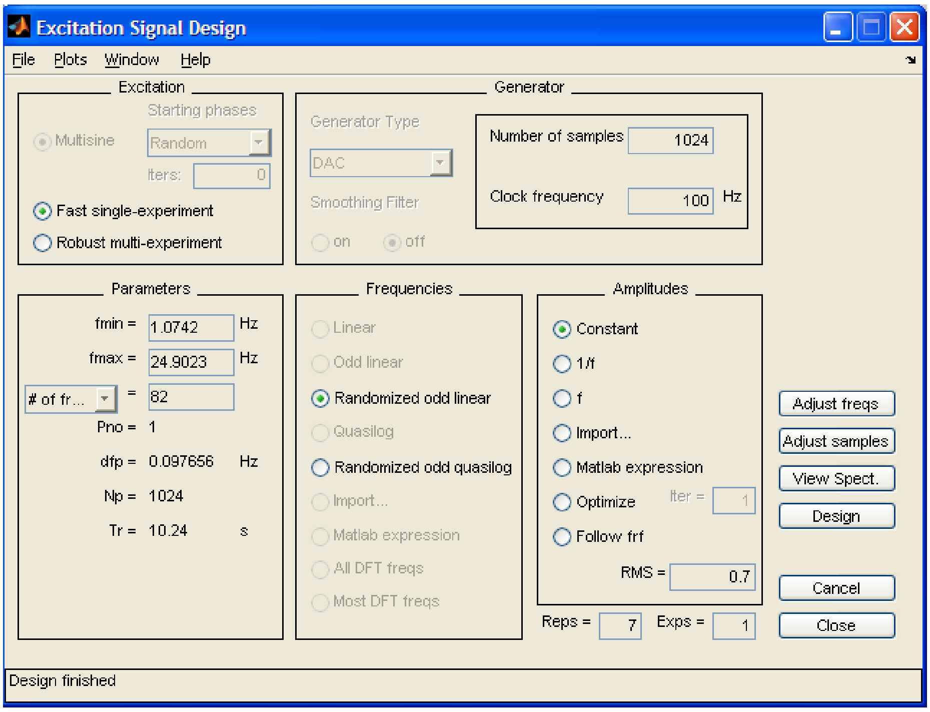
Save of Selected Model in Set #1 (order: 4/6) completed.

Fig. 6 Model validation using random phase multisines. Left the result using nonlinear error analysis, right the result based on pure linear analysis.

Let us observe that the two results (nonlinear and linear) do not really differ in the identified model. Although the residuals are not exactly the same, but they are at the same level. The main difference is in validation. Nonlinearities and noise do account for the residuals, while linear noise alone does not. Using the new technique, we can validate the model. Linear analysis shows too large residuals, but cannot go further.

3 RANDOMISED-GRID MULTISINE

The “fast and simple” method for nonlinear error characterisation requires the design of a randomised-grid multisine, that is, an odd-harmonic one with some non-excited frequencies scattered along the frequency axis.



Excitation

Starting phases

● Multisine Random

Iters: 0

● Fast single-experiment

○ Robust multi-experiment

Generator

Generator Type DAC

Smoothing Filter ○ on ● off

Number of samples 1024

Clock frequency 100 Hz

Parameters

fmin = 1.0742 Hz

fmax = 24.9023 Hz

# of fr... = 82

Pno = 1

dfp = 0.097656 Hz

Np = 1024

Tr = 10.24 s

Frequencies

○ Linear

○ Odd linear

● Randomized odd linear

○ Quasilog

○ Randomized odd quasilog

○ Import...

○ Matlab expression

○ All DFT freqs

○ Most DFT freqs

Amplitudes

● Constant

○ 1/f

○ f

○ Import...

○ Matlab expression

○ Optimize Iter = 1

○ Follow frf

RMS = 0.7

Reps = 7 Exps = 1

Design finished

Fig 7 Design of a randomised-grid multisine

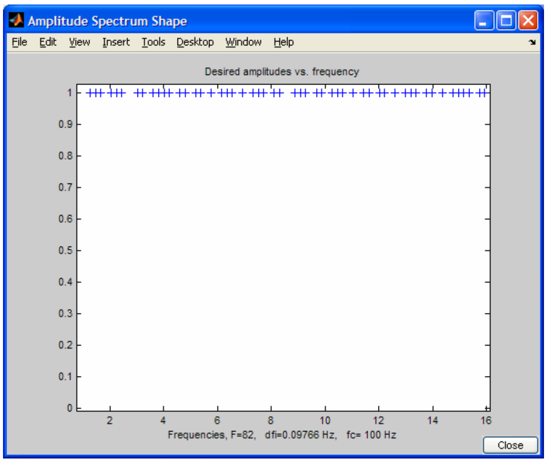
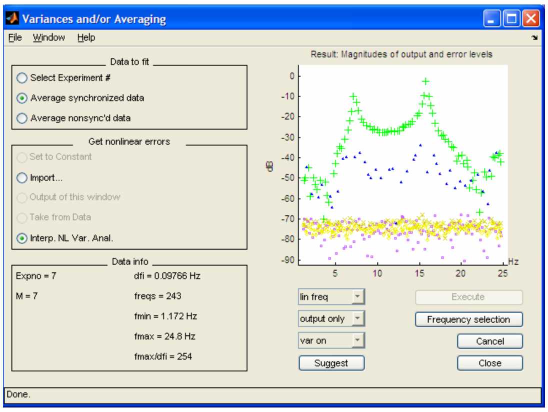


Fig. 8 Randomised frequency grid



Data to fit

○ Select Experiment #

● Average synchronized data

○ Average nonsync'd data

Get nonlinear errors

○ Set to Constant

○ Import...

○ Output of this window

○ Take from Data

● Interp. NL Var. Anal.

Data info

Expno = 7 dfi = 0.09766 Hz

M = 7 freqs = 243

fmin = 1.172 Hz

fmax = 24.8 Hz

fmax/dfi = 254

lin freq

output only

var on

Done.

Fig. 9 Variance analysis based on randomised-grid multisine

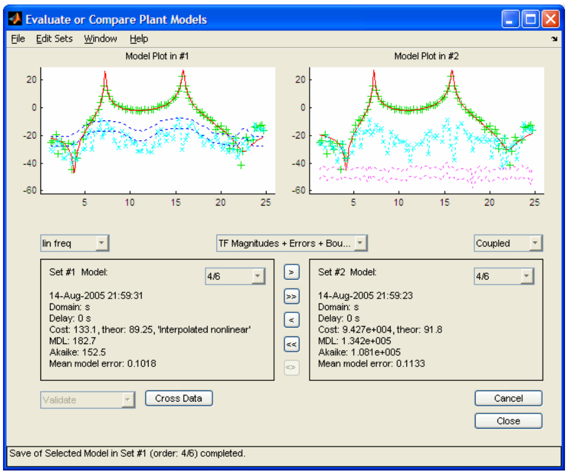
Users of the toolbox are not required to go into details of the whole signal design and analysis. It is only necessary to understand the results (Fig. 9).

The green[3] + marks denote the calculated FRF. The blue triangles illustrate the nonlinearity levels calculated at the non-excited odd lines. The yellow x marks show the noise level. The light blue squares show the nonlinearities at the even lines.

These are non-excited lines like the selected ones at odd frequencies. In our case they are at noise level, since the nonlinearity we applied to the signal is odd, so no even products are present.

Fig. 10 illustrates the results of model validation. This figure is very similar to Fig. 6. The nonlinearity-based model is validated (the residuals are below the theoretical limits), while the linear noise based model is not.

[3] The description refers to colors on the screen – for black and white printouts, form and darkness of the markers hopefully gives sufficient clue for the Reader.



Set #1 Model:

14-Aug-2005 21:59:31
Domain: s
Delay: 0 s
Cost: 133.1, theor: 89.25, 'Interpolated nonlinear'
MDL: 182.7
Akaike: 152.5
Mean model error: 0.1018

Set #2 Model:

14-Aug-2005 21:59:23
Domain: s
Delay: 0 s
Cost: 9.427e+004, theor: 91.8
MDL: 1.342e+005
Akaike: 1.081e+005
Mean model error: 0.1133

Save of Selected Model in Set #1 (order: 4/6) completed.

Fig. 10 Model validation using a randomised-grid multisine. Left the result using nonlinear error analysis, right the result based on pure linear analysis.

4 SUMMARY

The Frequency Domain System Identification Toolbox has been extended by tools for nonlinear system characterization. Linear models can be properly validated now even in the presence of nonlinear errors, and the error level is characterized. The key is to use special excitation signals: random-phase multisines or randomized-grid ones.

REFERENCES

The papers concerning the toolbox are available in electronic form from the WEB-page http://www.mit.bme.hu/~kollar/publist-KPA.htm

I. Kollár (1999-2005) *Frequency Domain System Identification Toolbox for Matlab.* `http://elec.vub.ac.be/fdident/`

I. Kollár, R. Pintelon and J. Schoukens (1991), “Frequency Domain System Identification Toolbox for Matlab”. *Proceedings of the 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation, Budapest, July 8-12, 1991.* Edited by Cs. Bányász and L. Keviczky. Pergamon Press, Oxford, England, pp. 1243-46.

I. Kollár, R. Pintelon and J. Schoukens (1994), “Frequency Domain System Identification Toolbox for Matlab: A Complex Application Example.” *Sysid'94, X. IFAC/IFORS International Symposium on System Identification and Parameter Estimation, Copenhagen, July 4-6, 1994.* Vol. 4, pp. 1619-24.

I. Kollár, R. Pintelon and J. Schoukens (1997), “Frequency Domain System Identification Toolbox for Matlab: Improvements and New Possibilities”. *Sysid'97, XI. IFAC/IFORS International Symposium on System Identification and Parameter Estimation, Fukuoka, 7-11 July, 1997.* Vol. 2, pp. 991-94.

I. Kollár, J. Schoukens, R. Pintelon, G. Simon and G. Román (2000), “Extension for the Frequency Domain System Identification Toolbox: Graphical User Interface, Objects, Improved Numerical Stability.” *Preprints of the IFAC Symposium on System Identification, SYSID 2000, 21-23 June 2000, Santa Barbara, CA, USA. CD-ROM.*

I. Kollár, R. Pintelon, Y. Rolain, J. Schoukens, and Gy. Simon (2003a), “Frequency Domain System Identification Toolbox for Matlab: Automatic Processing – From Data to Models.” *IFAC Symposium on System Identification, SYSID 2003*, Aug. 2003, Rotterdam. *CD-ROM.*

I. Kollár, R. Pintelon, J. Schoukens, and Gy. Simon (3003b), “Complicated Procedures Made Easy – A MATLAB Toolbox Using an Efficient GUI,” *IEEE Instrumentation and Measurement Magazine*, Vol. 6, No. 3, pp. Sep. 2003. pp. 19-26.

J. Németh, B. Vargha and I. Kollár, “Online Frequency Domain System Identification based on a Virtual Instrument,” *Proc. IEEE Instrumentation and Measurement Technology Conference, IMTC/99, Venice, Italy, May 24-26, 1999.* Vol. 2, pp. 1211-16.

R. Pintelon, and J. Schoukens (2001). *System Identification. A Frequency Domain Approach.* IEEE Press, Piscataway, NJ.

J. Schoukens, R. Pintelon, Y. Rolain, and T. P. Dobrowiecki (2001), “Frequency response functions measurements in the presence of nonlinear distortions,” Automatica, Vol. 37, pp. 939-946.

J. Schoukens, Y. Rolain, and Pintelon (2002), “Modified AIC rule for model selection in combination with prior estimated noise models,” Automatica, Vol. 38, No. 5, pp. 903-906.