

Fitting quad suspension high frequency dynamics with the FDIDENT tool

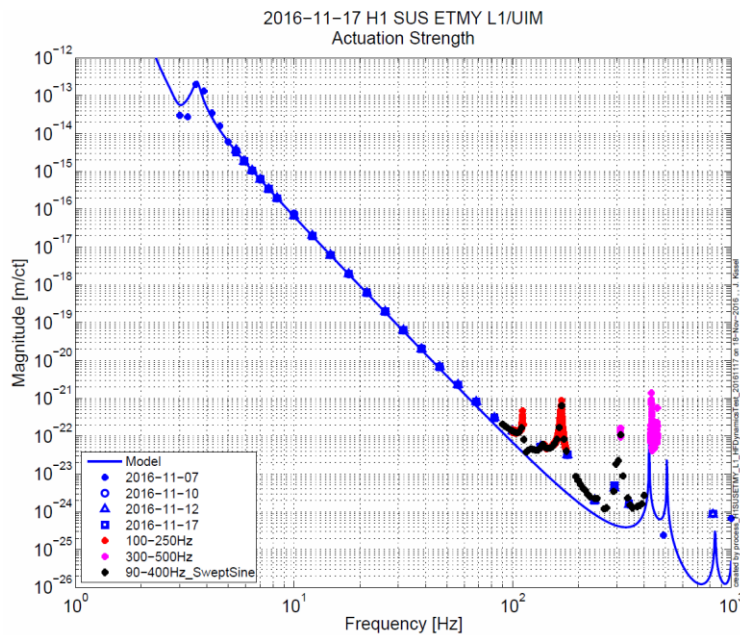
Dennis Coyne, 8-Mar-2017

Contents

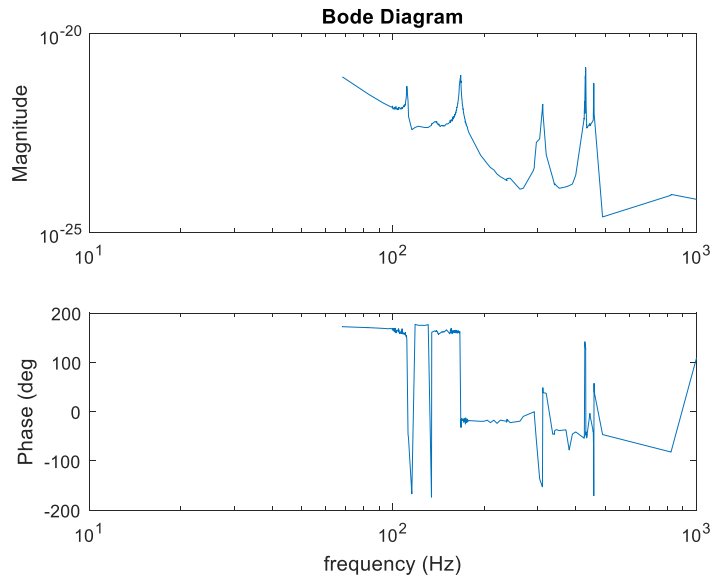
Quad suspension high frequency dynamics	1
FDIDENT	2
Unstable, Non-Minimum Phase	3
Stable, Minimum Phase	8
Stable, Non-Minimum Phase, with delay	8
Stable, Minimum Phase in two segments.....	11
Stable, Non-Minimum Phase, without delay.....	12
Matlab script.....	13

Quad suspension high frequency dynamics

Jeff Kissel and Darkhan Tuyenbayev found a number of dynamical, resonant features in the UIM/L1 actuation stage in the frequency range of ~100 to ~500 Hz, as reported in [LHO elog #31603](#). Their measurements are a combination of swept-sine (SS) and broad-band (BB) transfer function ratios, as indicated in the figure below.



They requested a pole-zero fit of this transfer function above 20 Hz that can be used as a "correction" filter to a model that falls off as f^6 . This filter does not need to perfectly resolve the details of all of the high-Q features, but it must track the overall frequency dependence over the 20 - 500 Hz region well. The measurements uploaded to the elog entry actually only go as low as 68 Hz, not 20 Hz. For the time being we'll work with this set, shown in the figure below, and assume an f^{-6} rolloff below, and above the frequency range of the measurements.



FDIDENT

The Frequency Domain System Identification toolbox (FDIDENT) is a third party Matlab toolbox which provides tools for identifying linear dynamic single-input/single-output (SISO) systems from time response or frequency response measurements. The toolbox is free for non-profit use.

https://www.mathworks.com/products/connections/product_detail/product_35570.html

<http://home.mit.bme.hu/~kollar/fdident/>

Measurement variances should be provided with the input & output measurements (in time or frequency domain). The FDIDENT code is based on the Estimation of Linear Systems (ELiS) program described in this text:

J. Schoukens, R. Pintelon, "Identification of Linear Systems: A Practical Guideline to Accurate Modeling", Pergamon Press.

ELiS uses the variances to determine how to best fit to the data. Since variance data was not available for the data set (or at least not posted in [LHO elog #31603](#)), I created fake variances.

In addition to the lack of measurement variances, the measurement set has 27 duplicate frequencies. Since the frd function can't handle duplicates, these measurements were simply dropped. (A better approach would be to average these measurements.)

Unstable, Non-Minimum Phase

If the fit that one seeks can be unstable and non-minimum phase and the delay is free parameter to fit, then one can get very good fitting results of very high order using fdident. For example, consider the fit shown in the figure below with a numerator of order 80 and a denominator of order 86 (with 6 poles at zero frequency). While the residual errors in magnitude and phase are low, the variation with frequency between measurement points is unlikely, or non-physical. In particular note that the two highest frequency points are well separated from the other measurements, which can lead to artificial dynamics in the fit.

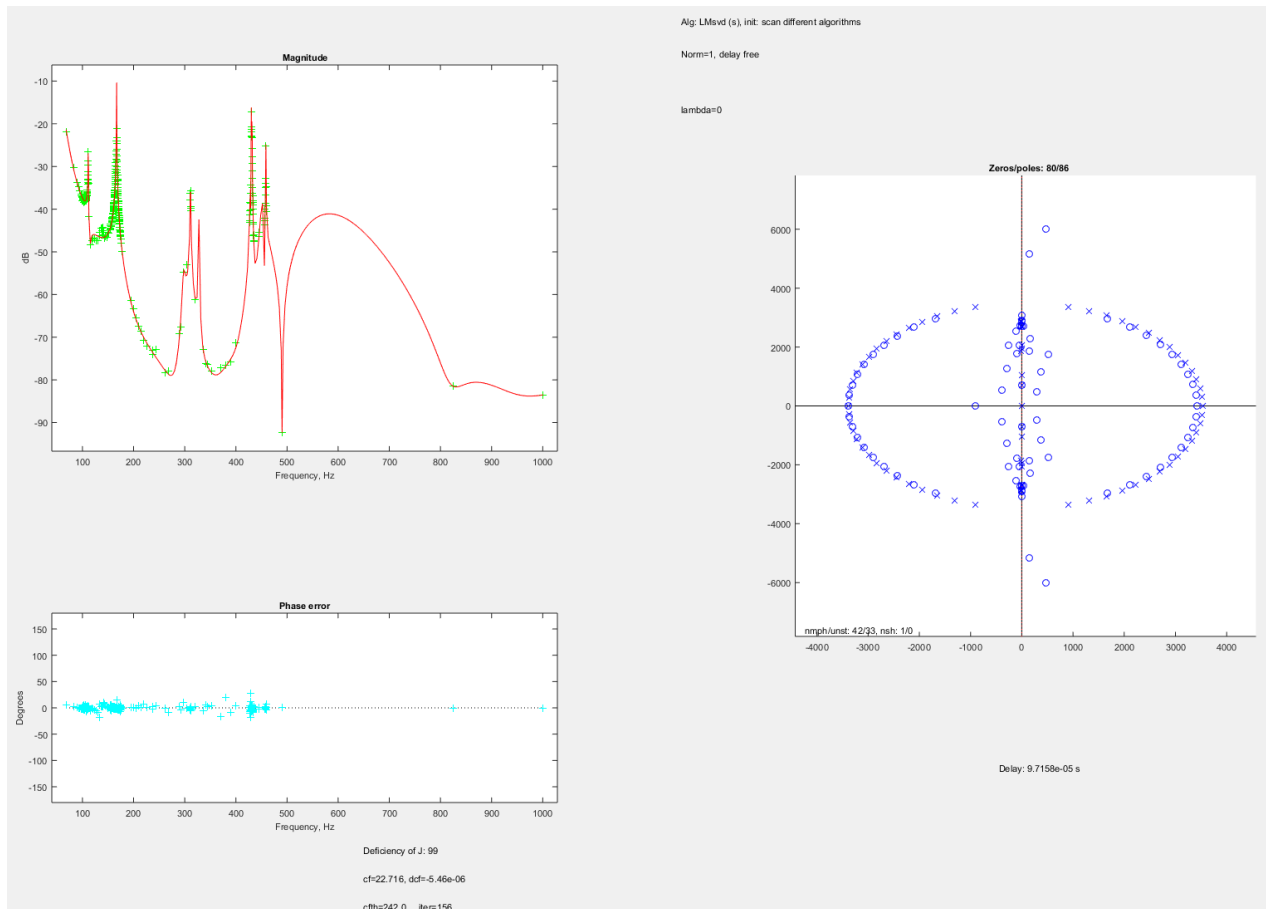
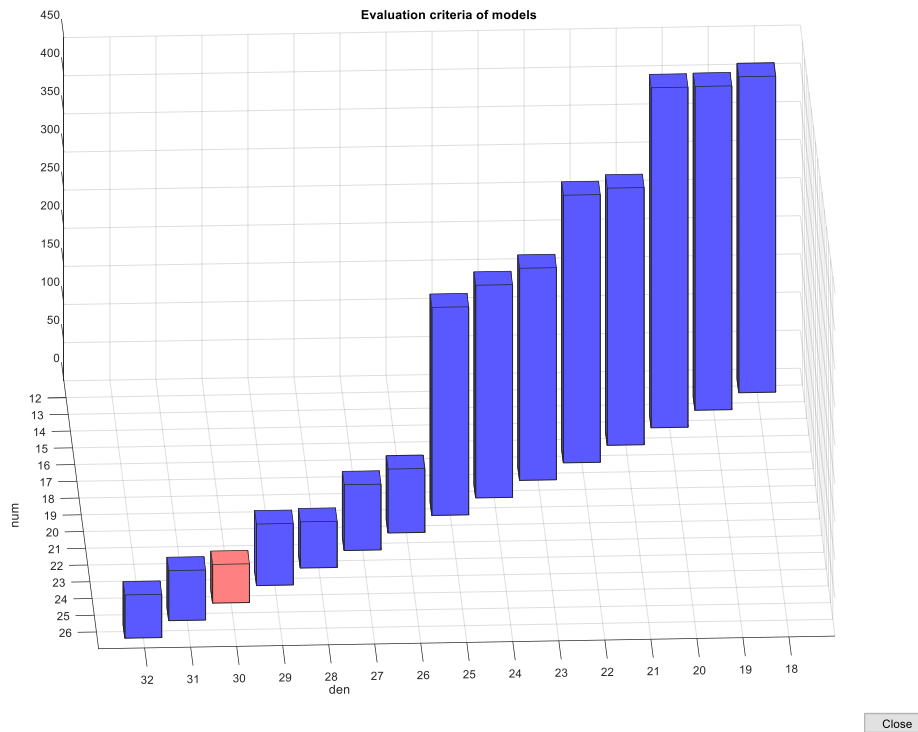


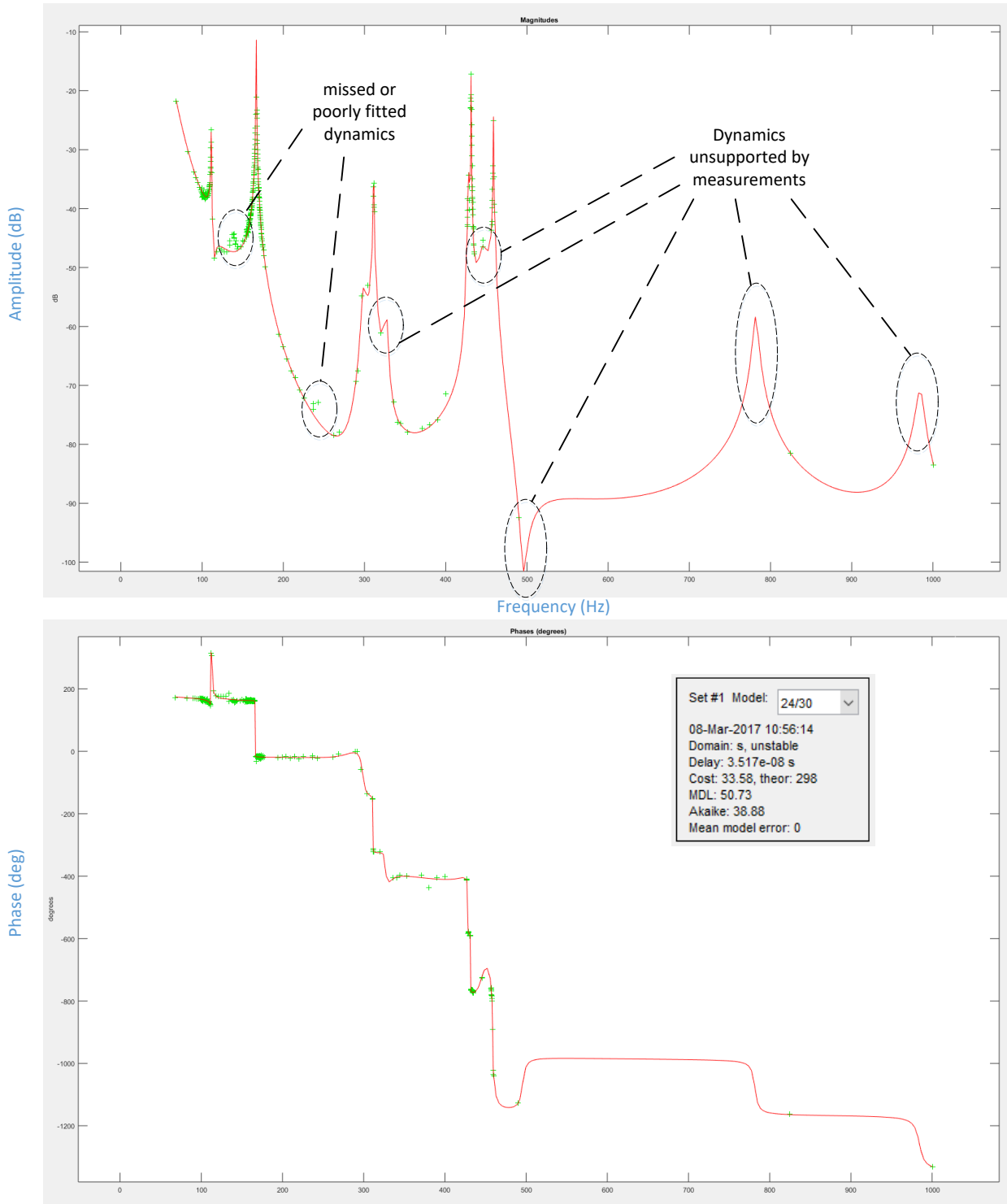
Figure 1 FDIDENT (Elis) result with a very high order model. Upper left is a plot of the magnitude with green “+” symbols denoting the measured points. The lower left plot indicates the phase errors of the fit. The right plot is a pole-zero diagram of the fit.

Using the automatic sequencing of models (“computer aided model scan”), with the constraint that the denominator has 6 poles at zero frequency, but the time delay is variable (can be fitted) results in a minimum cost function (best fit) with a numerator of 24th order and denominator of 30th order (the “unst, nmph, order 24:30 model”).

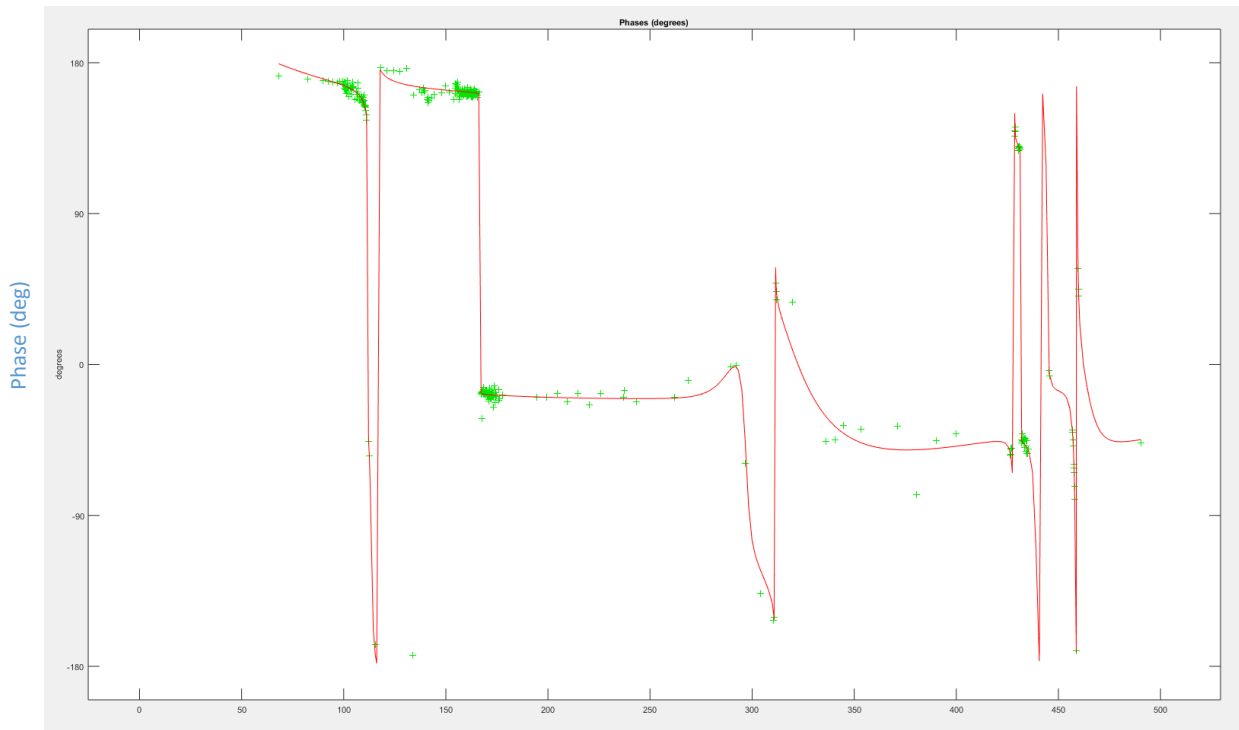
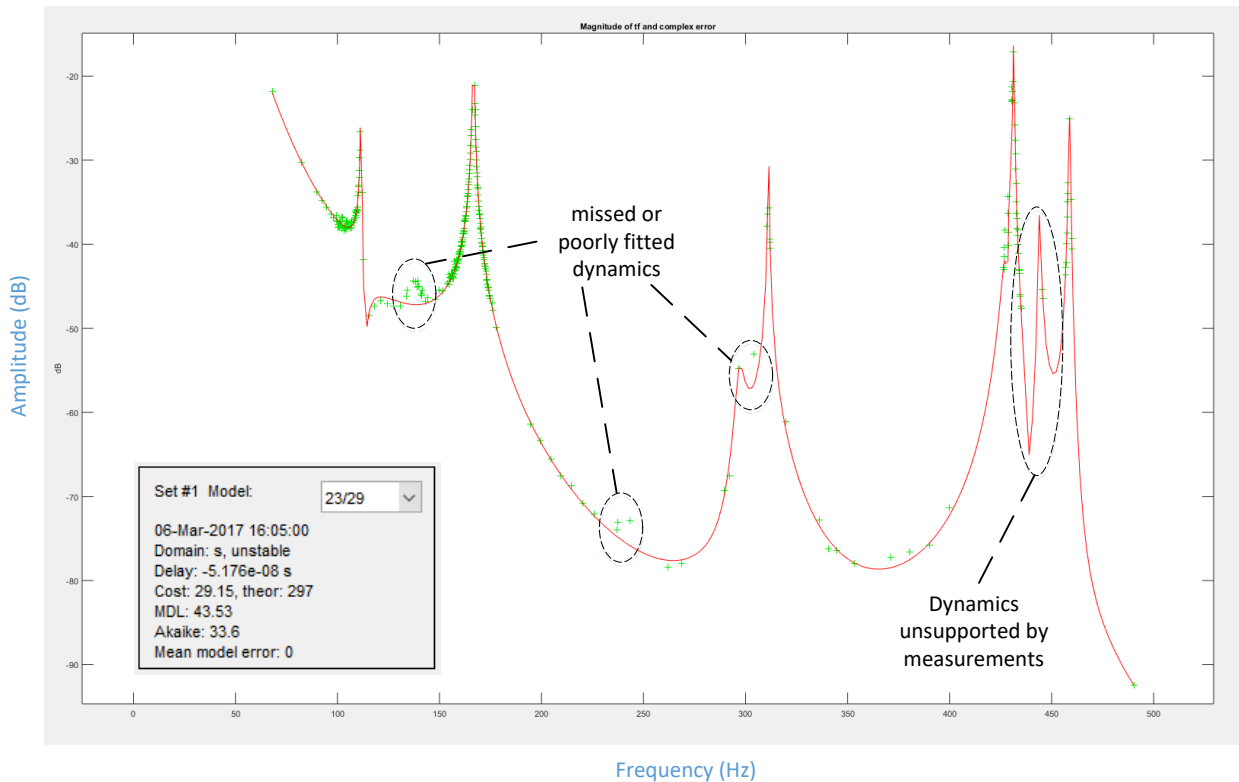


However the unstable, non-minimum phase, order 24:30 model fit is not very satisfying, as indicated in the bode diagram below. Most significantly, the model is not stable and is non-minimum phase. Certainly the physical plant/system is stable. Perhaps for use in calibration an unstable, but smooth function fit, is acceptable. However it would be better if the fit was physically realistic as well.

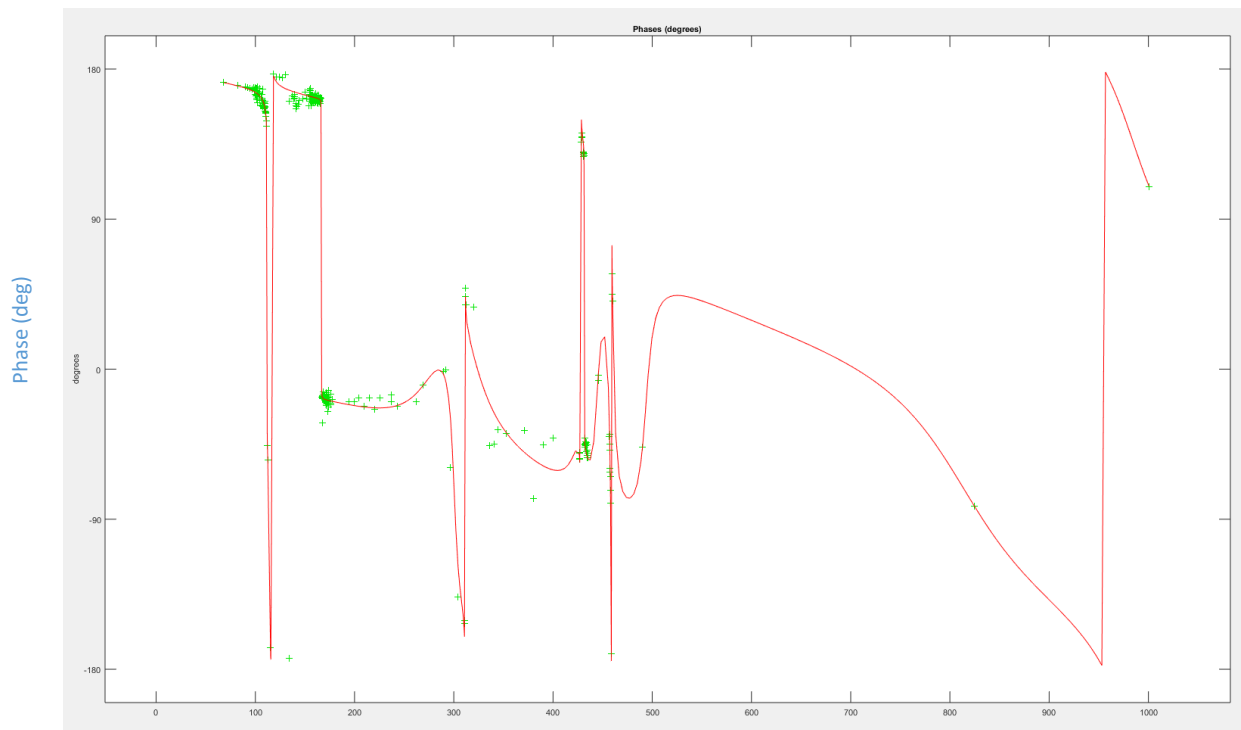
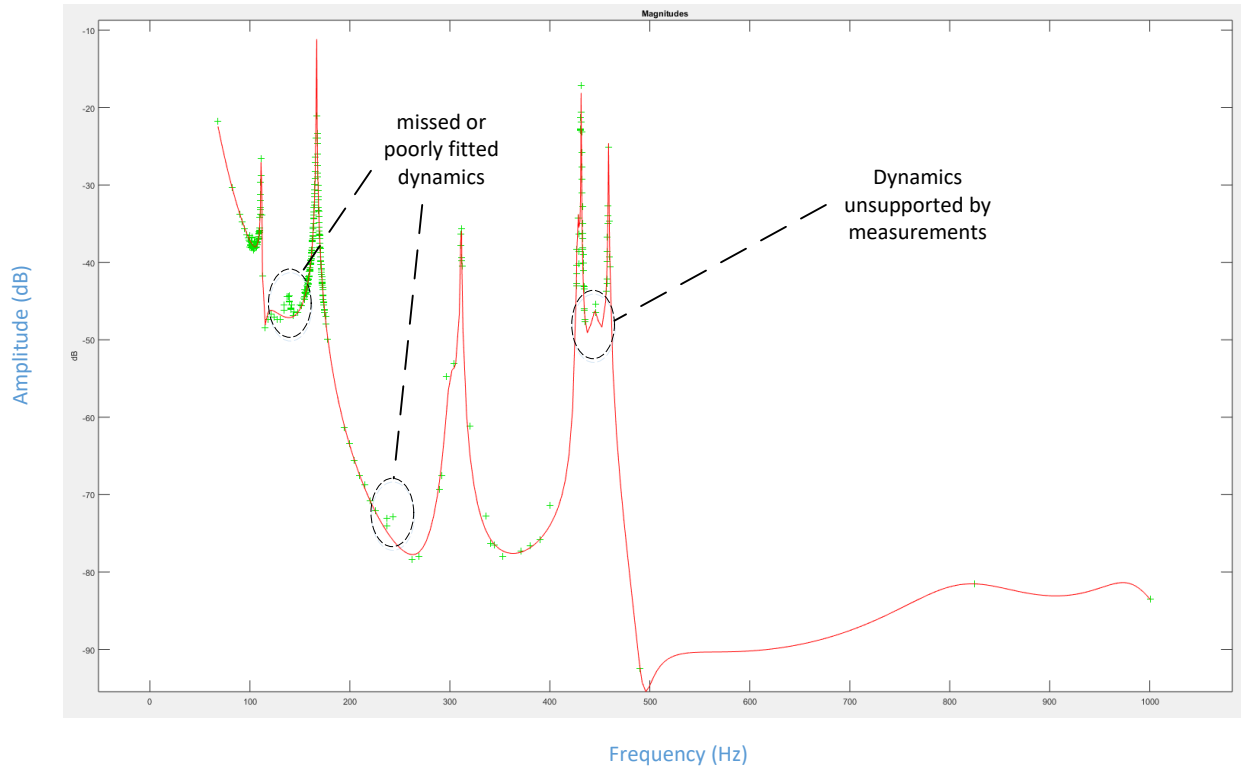
In addition, the fit introduces dynamics between some of the data points, in order to reduce the error at the frequency of the measurement. This results in large deviations from a likely transfer function especially between the last two widely spaced points.



If the two highest data points are dropped, the best unstable, non-minimum phase, fit is an order 24:30 model. However this fit is also not very satisfying (besides being unstable and non-minimum phase), as indicated in the bode diagram below.



The best unst, nmph fit, by eye (as opposed to minimizing the cost function) for all data points (other than duplicates), including the two highest frequency points is a 22:28 order model as shown below.



This model is saved as m2228. In order to use the model in Matlab, if you have fdident toolbox installed, then issue the following command:

```
>> load m2228fid
```

If you don't have fdident installed, then issue the command:

```
>> load m2228ss
```

Or

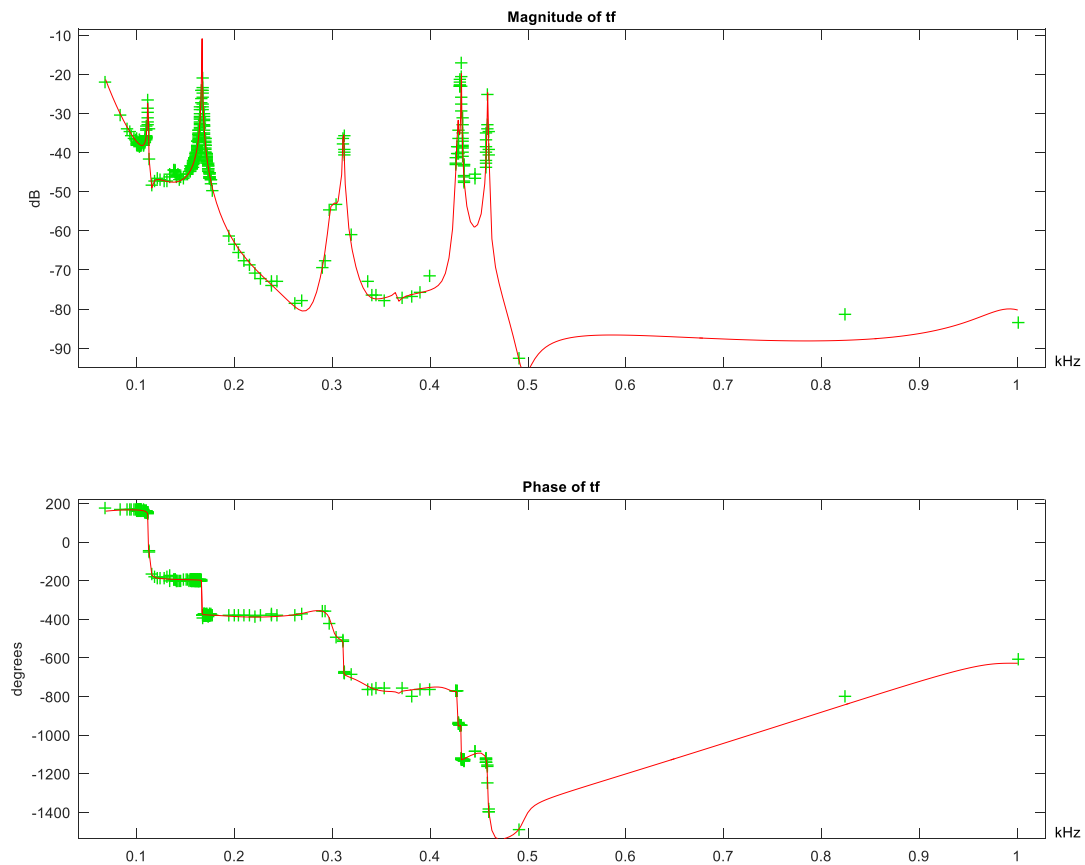
```
>> load m2228zpk
```

Stable, Minimum Phase

If we impose the requirement that the system be stable and minimum phase, no decent fits result. In addition an error ("elis error: Matrix dimensions must agree") occurs for a models of order 21:27 or higher.

Stable, Non-Minimum Phase, with delay

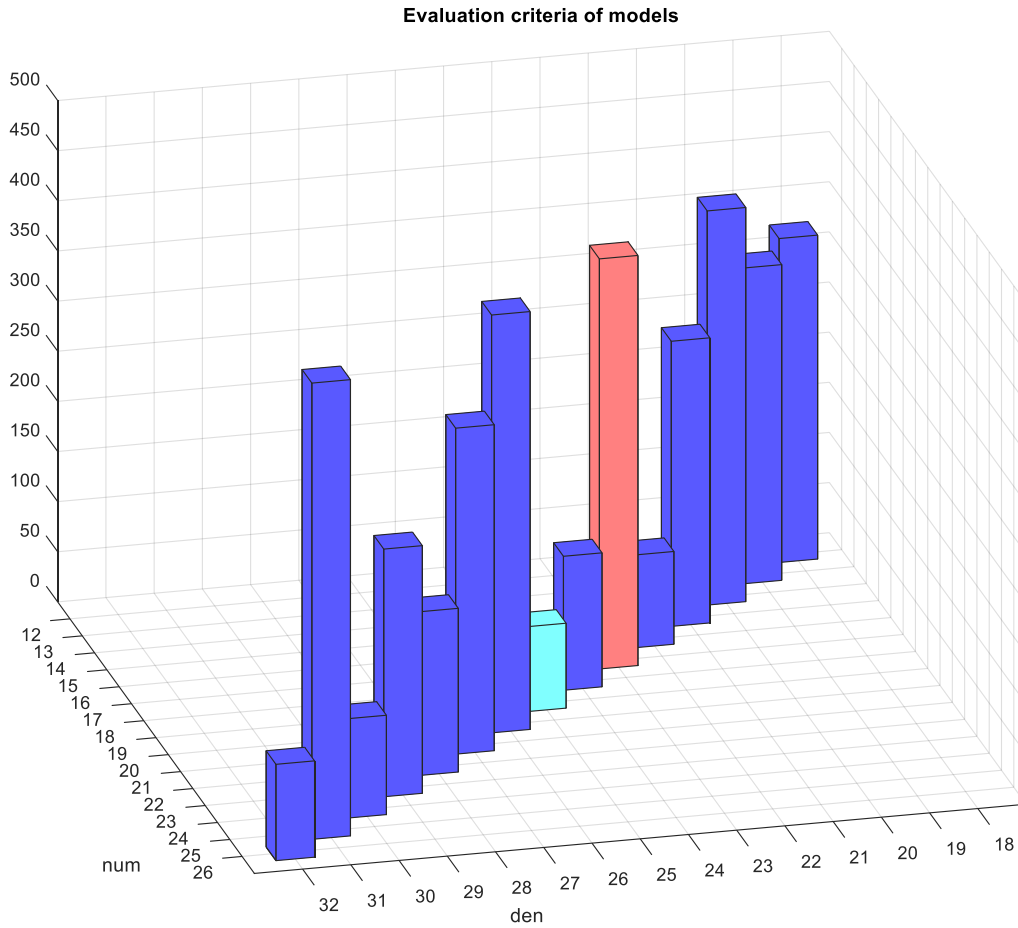
If we impose the requirement that the system be stable but allow non-minimum phase, reasonable fits result. It is possible that the system is in fact, non-minimum phase. The best fit for all data (except duplicate frequency data) is a 24:30 order model, with delay = -5.17 ms, as shown below.



This model is saved as m2430:

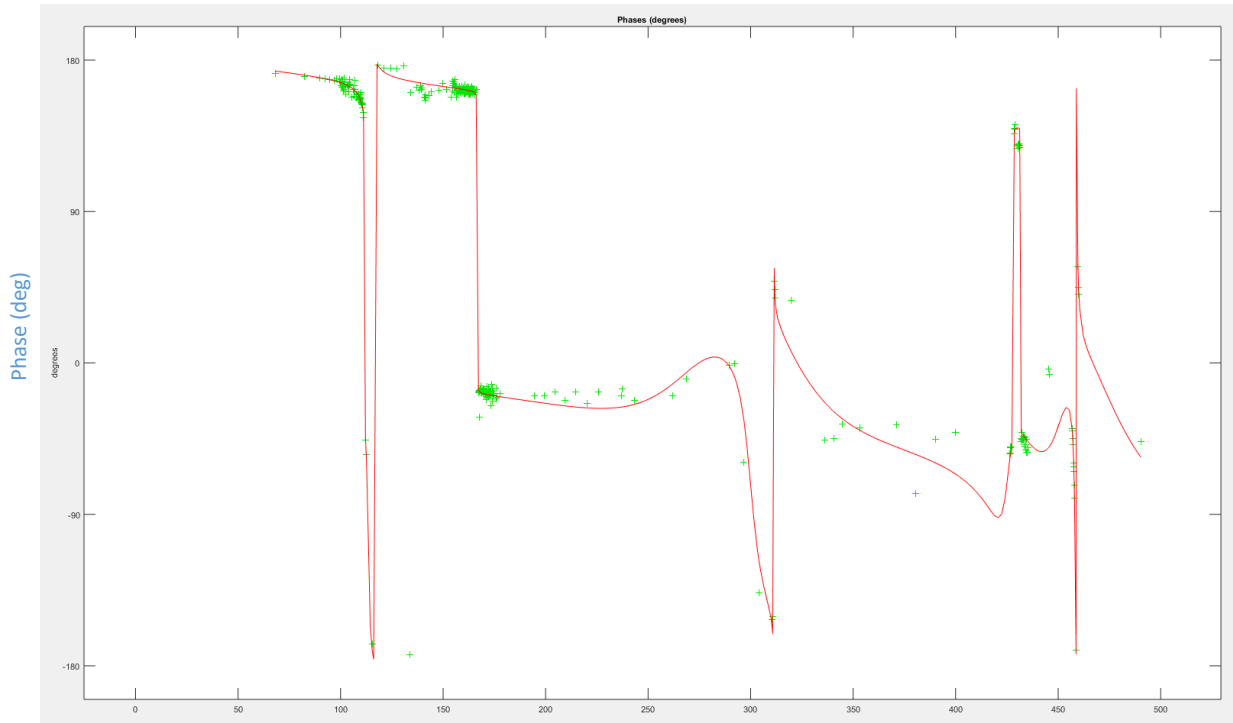
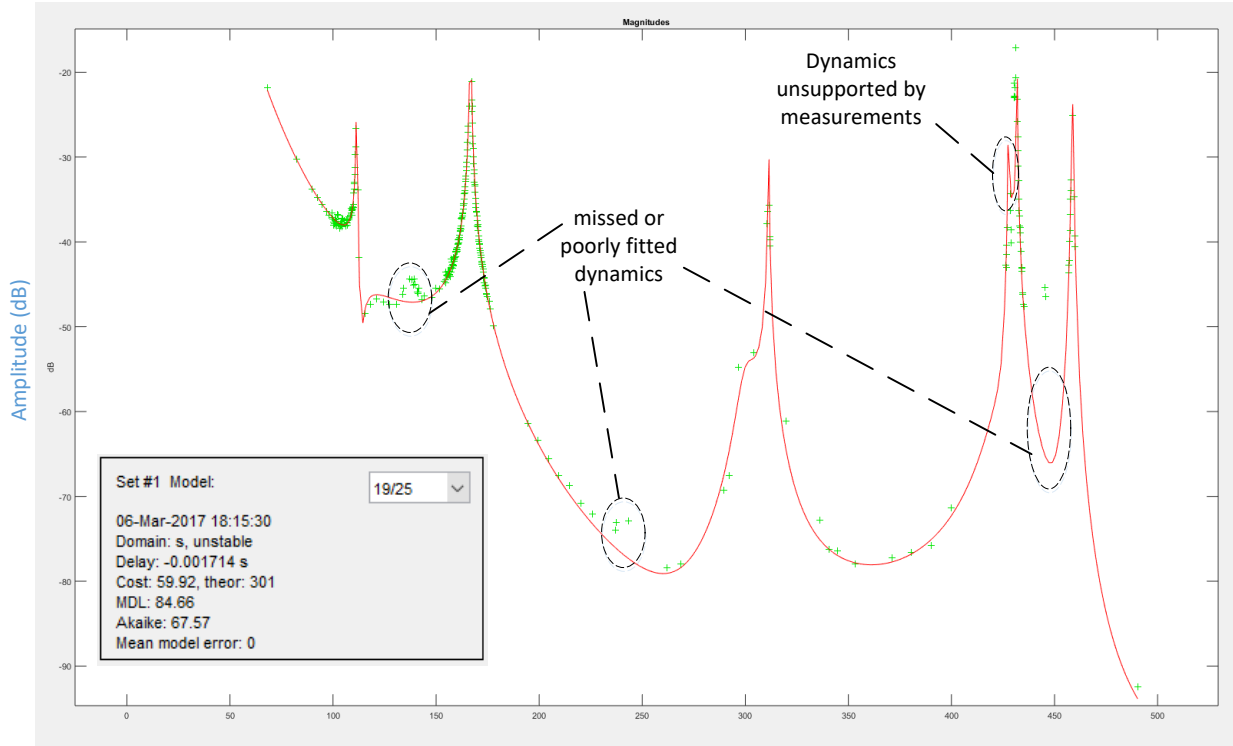

```
>> load m2430fid
>> load m2430ss
>> load m2430zpk
```

The best fit (without the two highest frequency points) is for the 19:25 order model.



Close

However the stable, non-minimum phase, order 19:25 model suffers from problems similar to the unst, nmp, order 23:29 model, as shown in the bode plot below.

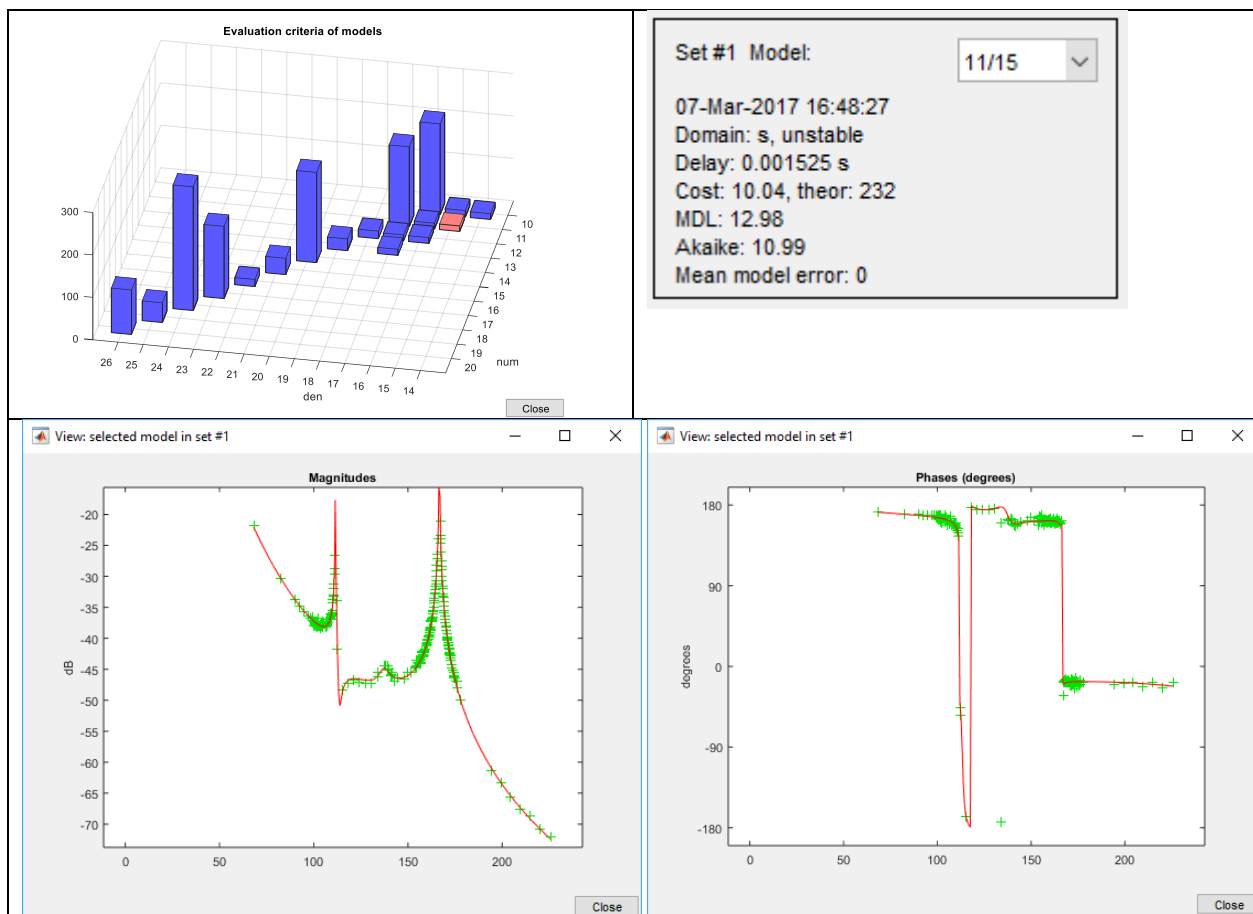


Stable, Minimum Phase in two segments

Since the high frequency dynamics are somewhat separated into two frequency ranges, 68 Hz to ~225 Hz and ~195 Hz to 490 Hz, we can consider trying to fit these two ranges separately and then ‘stiching’ these two fits together.

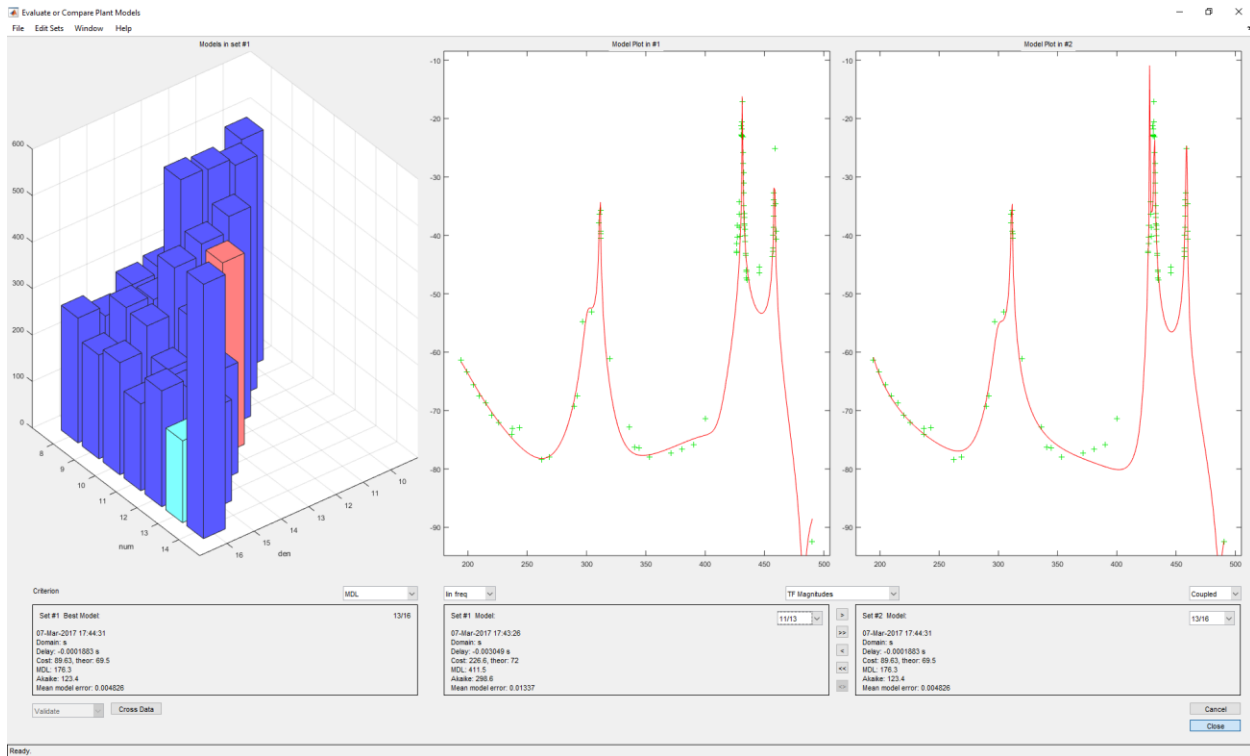
Again, attempts to fit a stable, non-minimum phase resulted in very poor results using the “reflection” or “contraction” methods. The “limitation” method results in very good magnitude fits, but spectacularly poor phase fits.

Stable, non-minimum phase fits were successful. For the first frequency range the best fit was a stable, non-minimum phase order 10:14 model (st,nmph,11:15 model), as indicated by the cost function plot below.



Note that the best model fit for the first (lower) frequency range (st, nmph, order 11:15 model) has a denominator order which is 4 higher than the numerator order. As a consequence we would like the fit for the second (higher) frequency range to have a denominator order which is 2 higher than the numerator order in order to have an f^{-6} roll-off at frequencies above these two fits.

The best fit for the second (higher) frequency range is a st, nmph, order 13:16 model. However, as indicated in the figure below, it suffers from similar fitting problems mentioned previously. In fact the st, nmph, order 11:13 model appears to fit better, or at least as well.



In principle these two model fits can be “stitched” together by ensuring matching slope in the overlapping frequency range (~195 Hz to ~225 Hz) and adjusting the overall gain to match the measurements. ... but I’ve not attempted this stitching as yet.

Stable, Non-Minimum Phase, without delay

This would seem to be the best approach – compatible with a Linear Time Invariant (LTI) representation. The best fit is a 22:28 order model, m2228, as shown in comparison to the measurements in the bode plot below.

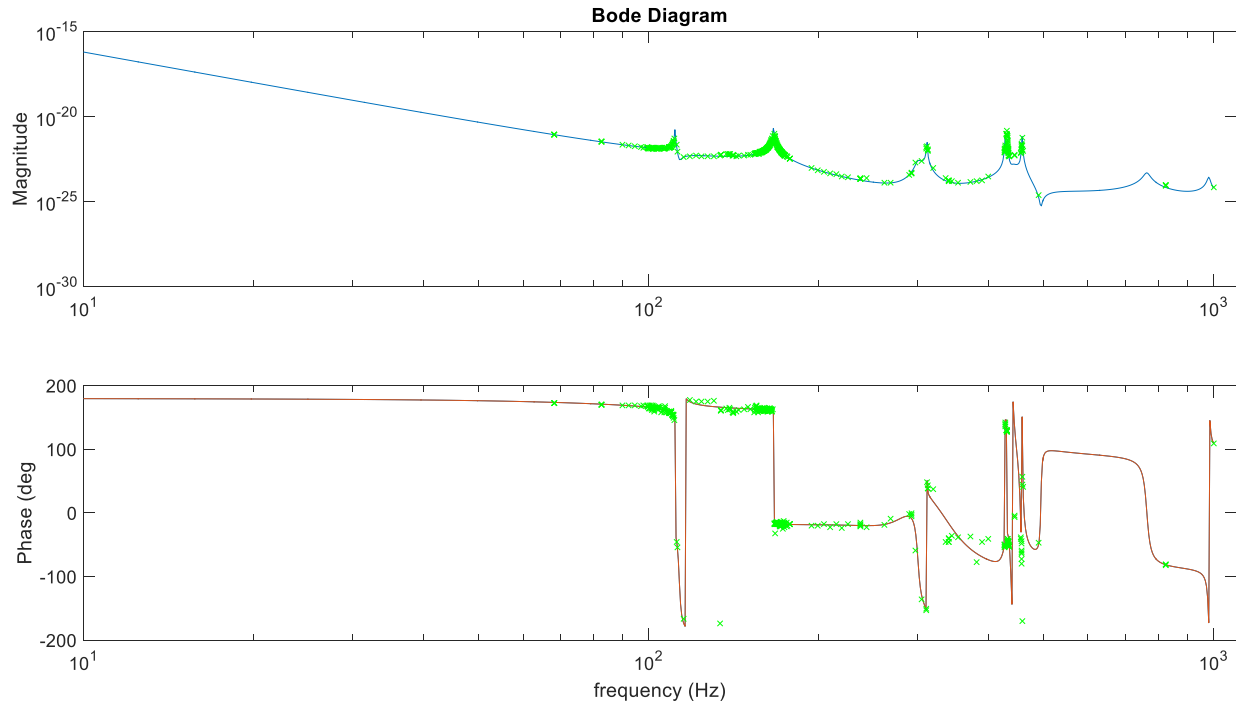
>> load m2430fid

>> load m2430ss

N.B.: Phases seem incorrect with this ss conversion.

>> load m2430zpk

N.B.: Best choice for LTI model.



Matlab script

The matlab m-file is given below. Note that the FDIDENT graphical user interface (GUI) was actually used to explore the fits.

```
function fdfit
% use of the FDIDENT toolbox to fit to high frequency suspension dynamics

% *****
% read and plot the transfer function
% *****

foo = load('2016-11-17_H1SUSEMY_L1_Actuation_HighFreqChar_asciidump.txt');
a = foo(:,1);
b = foo(:,2);
c = foo(:,3);
A = [a,b,c];
% a(1:8), 8 pts, 68.1292 to 1000.6 Hz
% a(9:17), 9 pts, 68.1292 to 824.2650 Hz
% a(18:26), 9 pts, 68.1292 to 824.2650 Hz
% a(27:35), 9 pts, 68.1292 to 824.2650 Hz
% a(36:297), 262 pts, 98 to 460 Hz
% a(298:348), 51 pts, 90 to 400 Hz
B = sortrows(A);
f = B(:,1);
C = B(:,2) + B(:,3) * i;
figure; subplot(2,1,1); loglog(f,abs(C));
title('Bode Diagram'); ylabel('Magnitude'); xlim([60 1000]); axis 'auto y';
subplot(2,1,2); semilogx(f,angle(C)*180/pi);
```

```

xlabel('frequency (Hz)'); ylabel('Phase (deg)'); xlim([60 1000]); axis 'auto
y';

% *****
% prepare the data
% *****

% frd can't handle duplicate frequencies
% for the moment drop duplicates
[uf,ia,iu]=unique(f);
ur=B(ia,2);
ui=B(ia,3);
% the last two frequency pts are distant from all others; drop
%uf=uf(1:end-2);
%ur=ur(1:end-2);
%ui=ui(1:end-2);

% increase the amplitude temporarily because the power spectrum magnitude
% is very small in this example
uC=10^20*(ur+ui*i);
um=abs(uC);
%up=angle(uC)*180/pi;

% define variances
sys=frd(uC,uf);
%sys2=idfrd(uC,uf);
sys.FrequencyUnit='Hz';
sys.Name='quadL1';
fdat=fiddata(sys);
fdat.InputVariance=0;
OutputVariance=0.1*um.*um;
% define high variance for the narrow peaks with apparent noise so that
% multiple high-Q peaks aren't attempted in the fit
OutputVariance(end-52:end-21)=2*um(end-52:end-21).*um(end-52:end-21);
fdat.OutputVariance=OutputVariance;

% bode plot system
opts = bodeoptions('cstprefs');
opts.FreqUnits = 'Hz';
opts.PhaseWrapping = 'on';
% phase wrapping control options don't seem to work
%opts.PhaseWrappingBranch = -180;
%opts.PhaseMatching = 'on';
%opts.PhaseMatchingFreq = 250;
%opts.PhaseMatchingValue = 0;
opts.MagUnits = 'abs';
opts.MagScale = 'log';
opts.XLim = {[60 1000]};
%opts.YLabel.String(1) = 'Magnitude (m/ct)';
figure; h = bodeplot(sys,opts);

figure; subplot(2,1,1); loglog(uf,abs(uC));
title('Bode Diagram'); ylabel('Magnitude'); xlim([60 1000]); axis 'auto y';
subplot(2,1,2); semilogx(uf,angle(uC)*180/pi);

```

```

xlabel('frequency (Hz)'); ylabel('Phase (deg)'); xlim([60 1000]); axis 'auto
y';

% *****
%fdtool - launches GUI for fdident toolbox
% *****
% set user level interactive
% open "read frequency domain data"
% import, from workspace, fdat
% close "read frequency domain data"
% open "variances and averaging"
% default is to take variances from data, simply execute & close
% open "estimate plant model" for manual fitting
% set: run modifiers, stabilization, to reflection
% set: max iterations to 2000
% set fixed poles at zero to 6
% set order of denominator to be 6 greater than numerator, e.g. 20:26
% set delay to variable with starting value 0 (zero)
% select "show ELiS", then Start
% close "variances and averaging", then open "evaluate or compare plant
% models", then save selected model, choose type fidmodel, LTI ss, or LTI zpk
% however delay is ignored for LTI models

% iterative fdtool fitting results saved
% unstable, non-minimum phase, with delay:
%     m2228fid.mat, m2228ss.mat, m2228zpk.mat
% stable, non-minimum phase, with delay:
%     m2430fid.mat, m2430ss.mat, m2430zpk.mat
% stable, non-minimum phase, without delay:
%     m2228fid.mat, m2228ss.mat, m2228zpk.mat
% N.B.: phases don't seem right with ss model - use zpk

load m2228fid;
delay = m2228fid.delay;
load m2228zpk;
fr = logspace(1,3,1000);
H = freqresp(m2228zpk/10^20,fr,'Hz');
Hm=squeeze(abs(H));
bias=0;
Hp=(squeeze(angle(H))+delay*fr)*180/pi+bias;
figure; subplot(2,1,1); loglog(fr,Hm,f,abs(C),'gx'); set(gca,'fontsize',18);
title('Bode Diagram'); ylabel('Magnitude'); xlim([10 1100]); axis 'auto y';
subplot(2,1,2);
semilogx(fr,Hp,fr,squeeze(angle(H))*180/pi,f,angle(C)*180/pi,'gx');
set(gca,'fontsize',18);
xlabel('frequency (Hz)'); ylabel('Phase (deg)'); xlim([10 1100]); axis 'auto
y';

% as an alternative to using the interactive tool, launch elis directly
% however I've not figured out how to impose N poles at zero frequency
% here's my attempt to impose 6 poles at zero frequency in a 22:28 model:
%fixobj = fidmodel('s',NaN*ones(1,22),[0 0 0 0 0 0 NaN NaN NaN NaN ...
%     NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN]);

```

```

%pdat=elis(fdat,'s',22,28,struct('delaytreat','v','stabilization','r', ...
%   'forceminimumphase','r','fixedpars',fixobj,'itmax',500));
%pdat=elis(fdat,'s',22,28,struct('delaytreat','v','stabilization','r', ...
%   'fixedpars',fixobj,'itmax',500));
%pdat=elis(fdat,'s',22,28,struct('delaytreat','v', ...
%   'fixedpars',fixobj,'itmax',500));
% figure; bode(pdat);

% *****
% try fitting the transfer function in two parts
% *****
% part A
sysA=frd(uC(1:243),uf(1:243));
ufA=uf(1:243);
umA=abs(uC(1:243));
upA=angle(uC(1:243))*180/pi;
%sys2=idfrd(uC,uf);
sysA.FrequencyUnit='rad/s';
sysA.Name='quadL1';
fdatA=fiddata(sysA);
fdatA.InputVariance=0;
OutputVariance=0.1*umA.*umA;
%OutputVariance(end-52:end-21)=2*um(end-52:end-21).*um(end-52:end-21);
%OutputVariance(end-20:end)=0.1*um(end-20:end).*um(end-20:end);
fdatA.OutputVariance=OutputVariance;
% best fit, stabilization = reflection, variable delay
% 6 poles at zero, 11 num, 15 denom
% NEED +2 in denom of part B
% saved as elisA (fidmodel)
% saved as elisAss (LTI ss)
% saved as elisAzpk (LTI zpk)

% part B
sysB=frd(uC(237:end),uf(237:end));
ufB=uf(237:end);
umB=abs(uC(237:end));
upB=angle(uC(237:end))*180/pi;
%sys2=idfrd(uC,uf);
sysB.FrequencyUnit='rad/s';
sysB.Name='quadL1';
fdatB=fiddata(sysB);
fdatB.InputVariance=0;
OutputVariance=0.1*umB.*umB;
%OutputVariance(end-52:end-21)=.5*um(end-52:end-21).*um(end-52:end-21);
%OutputVariance(end-22:end)=0.05*um(end-22:end).*um(end-22:end);
OutputVariance(end)=umB(end).*umB(end);
%OutputVariance(end-52:end-21)=2*um(end-52:end-21).*um(end-52:end-21);
%OutputVariance(end-20:end)=0.1*um(end-20:end).*um(end-20:end);
fdatB.OutputVariance=OutputVariance;
% best fit, stabilization = reflection, variable delay
% 0 poles at zero, 11 num, 13 denom
% saved as elisB (fidmodel)
% saved as elisBss (LTI ss)
% saved as elisBzpk (LTI zpk)

```



```
% match at overlapping frequencies?
%g = WHAT?
%elisC = g * elisA * elisB;

%fr = logspace(1,3,100);
%elisCss = ss(elisC);
%opts = bodeoptions('cstprefs');
%opts.FreqUnits = 'Hz';
%opts.PhaseWrapping = 'on';
%opts.MagUnits = 'abs';
%opts.MagScale = 'log';
%opts.XLim = {[50 1000]};
%figure; h = bodeplot(elisCss,sys,opts);

end
```