Mechanical Dynamics, Inc.

Frequency Dependent Elements in ADAMS

by
Jesper Slättengren
Mgr. Consulting
Mechanical Dynamics Sweden AB

Title: Frequency Dependent Bushing/Engine Mount/Tire Pre-processor

Purpose: To demonstrate and provide methodologies to represent the frequency dependent properties of isolation components.

Background: Many automobile companies are interested in performing noise, vibration, and harshness (NVH) studies of vehicles prior to the availability of detailed component system specification. During the design process, ADAMS models created for ride, handling, and durability studies are available. Although these models were not designed for the high-frequency requirements of NVH studies, they can be readily enhanced to include these capabilities. This will significantly reduce the time required to prepare NVH models and provide a common tool for vehicle dynamics/NVH trade-off studies.

In the automotive industry, NVH studies are critical to the successful design of a new platform. Design for noise reduction is essential in order to ensure customer satisfaction, but also to reduce the cost and weight associated with noise muffling and isolation devices. There are many reasons why vibration and harshness must be reduced, including passenger comfort and the reduction of component fatigue.

NVH engineers often have to wait until vehicle dynamics groups create their models before they can build models suitable for their studies. One possible solution to this problem is to use existing ride and handling models to conduct NVH evaluations. This will free NVH engineers from the burden of having to build parallel models to match the existing models from the vehicle dynamics groups. Previously, there was concern over ADAMS ability to produce good results in the required frequency range for vehicle NVH studies (0 - 200 Hz) or powertrain NVH analysis (0 - 40 Hz). Subsequently, MDI worked closely with NVH engineers to define the technical requirements, to define ADAMS capabilities in the 0 - 200 Hz frequency range, and to develop a methodology for adapting ride and handling models for NVH use.

Approach: Frequency dependent isolation components (bushings and engine mounts) are typically represented in ADAMS as linear bushings with frequency dependent force generation properties. The mean force produced by the bushing is calculated by the distance between the bushing's reference markers. The relative velocity is then used to modify the bushing material. Although the linear type bushing is generally useful for most dynamics modeling applications, NVH studies require a more realistic representation of bushing behavior. As such, a frequency-dependent bushing/engine mount capability was developed by MDI consultants. The bushing is implemented using the ADAMS Transfer Function, Single-Input, Single-Output feature (ADAMS statement TFSISO). The parameters of the TFSISO are calculated based on frequency response data from physical tests of the isolation component

The typical approach for the dynamic analysis of frequency dependent force elements is to construct an equivalent set of springs and dampers in series, which is in parallel with a second spring. The explicit formulations of the three components, spring constants, and damping coefficients are derived analytically as functions of the dimensions of the physical component of the properties being modeled. These formulations for a given system can be used efficiently for calculating the equivalent transfer function. The determination of these parameters generally requires the use of numerical procedures such as finite element analysis.

The focus of this work is to define an alternative method of using transfer functions generated directly from frequency response curves. This method relieves the burden of formulations to account for complex occurrences such as asymmetric bending and internal combined loading. With this method, an ADAMS TFSISO is created for each degree of freedom. These transfer functions are created automatically for each given frequency response curve.

The transfer functions uses a modification of a numerical algorithm for calculating the root locus of a given function in terms of the complex variable $s = x + iy$. This formulation takes the form of the following equation:

$$G(s) = \frac{F(s)}{D(s)} = K \frac{\prod_{m=1}^{M} (s - z_m)}{\prod_{j=1}^{N} (s - p_j)}$$

The amplitude of *G(s)* represents the stiffness of the bushing. Figures 1 and 2 show the stiffness vs. frequency and phase vs. frequency. For a given curve, the user needs to choose the proper order of the polynomial for the numerator and denominator and then calculate the coefficients. For example, Figures 1 and 2 show the amplitude and phase angle for first order polynomials. In general, higher orders will provide better results.

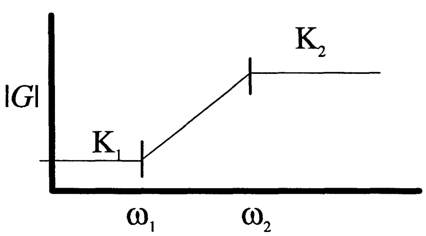


Figure 1: Stiffness vs. Frequency

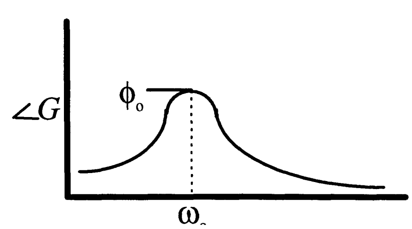


Figure 2: Phase Angle vs. Frequency

The following sections detail how to install, implement and use the macro to create these frequency dependent bushings/engine mounts.

1. INSTALLATION

To install this set of macros, unpack the tar-file named freqdep.tar, using the command *tar -xvf freqdep.tar*. Then go to the directory *freqdep/lib.* Once in this directory, execute the file *makefreq*. Next edit the file *freqdep.cmd* in the *freqdep* directory, and enter the installation directory for these files on the first executable line. You should now be ready to run an ADAMS/View session with the added macros.

2. Using the Macro

The first step that must be undertaken is to start an ADAMS/View session. Once into the session, read in the command file *freqdep/freqdep.cmd.* This can be done by selecting the FILES button, and then the COMMAND BUTTON. You may have to search for the command file in a higher directory.

To get to the panel containing the graphical design tool, the following marching menu commands should be used: MAIN MENU->PREPROCESSING->PART->CREATE (or MODIFY)->EQUATION->TRANSFER FUNCTION. This pathway is shown graphically in the figure below.

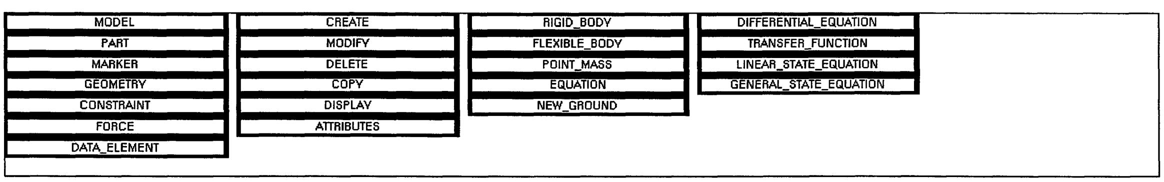


Figure 3

2.1 Transfer Function Panel

Once the PART CREATE EQUATION TRANSFER FUNCTION panel is accessed, it should look like the following:

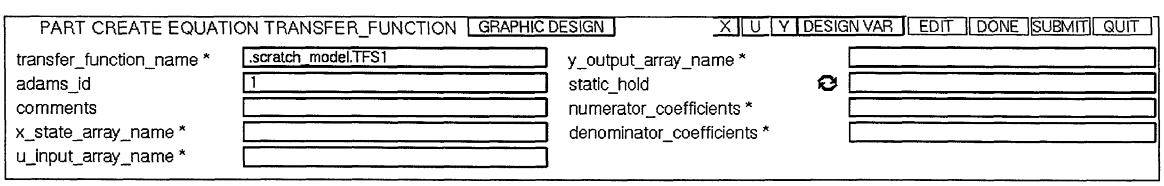


Figure 4

The button on the top of the panel, marked GRAPHIC DESIGN brings up the panel for the new macro, containing the graphic design options. This panel can be seen below.

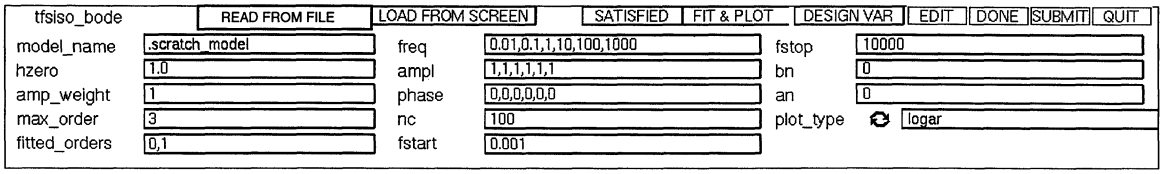


Figure 5

model_name:	name of the active model
hzero:	limiting value for the amplitude when the frequency approaches zero
phase_weight:	weighting factor for the scaled phase error relative to the scaled amplitude error
max_order:	maximum order of the inputted data curve (numerator and denominator)
fitted_order:	order of the fitted curve (numerator and denominator)
freq:	frequency values to be used in the model, in increasing order
ampl:	amplitude values corresponding to the frequency values, given as absolute values
phase:	phase values corresponding to the frequency values, given in degrees
nc:	number of points for the plot
fstart:	starting frequency, note that it must be a small number that is nonzero
fstop:	stopping frequency, note that this should correspond to the last frequency entered into the frequency field
bn:	fitted numerator coefficients, given as an output (initally zero)
an:	fitted denominator coefficients, given as an output (initally zero)

Do not have the an, bn or fitted_order fields highlighted when fitting, errors will result

plot_type:	cycle between lager and linear plot types

Once all of the proper fields have been changed to include the desired variables, click on the FIT&PLOT button. This button will bring up a message window stating that the fit mode has been activated. Once this message box appears, close it and use the SUBMIT button to create the desired plots.
The button LOAD FROM SCREEN can be used to load values that has been changed interactively. The button READ FROM FILE loads data from a file. The file should contain three columns (frequency, amplitude and phase).
When the plots have been created, it should be noted that certain fields in the graphic design panel have been changed. The fields that are of importance are the an and bn fields. Upon completion of the plots, these fields are changed to represent the denominator and numerator respectively of the transfer function.

The following section contains examples of how to create transfer functions.

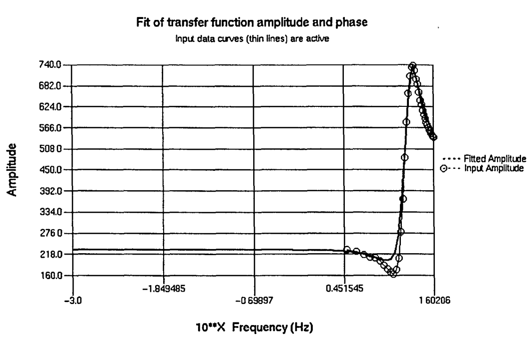
2.2 Example

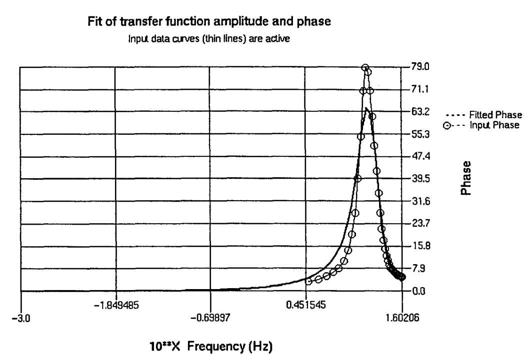
Example #1

Frequency	Amplitude	Phase
3	229.9	3.2
4	225.5	4.0
5	217.6	5.1
6	210.8	6.5
7	209.0	8.1
8	199.2	10.6
9	188.2	14.3
10	177.6	19.9
11	168.1	27.5
12	164.0	39.5
13	175.9	54.3
14	206.9	70.6
15	278.4	78.7
16	370.1	77.2
17	483.6	70.5
18	582.3	61.4
19	660.7	51.0
20	708.1	42.2
21	732.2	34.5

Frequency	Amplitude	Phase
22	738.0	27.4
23	724.4	21.8
24	700.8	17.9
25	685.9	14.7
26	664.2	12.5
27	642.0	10.7
28	626.9	9.5
29	614.1	8.3
30	602.3	7.7
31	591.0	7.2
32	581.6	6.7
33	574.7	6.3
34	567.0	6.0
35	561.2	5.7
36	556.6	5.5
37	550.8	5.3
38	544.7	5.2
39	541.9	5.0
40	539.7	4.0

The following fields were changed from their defaults to create the best fit through the data: hzero = 230, fstop = 40 and a phase weight of 0.5.





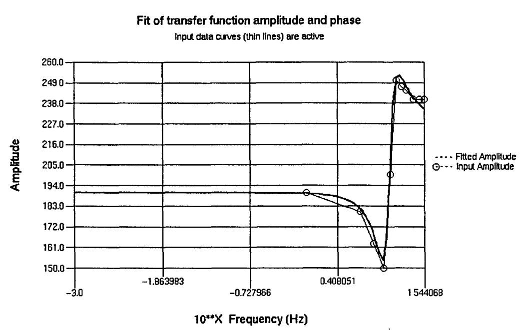
The fitted curves for both the amplitude and phase closely follow those for the inputted curves. Only two fields (other than the frequency, amplitude, and phase fields) had to be changed, so this was a relatively simple data set to run. The resulting transfer function coefficients are shown below.

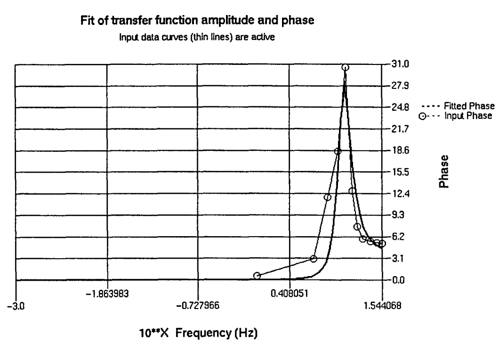
bn	=	218.875, 1.79586, 2.9308, 0.18421
an	=	0.951629, -0.70652, 0.0270157, 0.000253563

Example #2

Frequency	Amplitude	Phase
1	190	0.54
5	180	3.0
7.5	163	11.9
10	150	18.5
12.5	200	30.5
15	250	12.75
17. 5	247	7.6
20	245	5.86
25	240	5.49
30	240	5.38
35	240	5.23

The following fields were changed to create the best fit of the given data: hzero = 190, phase_weight = 0.6, fstop = 35





The parameters given in the problem statement were enough to get good fitted curves of the inputted values. The transfer function coeffiecients are shown below.

bn	=	-15.0108, -6.06806e-2, -2.90482e-3, 5.83115e-6
an	=	-7.90043e-2, -3.15806e-4, -1.21277e-5, 2.75394e-8

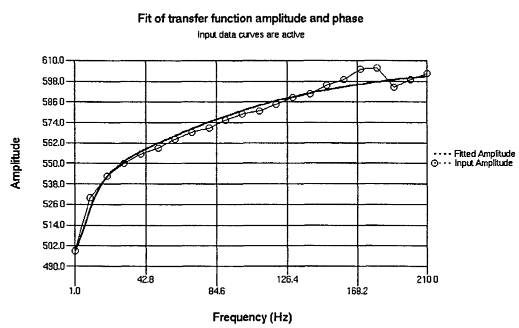
Example #3

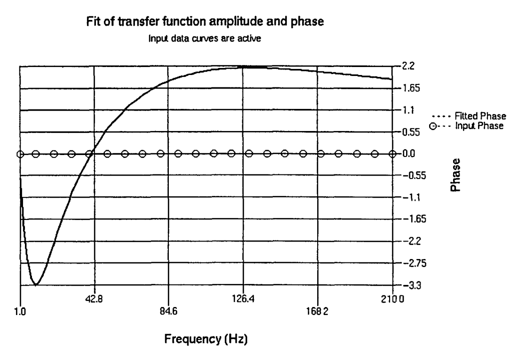
No phase included

Frequency	Amplitude
1	499.0287
10	529.9090
20	542.5764
30	550.1158
40	555.4789
50	559.1442
60	564.2199
70	568.4167
80	570.8100
90	575.3719
100	578.9592

Frequency	Amplitude
110	580.5960
120	584.6580
130	588.4928
140	590.8719
150	595.5749
160	598.8777
170	604.8204
180	605.7448
190	594.5515
200	599.0689
210	602.3665

Using the following variables, hzero = 499.0287, phase_weight = 0.2, max_order = 4, fstop = 210, and fitted_orders = 3,3 the following curves can be created.





As you can see, the fitted amplitude curve follows the inputted amplitude curve fairly well. The fitted phase curve does not follow the inputted phase curve because no phase was inputted. Since no phase was inputted, the phase_weight field was made vary low, so that the amplitude curve would carry the most weight. The transfer function coefficients are found below.

bn = -465.097, 3.04873e-2, 9.39883e-2, 1.50799e-4

an = -0.933787, -1.57021e-3, 1.67073e-4, 2.40621e-7

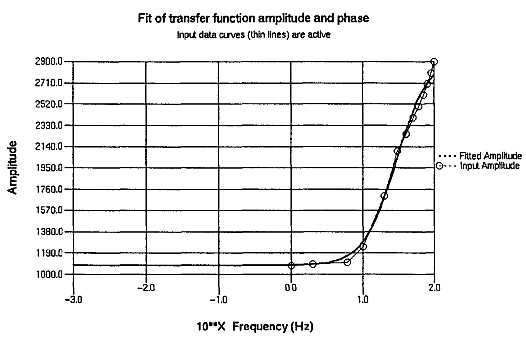
Example #4

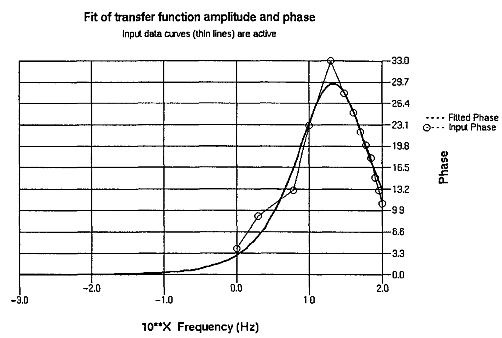
Data read off of a graph

Frequency	Amplitude	Phase
0	1080	4
2	1090	9
6	1110	13
10	1250	23
20	1700	33
30	2100	28
40	2250	25

Frequency	Amplitude	Phase
50	2400	22
60	2500	20
70	2600	18
80	2700	15
90	2800	13
100	2900	11

By changing the following fields the plots shown below can be created: hzero = 1080, fstop = 100.





In this case, the fitted curves can be made to follow the inputted curves with only a slight adjustment of the variables. Below is the transfer function coefficients corresponding to theses curves.

bn = -156.51, 3.45306e-2, 2.38943e-2
an = -0.144917, 1.24918e-3, 8.12535e-6