Instructions for the use of CalcHornAcousticImpedance.m

Below are a few steps to guide the user during the initial runs with the program.

1. First make a backup copy of the file and place it somewhere other than the directory from which you plan to run the examples and your other horn problems.

2. After opening Matlab make sure you see the file listed in the 'Current Folder' section of Matlab (typically far left panel).

3. Double click on the file to bring the file up in the editor. This is not necessary to run the file, but will allow you to read more of the comments and to see the location of the items discussed in this part of the Instructions.

4. The Matlab function CalcHornAcousticImpedance.m consists of 294 lines which you should see in the editor numbered in the editor.

5. You will note that line 1 is the function definition line that reads

   [realZa1, imagZa1]=CalcHornAcousticImpedance(fLow,fHigh,delf,ang)

   and is preceeded by the Matlab keyword 'function'. This alerts Matlab to the fact that this file is a Matlab m-file function. What is important to the inexperienced user is that the name of the file and the actual function name should be the same. So if you choose to modify the function call name, say to MyName(fLow,fHigh,delf,ang), you must also save this file under the name MyName.m.

6. The bracketed portion [realZa1,imagZa1] represents the 'return' variables that will be returned to the Matlab workspace area, and will be available for the user at the command line. So it is important to put the entire command starting from the left bracket until the right parenthesis followed by the semicolon at the command prompt and hit return to properly run the function (after first making sure you enter the correct values for fLow, fHigh, etc).

7. The variables realZa1 and imagZa1 are n x 2 matrices, where the first column contains the frequencies used in the calculation, and the second contain the real or imag part of the input acoustic impedance of the horn

model. They can be copied into a program like Excel and compared to other calculations, such as those produced by Hornresp (which allows export of csv files of its data). This is in fact how I produced the plot comparisons of the Matlab program and Hornresp.

8. When you run CalcHornAcousticImpedance it produces a plot of the real and imaginary portions of the input horn impedance over the frequency range you specified. This is essentially a plot of realZa1 and imagZa1 over the frequency range specified.

9. The core example included starts on line 39 and continues through line 99. The basic principle behind the program is to produce a separate 2 x 2 transfer matrix for each of the separate horn segments that make up the horn assembly. These matrices are called out explicitly in the example on lines 75 (T1), 79 (T2) and line 83 (T3). Each of these lines makes use of a function call to a specific horn type, viz., cylindrical = FiniteCylHorn(S1,L)), conical = FiniteConHorn(S1,S2,L), or exponential = (FiniteExpHorn(S1,S2,L). The actual functions are located at the bottom of the program, lines 195-273.

10. In the included example there are three segments, T1, T2 and T3. These are the calls that must be changed to run the 'trial example', or any further examples you may choose to run. I suggest (though it is not required) that you use the same idea as the example where T1 is the segment closest to the horn 'throat', T2 is the next after that, etc.

11. The program proceeds to calculate the final total product matrix, viz., see line 87: Ttot = @(k) T1(k)*T2(k)*T3(k) that is ultimately used to calculate the total horn impedance. This line must be changed correspondingly when additional matrices are added or subtracted depending on the horn example. For example to include a fourth segment we would have a line T4= , with a corresponding change in Ttot= @(k) T1(k)*T2(k)*T3(k)*T4(k), etc.

12. Finally, one other area should be noted that requires changing when a different example is run. These are lines 95 and 96. The initial horn 'throat' area S1 must be entered to agree with the input to T1, and the final 'mouth' area S99 must be entered to agree with the final horn segment output area.

13. As mentioned earlier the program uses a ‘piston in infinite baffle’ acoustic impedance that is assumed at the output of the last segment. Line 108 calculates the normalized acoustic impedance Zpn making use of a Struve H1 function approximation function courtesy of Robert McGough of Michgan State University (see lines 275-291). This was necessary as Matlab in the most current version that I have (R2012b) does not include Struve functions in the functions it supplies with the code. The final output impedance Zp is calculated on line 139. The solid angle (in units of pi steradians) subtended by this impedance is specified by the ang variable in the initial call statement to the function; choices include 0.5, 1.0, 2.0, 4.0, with any other choice defaulting to the choice 2.0. This section is covered by lines 100-140 in the program. None of this section needs to be changed to run different horn examples.

14. Lines 141-154 perform the necessary steps to get the horn input impedance Za1 into the correct format.

15. Lines 156-193 perform the task of plotting the data and preparing the return matrices, realZa1 and imagZa1. Note: Lines 187-192 can be modified to change the output text and graphing limits. The program defaults to semilog plots, but could be changed to regular plots by changing semilogx to plot on lines 178 and 181.

Running the Included Example

First, CalcHornAcousticImpedance.m is a Matlab function file that is invoked

by typing:

[realZa1,imagZa1]=CalcHornAcousticImpedance(fLow,fHigh,delf,ang);

where the user is expected to replace fLow, fHigh, delf, and ang with the

appropriate numbers: e.g, 10,20000,1,0.5, which translates into setting

fLow=10, fHigh=20000, delf=1 and ang=0.5. N.B., The semicolon at the end of the statement above, while not necessary, is highly recommended as without it Matlab will print out the real and imaginary parts to Za1 to the command line area.

This could result in close to 20000 lines of output for example above. Matlab uses the semicolon to suppress output to the command line work area.

Specifically fLow represents the low frequency range of the acoustic impedance calculation, fHigh represents the high frequency range of the acoustic impedance calculation, delf represents the increment between calculations, and ang represents the solid angle of interest in the output impedance. ang is similar to the variable used in David McBean's Hornresp program, and is the output solid angle in units of pi steradians.

Note, as illustrated above you do not type 0.5 pi for ang= 0.5 pi (sr) solid angle, you only type 0.5, the program knows the unit pi (sr) is understood.

As is, the function will run an 'example' horn file consisting of three segments:

(in order from 'throat' to 'mouth')

Segment 1 (matrix T1) is a cylindrical segment of area 5.07 cm2, and of length 7.62 cm.

Segment 2 (matrix T2) is a conical segment of input area 5.07 cm2, output area 45.6 cm2,

and of length 25 cm.

Segment 3 (matrix T3) is an exponential segment of input area 45.2 cm2, output area 1140.25 cm2, and of length, 25 cm.

The mouth segment is terminated in a 'piston in infinite baffle'. The program assumes S1 represents the 'throat' area, and S99 represents the 'mouth' area of the full horn assembly.

The example doesn't represent any known horn, it is just to illustrate the use of three different segment types.

The basic assumption is that the user will modify the function file supplied by changing the segments that are given in the example by adding, deleting and changing the type of segments to model.

While there is no limit to the number of segments set by the program, the practical side is that the calculation will take longer. I have run cases of seven segments without any problem with the program, how long it takes to complete a run depends not only on the number of segments, but on the frequency range and frequency increment chosen. A suggestion would be to use a frequency increment larger than you might want for a final run, say, 2 to 5 Hz, and then change it to match your needs.

I have included a plot of the output for the example given, and one for an additional 'trial example' for the user to try in order for the user to verify that he/she has worked the additional trial example correctly.

I've also included comparison plot overlays with this Matlab function and David McBean's Hornresp cases for both the included example and the trial example.

As the overlay plots show there is excellent agreement with Hornresp. I have checked numerous cases with Hornresp for a number of multi-segment horns and for different solid angles (ang values), and in all cases get essentially exact agreement with Hornresp. The exception is for a single segment horn where the Cir value (output circumference to cut off wavelength) exceeds 1. In these cases

Hornresp invokes an isophase solution that I have not been able to reproduce, although I am still working on this issue.

For an inexperienced Matlab user, the file, though heavily commented, might still be confusing. Matlab is very forgiving, and pretty good at pointing out errors and suggesting fixes (probably a good reason why it is one of the most used CAS at universities). Nevertheless, prudence suggests that the user make a copy of the original file, and to modify a separate one to run various cases.

That pretty much covers the gist of the program/function. The next part deals with the 'trial example' I suggest a user try to gain familiarity with making changes in the function.

Trial Example for User to try using CalcHornAcousticImpedance.m

The 'trial example' is a four conic segment model of the Klipsch-Jubilee horn design based on parameters as given in the article by Delgado and Klipsch (JAES,**48**,922(2000). All segments are taken as conics of form (S1,S2, L) with units given in cgs and with an ang value of 0.5 (i.e., corner placement)

Segment 1: 290, 610, 20.9

Segment 2: 610, 750, 53.2

Segment 3: 750, 2320, 66.1

N.B., These parameters are for a single driver, viz, ½ the actual total Klipsch-Jubilee which uses two drivers.

The calculated results appear in the plots included in the zip file, as well as a comparison with the results obtained Hornresp.

Written by Joe Ladish

February 12, 2013

joeladis@unm.edu

P.S. For those new to Matlab, a useful introduction text is the one by Amos Gilat, 'MATLAB: An Introduction with Applications',4th Ed., Wiley Publications, 2011, ISBN-13: 978-0-470-76785-6

This was the text used for an introduction to Matlab at the University of New Mexico.