

AkaAbak "For the cognitively challenged"

By Don Hills [<http://www.diyaudio.com/forums/showthread.php?postid=1840180#post1840180>]

I have used various "Box plot" programs for several years, but eventually I began to find them too limiting. I wanted to experiment with enclosures and configurations not found in their pre-defined libraries. I tried AkAbak, but like many other people I found the user interface and manuals too daunting and non-intuitive. Finally, I allocated time and worked my way through it.

Having done so, I realised that the program isn't as hard to use as it appears. Once a couple of key principles are understood, the rest falls into place. The tutorial that follows describes the steps that I followed to begin to use AkAbak productively. I hope they will be as useful to you.

The Discouragers

There were several things about AkAbak which initially discouraged me (and many other people). Later, I realised they were paper tigers (almost literally).

"Instead of the intuitive interface of most box-plot programs, AkAbak uses an arcane text-based interface."

AkAbak uses forms-based "Wizards" for all of its common functions. For example, you can design sealed and reflex sub boxes without typing a line of script. Even when using scripts, you can use the wizards to prompt you for the correct information which you can then paste into a script.

"The user manual for AkAbak is huge and I fell asleep after the first ten pages."

The manual should be used as a reference to be dipped into as needed, not read from cover to cover. There are about ten pages which are useful for first-time users. After that, whenever you come to something you don't understand, or you want to know how to do something, look it up in the contents and just read that section.

"The Examples manual assumes you already know how to use AkAbak and offers little help to the first-time user."

By the time you have finished this tutorial, you should have gained sufficient familiarity with AkAbak fundamentals to use the examples to aid in understanding.

No scripting required

Function for function, AkAbak is no harder to use than any other "box plot" program. It only appears difficult because of the sheer number of functions, with no "Start Here" button.

You do need to have used other "box plot" programs in order to have become familiar with the general speaker design process, and you need to know how to use Windows Cut and Paste functions.

You will be shown how to use AkAbak as a simple "box calculator", then how to use the results to move on to full AkAbak scripting.

To do this, we will use a common example: a simple subwoofer enclosure.

"Here's one I made earlier."

Using a "box calculator" program (Unibox, LspCAD, DosBox etc) that you are familiar with, design two subwoofer systems: a sealed enclosure and a vented enclosure, preferably using the same driver for each. Keep the results handy, you will need to refer to them later.

Sealed Sub - All Wizard, no scripting

We will now re-create the sealed box sub in AkAbak. Because this is a common item, AkAbak has a wizard-based way of creating one.

Start AkAbak. (AkAbak.exe)

From the top menu select "Def -> Def_BassUnit...".

A data entry window will open, called "Definition Def_BassUnit / Calculator".

In the top row of boxes, enter the parameters for the driver you are using. You should have everything except maybe the voice coil inductance.

Note the units choices for some of the boxes.

Akabak allows you to specify your driver in physical values (Mms, Cms, Bl etc) or Thiele-Small values (Qms, Qes, Vas etc), or a mixture of both. Right-button clicking on a box will pop up a list of parameters that Akabak will accept for that field.

For example, the second box from the right may be labelled either "Mechanical Mass (Mms)...", "Equ. vol. to compliance (Vas)...", or "Mechanical Compliance (Cms)...". You can either change it to the value you have by selecting from the right-click menu, or just enter the value you have with the correct unit suffix. To enter Vas, specify volume units, for example in cubic feet (2.4ft³) or litres (49L). Mms, Vas and Cms have different units of measurement so Akabak will know which value you have entered.

Also, the driver "Diaphragm diameter" (dD) is more commonly specified as "Diaphragm Area" (Sd). You can enter the value as a diameter (for example, 16.5cm) or area (for example, 200cm²).

In the second row of boxes you can enter the Xmax parameter if you have it, and the volume of the enclosure. The rest of the fields can be left as they are.

Check for Typos

Press the "Evaluate" button at the bottom of the window. If all of the required values have been entered, AkAbak will populate the two rows of values at the bottom left of the window. Values such as "fc", "Qtc", and "f3" should be close to those calculated by your other program.

If the fields do not populate, look to see if the cursor has moved to one of the input fields at the top of the window. The value may be missing (and required), or AkAbak can't understand the entered value.

If the evaluated values are very different than your pre-calculated box, then Akabak has likely misinterpreted one of the input values. For example, AkAbak defaults to "cubic metres" in the "Vb" field. Entering "165" will be interpreted as 165 cubic metres. Specify the unit, for example litres (165L) or cubic feet (6.2ft³).

Display the results

Press the "Diagram..." button to generate a graph. The frequency response should be similar to your pre-calculated box.

Press the "Alignments..." button to generate a list of possible boxes that will work with your driver.

Reflex box - still point and click

Next, we will generate a vented (reflex) enclosure:

Change the "Vb" to that suggested by your pre-calculated vented enclosure.

Tick the "Vented" box.

Enter the "Helmholtz Frequency (fb)". This is the box resonant frequency. If you have the port dimensions instead, right-click on the "fb" field. This will open a window where you can cross-calculate fb, box volume and port dimensions - enter what you have, and AkAbak will calculate the missing value.

Press "Evaluate". The results should be similar to those of your pre-calculated vented box.

Scripting the box

So far, AkAbak works like any other box calculator. You enter the numbers and get the results.

Now you can generate an AkAbak script from the data you have entered.

It's only Cut'n'Paste (but I like it)

Press the "Copy to clipboard and close" button.

From AkAbak's main menu, select "File->New Script".

A window will open containing a single line:

```
-----  
System 'S1'
```

Put the cursor at the beginning of the line and press Enter a couple of times to put some blank lines above the "System" line. Move the cursor to the top blank line and paste in the saved data ("Edit->Paste", or Ctrl-V). It should now look something like this: (Example from AkAbak's example script EX_11)

```
-----  
Def_BassUnit 'B1'  
dD=12.75cm |Piston  
fs=65Hz Vas=12.5L Qms=1.25  
Qes=0.6 Re=5.5ohm Le=2.35mH ExpoLe=0.618  
mb=0.95 Xms=4mm  
Vb=8L Qb/fo=0.1  
|Performance in sealed enclosure:  
| fc Qtc fD f3  
| 98.8Hz 0.639 857.1Hz 110.5Hz  
| SPLmax Pelmax Uomax t60 Ripple  
| 102.4dB 23.5W 16.07V 13.45ms 0
```

```
System 'S1'
```

Note that all the data you entered in the calculator is there, plus the evaluated results. The "vertical bar" symbol (|) indicates a comment, and will be ignored by AkAbak when it calculates. You can re-open the calculator with all this data in it by putting the cursor anywhere in the middle of the data, and selecting "Def->Def_BassUnit..." again. AkAbak will read the data out of the script and put it in the calculator.

AkAbak is flexible regarding the data layout. You can arrange it to make it easier for you to read. For example, the above data can be re-arranged like this:

```
Def_BassUnit 'B1'  
dD=12.75cm  
fs=65Hz  
Vas=12.5L  
Qms=1.25  
Qes=0.6  
Re=5.5ohm  
Le=2.35mH  
ExpoLe=0.618  
mb=0.95  
Xms=4mm  
Vb=8L  
Qb/fo=0.1
```

```
System 'S1'
```

The System

In AkAbak, all of the related components of a speaker are grouped as a "System". There can be more than one System in a speaker - for example, the woofer, its enclosure and crossover components can be one system, and the tweeter and its crossover components can be another system. Altering the components in one system will not affect the performance of the other.

Also, components that are commonly used can be defined outside of the system so that they are available to all systems. For example, if you are building a sub containing four identical drivers, you could paste the driver parameters into the "System" definition four times. Or you can do as per the example above, where you define the driver once (with a "Def_..." statement) and can refer to it many times within the system by name ('B1').

More GUI Wizardry

Next, you need to add the defined sub to your system.

Select "Net->Transducer->BassUnit..."

Leave the Node parameters as they are.

In the (Def=) field, select the unit you defined earlier (in def_BassUnit).

Click "Copy and close".

Put the cursor on a new line under the "System" line and paste ("Edit->Paste" or Ctrl-V).

It should now look something like this:

```

-----
Def_BassUnit 'B1'
dD=12.75cm |Piston
fs=65Hz Vas=12.5L Qms=1.25
Qes=0.6 Re=5.5ohm Le=2.35mH ExpoLe=0.618
mb=0.95 Xms=4mm
Vb=8L Qb/fo=0.1
|Performance in sealed enclosure:
| fc Qtc fD f3
| 98.8Hz 0.639 857.1Hz 110.5Hz
| SPLmax Pelmax Uomax t60 Ripple
| 102.4dB 23.5W 16.07V 13.45ms 0

```

```

System 'Bass'
BassUnit Def='B1' Node=1=0
x=0 y=0cm z=0 HAngle=0 VAngle=0

```

Your system now has a bass unit, using a driver/enclosure defined earlier called 'B1'. If you hadn't defined the driver/enclosure earlier, all of the values from "Def_BassUnit" would have to be included here.

"Node" will be explained later. The x, y, z and angle values refer to the position of the driver on the baffle. They can be ignored for now. They become significant when you design full-range systems with mid-bass and tweeters.

Still Point'n'Click

Now you can evaluate the speaker. Select "Sum->Acoustic Pressure..." from the menu. A dialog window will open that allows you to specify various parameters, such as the graph scales and the input power to the speaker. Just leave them at their defaults for now and press "OK". The resulting graph should be similar to the response graph from your pre-calculated speaker, allowing for scale differences.

If you want to make the graph look similar to that on your other box program, you can change the settings on the dialog window to the same scale values as used by the other box calculator program.

Other parameters can be displayed, either from options under the "Sum" menu or options under the "Inspect" menu. The main difference between "Sum" and "Inspect" is that "Sum" displays the behaviour of all of the "Systems" together, whereas under "Inspect" you can display the behaviour of individual systems or individual components of each system.

Almost there

There are two final steps to AkAbak enlightenment.

The first is to understand the concept of "Nodes".

Every component has one or more nodes or "connection points", rather like the connectors on Lego blocks.

Systems are built by defining components and connecting their nodes together, like clipping Lego blocks together.

For example, a resistor has two nodes (terminals).

An enclosure has one node.

A port has two nodes, one at each end.

A driver has at least 4 nodes: each voice coil has two nodes, the front of the diaphragm has a node, and the back of the diaphragm has a node.

All nodes are given numbers. All nodes with identical numbers are connected together.

There are two "special" nodes:

Node 1 is always the input node to the system (from the amplifier.)

Node 0 is always ground (earth).

The BassUnit we have defined above is a complete subwoofer with just a pair of terminals on the back to connect to the amplifier.

These are represented by the "Node=1=0" statement.

If you wished to connect two of these units in series across the amplifier, you would specify them like this:

System 'Bass'

BassUnit 'My_Bass1' Def='B1' Node=1=2

BassUnit 'My_Bass2' Def='B1' Node=2=0

The signal from the amplifier comes in at node 1, passes through the first unit and exits at node 2. Because all equal number nodes are connected together, it enters node 2 of the second unit and exits to ground at node 0.

The Final Concept

The last step is to understand the concept of "components".

So far, we have worked with a composite object, a complete speaker called a BassUnit.

This is built from a driver, an enclosure, a port and a couple of radiators. Each of these can be individually specified.

As described in the "Node" section above, a driver has a node for the front of the cone and a node for the back of the cone. To make sound waves, the driver has to have a radiator attached (by matching node numbers) to one side of the cone. The enclosure is attached to the other side of the cone.

If the enclosure is ported, a port (AkAbak calls it a "Duct") is attached to the enclosure at one end and has a radiator attached to the other end.

A vented box subwoofer defined from components would look something like this:

Def_Driver

| Specs of bare driver (no enclosure)

System

Driver Node=1=0=2=3

| Voice coil = nodes 1 and 0, front of cone = 2, rear of cone = 3

Radiator Node=2

| Attached to the front of the cone to generate sound waves.

Enclosure Node=3

| Attached to the back of the cone.

Duct Node=3=4

| One end attached to the enclosure, the other to a radiator.

Radiator Node=4

Attached to the end of the port to generate sound waves.

In the "Inspect" menu options, you can generate graphs for any defined node. For example, you can measure the voice coil impedance at node 1, pressure variations in the enclosure at node 3, the velocity of air in the port, or the individual SPL contributions of the driver radiator (node 2) or the port radiator (node 4).

The component concept can be taken even further. For example, a driver is a composite object - a magnet, a voice coil, a cone, a suspension etc. Each component can be individually specified. AkAbak can thus be used to design a driver to suit a specific enclosure, rather than the other way around.

Into Uncharted Waters

The power of AkAbak becomes apparent when you want to move on from the pre-defined enclosures available in "box calculator" programs. For example, there is a reflex enclosure variation called the "Double Chamber Reflex". A good analysis of it is at: <http://www.claudionegro.com/project...er/dcr/dcr.html>

It consists of a driver, two enclosures and three ports.

This is difficult to model in a simple "box calculator", but can be easily modeled in AkAbak. Here is an example, with the enclosures and ports named to match the illustration on the web page:

Def_Driver 'AD8066W8'

Meas_Dipole

dD=16.5cm

fs=39Hz Vas=49L Qms=4.2

Qes=0.52 Re=7.6ohm ExpoLe=0.618

System 'DCR'

| vt=77L, f2=34.04Hz, v1=51.3L, v2=25.7L, f1=30.2Hz

|-----
The speaker driver and the radiation from its diaphragm:

Driver 'Driver' Def='AD8066W8' Node=1=0=2=3

Radiator 'Rad_Driver' Def='Driver' Node=2

|-----
The first enclosure and its radiating port:

Enclosure 'V1' Node=3

Vb=51.3L Qb/fo=0.5 Lb=10cm

Duct 'P1' Node=3=4

dD=10cm Len=40.3cm QD/fo=2

Radiator 'Rad_P1' Def='P1' Node=4

|-----
The port connecting the two enclosures:

Duct 'P2' Node=3=5

dD=10cm Len=40.3cm QD/fo=2

|-----
| The second enclosure and its radiating port:

|-----
Enclosure 'V2' Node=5
Vb=25.7L Qb/fo=0.5 Lb=10cm

Duct 'P3' Node=5=6
dD=10cm Len=40.3cm QD/fo=2

Radiator 'Rad_P3' Def='P3' Node=6

Note the 'Def=...' statements. The radiator 'Rad_Driver' gets its dimensions from the 'Driver', which in turn gets its parameters from the definition 'AD8066W8'. The radiators for the ports get their dimensions from the ducts to which they are attached.

Graduation Day

That should be just about all you need to know to get started on basic design.

I suggest you now read pages 13 to 23 in the AkAbak main manual. Don't puzzle for long over sections that don't make immediate sense. Come back to them after you have used AkAbak a little more and they will fall into place. The "Examples" manual will teach you techniques when developing your own speakers. You will likely learn more by developing your own speakers using the examples as guides, rather than by working through the examples as exercises.