WinSound™

by William H. Spicher

User's Manual

printed on
May 11, 2001

THIS MANUAL AND THE ASSOCIATED PROGRAM IS AN UNPUBLISHED WORK FULLY PROTECTED BY THE UNITED STATES COPYRIGHT LAWS AND IS CONSIDERED A TRADE SECRET BELONGING

Contents

Foreward 3

Introduction 5

Tutorial 6
- Open WinSound 6
- Create speaker types. 6
- Place the speakers. 7
- Configure the Sound Man 7
- Launch the analysis. 8
- Save the setup 8
- Assess the results 8
- Fix the problem 10

Philosophy of WinSound 12

Preparing Models 15
- Creating Speaker Types 15
- Placing Speakers 20
- Creating Audience Boundaries 24
- Defining The SoundMan 27
- Model Files 28
- Launching Analyses 28

Interpreting The Results 31
- The Colorbar 31
- The Frequency Control 33
- The Sound Man 34
- The Speakers 35
- The Analyzer 36

Further Reading 37

Foreward

From 1989 to 1991 I worked in a local music store running the department of sound reinforcement. Eventually I got the opportunity to design a sound system for a small church, but I soon found that the aspect of choosing and locating speakers was far from easy. After reading a couple books on the subject, it became clear that the math was way over my head. The basic problem was that speakers possess directivity, and if a listener is at a location where the speaker is not directly aimed, the level will be reduced. But finding the amount of reduction seemed like a tortuous problem reserved for the mathematically elite. So I took night classes in Algebra and Trigonometry, and after a lot of work, I was able to hobble along somewhat with a programmable calculator.

But then I encountered another problem. The music store wanted to be reimbursed for the time I spent designing a system. At even at a modest $20 an hour (a handsome profit for the store after deducting my $6/hr), most small and medium churches were reluctant to make the investment. So my new-found skills mostly sat on the shelf, and the sound systems were built without the needed insight that I could have provided.

Yet I deeply loved what I was doing, so I attempted to pursue it. Major consulting firms told me that to get any further I would need a certain piece of paper—an engineering degree. While I hardly wanted to take a four-year detour, I knew that the education would come in handy, so I began school majoring in mechanical engineering. It was there that I discovered that I was one of those people that simply enjoys solving problems with calculators (geeks). Four years later, I had almost forgotten the world of sound that had inspired me to enter school.

When I graduated, I took a job with Caterpillar, and they placed me in a peculiarly familiar environment: Acoustics and Vibration. The world of decibels and frequency was reopened before me, and I fell in love with it all over again.

The church where I was attending and serving as a sound operator was having some complaints about the sound—it was usually excessively loud, bright, and poorly mixed. In a rare display of insanity, I thought, “I can tackle this.” So I sat down and drew pictures and wrote equations. But now I was armed with the powerful mathematics software package Matlab® (by the MathWorks, Inc.). Weeks later, after implementing the equations in code, I started to model the sound at my church. I soon discovered the basic problem: The sound operator—that is, me—couldn’t hear anything! The board had been stuck up in “the crow’s nest” where I could hardly see the speakers let alone hear them. The settings that resulted in decent sound for me were wreaking havoc on the ears of my audience.

As I started iterating through alternative designs, an enlightening thought occurred to me: I wasn’t using engineering math anymore. The process had been reduced to entering coordinates and running the analysis. I could have done this years ago at the music store without the engineering degree if the tools had been available! Then, in another rare display of insanity, I thought, “I could implement this stuff in a Windows application so that anyone can use it.”

After a bookshelf of programming manuals and two-hundred pages of C++ code, WinSound™ was born. As I showed it to everyone I could find that was crazy enough to try to use it, it became evident that few churches were reaping the benefit. And, in another rare display

of insanity, I thought, “I should publish WinSound so that every church at least has the opportunity to benefit from it”...working on it...

Introduction

The primary purpose of a church building is to provide a space where Truth can be heard and worship can be experienced. A crucial component in fulfilling that purpose is sound. The medium-sized church faces a tough problem when it comes to providing for sound. While a large church can afford to hire an acoustic consultant to work with the architect to produce a superb environment with a quality system, the very small church can rely on the unaided volume of the voice and instruments to get the job done. Somewhere in the middle is the medium church where a sound system is needed but the funds for professional consulting are difficult to produce.

The most challenging task (and the most important) is the selection and placement of speakers. This is where the professional is needed most because this task is the most time-consuming and requires the most expertise. Other aspects, such as choosing a mixing board and connecting effects to it, are easily handled with a little help from the dealer where the equipment is purchased. But few dealers have the expertise or are willing to invest the time to measure the site and properly design the speaker system.

Fortunately, the medium-sized church does have a powerful and valuable resource at its disposal—its members. With the right tools, a little guidance, and some time to contribute, a volunteer can close the gap between funds and quality. WinSound, together with this manual, is designed to provide the first two components in the equation. The basic idea is that every church has a volunteer with an interest in the subject and high-school math skills. WinSound is designed for such a person to use, and this manual is written to provide him (or her) with the guidance necessary to produce good results.

However, WinSound's benefits are not limited to churches. Anywhere sound needs to be delivered to a hundred people or more, WinSound can help. This includes school auditoriums, conference rooms, and concert events. Of special significance is the temporary event which changes with each application. In this case, WinSound can be a powerful tool in the hands of the sound vendor.

This manual is intended to be a vital component in the WinSound process. That is, WinSound is not written to be understood and used apart from a certain minimum amount of knowledge that cannot be conveyed on an as-you-go basis. Thus, it begins with a step-by-step tutorial on WinSound’s use and features followed by a more thorough discussion of the underlying philosophy alluded to above. The last two chapters describe in detail the two major activities in the WinSound process—creating WinSound models and interpreting WinSound’s results.

Tutorial

Understandably, many people would rather dive into a software package than read a manual about it first. Therefore, before we get into all the details about all the dialog boxes and so forth, let's get our feet wet by applying WinSound to a common sound reinforcement problem. However, if you are not yet versed in sound reinforcement jargon, just follow the directions even if you don't understand them. More explanation awaits you in later chapters.

1. Open WinSound

Open WinSound by double-clicking the file *winsound.exe* from the File Manager or the Windows Explorer. The working environment for WinSound is the main window. The colorful bar on the right is the scale for sound intensity. The "Mr. Bill" toward the bottom is the Sound Man[1]. The horizontal and vertical axes indicate position in feet. The scroll bar at the top is the frequency selector; it allows you to view the sound map for any of 29 third-octave frequencies.

2. Create speaker types.

The first step is to tell WinSound what *kind* of speakers you have by creating *speaker types*. Later, you will specify where these speakers are. By creating speaker types, multiple speakers can share the same type without it being created over and over. This saves you time and it saves the computer memory. Select Model | Create Speaker Types…, or press **F1**. In this dialog, you define your speaker type by specifying three things[2]: the speaker's efficiency, its frequency response, and its directivity. There are two options for specifying the directivity: dispersion angles or text files containing the polar patterns for the speakers. Check the Beam Widths radio button to use dispersion angles. Enter `120 x 75 at 2kHz` for the name of the pattern. Enter `120` for the horizontal, `75` for the vertical, and `2000` for the reference frequency. Next, leave sensitivity at 95 dB, and enter `jf100fr.txt` for the frequency response file. This specifies a file containing the frequency response for the EAW JF100. Press Create. WinSound will create a directivity pattern with a 6-dB drop at 60° off the horizontal axis (half of 120°) and 37.5° (half of 75°) off the vertical axis at 2000 Hz. At higher frequencies, the drop becomes steeper, and at 250 Hz and below the speaker is omnidirectional (no off-axis attenuation).

---

[1]The term "Sound Man" refers to the data object that performs the duties of a sound operator (that's actually its name in the source code). I don't mean to imply that a sound operator should be a man; women can do a fine job in this role and should be given the opportunity if they have the interest.

[2]For a more thorough discussion about the speaker type and its components, see the section *Creating Speaker Types* in the chapter titled, *Preparing Models*.

Create another type 60˚ x 40˚ with the name `60 x 40 at 2kHz`. Use the same sensitivity and frequency response. You can build as many speaker types as you wish, but for now press Close.

3. Place the speakers.

Select Model | Place Speakers…, or press **F2**. In this dialog you create actual speakers by specifying their position and spacial orientation. For this example you will create a single pair spaced 40 feet apart and mounted on the stage on stands. Enter `main pair` for the Name. In the Location group enter `20` for x, `35` for y, and `10` for Height. To tell WinSound how the speakers are aimed, in the Aim Point group enter `20` for x, `0` for y, and `4` for Height. Check the Twin check box, and select the speaker type named "60 x 40 at 2kHz" that you created earlier. Leave the Frequency Cutoffs and *Relative Level* at their default values[3]. Press Add. You have created the pair. They are 40 feet apart (20 feet from the centerline) and 10 feet high (4 feet for the stage and 6 feet for the stands). They aim straight outward at points in the audience 4 feet high, 20 feet from the centerline, and at $y = 0$. Thus, a person sitting at the point (20,0,4) would see the right speaker aiming straight at him. Press Close. At this point, WinSound appears as shown in Figure 1. The speakers are represented by rectangles with lines joining the corners; the aim points are shown as—what else?—bull's eyes; the speakers and aim points are joined by lines.

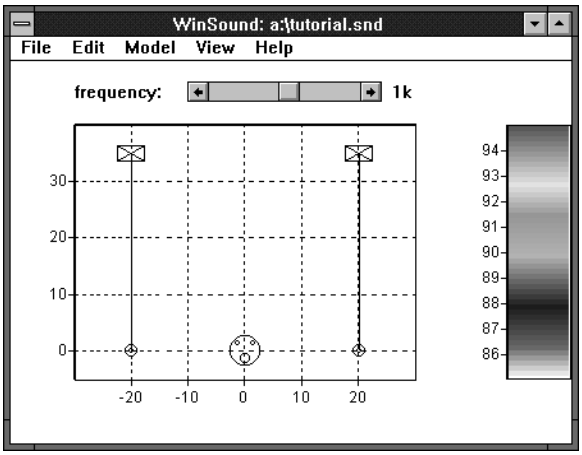


Figure 1 WinSound after placing speakers.

4. Configure the Sound Man

Select Model | SoundMan…, or press **F4**. Leave X and Y at 0, but enter `15` feet for Height to simulate a "sound booth" (also known as the "Crow's Nest") located in the central balcony. Leave Desired Operator Level at 90 dB. This means that the Sound Man will add sufficient power and equalization to receive 90 decibels at all frequencies. This is a central concept in WinSound: the sound operator will do everything he (or she) can to make it sound good to him. Therefore, his listening perspective is the reference against which all others are judged.

---

[3]These features allow you to refine the system when using multiple speakers and/or crossovers. We discuss them in more detail later in the section *Placing Speakers* in the chapter titled *Preparing Models*.

5. Launch the analysis.

Select Model | Launch Analysis…, or press **F5**. A message box appears reminding you that you have not yet told WinSound where the people are sitting, but it can generate a default "boundary" based on the axes' settings[4]. The listeners' ears are assumed to be four feet above the ground. Accept by clicking yes. The analysis dialog shows the range indicated by the horizontal and vertical axes so that you can change them here if desired. Accept these values, but enter 2 feet for the grid. This means that WinSound will calculate levels for all listening points within the range, and the points will be spaced 2 feet apart[5].

Leave Default Height at 5.5 feet. This is the height of listening locations that do not fall within the boundaries—in this example, there are none.

Enter 200 Hz for Phase Effects Below so that at low frequencies WinSound will consider phase interactions between the speakers. At higher frequencies, WinSound will simply sum the direct sound intensity from each speaker as energy. Press OK. A progress dialog will appear with the title, *WinSound is Number Crunching*. The black rectangle will be filled with red when the analysis is done. In the meantime, WinSound allows you to run other applications while it quietly crunches in the background.

6. Save the setup

Choose File | Save as… In the file box enter `tutorial.snd` and press OK. File saving accomplishes two things: It saves the work of reentering the setup data; and it allows the comparison of two or more possible setups. WinSound will warn you if you are going to overwrite an existing file. Also, you can select File | Save (or press **Ctrl-S**) to save the current file without having to reselect it in the dialog. Later, you can open the file with File | Open (or press **Ctrl-O**); if you try to open a non-WinSound file, WinSound will ignore it and will tell you that it is not a WinSound file.

7. Assess the results

If you have not moved the frequency scroll bar, it should be set at 1 kHz. The sound map is now a collection of small, colored squares where the color at a square indicates the total sound pressure level at that listening location at that frequency. The system has been adjusted and equalized so that the Sound Man receives 90 decibels at all frequencies, and the color bar has been autoscaled to show all levels at the current frequency of 1 kHz.

---

[4]WinSound offers sophisticated options with which you can tell it where people are sitting. See the section on *Creating Audience Boundaries* for more information.

[5]See the section on *Launching Analyses* more information on creating grids and other analysis options in this dialog.

Click on the thumb (little moving square) of the frequency scroll bar and drag it all the way to the left (31 Hz). Notice that the loudest points are closest to the speakers, but the quietest points lie near two curves that pass between the speakers. 31 Hz is a very low frequency—few speakers can actually produce it. At low frequencies, these speakers are "omnidirectional." In other words, the speaker radiates sound with the same intensity in all directions; thus, the level received only depends on distance to the speakers and not on where the speakers are aiming. However, certain listeners will be at locations where the wave from one speaker is directly "out of phase" with the wave from the other, and this causes the waves to cancel each other out. Also, recall that the Sound Man receives 90 dB. From the color bar you can see that listeners close to the speakers (where the color is red) are hearing 102 dB—12 more decibels than the Sound Man!

Remember that these speakers' directivity was defined using a reference frequency of 2000 Hz. Let's view the results at this frequency. Click on the thumb of the frequency scroll bar and drag it to the right until the text indicator to the right reads "2k", and let go. The sound map changes to show the results for 2 kHz. Now the positions of maximum level are directly in front of the speakers rather than around them. This is because of the directivity at 2 kHz; the speaker focuses the sound toward its front much the way a flashlight focuses light (objects in front of the flashlight's beam). Also, remember that the speakers are 10 feet above the ground, and the listeners are 4 feet, so the speakers are 6 feet above the analysis locations immediately surrounding it.

The color scale no longer reflects the actual range of levels here. Select View | Autoscale Color, or press **F6**. We can now see that at 2 kHz listeners directly in front of the speakers receive 100 dB—10 decibels more than the Sound Man. For more detail, pop up on one of the colored squares in front of one of the speakers and choose open analyzer from the popup menu. A window appears with a graph of the frequency response at that location. In other words, you can view the levels at this location for all 29 frequencies. Notice that the higher the frequency, the higher the level. Here's why: at higher frequencies, the directivity pattern is narrower (picture yourself "narrowing" the beam on an adjustable flashlight). This means that the Sound Man is receiving less level because he is significantly off the main axis of either speaker (like two flashlights not quite pointing at him). Thinking it sounds muddy, he turns up the high frequencies until it sounds right. But to the people in front of the speakers, it sounded fine *before* he turned the highs up.

Verify this by popping up on him and selecting view equalization. A frequency-response window appears showing the adjustments he made to the equalizer in order to receive a flat response. The numbers can be copied to the Clipboard by popping up in the graph area and selecting Edit | Copy; then they can be pasted into a spreadsheet for fancier graphics, printing, or more analysis. You can close the window by double-clicking the control box in its upper-left corner.

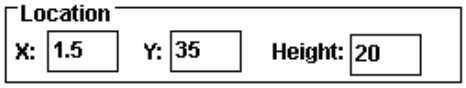
10 dB is too much of a difference. When the sound operator thinks the level is just right, it will be perceived as too loud by these listeners. Another implication of a difference this large is that the sound operator is probably listening to the reverberant field and can hardly hear the direct

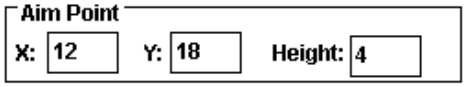
sound from the main pair.[6] The crux of the problem is that what the Sound Man hears is vastly different than what most of the audience hears. So when it sounds good to him, it probably sounds terrible for them.

8. Fix the problem

We will now reposition a few things to try to get a better sound distribution. By better we mean two things: (1) The sound will be evenly spread across the listening area, and (2) the Sound Man will hear the same levels as the audience.

Select View | Allow Movement (a checkmark will appear next to the menu) so that you can move things around with the mouse. Click on the Sound Man (Mr. Bill) and drag him forward about a third of the way toward the stage. Pop up on him and select set height. Enter 4 feet (we're putting him *in* the audience). Now click on the right speaker and drag it in toward the centerline until the pair is just next to each other. Then pop up on one of the speakers and select set height. A dialog appears allowing you to set the height. Enter 20 feet, and press OK. Physically this represents hanging the speakers from the ceiling in a "cluster". Finally, click on the right aim point (the little bull's eye) and drag it forward about half way toward the stage then drag it about half way to the left toward the centerline (about a third of the total distance from the right edge of the sound map. Select View | Allow Movement again (the checkmark will disappear) to prevent accidental movement. Now select Model | Place Speakers…, or press **F2**, and double click on main pair in the list box. The settings should read (approximately):





If they don't, you can enter these values. Select the other speaker type created earlier named 120 x 75 at 2kHz, and press Modify then Close. WinSound should now appear as shown in Figure 2.

Move the Analyzer's window and the main window so that you can see them both. Next, double-click the vertical axis in the frequency response window (or pop up in the axis and select Enter Limits), and change its range to a low of

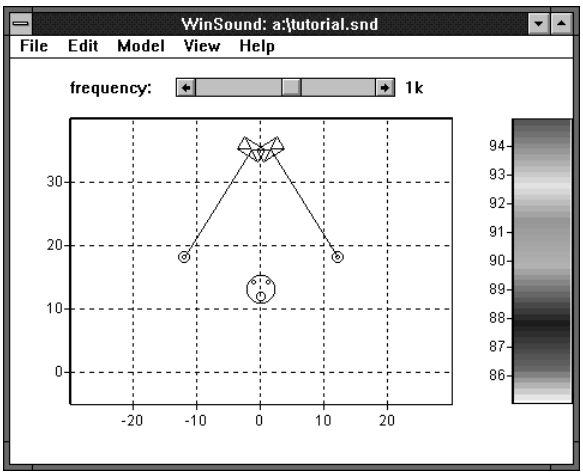


Figure 2 WinSound's appearance after modifications.

[6]WinSound does not compute reverberation, so it is up to us to be aware of its potential effects. For further reading about reverberation, see Davis and Jones 1989, 57.

75 dB and a high of 105 dB. Make sure the current frequency is still 2 kHz and launch another analysis by selecting Model | Launch Analysis…, or pressing **F5**. The original ranges and grid are displayed so press OK. Before number crunching, the Sound Man's equalization window and the Analyzer's response window are updated.

The sound map is now concentrated in the center of the room. The color scale reads from 82 to 92 dB, so the loudest points are only 2 dB higher than the Sound Man. The quiet points are at the back of the room and in the upper corners (verify by opening additional Analyzers as described above). Click *in* the frequency scroll bar just to the right of the thumb. The current frequency changes to 4 kHz--one octave higher than the last. Click again to 8 kHz and again to 16 kHz. The sound map splits into two hot spots corresponding to the two speakers' narrow beam widths at high frequencies. Click again to 20 kHz. This is the highest frequency available and can be heard by only the most sensitive ears (usually children). Even at such high frequencies the pattern resembles that for 2 kHz pretty well with no "dead spots" in the center. Drag the frequency scroll bar down to 250 Hz. At 250 Hz and below, these speakers are omnidirectional (due to the way we defined the directivity pattern). Inspect lower frequencies where phase is considered. Notice that these speakers are acting as one now because they are so close together. Phase interference has not been eliminated, but it will be much less of a problem now.

Now click and drag the Analyzer icon within the sound map (the circle with an X in it). The frequency response window dynamically updates so that you can inspect the frequency response at any point in the area quickly. You may have already noticed that this configuration is still less than perfect (all are to some degree). It is left as an exercise for the student to discover why and develop a better solution (hint: front-row-center). Some possibilities to consider may be,

- Wider dispersion angles (practical?).
- More speakers in cluster.
- Front fill speakers (use cutoff frequencies to remove low frequencies).

At low frequencies, where the speakers are omnidirectional, changing the aim points cannot have an effect on the sound map, so without moving the speakers and the Sound Man, the low-frequency map represents the best possible map for other frequencies[7]. Why? Because if the map is the same at all frequencies, then every listener will receive the same frequency response as the Sound Man; only the level would change. This allows those who want lower levels (how dare they?!) to sit in the back and those who want high levels to sit in the front, and everyone receives high-quality sound (assuming the Sound Man produces it). And that is the objective of WinSound.

[7]Of course, this is only true if the low-frequency distribution is well-controlled, as it now is in this example.

Philosophy of WinSound

WinSound is designed to address the need of the medium-sized church with the assumption that it will be used by a non-engineer volunteer. Real experts in the field have much more powerful—and more complicated and expensive—tools at their disposal. Hence, WinSound's user interface and its level of detail are kept as simple as possible while still providing answers to the most common sound problems in churches and other sound-reinforcement applications.

In short, WinSound answers the question, "How do I choose and use the right speakers?" It was created under the following basic theory: *Every church has at least one member who is proficient in high-school mathematics, possesses an interest in sound reinforcement principles, and has a little extra time to contribute toward making the system the best it can be.* Let's take a look at how these components affect the development of WinSound:

- **High-school mathematics:** Performing the computations by hand that WinSound does internally would require mastery of advanced mathematics—e.g., trigonometry, coordinate transformations, vector algebra, etc… WinSound presents a user interface that minimizes the math requirements. Still, one needs to know how to tell WinSound where various components are, and the language of location is the Cartesian coordinate system. Thus, if you can understand that the point (3,4,5) is 3 units to the right (along the x axis), 4 units forward (along the y axis), and 5 units above the origin (along the z axis), you are ready to plunge in.
- **An interest in sound reinforcement principles:** Paul says it best:

  > We have different gifts according to the grace given us. If a man's gift is prophesying, let him use it in proportion to his faith. If it is serving, let him serve; if it is teaching, let him teach; if it is encouraging, let him encourage; if it is contributing to the needs of others, let him give generously; if it is leadership, let him govern diligently; if it is showing mercy, let him do it cheerfully. Ro 12:6–8

  If permitted, I would add, *if it is designing sound systems, let him do it with WinSound*. The point is that not everyone will enjoy this activity, but I believe God will place someone in your church with the unique personality to enjoy and excel at this (such people do exist; I'm one of them). Such a person is not necessarily the sound operator either. A critical ear will do nothing to help you design your system until you actually assemble and test it. WinSound requires a critical mind. The basic activity is making judgements about the meaning of numbers and graphs as they relate to the quality of sound.
- **A little extra time:** The principle benefit of WinSound is the expense it saves by letting a volunteer—rather than a professional consultant—do the time-consuming tasks of measuring the room geometry, researching and entering speaker specifications, and iterating toward an acceptable design. If your church is large enough to afford a professional, by all means, use one. If your building has not been built yet, then *definitely* use one. WinSound will not alert you to a poorly designed room, and the expertise required to produce a good room is significant. It will be money well spent. In the mean time, you can still use WinSound to investigate the consultant's recommended design as to whether your needs have been clearly communicated and understood.

These fundamental principles have led to a computational approach to sound reinforcement that balances the need for accuracy against the pitfalls of complexity. This approach is characterized by the following features:

- **Direct Sound Intensity:** WinSound computes the total direct sound at each location. It uses the properties of the speakers (directivity, frequency response, and efficiency) along with the positions of the speakers, the Sound Man, and the listeners. Thus, it does not attempt to model reflections, reverberation, or room resonances. These are complex subjects and are beyond the scope of WinSound's intent. But even in the presence of these phenomena, the strategy remains the same: focus the sound on the audience—which is highly absorbent—and avoid spilling it onto reflective surfaces. And WinSound is designed to help you do just that.
- **The Sound Man:** The Sound Man is the heart of WinSound. The central idea is that a sound operator will do whatever he (or she) can to make it sound good to him. Usually, anyone who hears the same sound as the Sound Man hears the best sound that the system is capable of producing. Therefore, the Sound Man is the standard against which all other listening references are judged.
- **Third-Octave Frequencies:** WinSound is a third-octave-analysis program. That is, sound pressure levels are computed for 29 discrete frequencies ranging from 31 Hz to 20 kHz—generally the range of human hearing. These are the same frequencies that appear on third-octave equalizers and sound-level meters.
- **The 2D View, 3D results:** The main working area in WinSound appears as the top view of a listening area. The heights of speakers and listening regions cannot be seen directly, but the height is used in the computation and is always available through a mouse click. 3D graphics are an expense and complication that WinSound was intended to avoid.

The process of designing the sound system with WinSound is an iterative one, as indicated in Figure 3. The first step is creating the model—a mathematical representation of your system and listeners. This process consists of (1) telling WinSound what *type* of speakers you have, (2) telling WinSound where they are located, and (3) telling WinSound where the Sound Man and listeners are.

The next step is launching the analysis and interpreting the results. This is where you tell WinSound to crunch the numbers, and you

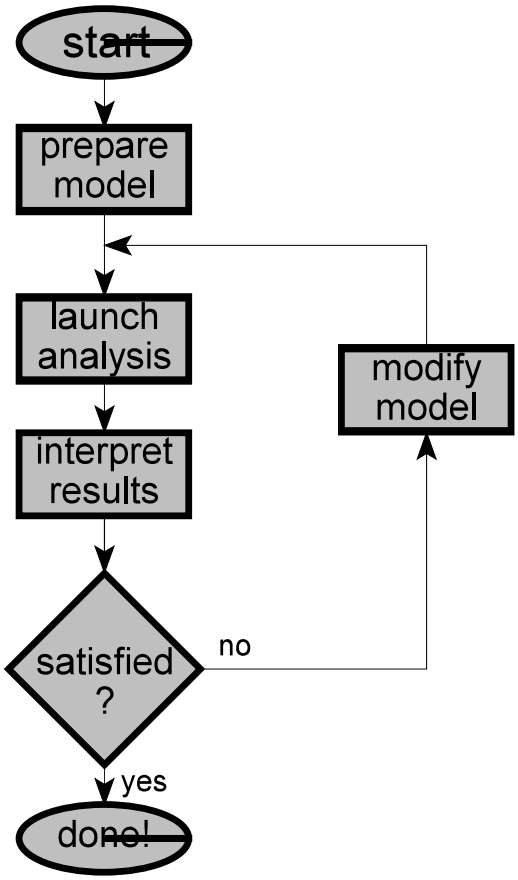


Figure 3 The iterative process of designing a system with WinSound.

decide what they mean. You use WinSound's graphics to decide whether everyone in the audience is getting good levels at all frequencies.

The chances are that your first attempt will not be satisfactory. If not, you follow the "no" arrow through the iteration loop of the process. The next step is to modify the model. You might simply re-aim your speakers to reach listeners in "dark" corners. You might add more speakers or change their type. You might move the audience boundaries around to locate people where the sound is best. You might move the Sound Man where he can hear better. With each change you make, you will launch a new analysis and evaluate the success of the change. Finally, you will decide that no more improvement is needed. You're done. Go build the system!

Preparing Models

The first step in designing a sound system with WinSound is preparing the model. This involves three things: (1) telling WinSound what *types* of speakers you have; (2) telling WinSound where they are placed and aimed; and (3) telling WinSound where the listeners are. However, before beginning, the best practice is to create a sketch of the environment showing where things are located. Using the example in the tutorial, the sketch may have appeared as follows:

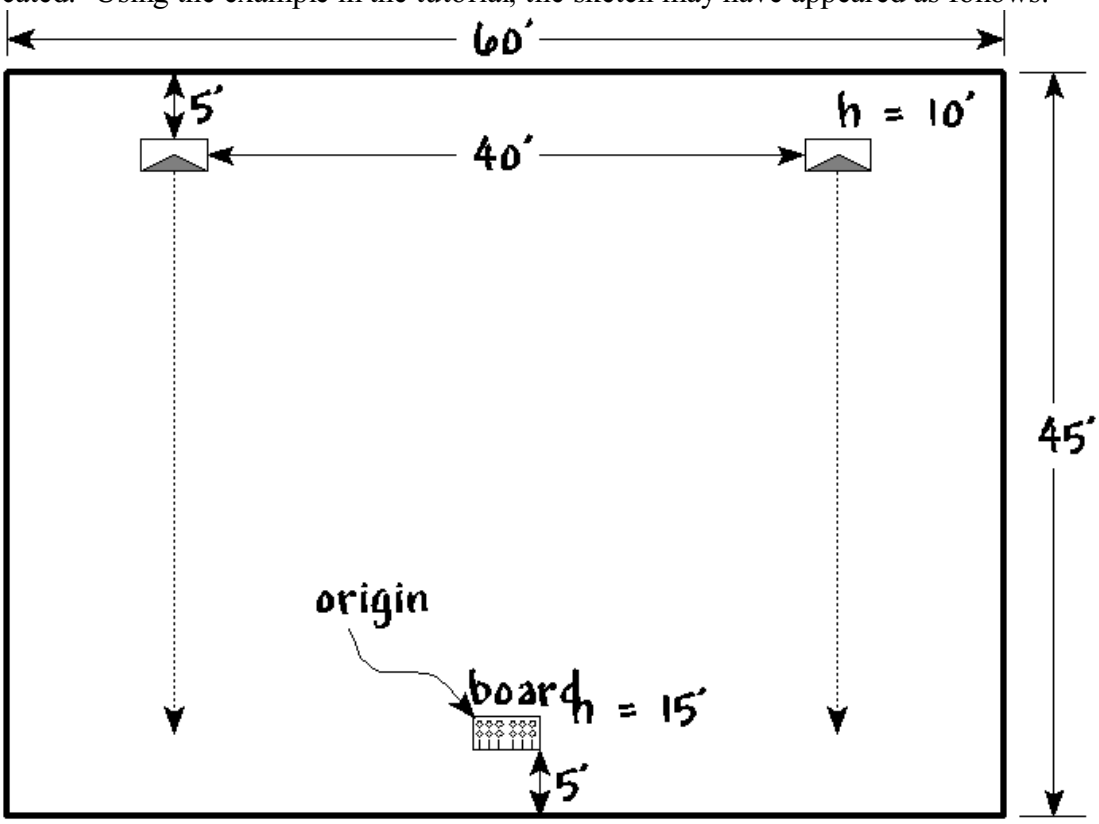


Figure 4 Sketch of system described in tutorial.

As shown, the origin (i.e., the point (0,0,0)) must be chosen so that the locations of items can be entered in WinSound. For instance, with the origin at the point indicated, the mixing board is at the point (0,0,15); the right speaker is at (20,35,10); etc... Choose an origin along a vertical centerline, if possible, so that symmetry can be taken advantage of as with the speakers (Recall that in the tutorial the pair was defined by only one speaker's coordinate).

Creating Speaker Types

The speaker type contains the information about how the speaker converts electrical energy into sound and how it radiates sound into space. A typical sound system may contain several speakers of the same model. In WinSound, these will all share the same speaker type. Speaker types are created through the dialog which appears when you select
Model | Create Speaker Types…, or press **F1**:

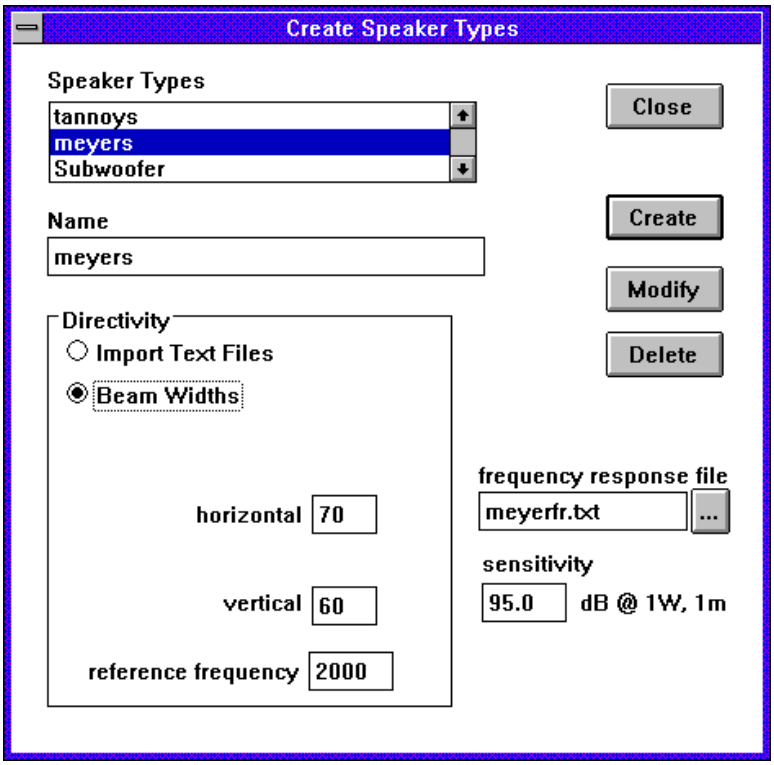


Figure 5 *Create Speaker types* dialog box.

A speaker type is defined by its sensitivity, frequency response, and directivity. The sensitivity is the sound pressure level in dB one meter in front of the speaker when one watt of electrical power is put into it. Thus, it has the unit of dB_{SPL} @ 1W, 1m. It is often reported as a single number, but, in truth, this is an average over a range of frequencies, so the frequency response gives us more detail. Thus, if the documentation tells you that the sensitivity is 95 dB, your first question should be, "at what frequency?" For example, suppose the sensitivity of a speaker is rated at 95 dB, and the frequency response is as shown in Figure 6. Wherever the frequency response graph reads 0 dB, the sensitivity is 95 dB—the "nominal" value. However, at 70 Hz the frequency response is –10 dB. This implies that the true sensitivity at 70 Hz is 95 – 10 = 85 dB. Likewise, at 12 kHz the frequency response is +5 dB, which implies that the true sensitivity at 12 kHz is 95 + 5 = 100 dB.

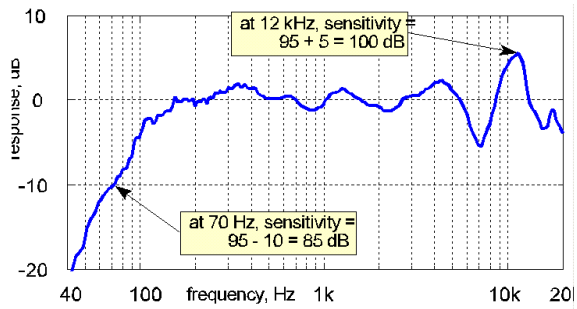


Figure 6 Relationship between sensitivity and frequency response. Wherever the frequency response is zero, the sensitivity is the nominal value of 95 dB.

Frequency response and sensitivity are entered by means of the two lower-right controls in the *Create Speaker Types* dialog shown in Figure 5. The sensitivity is entered directly, but the frequency response is entered as the name of a file. The file can be selected by clicking the small button just to the right of the text control. This file should contain two columns of text; the first column is the frequencies, and the second column is the response corresponding to those frequencies. A portion of the file used to create Figure 6 is shown in Table 1. The values in any row are separated by tabs or spaces. The constraints on this file are as follows: (1) the frequencies need to increase with each row; (2) the response is in dB; (3) each column must contain the same number of values. If the range of frequencies is less than the full bandwidth used by WinSound (i.e., 31 Hz to 20 kHz), WinSound will make some assumptions. Specifically, WinSound will apply 12 dB/octave rolloff for frequencies below the minimum and/or above the maximum frequencies in the file. Otherwise, WinSound will use linear interpolation for frequencies within the range of the file.

Notice the values in Table 1. The response values are all much greater than a few dB. A common way of measuring frequency response and sensitivity is to measure them together at all frequencies, as is done here. WinSound will notice if the average value is greater than 50 dB and will subtract it from the frequency response. It will ask if you would like to use the average as the nominal sensitivity rather than the value entered in the text box. This method is more accurate than entering the sensitivity. However, if for some reason you prefer to use the entered value, tell WinSound no, and it will use the entered value. But WinSound will still subtract the average value internally so that frequency response is still measured with respect to sensitivity.

The easiest way to create the file (if it doesn't already exist) is to use a spreadsheet (such as Microsoft Excel or Borland Quattro Pro). Select the two columns; copy them to the clipboard; open Notepad; choose Edit | Paste; and save the file.

40.0	73.7
41.2	74.3
42.5	74.8
43.7	75.8
45.0	76.8
46.2	77.0
47.5	77.4
48.7	77.7
50.0	78.6
51.5	79.7
53.0	80.7
54.5	81.1
56.0	81.5
58.0	82.0
60.0	83.1
61.5	83.3
63.0	83.8
65.0	84.2
67.0	84.5
69.0	84.7
71.0	84.9
73.0	85.3
75.0	86.0
77.5	86.1
80.0	86.8
82.5	86.9
85.0	87.2
87.5	88.1
90.0	88.4
92.5	90.0
95.0	90.5
97.5	90.5
100.0	90.8

Table 1 Portion of frequency response file showing 40 Hz to 100 Hz.

The next question is, what happens if we are not directly in front of the speaker. By experience we know that if we stand behind a speaker, the level is much lower than if we stand in front of it. And the more "in front of it" we are, the higher the level. This is the phenomenon known as directivity. A flashlight exhibits the same behavior with light.

WinSound offers two methods for building directivity patterns. Figure 7 shows the dialog for the simplest method of creating a directivity pattern—entering beam widths. When the Beam Widths radio button is chosen, the controls appear in the directivity box. Many manufacturers of professional loudspeakers do not provide full polar patterns. Instead, they will list the "beam widths" or "dispersion angles". For instance, the specification sheet may have an entry like this:

dispersion angles (-6 dB): 120H by 80V at 2 kHz

This says that in the horizontal plane at 60˚ (i.e., half of 120˚) off the front axis the sound intensity is 6 dB lower than directly in front of the speaker. Likewise, in the vertical plane the intensity is 6 dB lower at 40˚ (i.e., half of 80˚) off the front axis. The "6-dB-down angle" is half the beam width because the "beam" includes both sides of the axis. For instance, if we took our example speaker with its sensitivity of 95 dB and put one watt through it, we would measure 95 dB at one meter directly in front of it. Now suppose we walked around it 60˚ to one side so that we measure one meter away. Now our reading would be 95 – 6 = 89 dB. Alternatively, we could move the microphone up 40˚ above the center axis and again read 89 dB. The former reading is the "off-axis response" in the horizontal plane, and the reading could be repeated at all angles in this plane. Likewise, the latter reading is in the vertical plane, and the "off-axis response" could be measured at all angles in that plane. To show these measurements we would use "polar patterns" for the two planes. These polar patterns would look like the following:

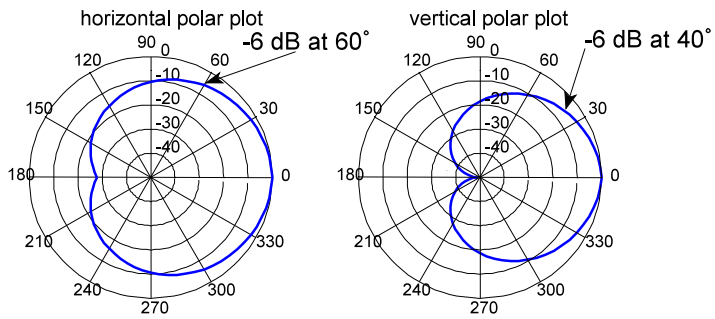


Figure 8 Horizontal and vertical polar plots for 120H-by-80V beam widths.

The polar plot is a graphical way of showing what happens to the level as we move the microphone around the speaker while holding it at the same distance. Zero degrees always refers to directly in front of the speaker, 180˚ always refers to directly behind the speaker (where the level drops the most). For the horizontal polar plot, 90˚ would be a position directly to the speaker's left, and 270˚ is to its right. For the vertical polar plot, 90˚ would be a position directly above the speaker, and 270˚ is directly beneath it.

However, these figures are only true at the reference frequency of 2000 Hz. Often the specification will omit the level drop (-6 dB) and the reference frequency. By convention, a 6 dB drop is implied. No convention exists for the reference frequency, but it is recommended that 2 kHz is used in WinSound if the reference frequency is not given. At frequencies higher than the reference, the beam width usually gets narrower—that is, the 6-dB-down angle gets smaller. At lower frequencies, it gets wider. WinSound will generate the polar patterns for all frequencies based on the horizontal and vertical beam widths and the reference frequency. For frequencies below 250 Hz, WinSound assumes the speaker is "omnidirectional." An omnidirectional speaker radiates sound the same in all directions, so the polar plot is just a circle[8]. The reference frequency should not be lower than 1000 Hz if the resulting pattern is to be realistic. Also, the beam widths should be no less than 40˚. Of course, the resulting pattern is only a crude estimate

[8]Using the analogy of light, a lightbulb is almost omnidirectional.

of the true directivity, but it is a useful tool for investigating possible types of speakers without the laborious work of entering the full polar patterns—which brings us to our next point.

The other method for building directivity patterns uses text files that contain the horizontal and vertical polar patterns for various frequencies. When the Import Text Files radio button is selected, the dialog has the following appearance:

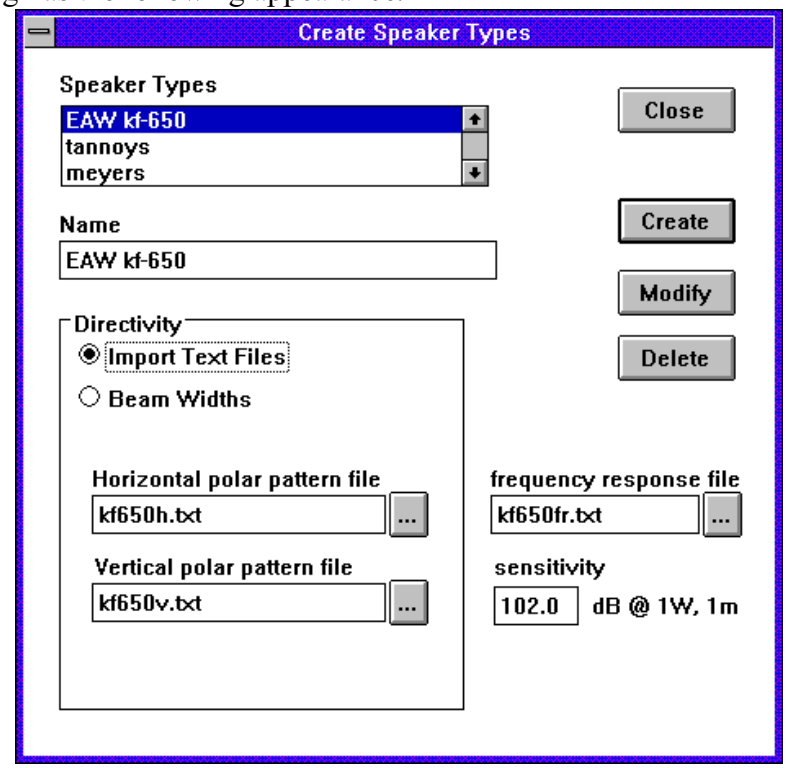


Figure 9 *Create Speaker types* dialog with *Import Text Files* method selected.

The names of the text files are entered in the text controls. The buttons to the right of the text controls bring up a file dialog so the file can be chosen. As in the case of frequency response, these files will contain numbers arranged in rows and columns. The first row should contain the angles starting from 0˚ and ending with 360˚ at which response has been measured. The first column should contain the frequencies for the various polar patterns. Thus, each row is a polar pattern for that specific frequency. IMPORTANT: The first number in the top row is also the first number in the left column. This number has no meaning, but it *must be present.* A zero is preferred. An example is shown below:

		angles									
	0	0	5	10	15	20	25	30	35	40	45
frequencies	100	0	0	0	-0.1	-0.3	-0.4	-0.3	-0.6	-0.9	-1
	125	0	0	0	-0.3	-0.5	-0.6	-0.8	-1.1	-1.1	-1.1
	160	0	-0.1	-0.1	0	-0.1	-0.3	-0.5	-0.8	-1	-1.3
	200	0	-0.2	-0.4	-0.5	-0.5	-0.6	-0.8	-1.1	-1.3	-1.5
	250	0	-0.1	-0.2	-0.3	-0.4	-0.6	-0.9	-1.2	-1.6	-2
	315	0	-0.1	-0.2	-0.3	-0.6	-0.8	-1.2	-1.6	-2.1	-2.5
	400	0	0	-0.2	-0.3	-0.6	-1.1	-1.5	-2	-2.6	-3.3
	500	0	0	-0.1	-0.4	-0.8	-1.2	-1.8	-2.6	-3.5	-4.5
	630	0	-0.1	-0.3	-0.7	-1.2	-2	-2.8	-3.6	-4.6	-5.6
	800	0	-0.1	-0.4	-1	-1.9	-3	-4.4	-5.8	-7.5	-9.5
	1000	0	-0.1	-0.6	-1.5	-2.8	-4.6	-6.7	-9.2	-12	-16

Table 2 Format example for a polar data file. Angles 0 through 45° shown. Notice the zero in the upper-left corner.

Of course, the lines, shading, and the words *angles* and *frequencies* are included only for illustration—the actual file would only contain the numbers, and the angles must continue to the right all the way through 360°. Also, it should contain polar data for higher frequencies as well. The numbers represent "off-axis attenuation"—thus they are usually negative, and all values for 0° and 360° are zero or at least very close to zero. For frequencies lower than the lowest contained in the file, an omnidirectional pattern is assumed. For frequencies higher than the highest contained in the file, the highest-frequency pattern is copied. Example files are included on the disk and have the names ?????h.txt and ?????v.txt for the horizontal and vertical patterns, respectively. Again, the best way to build these files is to copy the numbers into Notepad from a spreadsheet.

Once the frequency response, sensitivity, and directivity pattern have been set, give the speaker type a unique name—usually containing the brand and model name—and press Create. If there are any errors in the settings or files, they will be detected and a message will be issued. At this point the speaker type is loaded into WinSound and can be used in placing speakers. Its name appears in the Speaker Types list box. Double-clicking this name will fill the dialog with the properties of that type.

Placing Speakers

Once all the speaker types have been defined, speakers can be created and placed at three-dimensional locations. The Place Speakers dialog is accessed by selecting Model | Place Speakers…, or pressing **F2**, and is shown below:

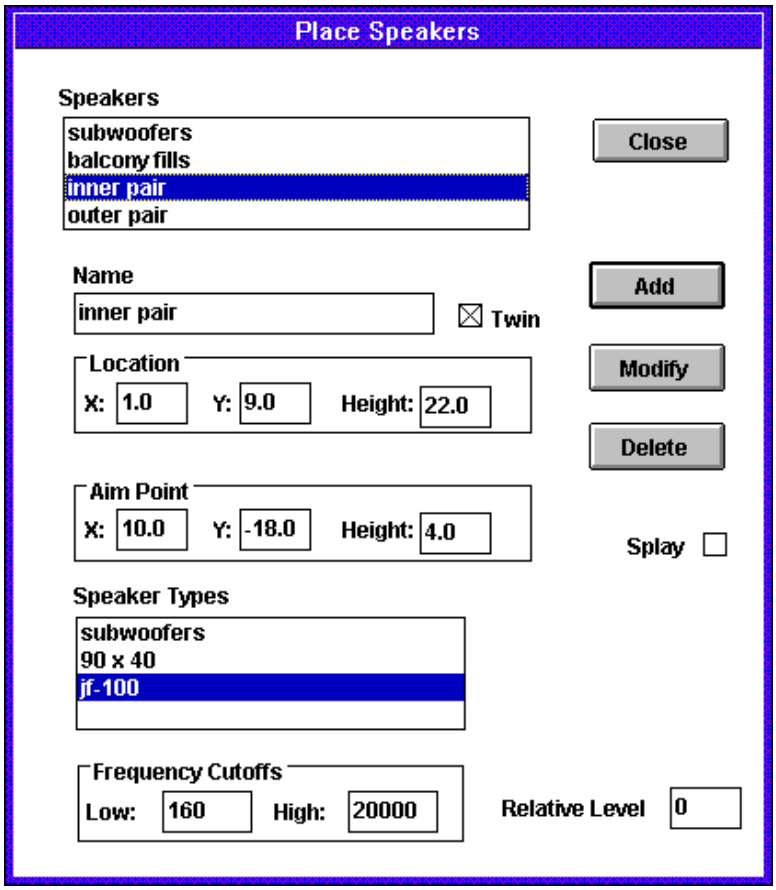


Figure 10 The *Place Speakers* dialog box

First, notice the Location controls group. These three coordinates define the position of the speaker. The x coordinate is positive to the right in the WinSound main window, and it is usually zero at a point of symmetry (more on symmetry later). The y coordinate is positive toward the top of the main window, and there is no rule on where it should be zero—you choose it for your own convenience. The Height coordinate (also known as *z*) is positive in the direction out of the screen. A height of zero should correspond to the lowest floor level. The coordinate system should be chosen in advance and should be carefully considered. It will be the same system used to locate the Sound Man and the audience in addition to the speakers.

An important companion of the Location group is the Aim Point group. This is how you tell WinSound how the speaker is oriented in space. The speaker will be oriented in such a manner that a person looking at the speaker from the aim point would see the speaker aiming straight at him and would appear to be sitting top-side-up. In other words, an upright speaker is first rotated toward the lister about its *z* axis and then it is tilted up or down so that the listener is on its central axis. Notice the convenience this provides: when you actually hang the speaker, you can know if it is aimed correctly by looking at it from the location of its aim point.

Next, notice the Twin checkbox. This is where symmetry becomes important and can make your life much easier. Most sound systems and audience areas are symmetric—that is, a line can be drawn down the middle so that the left side is the mirror image of the right side. The stage is

usually in the center, and speakers are placed and aimed the same way on the right as they are on the left. The Twin checkbox allows you to take advantage of this by entering data for one side only. WinSound will generate the mirror image, or, the "twin." The twin speaker will have the same height and the same *y* coordinate as the one you enter, but the *x* coordinate will be the negative of the one you enter, as indicated in Figure 11. The two will work together so that any changes made to the one will be made to the twin as well, but only the speaker you enter will appear in the *Place Speakers* dialog. If the pair is a cluster at the centerline, the x coordinate should be half the distance between them.

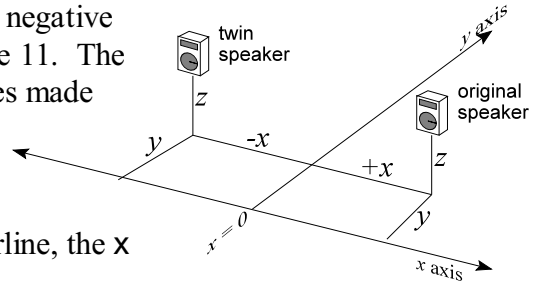


Figure 11 The "twin" speaker's location. *y* and *z* are the same, but *x* is negated.

Since a speaker is modeled as a point source, you need to pick a point within the speaker to represent the position of the entire speaker. The best choise is probably the center of the lowest-frequency driver. The reasoning here is that phase interference will be modeled based on this point[9]. Phase is generally more of a concern at lower frequencies, and WinSound allows you to account for phase at frequencies below a certain frequency which you choose. Ultimately, only you can decide how to model the speaker as you begin to understand more, and if you wish to model phase interference for higher frequencies, you need to choose the driver that produces them. But for now, the low-frequency driver is a good starting point.

In addition to locating and aiming the speaker, you must tell WinSound what type it is. The speaker types that you defined in the *Create Speaker Types* dialog are listed here in the listbox.

Many systems contain subwoofers at different locations than the main speakers. This is because subwoofers are heavy, low frequencies have little directionality, and subwoofers are more efficient when placed on the floor or against a stiff wall. The Frequency Cutoffs control allows you to model this behavior. For example, consider a system with a crossover set at 100 Hz. The subwoofer would have low and high entries of 31 Hz and 100 Hz, respectively (31 Hz is the lowest frequency modeled by WinSound). At 100 Hz, the power sent to the subwoofer is reduced by 3 dB, and for frequencies higher than 100 Hz, it is reduced by an additional 12 dB per octave[–15 dB at 200 Hz, –27 dB at 400 Hz, etc…]. The main speaker would have low and high entries of 100 Hz and 20 kHz, respectively (20 kHz is the highest frequency modeled by WinSound). At 100 Hz, the power sent to the main speaker is reduced by 3 dB, and for frequencies lower than 100 Hz, it is reduced by an additional 12 dB per octave [–15 dB at 50 Hz, –27 dB at 25 Hz, etc…].

Finally, if your system contains multiple speakers covering various audience areas, you may wish to adjust the volume to one or more speakers. The Relative Level control allows you to do this. Most speakers should be left at zero. Entering the relative level in dB will cause that much more power to be sent to that speaker. A negative value will cause that much less power to be

[9]More on 'phase' in the section "Launching Analyses."

sent. Usually this will be left at zero when the speakers are created, and it can be modified later during analysis by accessing the speaker's popup menu.

An advanced option for placing speakers is provided by the checkbox labeled Splay. Splayed speakers are physically connected together side-by-side, so moving one speaker requires moving all the speakers in the splayed cluster. When the Splay checkbox is checked, additional controls appear allowing you to enter the splay angle and the number of speakers splayed to the left and to the right of the center speaker.

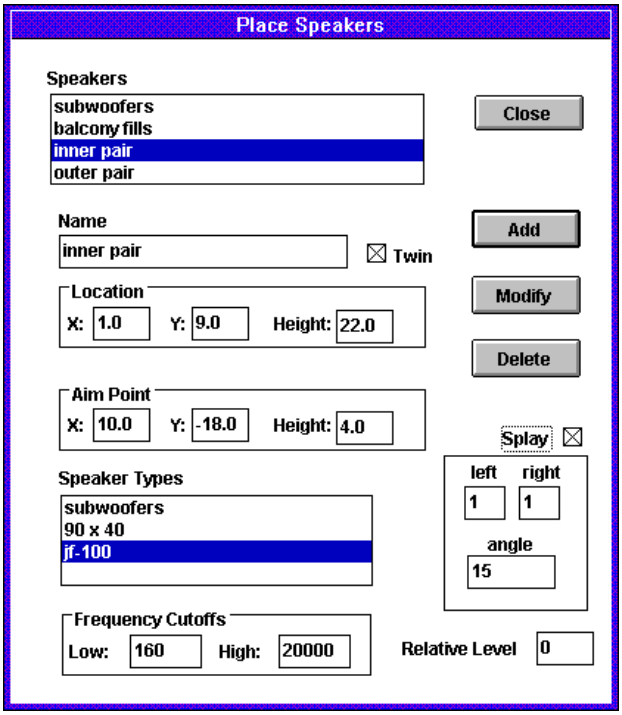


Figure 12 The *Place Speakers* dialog with the splay controls active.

The center speaker is the one that is defined in the Location and Aim Point controls. The controls labeled left and right are the number of speakers to the left and right of the center speaker, respectively (they need not be equal or nonzero). The spay angle is the angle, in degrees, between the direct axes of two neighboring speakers. The aim point of a splayed speaker is calculated by WinSound based on the splay angle and the number of speakers between it and the center speaker. All

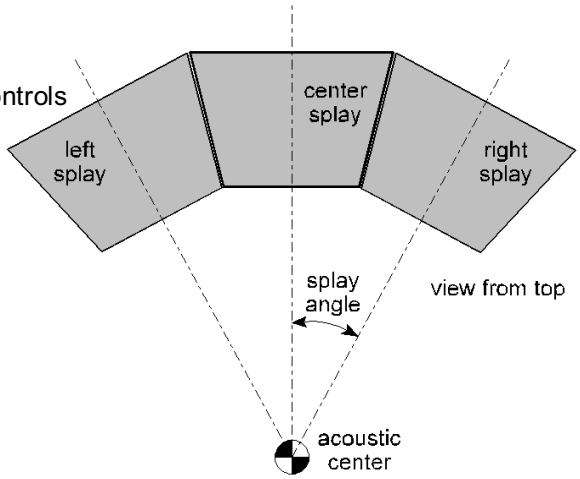


Figure 13 A splayed speaker cluster with one left and one right splay. The splay angle is the angle between the direct speaker axes.

speakers in a splayed cluster have the same location. This is because the intent of a splayed cluster is that the cluster behaves as one speaker with an "acoustic center", which is at the intersection of the direct axes. The arrangement is illustrated in Figure 13. Notice that "left" and "right" are defined from the vantage point of the center speaker. If the Twin checkbox is checked, an identical cluster will be created, but the number of splays to the left and right are reversed so that the twin cluster is the mirror image of the original cluster.

After adding a unique name for the speaker (usually indicating the type and location), pressing Add will create the speaker. Its name will appear in the listbox at the top. Double-clicking on a speaker in the list causes the dialog's controls to be filled with that speaker's properties. Pressing Delete deletes the selected speaker, and pressing Modify causes the selected speaker to be replaced with the one defined in the controls. Thus, the best way to make a small change to a speaker is to double-click it, make the change, and press Modify.

Creating Audience Boundaries

Telling WinSound where the audience is sitting is surprisingly easy even though powerful options are available. In WinSound the audience is located in a set of planar regions called "boundaries." The term "boundaries" indicates the fact that the region is defined by a set of vertices joined together by straight lines—polygons. Each region lies in a plane defined by three of the vertices. This allows one to model sloped balconies and similar seating arrangements since the three vertices need not have the same height. WinSound uses the boundaries to determine the height of listeners that fall inside the boundary. It also draws the boundary lines for a visual aid in interpreting the results. The *Boundary Editor* dialog is shown below:

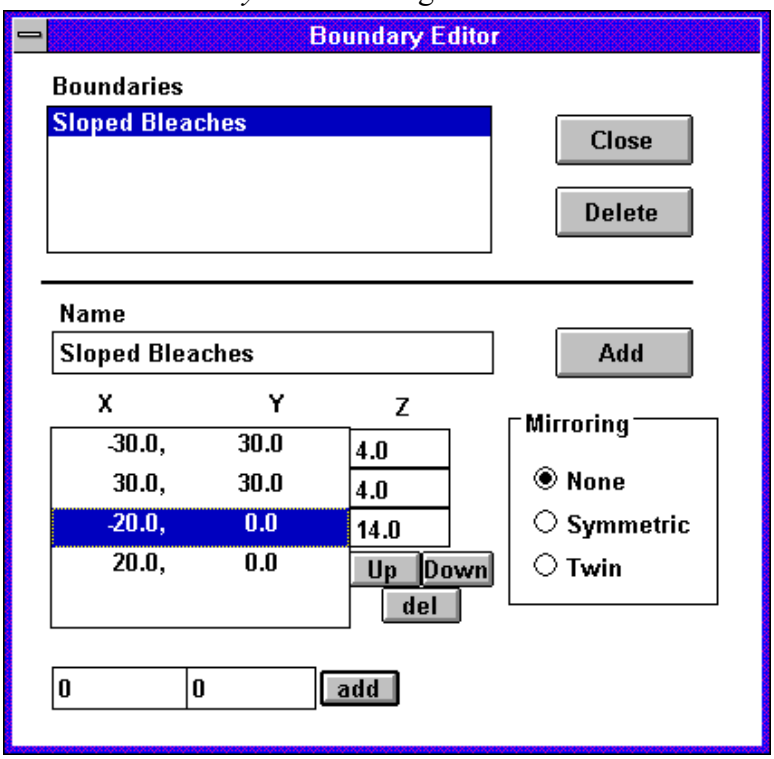


Figure 14 *Boundary Editor* dialog

There are three types of boundaries, indicated by the three radio buttons in the mirroring control group. The first, labeled none, has no mirroring (more on that later). The *x-y* coordinates of the vertices are entered via the two text controls at the bottom. Pressing the small add button on the right adds the *x-y* coordinate to the list. For efficiency, pressing the **enter** key adds the *x-y* coordinate and returns the cursor to the x coordinate text control—this way you can enter all coordinates without having to use the mouse. Once the vertices are entered, you need to tell WinSound which three points define the plane that the region lies in. This is done using the three text controls under the label Z where you enter the three heights. The first height corresponds to the first vertex in the list, the second to the second, and the third to the third. These should correspond to the heights of the listeners' ears. To match the vertices to the heights, use the Up and Down buttons to move the selected vertex until it lines up with the desired height. Finally, add a unique name for the boundary and press the large Add button on the right of the dialog. WinSound will sort the vertices, connect them with lines, and create the boundary.

As an example, consider the sloped bleaches modeled in Figure 15. The vertices are in Cartesian coordinate notation (*x,y,z*). The front row seats (at $y = 30$) are all at a height of 4 feet while the back row is at 14 feet. This boundary would be entered as shown in Figure 14. In this case, any three of the four points can be used to define the plane. But in general, care should be used: The 3 points must not lie in the same line. In fact, they should not even be close to any one line but should form as equilateral a triangle as possible. Of course, if the region is not sloped, all points will have the same height, but the height must still be entered. Also, if the slope is steep, be careful to enter the *x-y* coordinates of the vertices with respect to the flat coordinate system rather than with respect to the plane that the region lies in. For instance, using the above example, if you measure the distance from the front to the back row by dragging a tape measure from one seat to another—allowing it to slope with the bleaches—you will erroneously measure 31.6 feet rather than the correct value of 30 feet, which would result by holding the tape horizontal.

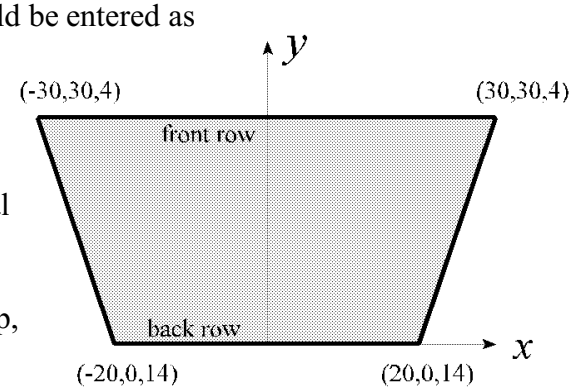


Figure 15 Schematic for modeling a sloped audience area.

The sloped-bleaches example serves to illustrate the second type of boundary—the symmetric boundary. The sloped bleaches are symmetric about the *y* axis, and we can take advantage of this. If we choose the symmetric radio button from the mirroring group, WinSound will "mirror" the points we enter about the *y* axis for us; we only need to enter the right-hand side (positive *x* values). Thus, we could create the same boundary with the following settings:

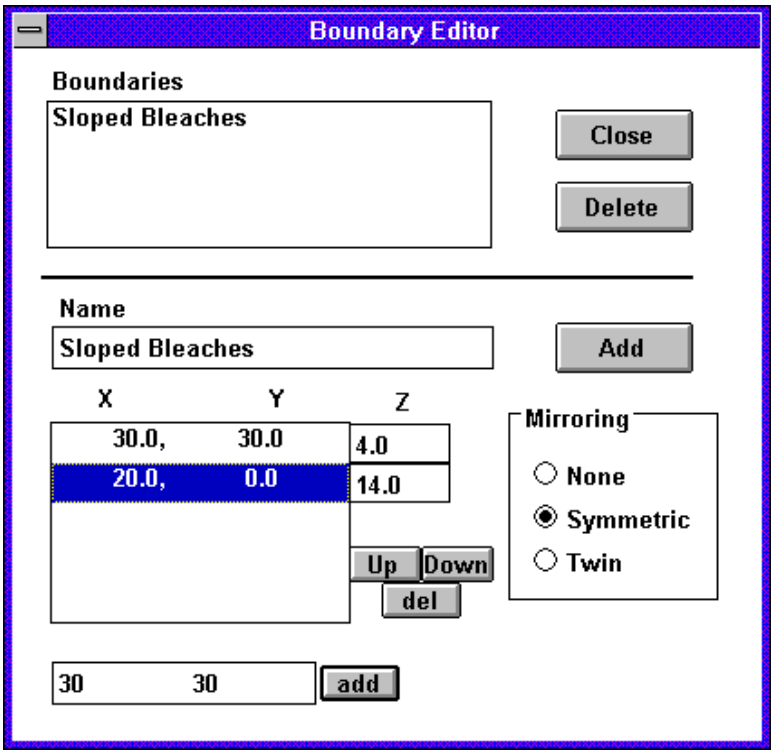


Figure 16 Entering a symmetric boundary.

With the symmetric boundary, we only need to enter two heights since those two points are mirrored across the *y* axis.

The third type is the twin boundary. In this case, you enter all the coordinates of an arbitrary boundary, just as with the none type. However, pressing Add creates two boundaries; one is the one you entered, and the other is the mirror image across the *y* axis. Generally the vertices entered should all be in the right-hand side (positive *x* coordinates) because if the boundary crosses the *y* axis, it will overlap with its twin resulting in confusion.

There are a few general rules to keep in mind when creating boundaries:

- WinSound uses the "centroid", or average of the vertices when it sorts the points and when it decides whether a point is inside the boundary or not. The centroid must not coincide with one of the vertices. For example, the boundary shown in Figure 17 cannot be entered. The centroid is at (0,10,4)—one of the vertices. It must be split into two boundaries—for instance, along the dashed line. WinSound will tell you if this happens.

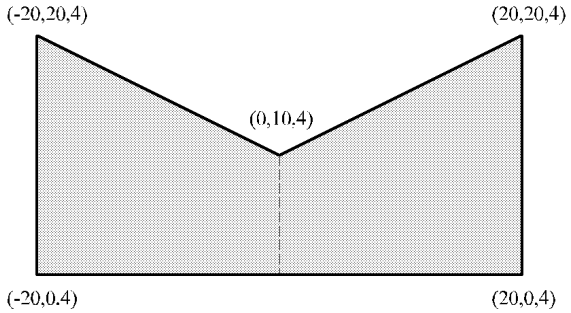


Figure 17 Example of a bad boundary. The centroid coincides with a vertex.

- WinSound will try to sort the vertices so that a line from the centroid to a vertex will never cross a boundary line. Such a boundary is said to be "starlike." If your boundary is such that one of the boundary lines lies in one such line, WinSound will issue the error message, "The boundary is not starlike." An example of such a boundary is shown in Figure 18. The centroid is at the point (0,0,4). The diagonal, dotted line shows how a line connecting the centroid and a vertex can contain one of the boundary lines. One solution would be to split this into two triangular boundaries along this line.
- If boundaries overlap, only the boundary toward the top of the list will own the points within the overlap region.

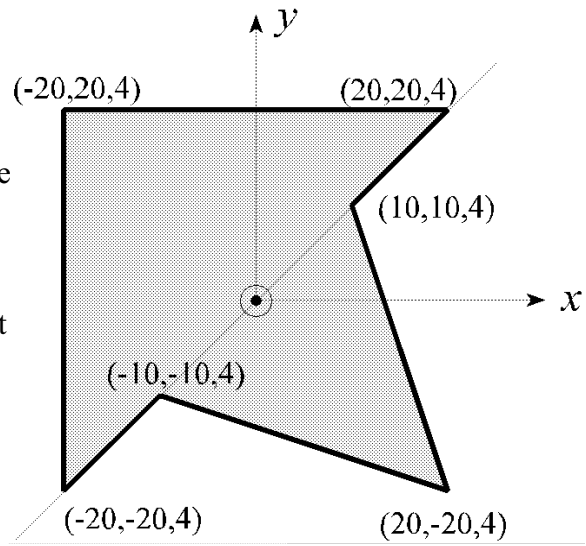


Figure 18 A boundary which is not "starlike."

Defining The SoundMan

The location and desired level of the Sound Man are accessed by selecting Model | Soundman…, or pressing **F4**. The following dialog appears:

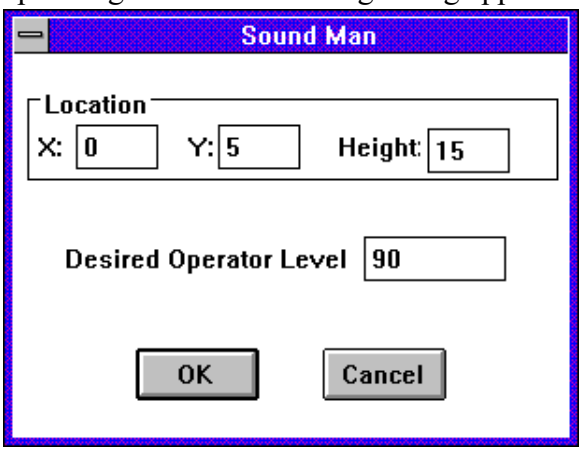


Figure 19 Sound Man dialog.

As with a speaker, the Location controls define the position of the Sound Man. The Height should be that of his ears.

The Desired Operator Level is the level in dB that the Sound Man will receive after adding power and equalization. Here's how it works: Initially each speaker has one watt running through it with no equalization. Those speakers whose reference level is nonzero are adjusted to have the correct power, and cutoff frequencies are applied if necessary. Then the Sound Man receives the total sound from all speakers. For each frequency, he adjusts the equalizer and volume until he hears the desired level.

Model Files

Once the major data has been entered as described above, you will wish to save your work so that you don't have to go through that again. Choose File | Save as…; enter a name with the `.wsd` extension; and press OK. WinSound will save your boundaries, your speakers, your speaker types, and the location and desired level of the SoundMan. Open the file by choosing File | Open, or pressing **`Ctrl-O`**. The items listed above will be replaced by the file contents. File saving can also be handy for comparing two or more proposed designs by opening two instances of WinSound.

Launching Analyses

When you have entered all the model data, it is time to let WinSound crunch the numbers for you. Choose Model | Launch Analysis…, or press **`F5`**. The following dialog appears:

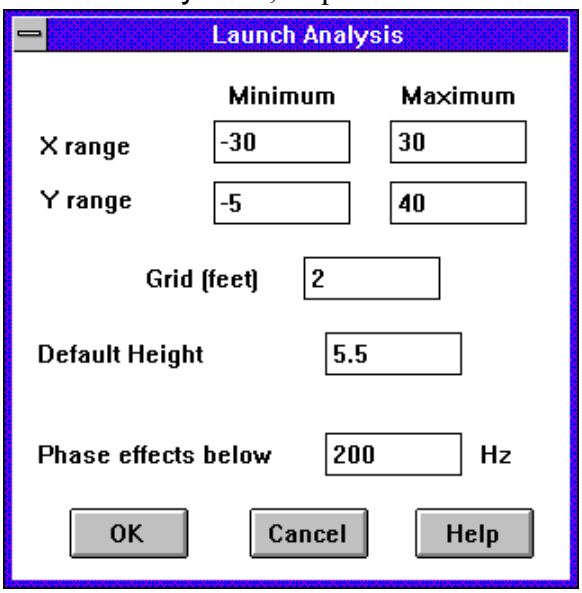


Figure 20 The *Launch Analysis* dialog.

The top five controls specify a rectangular grid within which sound levels will be calculated for all frequencies. The *X* and *Y* ranges are initially filled with the ranges of the *x* and *y* scales in the main window. All locations analyzed will be within these limits even if the boundaries extend outside the rectangle. Since some points within the analysis range may not lie within a defined boundary, the Default Height defines the height that will be used for these points.

The Grid control specifies how far apart (in feet) the locations will be and should be set with some thought. Entering too large a grid will not give enough resolution—that is, it will be hard to know what the level is between locations. On the other hand, entering too small a grid will tax the computer resources heavily, so the analysis will take too long to complete. WinSound will limit the maximum number of locations based on your availabe RAM (4 Mb of RAM will permit about thirty thousand locations—a hefty amount). If you exceed this maximum, it will issue a message recommending the maximum grid size. The number of locations grows with the inverse of the square of the grid—in other words, halving the grid size quadruples the number of locations. Try starting with a 5-foot grid.

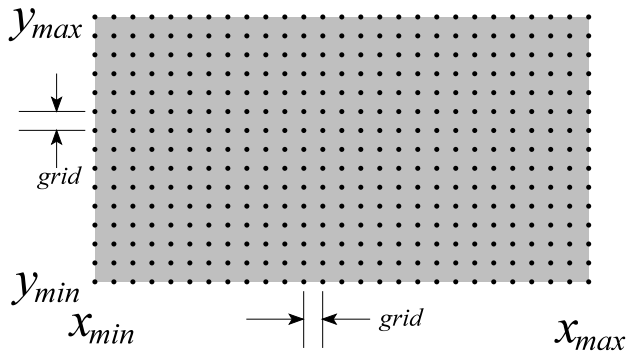


Figure 21 Listening locations created from the range and grid controls.

The last control, Phase effects below, allows you to choose how the sound from all the speakers is added together. For frequencies above this value, the sound is modeled as acoustic energy. Thus, if two speakers produce the same intensity at a location, they will combine to increase the level by about 3 dB above the level created by just one speaker.

For frequencies up to and below this value, the sound is modeled as a spherical pressure wave. Two waves may add together to increase the level by 6 dB if they add *in phase*. Alternatively, they may cancel each other out if they add *out of phase*. Figure 22 illustrates this effect. The wavy lines at the top of the graph represent the pressure waves from each speaker. The waves travel outward at the speed of sound, as indicated by the arrows. The waves are represented by the solid and dashed circular lines around each speaker. Imagine dropping two stones into a pond and watching the ripples radiate from the two locations until they reached each other and began to interact. The solid lines indicate the crest of the wave—or high pressure. The dashed lines indicate the troughs of the wave—or low pressure. Wherever the crest of one wave intersects the crest of the other wave, the pressures will add (if they are equal the pressure will double). However, if the crest of one wave

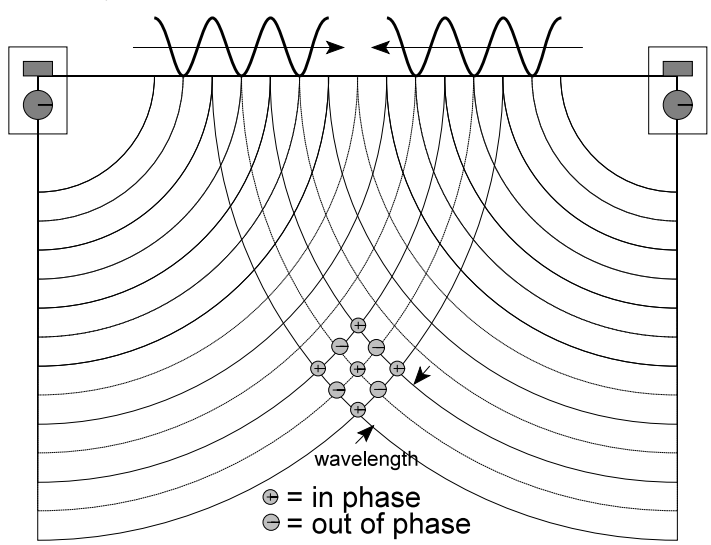


Figure 22 Phase addition and cancellation between two speakers.

intersects the trough of the other wave, the pressures will subtract (if they are equal, the result is zero). This is indicated by the circles with pluses and minuses for a few locations. These are said to be locations of *constructive addition* (in phase) and *destructive addition* (out of phase).

Notice that a plus is also plotted at the intersection of the two troughs. Recall that the waves are moving outward at the speed of sound. The distance between two adjacent crests is the wavelength (represented by λ) which is equal to the speed of sound divided by the frequency:

$$\lambda = \frac{v}{f}$$

where v = 1115 ft/second. At the next instant, the solid lines will travel outward to where the dashed lines are. As can be seen from the illustration, the point where two dashed lines intersect will become a point where two solid lines intersect. By the same token, the intersection of a solid line and a dashed line will become the intersection of a dashed line and a solid line. The point is that the locations of constructive and destructive addition do not move, even though the waves are moving.

However, these locations depend on the wavelength, so they are different at different frequencies. For example, the wavelength for 100 Hz is about 11 ft long. Thus, the distance between a constructive-addition point and a destructive-addition point is about 6.5 feet (half a wavelength). If your grid is one foot, you will see a gradual change in level between these locations. But now consider 10 kHz; the wavelength is about an inch and a half. From one location to the other is about 8 wavelengths. In other words, moving from a constructive-addition location to a destructive-addition location only requires a tilt of your head! Again, if your grid is one foot, two adjacent locations may have constructive and destructive addition, so your results will look like a random-color checkerboard.

This is why phase is only considered below the frequency you enter. At higher frequencies, the resulting sound pressure level is found by simply adding the acoustic energy from each speaker. This tends to average the effects of phase addition and cancellation over frequencies near the one being analyzed and gives a better correlation to the effective quality of the sound. 200 Hz (λ = 6.5 ft) is recommended as a general maximum frequency for modeling phase effects. Beyond this it becomes doubtful that we even know the exact positions of the speakers or the listeners to a fraction of the wavelength.

Now you have entered your grid; you have specified the sound pressure level for the Sound Man; and you have chosen which frequencies should model phase effects. You are ready to let WinSound crunch the numbers for you. Press OK. If there are any modeling errors with your speakers, WinSound will protest. Otherwise, a progress dialog will appear with the title, *WinSound is Number Crunching*. The black rectangle will be filled with red when the analysis is done. In the meantime, WinSound allows you to run other applications while it quietly crunches in the background.

Interpreting The Results

So WinSound has just finished crunching all the numbers, and you feel like you are staring at a Rorschach test. What do all these colors mean anyway? Let's start with the squares. Remember that in the *Launch Analysis* dialog you told WinSound To create a rectangular grid of locations spaced *grid* feet apart. Well each colored square represents one of those locations. The exact location is in the center of the square, but since WinSound has no results for the rest of the square, it regards the entire square as being the same as the center. This allows you to get a feel for how well you have chosen your grid size. If your grid was too large, WinSound probably painted the picture in a fraction of a second, and the color changes drastically from one square to the next. If your grid was too small, you probably had to take a coffee break while WinSound plotted thousands of tiny dots, and the color is so smooth it looks like it was painted with an airbrush. The ideal grid lies somewhere in between: The color is adequately smooth (though you can still see squares), and WinSound takes about a second to draw the picture.

The Colorbar

Now about the color—it represents direct sound pressure level. Each square is filled with the color that represents the level at the location in the center of the square. The level represented by the color can be found on the colorbar. The distribution of color that looks like a Rorschach test is a map that gives a quick visual indication of how the sound is distributed to your audience. Use it to zero in on trouble spots—deafeningly high levels or dead zones.

You may ask, so what is "direct sound pressure level?" Basically, it is a measure of how loud it is. In WinSound, direct sound pressure level at a given location is the total sound energy delivered by all speakers to that location. For a given listener's location, every speaker will contribute *some* sound—even if aimed at a totally remote area. The level indicated by the color is the summed result of the sound from all speakers.

There are some things not included in direct sound that WinSound cannot take into account, and they can be very important. For instance, if a wall is between a speaker and a location, WinSound can't know about it; it assumes the wall is not there. Also, reflections from walls and ceilings are not considered by WinSound. Finally, WinSound does not predict reverberation, which is very important indoors. At remote locations where the direct sound is low, the reverberant sound can be the vast majority of what people hear there. It is up to you to anticipate these effects and judge their relative worth. So why use WinSound if it can't model all this stuff? Because WinSound *does* allow you to employ the strategy to cope with acoustic anomolies:

- **Acoustic shading:** If the most prominent speaker supplying sound to a given audience area is blocked by a wall, the design needs improving. A speaker must be in the line of sight to do well. If the speaker that is shaded by a wall is not the prominent speaker, the error introduced by WinSound's ignorance is probably small enough to ignore. For example: Suppose the prominent speaker supplies 90 dB and the shaded speaker would supply 80 dB without the wall. Due the mathematics of decibels, WinSound adds the two together to

obtain 90.4 dB as opposed to the [more] correct value of 90.0 dB. So WinSound's error is only a fraction of a decibel.

- **Reflections and reverberation:** In the presence of reflective surfaces and/or a reverberant room, the strategy is the the same: Concentrate the sound onto the audience—which is almost perfectly absorptive—and away from the walls and ceiling. WinSound is an excellent tool for doing this. Just draw your boundaries so that you know where the walls are, and avoid letting frequencies above 1 kHz stray outside of them. If in practice this does not work, you have a problem that needs the attention of a skilled acoustician. No speaker or equalizer can overcome a severely reverberant room.

Now that you know the sound pressure level at a location, let's look at the implications of that number. The essence of sound is a variation in pressure, which is a force distributed over an area. Thus, we measure pressure in pounds per square inch, or newtons per square meter, etc… Pressure is always present; our atmosphere is usually about 14.7 psi (1 psi = 1 lb/in^2), but that is the average value. In the presence of sound, the pressure is actually jumping up and down by a small amount. The amount that it jumps up and down is called the *amplitude*, and that's where we compute the level from. For instance, 90 dB_{SPL} corresponds to an amplitude of roughly 9×10^{-5} psi. So the pressure is oscillating between 14.69991 and 14.70009 psi, as shown in Figure 23 (the total variation is twice the amplitude). If the frequency of this sound is 2 kHz (2000 Hz), it will oscillate through two complete "cycles" in one millisecond (a thousandth of a second), as illustrated.

The amplitude is converted to a decibel value based on its size relative to the reference amplitude of 2.9×10^{-9} psi, which is the smallest amplitude the average person can hear. The formula uses the base-ten logarithm:

$$dB = 20\log_{10}\left(\frac{p}{p_0}\right)$$

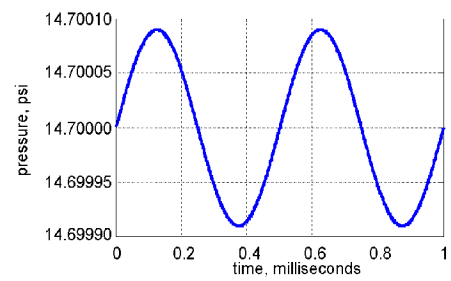


Figure 23 The pressure trace for a 2 kHz, 90 dB_{SPL} sound.

where p_o is the reference level. This formula is used because it produces "nice" numbers. Some rules of thumb with decibels are:

- A change of 1 dB is the smallest perceptible change in level.

- A change of 3 dB requires a doubling of power yet 3 dB is still perceived as a small change in level.
- A change of 10 dB is perceived as a doubling in volume.
- Suppose the pressure were to fluctuate from zero to double atmosphere pressure. Using this equation, the level would be 194 dB. Since negative pressures are impossible, this is the highest theoretical sound level for a pure tone.

Some common levels are shown in Figure 24. Perhaps the best way to develop a feel for sound level is to purchase an inexpensive sound level meter and take some readings during a typical event (e.g., concert or worship service).

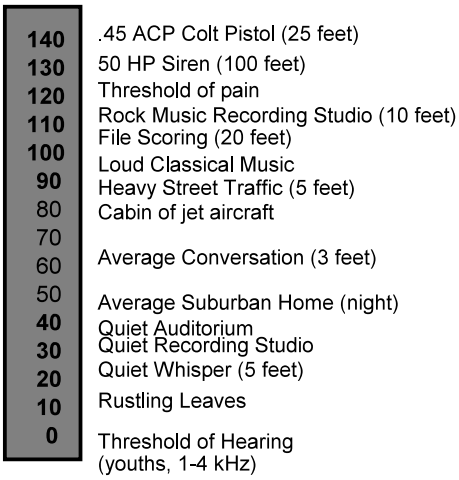


Figure 24 Some common dB_{SPL} levels (from Davis and Jones 1989, 26).

With these relationships in mind, you can use the colorbar intelligently. If, for example, you don't want the quiet portions of the audience to sound less than "half as loud" as the loud portions, you may want to set the minimum level on the colorbar to 10 dB below the maximum. Then any locations with no color (white) must be 10 dB or more below the maximum. You can also autoscale the colorbar (select View | Autoscale Color or press **F6**). All levels analyzed will then be included, and you get a quick indication of the total range. One word of caution though: Any levels above the maximum will be painted red—the same as the maximum—so you may wish to autoscale first before setting the minimum.

Arguably the most important aspect of sound pressure level is the potential for hearing damage. The higher the level, the quicker damage is likely to occur. Table 3 shows one expert's recommendation for maximum exposure times for various levels. Since the music portion of a typical worship service lasts between 30 minutes and an hour, the audience should not be subjected to more than an average of 95 dB. This is especially true for midrange frequencies (from 500 Hz to 4000 Hz) where the ear is most sensitive. Beyond the recommended exposure time, the probability of hearing damage becomes increasingly significant. And tragically, once hearing damage has occured, it cannot be remedied.

level dB_A	*max exposure*
85	2 hr
90	1 hr
95	30 min
100	15 min

Table 3 Recommended sound level limits for typical exposure times.

The Frequency Control

The sound map displayed is a map only for one frequency, so if a location's color indicates 85 dB, we only know it is 85 dB at this particular frequency. To see other frequencies, we need to adjust the frequency control. WinSound is a "third-octave" analysis program. In other words, every two neighboring frequencies are a third of an octave apart (the higher frequency is 1.26

times the lower). The total range is from 31 Hz to 20 kHz—the range of the human ear, and the frequencies coincide with those found on a third-octave equalizer. Clicking one of the scroll buttons will advance to the next available frequency. Clicking *in* the scroll bar will advance one octave in the direction clicked—that is, it will skip the next two frequencies and land on one that is half or double the current frequency. If you drag the thumb (little moving button in the scrollbar), the frequency text indicator will update with the frequency that will be selected if you let go of the thumb. Changing the frequency does not affect the range of the colorbar.

One of the subtler issues to bear in mind when viewing different frequencies is the directionality of the speakers. At low frequencies (31 to 250 Hz), speakers are usually "omnidirectional" (see the section on speaker types for more discussion). Because of this, the resulting sound map comes from the speakers' locations and has nothing to do with how they are aimed, so re-aiming them will not help a problem at these frequencies. Furthermore, keep in mind the frequency below which you choose to model phase. Table 4 shows some common sound sources and their associated frequency ranges. The primary range refers to a sort of "bare minimum" required where the vast majority of energy is concentrated. The extended range refers to that needed to do a good job of recreating the sound. For instance, if a listener gets good response only between 200 and 2000 Hz, you can conclude that voice reproduction will be adequate. But to make a quality reproduction of the voice, that range should be expanded to 80 to 10 kHz. Notice that drums are the most demanding. This is due to the different types of elements used—bass drum, cymbals, snare, high-hats, etc…[10]

source	*primary range*	*extended range*
voice	200 – 2k	80 – 10k
bass guitar	50 – 400	31 – 5k
drums	80 – 2k	31 – 20k
guitar	200 – 2k	100 – 10k
piano	100 – 1k	50 – 4k
strings	200 – 2k	50 – 20k

Table 4 Common sound sources and associated frequency ranges.

The Sound Man

As stated earlier, the Sound Man is the heart of WinSound's analysis. If it sounds too loud, he (or she) will turn it down; if the highs are too soft, he will turn them up through equalization; and so forth… Thus, the assumtion is that the Sound Man has the best seat in the house. This becomes the standard against which all other seats are judged.

The Sound Man has power and equalization at his disposal, and he applies them to receive a given sound level at all frequencies. In other words, he is guaranteed to receive a "flat" response. The power is an average over all the frequencies and gives a ballpark idea of the demand he will likely place on the system. If the SoundMan is adding an average of 200 watts of power, and your speakers can only handle 50 watts, you know that your system needs to be expanded or the

[10]For an excellent discussion of frequency and its implications, see Davis and Jones 1989.

desired level must be reduced. The average power can be viewed by accessing his popup menu—click the right mouse button over him, and choose view power.

Equalization can be viewed by choosing view equalization from the Sound Man's popup menu. Equalization is plotted in dB for all frequencies, so the plot may look like the equalizer when the Sound Man is through with it. The equalization contains clues about the suitability of the Sound Man's location. If, for example, the equalization at low frequencies is 30 dB, you know that the system is inadequate at low frequencies and needs improvement.

The tutorial describes a common fatal mistake made in countless churches: The Sound Man is placed in a location remote from the focus of the speakers. As a consequence, it sounds muddy (no high frequencies) and soft (low levels), and he compensates by turning up the overall volume and the high frequencies. Listeners down where the action is—totally oblivious to these issues—are shocked that anyone would mix that way. WinSound provides two clues that can indicate this situation: One, if the maximum level in the audience is greater than 4 dB above the level set for the Sound Man; and, two, if the equalization is greater than 3 dB for high frequencies (Look for these clues at several frequencies; sometimes phase interaction or extremely odd speaker directivities may produce these conditions at one frequency).

The Sound Man's height is also set from the popup menu. Be sure to remember to set it relative to the main floor if he is located in a balcony. The Sound Man can be moved by dragging with the mouse if movement is allowed, as indicated by the menu item View | Allow Movement. It will have a check mark next to it if movement is allowed. The state is toggled each time it is selected.

The Speakers

Arriving at a satisfactory design involves a lot of experimentation with the speakers. WinSound offers some convenient ways to work with the speakers without having to calculate coordinates and enter them in the *Place Speakers* dialog. The goal is to obtain a smooth coverage (the variation in level is minimized). An area may receive too much sound for a variety of reasons: The speaker may be too close to the locations, or the relative volume may be too high. Or, perhaps, it is aimed too exclusively at that location. By the same token, an area may have inadequate coverage because it is far away from the speaker, or the speaker is aimed away from it, or its relative volume is too low.

The speaker can be moved in the *x-y* plane by dragging with the mouse (if movement is allowed, as indicated by the menu item View | Allow Movement). If the speaker has a twin, the twin will be moved accordingly (see the section on the *Place Speakers* dialog for a full explanation of twins). The height can be changed by accessing the speaker's popup menu. Also, the speaker can be re-aimed by dragging its aim point. The height of the aim point is not affected; it can be set by accessing the aim point's popup menu. Note: if you intend that the aim point is a location in the audience, and the audience area is sloped, you will have to set the height after moving the aim point. The height can be inferred by placing an analyzer on the location and reading the title bar of the Analyzer's response window (see below).

The speaker's relative volume can also be set by accessing either the speaker's or its aim point's popup menu. Remember that this level is only relative to other speakers. Thus, if this is

the only speaker in the model (together with its twin if applicable), the relative level will have no effect on the results.

Finally, the electrical power delivered to the speaker can be viewed by choosing view power from either the speaker's or its aim point's popup menu. The power in watts is plotted against frequency. This is where you must learn to use the "eyeball average." The power curve may have large spikes at one or two frequencies due to phase interactions or severe directivity effects, and a real sound operator will rarely tune the system with this much precision. But to assess the true power required, try to blur your eyes and estimate where most frequencies are on the curve. However, don't take this number to the audio store and ask for speakers with this power rating. Even if your analysis is based on the loudest expected conditions, this number represents a time average of the power content of your application. The peaks (kick drum shots, bass slaps, etc…) will likely be much higher. A good rule of thumb is to allow at least 6 dB of "headroom" for these peaks, and in the world of electrical power, 6 dB represents a factor of four. Thus, if your "eyeball" average is 50 watts, your speakers should be rated to handle peaks of 200 watts or more.

The Analyzer

While the sound map displays results for all locations but only one frequency, the Analyzer allows you to view the results for all frequencies but at only one location. Multiple Analyzers can be created allowing comparison between various locations. The Analyzer is created by popping up at the location at which you want to view the Analyzer data and choosing open analyzer. The title bar of the Analyzer's window displays the *x-y-z* coordinates of its associated location.

The title is taken from a common tool in audio engineering. A microphone is connected to the analyzer which sends pink noise (pink noise contains all frequencies at equal energy) to the speaker, and the analyzer displays the frequency response received by the microphone. This allows the technician to adjust the equalizer until the response is flat.

In WinSound, the Analyzer creates a graph window displaying the direct sound pressure level at its associated location over all frequencies. The ideal graph is a flat line at the Sound Man's desired level. If it is a flat line but the level is 10 dB higher, the quality of the sound will be good, but it will likely be too loud (as discussed above). Conversely, the average level may be the same, but the high frequencies may be several dB lower. The sound will probably be muddy—music will lack clarity and brilliance, and voice will lack intelligibility. Perhaps speaker directivity causes the mid frequencies to be several dB higher than the rest. The sound will probably have a piercing, glaring quality to it. Maybe phase interaction causes a jump from 200 to 500 Hz; sounds will be "thick". If you have the opportunity, a good way to get a feel for various frequency responses is to play with the sweepable mid EQ control on the mixer (most intermediate boards have one). Turn the boost all the way up while playing something through the channel, and sweep the frequency control around and listen to the effects.

One handy feature of the Analyzer is its ability to draw the frequency response dynamically while it is dragged around the sound map. Position the main window and the response window so that you can see them both. Click on the Analyzer and drag it around (holding the mouse button

down). Watch the response window to get a quick overview of frequency response trends in different audience areas.

Further Reading

Davis, Gary and Ralph Jones. Sound Reinforcement Handbook. 2d ed., Milwaukee, WI: Hal Leonard, 1989.

Davis, Don and Carolyn. Sound System Engineering. Indianapolis, IA: Howard W. Sams & Co., 1975.