Proceedings of the Institute of Acoustics

THE GENERATION AND ADJUSTMENT OF AN AMBISONIC SOUND FIELD USING A PC

Author: Robin Cross.
Sound and Vibration Unit, British Telecom Research Laboratories.

INTRODUCTION

This project was carried out specifically to enable the measurement of a telephone parameter called the Listener Sidetone Loudness Rating (LSTR), ie the amount of room noise arriving at the telephone users ear via the telephone sidetone path. This quantity has an input into the way the user perceives the quality of the telephone / connection. However, this paper will deal with the measurement system rather than the telephony requirements.

The software module is written in the C programming language and enables a P.C. to generate an ambisonic noise soundfield in a room or enclosure. Concurrently, a software non-real-time One third octave analyser aquires and measures the sound pressure via a calibrated microphone. Additionally, the soundfield can be adjusted to within +/- 0.2 dB of a look-up target in third octaves. Because the noise data is held in text files. It is a relatively simple matter to update the type of noise used, eg an artificial room noise source, reflecting the frequency spectrum and dynamics of real room noise or an ambisonic recording of real room noise. The software is compiled using the Turbo C compiler.

The author assumes that the reader is familiar with ambisonic encoding and replay techniques and programming in C.

STRATEGY

An earlier feasibility experiment used ambisonic recordings of noise in one third octave bands. The noise bands were used sequentially to enable the measurement to be made without the need for an expensive one third octave analyser. It was envisaged that the final solution would use full bandwidth noise and employ fourier techniques to adjust the sound field.

The decision was taken to incorporate all of the measurement technique into the software, thus enhancing the system flexibility and keeping the hardware cost and fault liability down. Indeed, the hardware consists of an ambisonic decoder / loudspeaker amplifier module and a commercially available four channel 12 bit D to A converter for the PC. A 16 bit A to D converter is also required but is already fitted to the PC.

Future telephones may have 'smart' circuitry to adjust their sending / receiving gain in noisy environments. This would require the generated room noise to have the appropriate dynamic and frequency characteristics to fully exercise the telephone. Because the ambisonic noise files are stored on disc, it is a simple matter to change the characteristics of the signal to suite the requirements. We have an artificial voice signal available, why not an artificial room noise signal?

SOURCE RECORDING

The Sound and vibration unit at BTRL has a Reverberation chamber. The reverberant field has a net energy flow or intensity of zero at any point. If this field is recorded and replayed accurately then the target replay room can exhibit these features at a point. The ambisonic 'B' format records and replays three orthogonal velocity signals and a pressure signal. This is known as a periphonic recording system with a spin of 1, and of weight 1 using spherical harmonic nomenclature.

Figure 1 shows the production of the source recording.

Circuit diagram for recording wideband noise stimuli in Ambisonic 'B' format with full periphony.

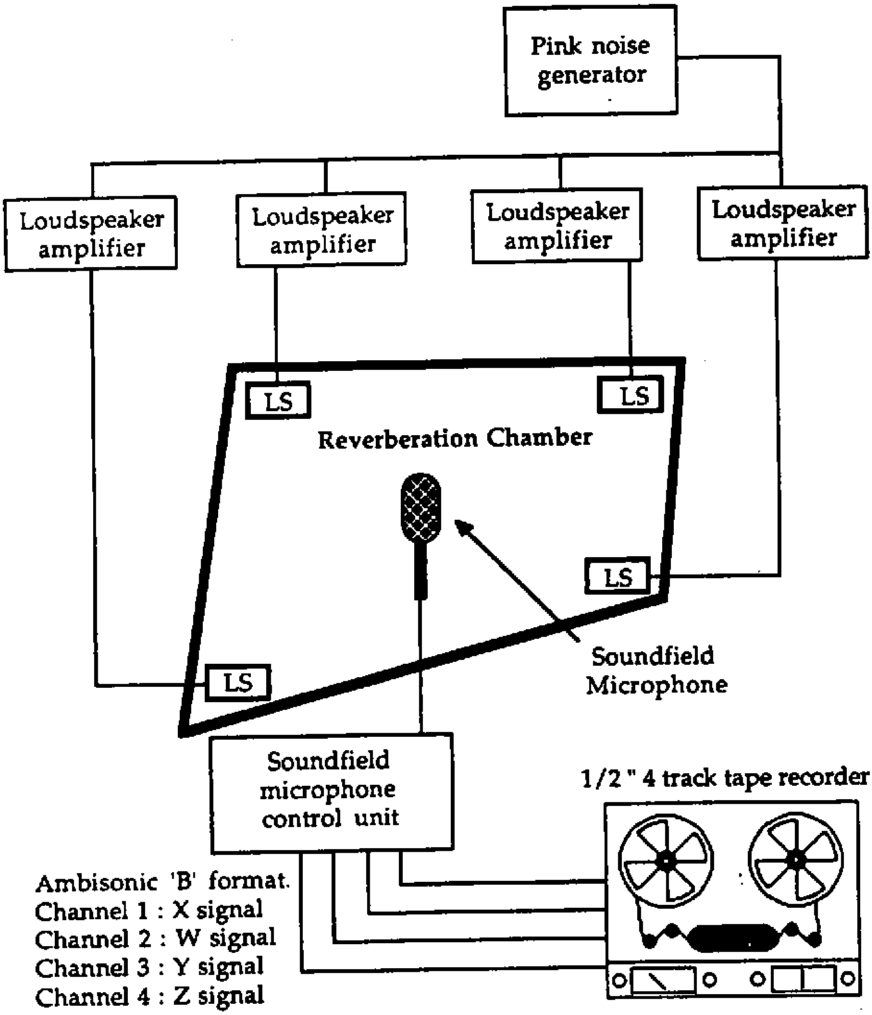
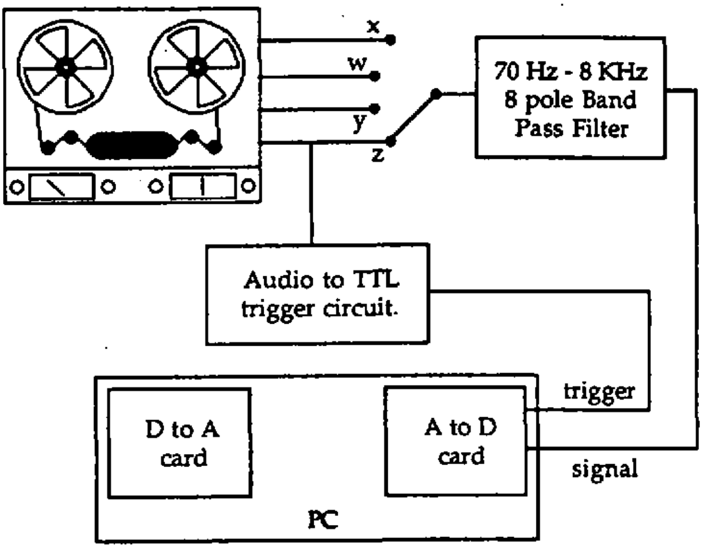


Figure 1

Decoding the ambisonic recording requires the four channels to be synchronised and this placed timing constraints on the digitising process. Figure 2 shows the principle.

THE GENERATION AND ADJUSTMENT OF AN AMBISONIC SOUND FIELD USING A PC

Diagram to show digitising of ambisonic recording.



Digitised samples stored in ascii files;

Figure 2.

The tape consists of 10 minutes of ambisonic noise. It was prepared for digitising by erasing 20 seconds of channel 4 only (the z channel). Thus when the tape is replayed, the trigger circuit operates at the same point in time, thus maintaining synchronisation. The z channel is used for triggering as it contains the ambisonic vertical information (This is the minimum disturbance to the tape recording as the z channel is not used in a horizontal surround sound ambisonic system.

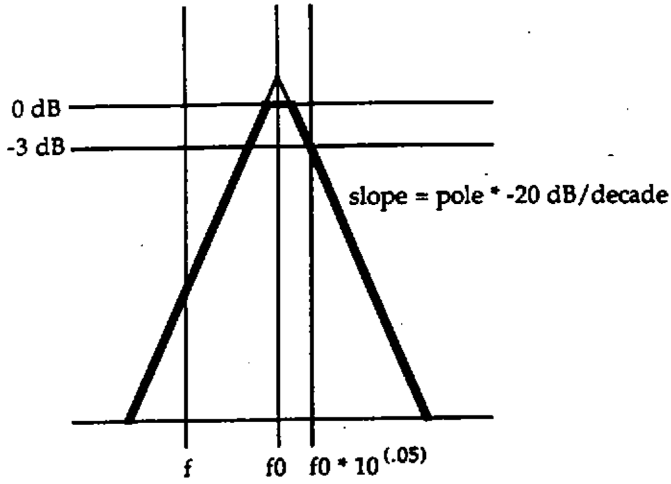
SOFTWARE

To implement the system two C functions cal_lstr() and meas_lstr() were required to integrate into the software. The first is used to calibrate the soundfield and the second to perform the measurement. The function third_oct() and the file correction process will be described in detail. The support functions are listed by name.

Function third_oct.

All time waveform datafiles are clocked at 16 kHz and consist of 4096 16 bit signed samples. The A to D puts its output into a file called adop.txt. Memory is allocated on the heap as 8192 floating point locations pointed to by *data. The support function get_sample places the file data into the memory at the even (real) locations, the odd (imaginary) locations are set to zero. This is a standard form for the storage of real and imaginary data. The data is then Fourier transformed into the frequency domain in place, ie the data is returned to the same memory location thus saving memory space.

The third octave levels are derived by applying third octave filter windows to the frequency data and calculating the RMS value of the data contained within the window. The filter window is constructed from a simple linear equation with the -3 dB point at $10^{(0.05)}$ spacing away from the centre frequency and all values greater than one are set to one. It is possible to adjust the slope (number of poles) of the window but in this case has been set to 10 to accord with the ISO one third octave filter standard. Two frequency arrays are used by the software, display_freq [ ] contains the ISO 10 series preferred numbers to represent the third octaves. For calculation purposes an array called freq [ ] contains frequencies spaced by $10^{(0.1)}$ or 1.258925. This gives accurate centre and band edge frequencies. Figure 3 is the geometric representation of the window.



window gain = -20 * pole (abs (log 10 (f/f0))) + (pole -3)

if window gain > 1 ; window gain = 1

figure 3.

In order to reduce computation time, each window is scanned from a low point to a high point which represents an equal distance either side of the centre frequency, in this case a value of $f/f0 = 1.5$ is used, equivalent to 28.2 dB down.

File correction process.

The target one third octave sound pressure levels are held in an array. Once a third octave analysis of the actual sound pressure has been made a correction factor array is produced. In this case the ambisonic data files are sequentially transformed into the frequency domain. Either the original or equalised files can be chosen. In the frequency domain the data points corresponding to the one third octave bands are modified by the appropriate correction factor.

Support functions.

Name: load_ambi(select, gain)

THE GENERATION AND ADJUSTMENT OF AN AMBISONIC SOUND FIELD USING A PC

The modified data points are then mirrored into the top half of the transform, inverting the imaginary component to produce the complex conjugate. This structure has the property of producing real only data when the inverse transform is subsequently applied. The 8192 data points are compacted into 4096 by discarding the redundant imaginary data and storing as equalised data files. A check is made at this point to ensure that the integer ceiling of the new file is within bounds. These new files are then clocked out of the D to A's to produce the new sound field.

Use: To load either original or equalised ambisonic data files into the array noiseout[][]. 16 bit data is split into 12 bit byte and nibble for subsequent clocking out. (If this were done at the clocking out stage the required clock rate would not be achieved due to excessive processing time.

Name **ambi_out(presig)**
Use Outputs noise to four channel D to A.

Name **get_sample(data, j)**
Use Get a time sample from file and load into memory.

Name **prnt(data)**
Use Used during development to print Fourier data to screen.
Parameter. data: pointer to start of memory location.

Name **fourier(data, nn, isign)**
Use Performs fourier or inverse fourier transform.

Name **filter(j, f, pole)**
Use To window the frequency domain data to give performance equivalent to a true digital filter operating in the time domain. (modulus only required).

Name **collect_and_store_data( )**
Use To get the data generated by the A to D (held in its own memory) and store in file 0. (Is actually 7039 samples of which the first 4096 are used.)

Name **calc_lstr(level)**
Use To calculate the loss from the array sound_pressure [0][i] (sound field) and sound_pressure[1][i] (output from artificial ear). Calculate and return the value of LSTR.

Name **open_read_file(j)**
Use Sets file handle to open datafile for reading.

Name **open_write_file(j)**
Use Sets file handle to open datafile for writing.

Name **band_limit_set (band_edge)**
Use Calculate and store datapoints between which the third octave is defined.

HARDWARE

In keeping with the strategy the hardware has been kept as simple as possible. Instead of decoding from four to eight channels it is possible to decode from four to six, and use the loudspeaker interconnection to provide the final decoding, thus saving two power amplifiers. Figure 4 shows the P.C. to decoder, via 5 pole, 8 kHz reconstruction filters. The filters are proprietary switched capacitor I.C.'s with a common 800 kHz clock. Figure 5 shows the amplifier and loudspeaker interconnection arrangement. These are activated by a control line.

THE GENERATION AND ADJUSTMENT OF AN AMBISONIC SOUND FIELD USING A PC

Block Diagram of Ambisonic Decoder.

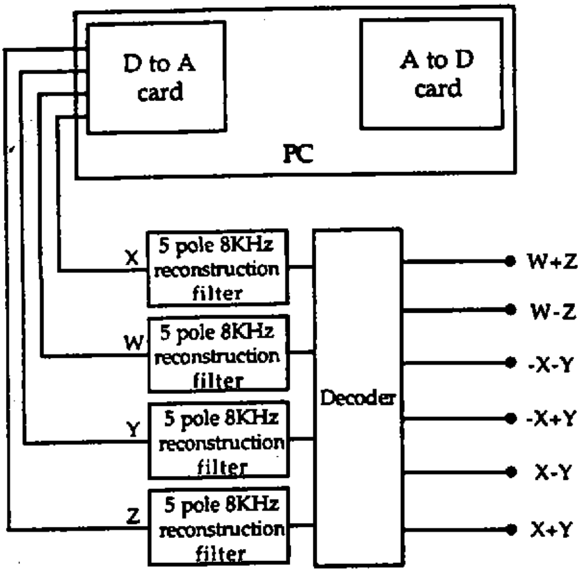


Figure 4.

Block diagram of Amplifier / Loudspeaker interconnection.

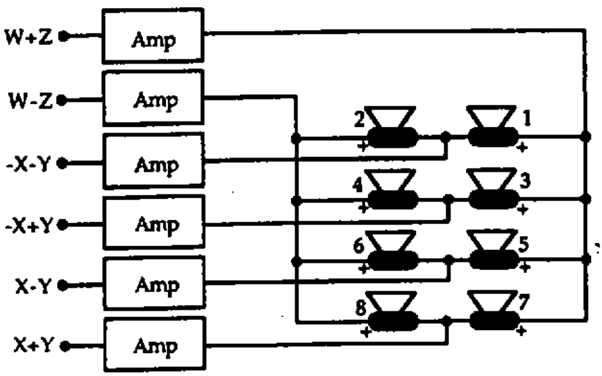


Figure 5.

Note that the return paths of the six, six watt power amplifiers are not used. The W signal is omitted as it is added to all signal combinations. The loudspeaker signals are shown in the following Karnaugh diagram, figure 6.

THE GENERATION AND ADJUSTMENT OF AN AMBISONIC SOUND FIELD USING A PC

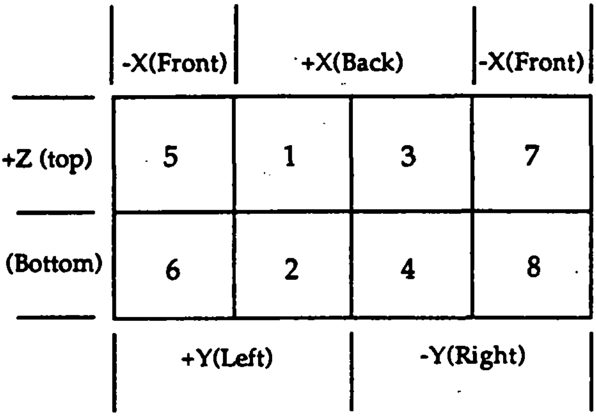


Figure 6.

Each loudspeaker is a proprietary unit consisting of a 70 mm diameter drive unit housed in a sealed cylindrical plastic case measuring 95 mm diameter by 125 mm long. The loudspeakers are mounted in the corners of the test cabinet and supported by an aluminium angle frame. Long term sensitivity stability is rendered un-necessary by the calibration process.

PERFORMANCE

Third octave analyser accuracy.

By comparing soundfields measured with a Bruel & Kjær 2131 analyser and the software analyser the difference was found to be +/- 0.2 dB over the range 100 Hz to 8 kHz.

Speed of adaption.

The un-equalised ambisonic data files give a soundfield deviation of 22.4 dB over the range 100 Hz to 8 kHz at a level of 75 dBA. From then on the soundfield deviation is 2.32, 0.58, 0.34, 0.32, 0.28 dB from the target. The maximum soundfield deviation has been set to 1 dB so that the soundfield will be +/- 0.7 dB including the accuracy of the analyser. The specification calls for +/- 1dB.

Overall accuracy.

LSTR measurements are recommended to be made at 60 dBA. However, if the soundfield is calibrated at this level the acoustic noise level in the blue box begins to affect the speed of adaption. For this reason, the soundfield is calibrated at a level of Hoth 75 dBA which is a reasonable compromise. Indeed, if the soundfield is calibrated at 80 dBA the soundfield deviation reduces to within 0.04 dB of the target, showing the effect of improved signal to noise.

CONCLUSION

Two functions have been produced to run on a PC to integrate into a Digital Telephone Tester. Associated hardware has been designed and built to enable LSTR measurements to be made on telephones (analogue and digital). Because the target soundfield is stored as a table, the system can produce any arbitrary noise soundfield. Examples are Hoth, aircraft, car and traffic noise or a flat soundfield.