Proceedings of the Institute of Acoustics

CONTROL SOFTWARE FOR A PROGRAMMABLE SOUNDFIELD CONTROLLER

D.G Malham (1), John Clarke (2)

(1) Music Technology Group,Department of Music,University of York,Heslington,York YO1 5DD. E-Mail:- DGM2@UK.AC.YORK.VAXA
(2)BP Oil UK Ltd., Breakspear Way, Hemel Hempstead, Herts., HP2 4UL

1. INTRODUCTION

This paper describes software which was designed to control various forms of Ambisonic sound image manipulation in real time, using a mouse pointer control interface. The software runs on an Atari ST computer and drives external hardware to control the actual audio signals. Non-Ambisonic sources can be processed into Ambisonic B-format ones and B-format signals can be further modified. The second author wrote the original version of the software as part of his Music Technology MSc project, whilst the first author developed the hardware and produced the current version of the software.

2.HARDWARE - THE DIGITALLY PROGRAMMABLE SOUNDFIELD CONTROLLER

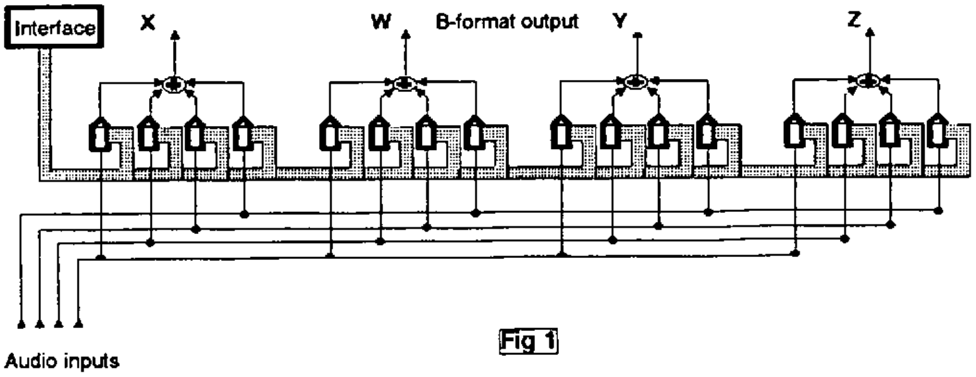


Fig 1

In the early 1980's the first author designed an experimental tool for investigating the production and manipulation of Ambisonically encoded sounds. Called the 'digitally programmable soundfield controller' [1], it was essentially a box with four audio inputs and four audio outputs. Each output could have routed to it a mix of the four inputs, with any desired combination of gains and polarities. The gains were set by sending each input to the reference input of a set of four, 14 bit multiplying DACS and the polarities were set with electronically controlled sign bit amplifiers (fig 1). Each input and its four associated sign amps and DACS were contained on one card. Each of the four DAC outputs was routed to a separate audio bus and thence to a mixing card where the corresponding outputs of each input card were mixed together. The levels out of the multiplying DACS and their polarities were controlled from a separate computer via an interface card, according to the following equation;
OUTPUT = INPUT * P * A/16383
where the gain coefficient $A > 0 \leq 16383$ and P is the polarity.

3. SOFTWARE

3.1 Project Aims

Originally, the controlling computer was an 8 bit CP/M machine and the software was written in Pascal. Whilst this was adequate for experimental determination of minimum word lengths for controlling coefficients and similar tasks, its slowness and poor user interface rendered it useless for studio use. In the summer of 1990, the second author, who was then an MSc student studying Music Technology at the University of York, chose as the subject of his final project the development of a performance interface for Ambisonic sound control. The intention was to develop an electromagnetic position sensor which a performer could hold and move around in space, with the sounds tracking its position - a sort of 3D mouse.

3.2 Mouse Control Interface

As part of the investigation into how this might work in practice, a new piece of control software for the Soundfield controller was written, running on a much faster 16 bit computer, the Atari ST. Using this software it is possible to select and use many different types of Ambisonic sound manipulations. These are controlled with the computers' mouse. Because of the way the controller is constructed, it can be used either to manipulate a full, four channel B format soundfield, or it can place up to four individual sound sources into B format soundfield configuration. Accordingly, the software provides a number of options (fig 2) for controlling either B format signals or non-Ambisonic sources.

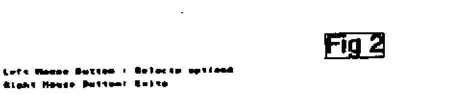


Fig 2

3.2.1 Types of Interaction. Two types of mouse/screen interaction are used in different parts of software. In the main, circles are drawn on the screen to represent horizontal and, where appropriate, vertical planes, manipulation of sound being by pointing at a position within a circle and clicking the left mouse button to activate the control. The other type of control, which is used in some screens, is a set of sliders for setting x, y and z coordinates or in the case of the 'SATELLITE' option, angles. In order to facilitate the description of manipulations of full soundfields, a set of conventions was devised (fig 3 and appendix A). ROTATION is about the vertical (z) axis and is positive for anti clockwise movement, TILT is about the front-back (X) axis and is positive for a tilt to the right, TUMBLE is about the left-right (Y) axis and is positive for a backward tumble.

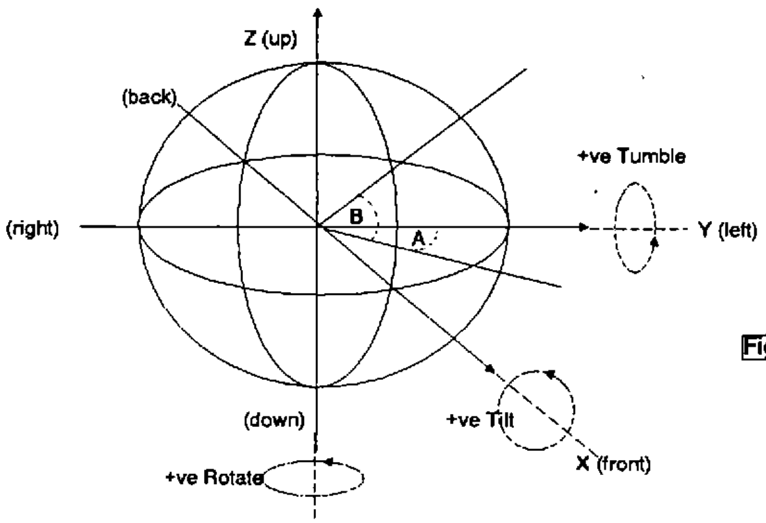
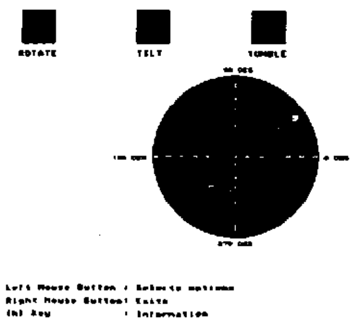


Fig 3

3.2.2 B-format controls.Movement control in the four main B format manipulations (ROTATE/ TILT/TUMBLE/ARBITARY) is implemented in a consistent manner (Fig 4). The user is presented with a circle on the screen in which the zero degree position represents the unmodified condition. The axis about which the soundfield will be rotated is assumed to be in the centre of the circle perpendicular to the screen, so in the case of ROTATE, the circle represents the horizontal plane, in TILT the circle represents a vertical plane through the X axis and in TUMBLE, a vertical plane through the Y axis. In the ARBITARY screen, this axis of rotation is set using X, Y, Z sliders which define a point through which the axis must pass.

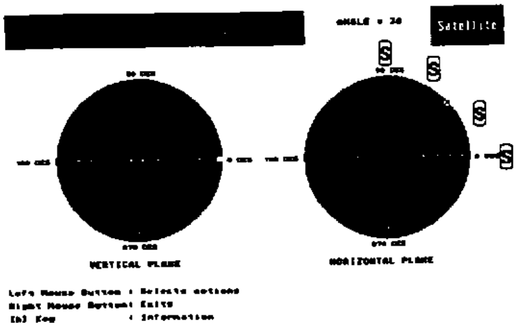


```
Left Mouse Button : Selects options
Right Mouse Button: Exits
Del Key           : Information
```

Fig 4

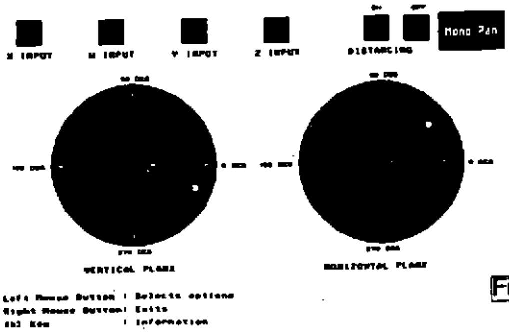
The soundfield manipulations are executed in software using a four by four matrix multiply which generates sixteen coefficients ([2] and Appendix B). These coefficients fed via a parallel interface to the DAC's and sign bit amplifiers in the programmable soundfield controller.

3.2.3 Non-Ambisonic inputs. A similar process is used for use with non ambisonic inputs, except that the user is presented with two circles, a horizontal one which is fixed and a vertical one which is perpendicular to it and aligned with the horizontal position of the sound source (fig 5). The four inputs can be treated as four individual sound sources, independently panable, as two sets of 60 spaced stereo images independently panable or as a quad source, panable as a single unit. In the latter case, the angles separating the four sources are selectable with a slider, the four sources Ⓢ being panned as a single unit with the centre point between the middle two images being the panning point.



Left Mouse Button : Selects options
Right Mouse Button: Exits
[h] Key : Information

Fig 5



Left Mouse Button : Selects options
Right Mouse Button: Exits
[h] Key : Information

Fig 6

In MONO PAN, (Fig 6) the software allows the use of standard Ambisonic pan pot coding in which the overall level of the sound stays roughly constant, even if the source is moved towards the centre point or a newly developed distance compensated version. This was developed by the first author as a result of the observation that interior effects as produced by existing Ambisonic pan pots, whether implemented in analog hardware [3] [4] or software [2], are not consistent with the changes in the B format signals from a soundfield microphone as sound sources approach it. With such conventional devices, the omnidirectional (X) signal increases as the sound source moves towards the centre to compensate for the drop in level of the directional (X, Y, Z) signals. As a result, the image becomes more and more diffuse as the sound is panned towards the centre. In contrast to this, the levels of all four components increase as a real sound source approaches a soundfield microphone until at closest approach, the relevant directional components undergo a rather rapid phase (or more accurately polarity) reversal, whereafter the levels of all components start reducing again as the source moves away on the opposite of the microphone (fig 7). The DISTANCING option, when turned on, emulates this behaviour. The optimisation of the precise law the amplitude curves should follow to match the subjective effects is still under investigation and we hope to report further at a later date. Early results are, however, very promising, with further improvements expected as we move the whole environment onto a full digital platform, such as the Atari Falcon computer, where control of reverberation, HF roll off and early reflections will add considerably to the realism.

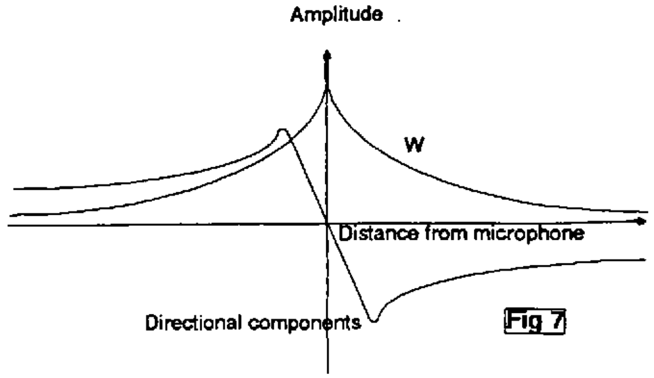


Fig 7

4.CONCLUSIONS

Although it cannot be pretended that a mouse driven control system for Ambisonic manipulations is by any means optimum, until 3D sensor/input systems such as currently used in virtual reality systems become cheaper, it does represent an eminently useable and extremely cost effect option.

5. REFERENCES

[1] MALHAM D.G. "Digitally Programmable Soundfield Controller" STUDIO SOUND, Vol 26 No 2 pp 75, 1984.
[2] MALHAM, D.G. "Computer Control of Ambisonic Soundfields" Preprint No. 2463(H2) presented at the 82nd AES convention 1987 10-13 March, London.
3) GERZON, M.A. "Ambisonics Part two: Studio Techniques" STUDIO SOUND, August 1975. pp24 - 30.
[4] GERZON, M.A. "Panpot and Soundfield Controls" NRDC Ambisonic Technology Report No. 3, August 1975.

APPENDIX A.ENCODING A MONOPHONIC SOUND INTO AMBISONIC B-FORMAT.

Since the decoder designs are predicated on the basis that sounds being positioned in Ambisonic B-format are placed on the surface of or within a notional unit sphere the maximum radius a sound may be placed at is 1. If the sound is moved outside this sphere the directional information will not be decoded correctly and sounds will tend to pull to the nearest speaker. This means that within a coordinate system sound source coordinates must obey the following rule;

(x^2 * y^2 * z^2) <= 1

Initially all transformations will place sounds on the surface of the unit sphere.
If a monophonic signal is to be placed on the surface of a unit sphere, then its coordinates will be, referenced to centre front;

x = Cos A * Cos B
y = Sin A * Cos B
z = Sin B

These coordinates directly relate to the B-format signal levels thus;

X = input signal * Cos A * Cos B
Y = input signal * Sin A * Cos B
Z = input signal * Sin B
W = input signal * 0.707

The 0.707 multiplier on W is there as a result of engineering considerations related to getting a more even distribution of signal levels within the four channels when taking live sound from a Soundfield microphone. A is the anticlockwise angle of rotation from the centre front and B is the angle of elevation from the horizontal plane. These multiplying coefficients, (Cos A * Cos B etc.), will position the monophonic sound anywhere on the surface of the soundfield, producing the B-format encoded output signals.

APPENDIX B. SOUNDFIELD MANIPULATIONS

B.1 Definition of the coordinate system for B-format soundfield manipulations.
If a B-format signal is to be transformed, for example rotated and tilted, then the four channel of the

signal must be scaled by the correct coefficients. The following standard definitions are made about the way the sound will move to a new position are provided in order to keep the equations coherent and minimise the confusion that can all too easily occur.

B1.1 Positive angles of rotation are anti_clockwise or by convention rotation to the left.

B1.2 A rotation is defined as a circular movement about a pre-defined axis, normally taken as the Z-axis, this being the same as an anti-clockwise movement in the horizontal plane.

B1.3 A tilt is defined as a rotation about an axis lying on the horizontal plane, for example the x-axis which is the same as an anti-clockwise movement in the vertical left-right plane.

B1.4 A tumble is defined as a rotation about the Y-axis. This is the same as an anti-clockwise movement in the vertical front-back plane. Note that a tumble is the same as a tilt that has first been rotated by 90 degrees about the Z-axis.
figure 3 shows the graphical representation of this where A = the angle of rotation , B = the angle of elevation .

B.2 ROTATING A POINT ABOUT THE Z-AXIS

If A is the positive angle of rotation and C is the angle between the X-axis and the untransformed position, (x,y), we have;

$x = r*\cos C$, $y = r*\sin C$
$x' = r*\cos (A+C)$, $y' = r*\sin (A+C)$

simplifying;

$x' = r*\cos C * \cos A - r*\sin C * \sin A$
$y' = r*\cos C * \sin A + r*\sin C * \cos A$

and substituting for x and y

$x' = x * \cos A - y * \sin A$ $y' = x * \sin A + y * \cos A$

w and z remain unchanged since the rotation is about the Z-axis, for points on the surface of the unit sphere $w = 0.707$. If the same procedure is applied to to the tilt and rotate equations this gives the following;

TILT.

$x' = x$
$w' = 1.0$
$y' = y * \cos B - z * \sin B$
$z' = y * \sin B + z * \cos B$

TUMBLE.

$x' = x * \cos B - z * \sin B$
$w' = 1.0$
$y' = y$
$z' = x * \sin B + z * \cos B$

These equations can now be combined to perform transformations such as rotate-tilt which give an angular rotation of the whole input soundfield to the left by an angle of A from the centre front. Then it tilts the B-format soundfield by an angle B from the horizontal.

ROTATE-TILT.

$x' = x * \cos A - y * \sin A$
$w' = 1.0$
$y' = x * \sin A * \cos B + y * \cos A * \cos B - z * \sin B$
$z' = x * \sin A * \sin B + y * \cos A * \sin B + z * \cos B$

Any combination of the many possible soundfield manipulations can be realised by using one matrix of scaling coefficients thus;

$X' = K1.X + K2.W + K3.Y + K4.Z$ $W' = K5.X + K6.W + K7.Y + K8.Z$
$Y' = K9.X + K10.W + K11.Y + K12.Z$
$Z' = K13.X + K14.W + K15.Y + K16.Z$

where K1 - K16 are the scaling coefficients formed by the soundfield manipulations applied to the incoming signals X, W, Y, and Z. X', W', Y' and Z' are the resultant B-format output signals.

ACKNOWLEDGEMENTS

The authors would like to thank Tim Anderson of the Music Technology Group at York for his work in "modularising" the original code and adding MIDI functions to it.