FINAL WRITTEN REVIEW: DEVELOPMENT PROPOSAL FOR VARIABLE TWO-WAY ROTARY LOUDSPEAKER DIGITAL AUDIO EFFECT

Matt Croteau 310303923

Digital Audio Systems, DESC9115, Semester 1 2012
Graduate Program in Audio and Acoustics
Faculty of Architecture, Design and Planning, The University of Sydney

1. DESCRIPTION

A commonly desired effect for digital audio processing is to emulate the electroacoustic characteristics of a rotary loudspeaker (such as Leslie or Motion Sound models). Although there are various implementations available, many of these are locked into a larger, more comprehensive piece of software (often including virtual instruments and other periphery processors). I am proposing a software implementation that will be user-friendly, simple, light on processing power, realistic but with appropriate level of user control, and easily implemented as a single standalone process.

The process involved requires an appropriate understanding of the functionality, sound, musical context, and acoustic properties of such a device. With these things in mind I have developed a strategy for designing and developing an algorithm to create such a piece of software.

Although the overall characteristic of a rotary loudspeaker is complex, the controls needed to simulate one are relatively simple. It is essential that the software has not only an attractive and believable sound, but is also an easy user-interface that remains true to the original. To accomplish this a degree of complex processing must be implemented behind-the-scenes with a simple user interface to make the software appear simple and understandable.

.

2. SPECIFICATION

The rotary loudspeaker I am aiming to emulate with my software consists of two rotating loudspeaker drivers. Figure 1 details the basic construction of a typical rotary loudspeaker.

As is dictated by common practise in music production, these loudspeakers typically have a mono input and produce a stereo output. The horn has adjustable speeds and the drum normally has one, slower speed. This creates a wide, rich, modulating sound. As both the horns and the drum rotate in a circle, the modulation is regular and sinusoidal in both amplitude and pitch.

All of these considerations have been made in designing the DSP for this software.

2.1. Summary of Electroacoustic Characteristics

The main sonic characteristics of this loudspeaker type that need to be included in the effect are summarised as follows:

- Sinusoidal vibrato/pitch modulation (Doppler effect)
- Sinusoidal amplitude modulation
- Mono signal input
- Stereo signal output with inverse modulation between channels (opposite to each other)
- Crosstalk simulation between channels for realistic stereo imaging
- Doppler/frequency modulation for the drum portion of the signal but without amplitude modulation (due to its increased power and low frequency content)

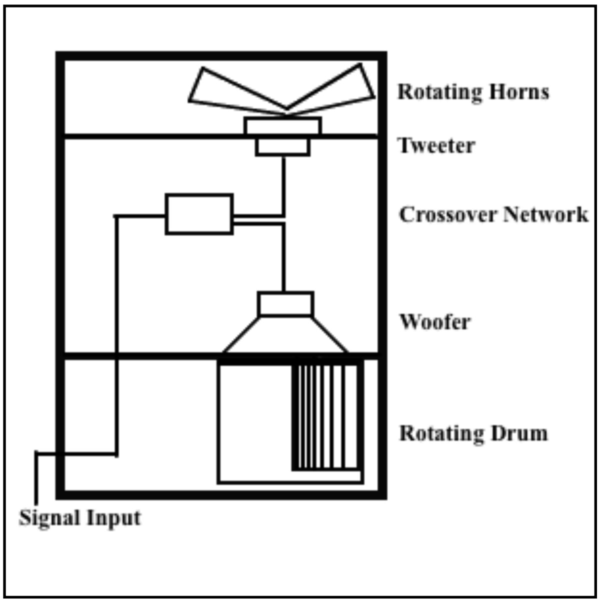

Figure 1. *The analogue two-driver rotary loudspeaker to be simulated with ad DSP algorithm.*

3. IMPLEMENTATION

The first step in understanding the process needed is to create a vibrato algorithm to simulate the Doppler effect of the rotating speaker. This must be processed as two separate signals (described in this proposal as A and B). The signals must also be amplitude modulated to create the alternating level changes between the channels. Both modulations must be done with sinusoidal oscillation inverse to one another to recreate the circular rotating effect in stereo as shown in Figure 2.

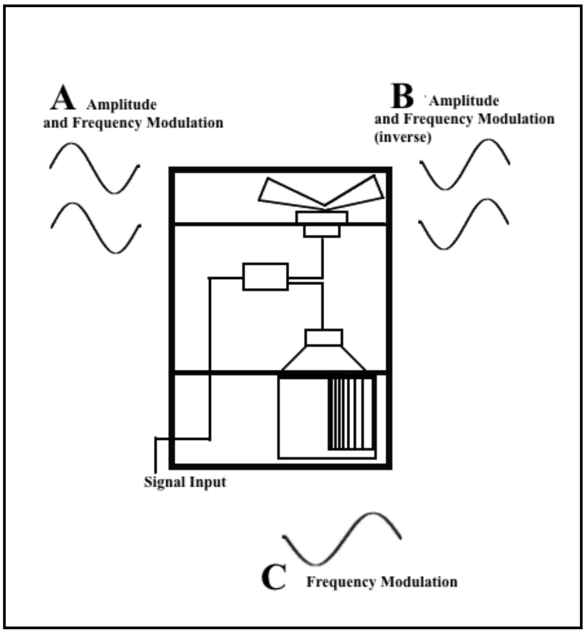

Figure 2. *The three main signal components of the rotary loudspeaker effect..*

To accomplish the desired effect, the following process must be followed. A sine for frequency modulation (vibrato, or Doppler effect) must be created at an appropriate frequency (that oscillates at the typical rate of a rotary loudspeaker horn). The A and B signals must have their sample rates multiplied by this sine (and inverse sine, respectively). This will create the inversely oscillating pitches. The next step is to multiply the amplitude of these signals by the same sine and inverse sine (so the signals have matching amplitude and pitch modulation frequency and phase.

The C signal needs to include slower modulation to interact with the faster modulation of the horns. This can be done by decreasing the modulation frequency for that component of the signal. In my initial development run I have set this to always be $1/3^{rd}$ the speed of the initial sine modulations for signals A and B. This will always scale the C signal to be slower no matter how the initial frequency is changed. This is a parameter that can easily be given a separate control by following the same processing principles.

To simulate crosstalk between the channels and create a believable stereo image, it will be necessary to combine the A and B signals with one another to a small percent (70% used in this example). The C signal will finally be introduced by a certain percent (40% in my initial run) to the main stereo outputs. This also can be given a parameter control for user flexibility. Figure 3 below outlines the basic signal flow in accomplishing the process.

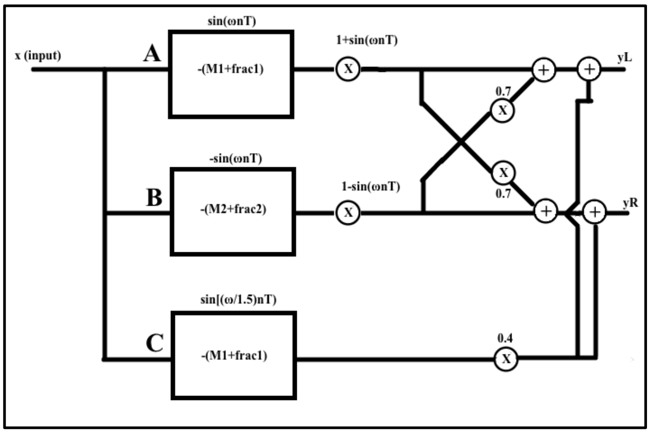

Figure 3. *Signal flow diagram of the complete rotary loudspeaker algorithm, beginning with a detail of frequency modulation and finally the processing of the stereo signals.*

In Figure 3 above, the input (x) is modulated by a sinusoidal signal (m) resulting in an output (y) of the original signal x with the frequency characteristics of signal m. Greek letter omega (ω) is the frequency of the sine, which is proportionally slower for signal C. In the second portion of the diagram this process is expressed as M1+frac1 and M2+frac2, to derive A and B, and C with their respective modulator waves indicated above. This delay is based on sinusoidal and inverse sinusoidal frequency at a set width and depth (expressed as n and T).

The A and B signals are then multiplied by the sine/inverse sine to create the amplitude modulation required. This is not required of signal C. Finally, these signals are added to each other at a ratio of 0.7/1 for the horn signals and 0.4/1 for the drum signal to create the final output.

This process will effectively simulate the acoustic effects of a rotary loudspeaker with simple user controls and the opportunity to expand the functionality quite easily.

4. EVALUATION

This process will meet all of the goals outlined above to simulate a rotary loudspeaker as described, with variable pitch modulation, frequency of modulation, and amplitude modulation controls. The desire to create detailed filtered signals (i.e. the crossover network allowing high frequencies for the horns and low frequencies for the drum) could be implemented based on user evaluations.

4.1. Evaluation Strategy

In order to test the success of the process it would be desirable to have user feedback on not only the control parameters but also the sonic quality of the software. This should hopefully determine if the processor is sufficient in its design, not just technically but also aesthetically.

I propose two approaches of evaluation:

- A qualitative listening assessment allowing users to provide a range of values according to how satisfied they are with the sound produced by the processor.
- A usability assessment in which users manipulate the parameters on their own and provide feedback on the functionality of the processor.

4.1.1. Listening Assessment

In this evaluation, listeners shall be played various audio tracks. There will be 5 sets of 5 tracks. Each set will start with an example of unaffected audio and then play the same audio with rotary loudspeaker effects placed on them. Only one of these tracks per set will be the process proposed, the others will be produced from commercially available processors.

The audio tracks should vary in content but be relevant to the target audience of the processor (guitar, vocal, mixed music, etc would be appropriate). The listeners should also vary in background and experience, from trained musicians and audio engineers to people with little background in audio production or music.

The subjects will then be asked to evaluate, on an aesthetic basis, their preferences on a scale of 1 to 4. This should provide valuable insight by comparison, and allow the development team to make adjustments to better conform or avoid certain sonic characteristics.

4.1.2. Usability Assessment

In this evaluation, users will be asked to control the parameters of the processor themselves. The initial user controls will be speed and amount of "drum", or signal component C, to add to the output. A rudimentary assessment can be made by asking the users to rate the product on:

- Ease of use (1-10)
- Sufficient user control options (1-10)
- Overall understanding of the controls and what they do (1-10).

A short questionnaire should be provided with a list of additional parameters that can be included in the design of the GUI. These parameters can be as follows:

- Pitch Modulation Amount
- Amplitude Modulation Amount
- Crossover Frequency (to be developed)
- Initial Delay of Modulation
- Discreet speed control of the C signal component

They will be asked to prioritise these parameters on a scale of 1 to 5 based on their desire to control them. Again the users should be chosen to have varying backgrounds and familiarity with audio processing software. The feedback provided here will allow us to implement an appropriate level of control without making the GUI overly complicated.

Once an initial revision has been made to the GUI, another test should be run following the same process to determine that indeed the controls provided reflect the desires of the end-user.

5. CONCLUSION

I believe it would be not only possible, but worthwhile to pursue the development of this rotary loudspeaker emulator. With proper development and testing, this very popular and sought-after effect can be implemented in such a way as to be attractive and not intimidating to users. With efficient development of the code and easily expandable parameters, a highly competitive product can be made available without the complications or expense of bundling it with an expensive or otherwise complicated software package.