Surround Sound: Passive Encoding and Decoding

Version 2.9: 2002-09-05

Swaroop Appadwedula
Douglas Jones
Michael Kramer
Dima Moussa
Brian Wade
Jake Janevitz
Daniel Sachs
Matthew Berry
Mark Haun
Produced by The Connexions Project

Abstract

You will implement a Dolby Pro Logic encoder in MATLAB and a passive surround sound decoder on DSP hardware.

Surround Sound: Passive Encoding and Decoding

1 Introduction

To begin understanding how to decode the Dolby Pro Logic Surround Sound standard, you will implement a Pro Logic encoder and a passive surround sound decoder. This decoder operates on many of the same principles as the more sophisticated commercial systems. Significantly more technical information regarding Dolby Pro Logic can be found at Gundry (pg **??**).

2 Encoder

You will create a MATLAB implementation of the passive encoder given by the block diagram in Figure 1.

The encoder block diagram shows four input signals: Left, Center, Right, and Surround. These are audio signals created by a sound designer during movie production that are intended to play back from speakers positioned at the left side, at the front-center, at the right side, and at the rear of a home theater. The system in the block diagram encodes these four channels of audio on two output channels, Lt and Rt, in such a way that an appropriately

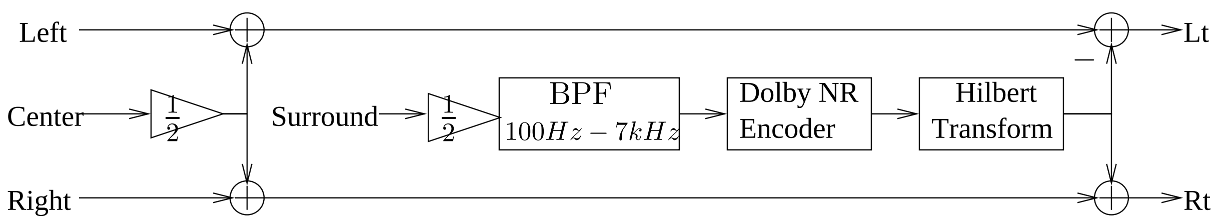

Figure 1: Dolby Pro Logic Encoder

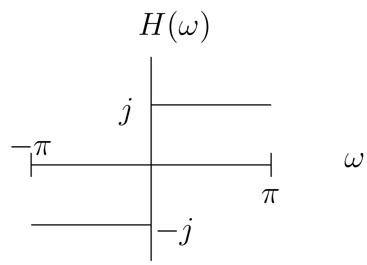

Figure 2: Hilbert transform transfer function

designed decoder can approximately recover the original four channels. Additionally, to accommodate those who do not use a surround sound receiver, the encoder outputs are listenable when played back on a stereo (two-channel) system, even retaining the correct left-right balance.

The basic components of the encoder are multipliers, adders, a Hilbert transform, a band-pass filter, and a Dolby Noise Reduction encoder. If you wish to implement Dolby Noise Reduction, refer to Dressler (pg **??**). The other components are discussed below.

The transfer function of the Hilbert Transform is shown in Figure 2. The Hilbert Transform is an ideal (unrealizable) all-pass filter with a phase shift of -90°. Observe that a cosine input becomes a sine and a sine input becomes a negative cosine. In MATLAB, generate a cosine and sine signal of some frequency and use the `hilbert` function to perform on each signal an approximation to the Hilbert Transform. (Why is the Hilbert Transform unrealizable?) The imaginary part of the Hilbert Transform output (i.e., `imag(hilbert(signal))`) will be the -90° phase-shifted version of the original signal. Plot each signal to confirm your expectations.

For the band-pass filter, design a second-order Butterworth filter using the `butter` function in MATLAB.

2.1 Generating a surround signal

Create four channels of audio to encode as a Pro Logic Surround Signal. Use simple mixing techniques to generate the four channels. For example, use a voice signal for the center channel and fade a roaming sound such as a helicopter from left to right and front to back.

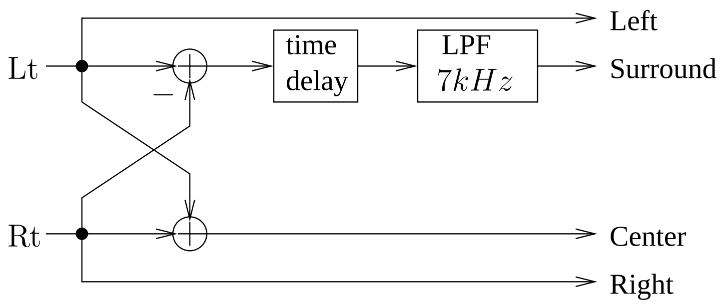

Figure 3: Dolby Pro Logic Passive Decoder

In MATLAB, use the `wavread` and `auread` functions to read `.wav` and `.au` audio files which can be found on the Internet.

3 Decoder

Implement the passive decoder shown in Figure 3 on the DSP. Use an appropriate time delay based on the distance between the front and back speakers and the speed of sound.

Is there significant crosstalk between the front and surround speakers? Do you get good separation between left and right speakers? Can you explain how the decoder recovers approximations to the original four channels?

4 Extensions

Differences in power levels between channels are used to enhance the directional effect in what is called "active decoding." One way to find the power level in a signal is to square it and pass the squared signal through a very narrow-band low-pass filter ($f \leq 80$Hz). How is the low-frequency content of the squared signal related to the power of the original signal? Remember that squaring a signal in the time domain is equivalent to convolving the signal with itself in the frequency domain.

To implement a very narrow-band low-pass filter, you may consider using the Chamberlin filter topology, described in Surround Sound: Chamberlin Filters[1].

5 References

- - K. Gundry, "An Introduction to Noise Reduction", http://www.dolby.com/ken/.
- - R. Dressler, "Dolby Prologic Surround Decoder Principles of Operation,"http://www.dolby.com/tech/whtppr.

[1]http://cnx.rice.edu/content/m10479/latest/