Stereo Radio Signals

STEPHEN ANSEL CAMPBELL
sac170630@utdallas.edu
EE 4361 — INTRO TO DIGITAL SIGNAL PROCESSING

***Abstract*—Methods to simultaneously transmit signal (mono) and dual (stereo) channel audio signals are valuable and important applications of signal processing. This report details one such method which utilizes fast analog to digital conversion with a digital signal processor. By transforming our signal into the digital domain, this method leverages the techniques discussed and elaborated on in the course *EE4361* taught by Dr. Panahi.**

I. Introduction

BECAUSE using physical digital signal processors currently on the market would add cost and complexity to implementing a particular method, *MATLAB* is utilized to simulate, design, and prototype a digital signal processing pipeline that can act as a stereo receiver. This pipeline transforms the signal received by an antenna and to a separate left channel and right channel. The formulation of the problem is given and followed by the design of the digital signal pipeline. Finally, an example is discussed, showing the pipeline indeed transforms the modulated signal is into its desired audio outputs.

II. Formulation

The signals start as analog voltage magnitudes with the notation given below.

$$x_{1,a}(t) := \text{male speaker}$$
$$x_{2,a}(t) := \text{female speaker}$$

The signals are then low pass filtered such that they are bandlimited to 5kHz. This is shown with the signals' convolution with $h_{lp,f_c=5\text{kHz}}(t)$.

$$x_{1,\text{bandlimited}}(t) = x_{1,a}(t) * h_{lp,f_c=5\text{kHz}}(t)$$
$$x_{2,\text{bandlimited}}(t) = x_{2,a}(t) * h_{lp,f_c=5\text{kHz}}(t)$$

These resultant signals are sampled at twice their bandwidth at $F_{s,\text{mic}} = 16 \times 10^3\text{Hz},\ T_{s,\text{mic}} = 1/F_{s,\text{mic}}$. The sampling is stopped after a time T_f which is equivalent to rectangular windowing between 0 and T_F.

$$x_1(n) = x_{1,\text{bandlimited}}(t)\Big|_{t=nT_{s,\text{mic}}},\ 0 < nT_{s,\text{mic}} < T_F$$
$$x_2(n) = x_{2,\text{bandlimited}}(t)\ \Big|_{t=nT_{s,\text{mic}}},\ 0 < nT_{s,\text{mic}} < T_F$$

In the digital transmission pipeline, the signals are added and subtracted. The sum signal is the signal channel (mono) signal. The difference signal will eventually allow the left and right channels to be separated on stereo receivers.

$$s_1(n) = x_1(n) + x_2(n)$$
$$s_2(n) = x_1(n) - x_2(n)$$

The signals are then digitally modulated to prepare for transmission. This modulation occurs at $f_{\text{carrier}} = 70\text{kHz}$ and $f_{\text{carrier}} + f_\Delta = 90\text{kHz}$.

$$0 < nT_{s,\text{mic}} < T_F$$
$$s_{1,\text{modulated}}(n) = s_1(n)\cos\left(\frac{2\pi f_{\text{carrier}}}{F_{s,\text{mic}}}n\right)$$
$$s_{2,\text{modulated}}(n) = s_2(n)\cos\left(\frac{2\pi(f_{\text{carrier}} + f_\Delta)}{F_{s,\text{mic}}}n\right)$$

The final step in the digital transmission pipeline is adding the sum and difference signals.

$$TX(n) = s_{1,\text{modulated}}(n) + s_{2,\text{modulated}}(n)$$

The completed digital transmission signal must be converted to analog for transmission. In this model, an ideal D/A is used.

$$TX(t) = \sum_{k=-\infty}^{\infty} TX(k)\delta(t + kT_s) * h_{lp,f_c=90\text{kHz}}(t)$$

For this model, the analog transmission is assumed to be lossless.

$$RX(t) = TX(t)$$

After transmission, the receiver will sample the signal with an analog to digital converter with sampling frequency and sample interval $F_{s,\text{receiver}} = 400 \times 10^3\text{Hz},\ T_{s,\text{receiver}} = 1/F_{s,\text{receiver}}$, respectively. Again, because the recorded signal was finite, the signal is windowed between 0 and T_F

$$RX(n) = RX(t)\Big|_{t=nT_{s,\text{receiver}}},\ 0 < nT_{s,\text{receiver}} < T_F$$

We now come to the main focus of the design. This received discrete time signal must be transformed to output reconstructed versions of the left channel and right channel. For now, this is shown with a black box function RECV

$$\text{RECV} := \text{Receiver to be designed}$$
$$\tilde{x}_1(n), \tilde{x}_2(n) = \text{RECV}\left(RX(n)\right)$$

By reconstructing the original signals, the signal processing pipeline is complete.

III. Design

The RECV operation is given as the below signal processing pipeline. The pipeline goes through stages of demodulation, filtering, addition/subtraction, and gain.

A. Demodulation

The signals of importance lie in the frequency domain at 70kHz and 90kHz. In order to extract the signal information at those frequencies, the received signal needs to be demodulated at both 70kHz and 90kHz. The *MATLAB* implementation is shown in Fig. 4.

$$
\begin{aligned}
0 < nT_{s,\text{receiver}} &< T_F \\
RX_{\text{demodulated, 70kHz}}(n) &= RX(n)\cos\left(\frac{2\pi f_{\text{carrier}}}{F_{s,\text{receiver}}}n\right) \\
RX_{\text{demodulated, 90kHz}}(n) &= RX(n)\cos\left(\frac{2\pi (f_{\text{carrier}} + f_{\Delta})}{F_{s,\text{receiver}}}n\right)
\end{aligned}
$$

B. Filtering

Unfortunately, demodulation also results in frequency shifted replicas of the original received signals appearing in the resultant signals. These replicas can be removed via filtering. Because the information signal was originally bandlimited to 5kHz, to reconstruct the signals faithfully the demodulated signal was also be low pass filtered to that frequency. The *MATLAB* implementation of this filtering is show in Fig. 3.

$$
\begin{aligned}
N &:= \text{length}\left(\tilde{s}_1(n)\right) \\
\tilde{s}_1(n) &= RX_{\text{demodulated, 70kHz}}(n) \; N \; h_{lp,f_c=5\text{kHz}}(n) \\
\tilde{s}_2(n) &= RX_{\text{demodulated, 90kHz}}(n) \; N \; {}_{lp,f_c=5\text{kHz}}(n)
\end{aligned}
$$

C. Addition/Subtraction and Gain

With the sum and difference signals now reconstructed, the left and right channels can be acquired. Both gains must be $\frac{1}{2}$ because each

$$
\begin{aligned}
\tilde{x}_1(n) &= G_1\left(\tilde{s}_1(n) + \tilde{s}_2(n)\right) \\
\tilde{x}_2(n) &= G_2\left(\tilde{s}_1(n) - \tilde{s}_2(n)\right) \\
G_1, G_2 &\rightarrow \frac{1}{2}
\end{aligned}
$$

IV. Results

The example data file distributed through email is transformed and used as an example of the functional stereo receiver digital signal processing pipeline. The data file was analyzed the resulting signals $x_1(n)$ and $x_2(n)$ were plotted in Fig. 1. Additionally, their respective spectrums as given by the DFT were calculated and plotted in Fig. 2. The right, female channel says "The clothes dried on a thin wooden rack. " While the left, male speakers says, "The birch canoe slid on the smooth planks. "

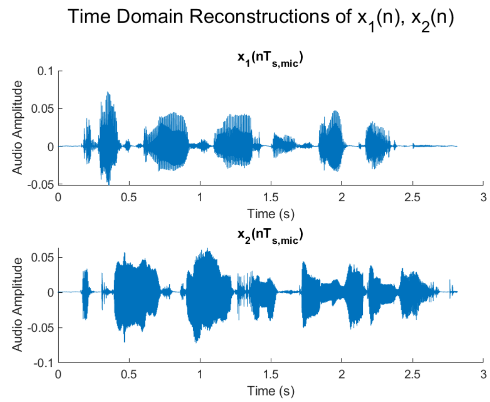

Fig. 1. Time Domain Representation of Reconstructed Signals $x_1(n)$ and $x_2(n)$

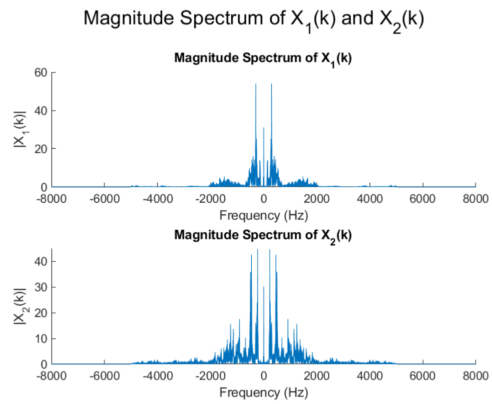

Fig. 2. Magnitude of DFT of Reconstructed Signals $x_1(n)$ and $x_2(n)$

V. Discussion and Conclusion

Through the discussion of this emphatically digital method to transmit and receive both mono and stereo audio, immense familiarity has been gained with the fundamentals of digital signal processing.

Appendix A
MATLAB Implementations

Fig. 3. *MATLAB* Implementation of Filtering

```
function filtered_sig = ideal_lowpass(signal_fft,
    cutoff_freq,Fs)
    num_samples = length(signal_fft);
    passband_freq_index = floor(cutoff_freq*
        num_samples/Fs)
    rectangle = zeros(size(signal_fft));
    rectangle(1:passband_freq_index+1) = 1;
    rectangle(end-passband_freq_index+1:end) = 1;
    filtered_sig = rectangle .* signal_fft;
end
```

Fig. 4. *MATLAB* Implementation of Demodulation

```
function demodsig = demodulate_signal(time_sig,
     shift_down_freq, Fs)
    num_samples = length(time_sig);
    carrier = cos(2*pi*shift_down_freq*(0:
         num_samples-1)/Fs)'
    demod_time_sig = time_sig .* carrier;
    demodsig = demod_time_sig
end
```

APPENDIX B
REPOSITORY

This source of the latex report, the *MATLAB* code, and all references can be found at https://github.com/Stephen-Campbell-UTD/DSP_Project