DTV Comparison: OFDM v. 8VSB

Annie Martin, Sarah Zimmermann

***Abstract*— Two digital television coding schemes have been proposed both in America and around the world. These schemes, 8-Vestigial Sideband Modulation (8-VSB) and Coded Orthogonal Frequency Division Multiplexing (COFDM) may have different advantages in different types of environments. We have designed models of each of these systems, and used them to test the advantages of each coding scheme.**

I. Introduction

A. Background

In 1987 broadcasters petitioned Federal Communications Commission (FCC) to reserve a spectrum for Digital Television (DTV). It was believed at that time that DTV would require new spectrum distribution as television stations moved towards digital television. By 1988, PBS had debuted DTV transmission, although no commercial DTV receivers had been sent to market. In 1992, the first all-digital modulation scheme was announced. This scheme fit in the original 6MHz channel and allowed for full HD television to be broadcast, which signaled the end for analog TV transmission as the new digital standard could fit in the older analog spectrum, and produce better results. Seven companies working on this technology, formed a ”Grand Alliance” to create the ”best” modulation scheme. The Grand Alliance developed the 8-VSB technology which was documented under the auspices of the Advanced Television Systems Committee (ATSC). In 1996, after extensive testing, the FCC adopted the ATSC transmission Standard. As soon as 1998, 7 digital TV stations were on the air.(CinemaSource 2002) In the following year, Sinclair Broadcasting Group created a stir in the broadcast industry when they conducted field trials of the 8-VSB standard (FCC 1999). They reported that the 8-VSB standard was inherently flawed. Sinclair claimed that 8-VSB receiver antennas were unable to consistently receive signal in a city environment indoors (FCC 1999). Sinclair proposed a system similar to Europe’s DVB-T (Digital Video Broadcasting - Terrestrial) standard, which is based on Coded Orthogonal Frequency Division Multiplexing (COFDM). Sinclair Broadcasting, along with other TV broadcasters, petitioned the FCC to re-evaluate the DTV Standard. The major points of debate involve the multipath handling of 8-VSB, the power consumption of COFDM, and the financial and political cost of the change. A careful evaluation of these points of disagreement inevitably leads to an affirmation of the decision made by the FCC to stick with the 8-VSB standard. Here, we revisit that debate.

B. The 8-VSB Model

The 8-VSB modulation scheme is the current standard for American Digital Television. It was adopted over a COFDM scheme because it was found to be more power-efficient over large rural areas that are common in the United States.

8-VSB uses a 8 levels of pulse amplitude modulation. In the frequency domain, this series of pulses creates a real signal at baseband, which means that it is symmetric across the f=0 axis. The series of pulses is then modulated to the appropriate transmit frequency. This creates four copies of the data in the frequency domain- two from the signal at baseband, which are doubled when modulated by a cosine. A VSB filter removes two of these copies without disrupting the reality of the signal (See Figure 1). This allows the transmit power to be greatly reduced without reducing the amount of data being sent. Note that it would be technically more power-efficient to use a vertical filter to send only one sideband of the symbol (”Single Sideband Modulation”). However, creating such a high order filter requires extremely precise equipment, and is not practical in reality.

The 8-VSB standards adopted by the FCC use 8-PAM encoding. Additionally, to increase efficiency, a considerable amount of signal compression is used. Block coding is used to create time diversity among the symbols. Error correction codes are used at several levels, both in the data and at the block levels. We chose not to model any of these sophisticated methods in order to increase the simplicity of our model. Instead, we modeled the most basic 8-VSB encoding scheme using 8-PAM modulation and a VSB filter, to determine what the effects of the actual encoding scheme are on the transmitted data.

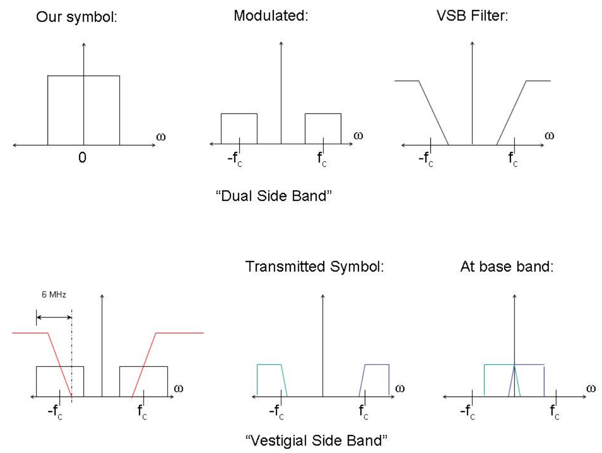

Fig. 1. VSB Modulation. a)Symbol at baseband. b)Modulated symbol. c)VSB filter. d)VSB filter overlayed on the modulated symbol. e)The symbol after being filtered. f)The receive-side demodulated symbol.

C. The OFDM Model

The European Union has adopted the DVB-T standard, which uses COFDM for the broadcast of digital television. The European Union adopted this standard because it was found to be more efficient in urban environments, which characterizes much of the densely populated European content. It uses 16, 64, or 256QAM encoding on each of 2000 or 8000 carriers. The inverse ifft is taken to determine the signal to be sent in the time domain. Cyclic prefixing is then added to avoid inter-symbol interference when going through the channel, by creating a circular convolution instead of a partial linear convolution. The DVB-T standard also includes some error correction and compression, as it is important for a signal to be appear random when going through a channel. Our model uses 16-QAM on 4096 carriers. We do cyclic prefixing, however we use no error correction or data compression.

II. Channel Models

A very general channel model was needed to test the two modulation schemes. The Stanford University Interim Model was chosen for its versatility and ease of use. The Stanford University Interim Model (SUI model) is based off statistical channel model that incorporates various models for different channel factors. Each of these factor models have been experimentally derived and evaluated but a full scale evaluation of the statistical model has not been extensively tested. These factor models Include models for path loss of shadowing effect, multipath delay, K-factor (scattering factor), and doppler spread. The path loss model is a modified Hata- Okumura model tested by the AT&T Wireless Services across the United States and is accurate for frequencies from 500 MHz to 1.9GHz. They multipath delay profile was drawn from a large number of scholarly reports and the K-factor used an independent set of data collected in the San Francisco Bay Area to validate the model. The Doppler spread is a COST 207 model but the Doppler effect is neglected in our use of the channel model and is inconsequential. The versatility of the SUI model is contained in the fact that model parameters are defined for 6 different channel types. These 6 channels are defined by a set of 4 characteristics: terrain type, scattering factors, delay spread, and Doppler spread. The three terrain types are A, B, and C. Terrain type A is a hilly terrain with a high tree density. Terrain type B is a moderate terrain and C is a flat terrain with a low tree density. The scattering or K-factor is a measure of ”fixed power”/”scatter power.” This K-factor is some measure of signal attenuation. One contributor to the k-factor is season. The K-factor is higher in the winter because there are no leaves to further scatter the signal. The delay spread is the multipath delay profile. Urban environments have a higher multipath profile than rural environments. Again, the Doppler spread is irrelevant for the purposes of this simulation.

Model	Terrain	K-factor	Delay Spread	Doppler Spread
1	C	High	Low	Low
2	C	High	Low	Low
3	B	Low	Low	Low
4	B	Low	Moderate	High
5	A	Low	High	Low
6	A	Low	High	High

We can see from the above chart that the 6 channel models are common channels in the United States. The SUI model was chosen for two primary reasons. The first was convenience. Matlab code was freely available to simulate the channel. This greatly spread the process of simulation and moved the project on to analysis much faster than writing a new channel model in matlab. The second and more important reason for the use of the SUI model was the versatility of the model. The model needed to work for both the 8-VSB and COFDM modulation structures and include channels for both urban and rural environments. The SUI model allowed for simulation of urban, rural, and intermediate environments in a simple and quick to implement package.

III. Results and Conclusions

We simulated the COFDM modulation scheme and VSB modulation scheme for each of the six SUI channels. We varied the distance over which each signal traveled, which changed the SNR of the signal, and measured the bit error rate (BER) of the signal at the receiver.

We expect that the COFDM modulation scheme will perform with a lower BER rate for SUI-1 and SUI-2, the channels based on urban environments. We expect the VSB modulation scheme will perform with a lower BER rate (and therefore be more power-efficient) for the channels modeled by SUI-5 and SUI-6, as those are the channels based on rural environments.

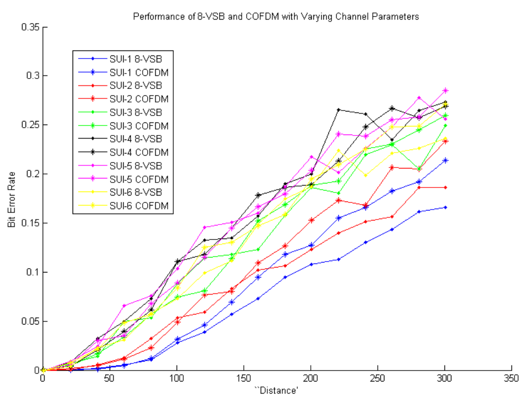

Fig. 2. OFDM and VSB responses at varying reciever distances

We can analyze the difference between the OFDM and VSB responses by subtracting the VSB BER from the OFDM BER at each receiver distance for each channel (Figure 3).

Interestingly, the VSB model seems to perform better across all channel models. Note however, that the improved performance is less consistent in the high SNR (short distance) range. We can repeat our experiment for these shorter distances (Figure 4).

We can again analyze the difference between the OFDM and VSB responses by subtracting the VSB BER from the OFDM BER at each receiver distance for each channel (Figure 5).

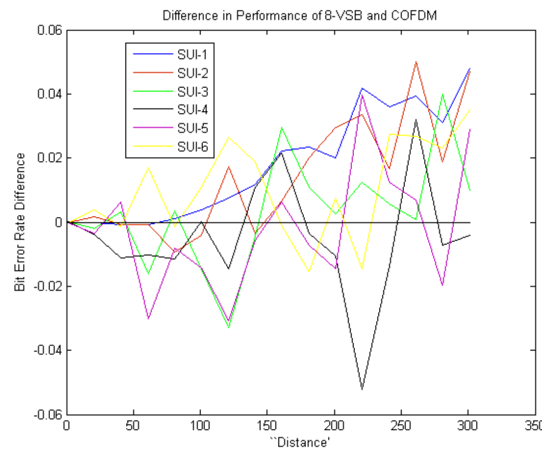

Fig. 3. OFDM and VSB Difference: BER of OFDM - BER of VSB

Interestingly, here the COFDM model seems to perform better across all channel models, with some drop-off for channels 1 and 2 at higher distance (lower SNR) values.

Our results seem to suggest that and COFDM modulation scheme works better at high SNR. Such an environment may include a city-like terrain, where the shorter distances between broadcast towers allow for less fading, and therefore higher SNR. The VSB modulation scheme works better at low SNR, which may be true rural environments that have long distances and dense tree distribution to increase the noise in a system. However, our model made many unrealistic simplifications, and it is difficult to draw any conclusions relevant to the OFDM/VSB debate because of this.

IV. Future Directions

There are many ways to expand the analysis presenting in this simulation of the 8-VSB and COFDM modulation schemes. Using a more complicated and more experimentally accurate model would presumably validate the results in a more rigorous manner. Experimental data itself would serve to completely justify the conclusions drawn in this analysis. Additionally, more rigorous power analysis and range analysis would be necessary to truly analyze the two DTV schemes in question.

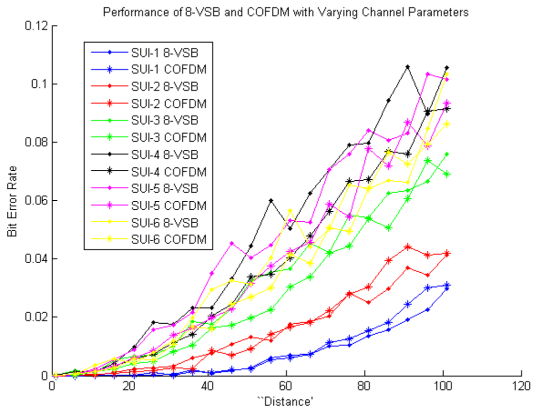

Fig. 4. OFDM and VSB responses at varying reciever distances

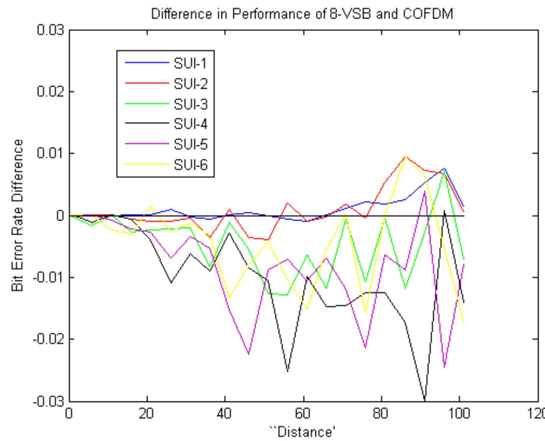

Fig. 5. OFDM and VSB Difference: BER of OFDM - BER of VSB

Several large assumptions were made in the simulation presented. One such assumption was that all data occurred in a single coherence bandwidth. Using actual block coding, analysis on data throughput and total power consumption over time including the variety of training measures could be preformed.

Additionally, there are many parts of the ATSC and DVB-T modulation schemes that were omitted from this analysis. Simulating the entire protocol including error correction coding would provide an analysis of the protocols themselves and not just the modulation schemes in place.

The simulation presented is just a small part of the total analysis possible for these two schemes. Not only is a more in-depth look at the modulation schemes necessary, but error correction coding, hardware requirements, and politics all play a huge role in deciding on a broadcast standard. Rigorous analysis across many disciplines is necessary in order to decide on the optimal standard for commercial deployment.

V. References

1) Sporano, David. ”What is 8-VSB Anyways?” Miller Freedman Guide to Digital Television. 1997. CinemaSource Technical Bulletins. Copyright 2002.
2) Wu, Yiyan. ”Perfromance Comparison of atsc 8-VSB and DVB-T COFDM transmission systems for Digital Television Terrestrial Broadcasting.” IEEE Transactions on Consumer Electronics, Vol. 45, No. 3, AUGUST 1999 .
3) Tse, David, Viswanath, Pramod. Fundamentals of Wireless Communication. Cambridge University Press, 2005.
4) ”Channel Models for Fixed Wireless Applications” IEEE 802.16.3c-01/29r4

VI. Acknowledgements

We would like to thank our professor, Siddhartan Govidasamy, for teaching us and inspiring us in the field of communications.

```
function ber = ofdm(sui,distance)

%Framing of COFDM

%Length of transmission in bits
x = 10000;
powerscale=100;
%Random Generated Bit Array
data=floor(rand(x,1)*2);

%Modulate through 16QAM
T = The16QAMifyer(data);

%ifft to take it into time

npow = 2^nextpow2(T);
t=ifft(T,npow);

%Cyclic Padding
L=ceil(length(t)/5);%use 1/4 of t as cyclic padding for now, this should be
changed when we know more about the channel.
x=powerscale*OFDM_addcyclicpadding(t,L);

%Goes through Channel
[tap1, tap2, tap3] = SUI_channel(sui);
channel = [tap1(1) tap2(1) tap3(1)];

xnew = conv(x,channel);
xnew = xnew + distance*.01/sqrt(2)*(randn(size(xnew)) + j*randn(size(xnew)));


%Un-CyclicPad
xest=OFDM_removecyclicpadding(xnew,L)/powerscale;

%fft to take it back into freq
 xest = fft(xest,npow)./fft(channel,npow);

% Xestlong=fft(xest,npow);
Xest=xest(1:length(T));

%Decision Boundaries based on QAM
dataest = TheDE16QAMifyer(Xest);

% calculate percent error
count = 0;
for i = 1:length(data)
    if data(i)~= dataest(i)
        count = count+1;
    end
end
```

```
Percent_error = count/length(data);

ber = Percent_error;
```

```
function Percent_error=VSB_SUI_function_new(SUI,distance)
%Framing of 8VSB; takes SUI type and distance and returns percent
%error for a single test run.

%Length of transmission in bits
d = 10000;

%% Random Generated Bit Array
data=floor(rand(1,d)*2);

%% Modulate through 8PAM
PAMed = PAM8ify(data);
extendeddata=zeros(1,d);
for n=1:length(PAMed)
    extendeddata((n-1)*10+1:n*10)=PAMed(n)*ones(1,10);
end

%% VSB filter
x=VSB_filter(extendeddata);

%% Goes through Channel
[tap1, tap2, tap3] = SUI_channel(SUI);
y = conv(x, [tap1 tap2 tap3]);
noisepower=.01;
y=y+noisepower*distance*(randn(size(y))+j*randn(size(y)))/2^.5;

%% Un-VSB
npow=2^16;
y_undochannel=ifft(fft(y,npow)./fft([tap1,tap2,tap3],npow),npow);
y_undochannel=y_undochannel(1:length(x));
y2=VSB_lpf(y_undochannel,0);

%% VSB boundaries
xest=abs(VSB_filterboundaries(real(y2),9));
xest=xest(1:ceil(d/3));
shift=1.44;
scale=26.68;
xest=(xest-shift)/scale*14+1;

%% Decision Boundaries based on PAM
dePAMed = dePAM8ify(xest);

%% calculate percent error
count = 0;
for i = 1:length(data)
    if data(i)~= dePAMed(i)
        count = count+1;
    end
end

Percent_error = count/length(data);
end
```

```
function [tap1,tap2,tap3] = SUI_channel(SUI)

N=1; %10000;
OR=20;
M=256;
Dop_res=.1;
res_accu=20;
%SUI=6;
switch SUI
    case 1
        %disp('SUI-1'); %terrain C(flat terrain, low tree density) low dopler
& delay spread: suburban & urban areas with uniform building height
        P=[0,-15,-20];
        K=[4,0,0];
        tau=[0.0,0.4,0.9];
        Dop=[.4,.3,.5];
        ant_corr=.7;
        Fnorm=-.1771;
    case 2
        %disp('SUI-2'); %terrain C(flat terrain, low tree density), low
dopler & delay spread: suburban & urban areas with uniform building height
        P=[0,-12,-15];
        K=[2,0,0];
        tau=[0.0,0.4,1.1];
        Dop=[.2,.15,.25];
        ant_corr=.5;
        Fnorm=-.393;
    case 3
        %disp('SUI-3'); %terrain B(moderate tree density), low dopler, low
delay spread:
        P=[0,-5,-10];
        K=[1,0,0];
        tau=[0.0,0.4,.9];
        Dop=[.4,.3,.5];
        ant_corr=.4;
        Fnorm=-1.5113;
    case 4
        %disp('SUI-4'); %terrain B(moderate tree density), high dopler,
moderate delay spread:
        P=[0,-4,-8];
        K=[0,0,0];
        tau=[0.0,1.5,4.0];
        Dop=[.2,.15,.25];
        ant_corr=.3;
        Fnorm=-1.9218;
    case 5
        %disp('SUI-5'); %terrain A(hilly, moderate to heavy tree density),
low dopler, high delay spread:
        P=[0,-5,-10];
        K=[0,0,0];
        tau=[0.0,4,10];
        Dop=[2,1.5,2.5];
        ant_corr=.3;
        Fnorm=-1.5113;
    otherwise
```

```
        %disp('SUI-6'); %terrain A(hilly, moderate to heavy tree density),
high dopler, high delay spread:
        P=[0,-10,-14];
        K=[0,0,0];
        tau=[0.0,14,20];
        Dop=[.4,.3,.5];
        ant_corr=.3;
        Fnorm=-.5683;
end
%calculate power in constant and random Rice distribution for each tap
P=10.^(P/10); %linear power
s2=P./(K+1); %variance
m2=P.*(K./(K+1)); %constant power
m=sqrt(m2);
%Create Ricean channels
L=length(P);
paths_r=sqrt(.5)*(randn(L,N) + j*randn(L,N)).*((sqrt(s2))'*ones(1,N));
paths_c=m'*ones(1,N);
%normalize filter in the time domain
for p = 1:L
    D=Dop(p)/max(Dop)/2;
    f0=[0:M*D]/(M*D);
    PSD=.785*f0.^4 - 1.72*f0.^2 +1;
    filt=[PSD(1:end-1),zeros(1,round(M-2*M*D)),PSD(end:-1:2)];
    filt=real(ifftshift(ifft(sqrt(filt))));
    filt=filt/sqrt(sum(filt.^2));
    path=fftfilt(filt, [paths_r(p,:), zeros(1,M)]);
    paths_r(p,:)=path(1+M/2:end-M/2);
end;
paths=(paths_r+paths_c)*10^(Fnorm/20);
SR=max(Dop)*2;
m=lcm(SR/Dop_res, OR/Dop_res);
P= m/SR*Dop_res;
Q=m/OR*Dop_res;
paths_OR=zeros(L,ceil(N*P/Q));
for p=1:L
    paths_OR(p,:) = resample(paths(p,:),P, Q, res_accu);
end
tap1 =paths_OR(1,1);
tap2 =paths_OR(2,1);
tap3 =paths_OR(3,1);
end
```

```
function [paths, paths_r, paths_c] = calcRandomPath(N, OR, M, Dop_res,
res_accu, P, K, tau, Dop, ant_corr, Fnorm, plotplease)

P = 10.^(P/10); % calculate linear power
s2 = P./(K+1); % calculate variance
m2 = P.*(K./(K+1)); % calculate constant power
m = sqrt(m2); % calculate constant part



%%
rmsdel = sqrt( sum(P.*(tau.^2))/sum(P) - (sum(P.*tau)/sum(P))^2 );
fprintf('rms delay spread %6.3f ?s\n', rmsdel);
%%
L = length(P); % number of taps
paths_r = sqrt(1/2)*(randn(L,N) + j*randn(L,N)).*((sqrt(s2))' * ones(1,N));
paths_c = m' * ones(1,N);

for p = 1:L
D = Dop(p) / max(Dop) / 2; % normalize to highest Doppler
f0 = [0:M*D]/(M*D); % frequency vector
PSD = 0.785*f0.^4 - 1.72*f0.^2 + 1.0; % PSD approximation
filt = [ PSD(1:end-1) zeros(1,M-2*M*D) PSD(end:-1:2) ]; % S(f)
filt = sqrt(filt); % from S(f) to |H(f)|
filt = ifftshift(ifft(filt)); % get impulse response
filt = real(filt); % want a real-valued filter
filt = filt / sqrt(sum(filt.^2)); % normalize filter
path = fftfilt(filt, [ paths_r(p,:) zeros(1,M) ]);
paths_r(p,:) = path(1+M/2:end-M/2);
end;
paths = paths_r + paths_c;


paths = paths * 10^(Fnorm/20); % multiply all coefficients with F


Pest = mean(abs(paths).^2, 2);
fprintf('tap mean power level: %0.2f dB\n', 10*log10(Pest));

if plotplease ==1
figure, psd(paths(1,:), 512, max(Dop));
end
end
```