

## Speech Processing for communication

1. Generate your own voice of 'your name' as a wav file.
2. Change to an ascii file.
3. Plot the speech of your name and produce the sound.
4. Complete dm\_decoder. (Complete if, else statement.)
5. Apply your voice to the DM encoder and decoder. The DM encoded signal is the input to the DM decoder.
6. Plot the speech of your name and produce the sound after DM decoder

You can use MATLAB to encode.

Please turn in your source code as well as the outcome It is acceptable to turn in the home work in electronic form.

%Demo matlab code

```
%example %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
load('your_name.dat')
xx=your_name;
maxx=max(abs(xx));
s=xx/maxx;
fs=10000;
sound(s, fs);
pause
% Delta modulation-demodulation
StepSize=1/15;
%encoder
cn=dm_encoder(s, StepSize);
%decoder
[Sn]=dm_decoder(StepSize,fs, cn);
%Sa:reconstructed signal after LPF.
Sa=lpf(100, .4, Sn);
%plot the input and accumulator output
subplot(3,1,1);plot(s,'r');
grid on
title('DM (red;input signal, green;decoder output, blue; LPF output)');
ylabel('amplitude');
subplot(3,1,2);plot(Sn,'g');
grid on
ylabel('amplitude');
subplot(3,1,3);plot(Sa,'b');
grid on
xlabel('index, n');
ylabel('amplitude');
sound(Sa,fs);
```

```

function cn=dm_encoder(x, StepSize)
%DM_encoder constant step size Delta Modulator.
%StepSize: Step size for the delta modulator.
%x: sampled input signal, where fs>=5*Nyquist rate.
%cn:encoded binary data.
% Reference[1]: Martin S. Roden, Digital and Data
% Communication Systems, Prentice-Hall, 1982, pp. 105-110.
% Reference[2]: L. R. Rabiner, and R. W. Schafer, Digital
% Processing of Speech Signals, Prentice-Hall, 1978, pp. 216-225.

%staircase generator and comparator (quantizer)
xlen = length(x);
accum(1) = 0;
for i=1:xlen
    if(x(i)>=accum(i))
        e_tilda_n(i)=1;
        accum(i+1) = accum(i) + e_tilda_n(i) * StepSize;
    else
        e_tilda_n(i)=-1;
        accum(i+1) = accum(i) + e_tilda_n(i) * StepSize;
        tx(i)=0;
    end
end

%encoder
cn = e_tilda_n < 0;

function [Sn]=dm_decoder(StepSize,fs, cn)
%StepSize: Step size for the delta demomodulator.
%fs: sampling frequency.
%cn: received (encoded) signal.
%Sn:reconstructed quantized signal before LPF.

xlen = length(cn);
Ts=1/fs;
n=0:Ts:Ts*(xlen-1);

%decoder and accumulator
xlen = length(cn);
accum(1) = 0;

for i=1:xlen
    if

        else

    end
end

% quantized output
[xx,yy]=stairs(n,accum(2:xlen+1));
Sn=accum(2:xlen+1);

```