Mobile Communications

Case Study: Software Simulation of Digital Continuous Modulation for Mobile Communications

1 Introduction: In recent years there has been a rapid growth in application of computer simulation in communication engineering. With the hardware becoming more complex and costly, a way forward to many researcher and teachers is to implement ideas in the software environment. Computer simulations in many cases are an attractive alternative to hardware implementation. The advantage would be that one is able to make significant changes readily. This allows testing of the system using idealised processing elements, which may take a significant time to design and realise in hardware. In addition, simulation can support the hardware design by giving optimised component values, for the critical parts, and an early indication of the performance of the system. In simulation information can be tabulated and plotted with ease. There are several commercial packages, which enable complex systems to be modelled. In this study a commercially available software package called Matlab is used to demonstrate how complex communication systems can be modelled and their behaviour explored. It is the intention of this case study to demonstrate that MATLAB can be used to implement complete communication systems as shown in Fig. 1.

![A typical communication system block diagram.](image)

2 Modulation: Modulation is an essential process in telecommunication since it enables multiple signals to be transmitted simultaneously over a common medium or communication channel. The process involves transferring the information, data, voice, video (i.e. the modulating signal) to a high-frequency signal known as the carrier. For a sinusoidal carrier signal, modulation is achieved by changing the carrier amplitude, frequency or phase with the modulating signal, as outlined below:

- When the amplitude of the carrier signal is varied in accordance with the modulating signal, the form of modulation is known as amplitude modulation (AM).
- When the frequency of the carrier signal is varied, the result is frequency modulation (FM).
- When the phase of the carrier signal is varied, the result is phase modulation (PM).

2.1 Amplitude modulation: Amplitude modulation (AM) is used in applications such as radio and television Broadcasting. An amplitude modulated carrier signal $e(t)$ can be expressed as [Young, 1990]:

$$e(t) = E_c[1 + M \cos(2\pi f_m t)] \cos(2\pi f_c t)$$  \quad (2)

Where, $E_c$ and $f_c$ are the peak amplitude and frequency of the carrier signal, respectively, $M = E_c/E_M$ is the modulation index, and $E_m$ and $f_m$ are the peak amplitude and frequency of the modulating signal, respectively. Note that (2) also know as AM with the carrier contains three terms:

1. Carrier: $E_c \cos 2\pi f_c t$
2. Upper side bands (USB): $0.5E_c M[\cos(2\pi f_c + 2\pi f_m )t$
3. Lower side band (LSB): $0.5E_c M[\cos(2\pi f_c - 2\pi f_m )t$. 

Figure 2a and 2b shows the time domain waveforms for the modulating, modulated AM signal and its spectrum, and over-modulated AM waveform.

Fig. 2 AM simulation results: (a) modulating signal, (b) AM waveform, (c) AM spectrum and (d) over modulated AM waveform.

2.2 Case Study

2.2.1 Amplitude Shift Keying: A digital signal can be used to modulate the amplitude, frequency or phase of a sinusoidal carrier wave. If the modulating signal (i.e. digital data) format is non-return to zero (NRZ), then the carrier will be basically switched or keyed from one discrete level to another. If the digital data is used to change the carrier voltage level, then the modulated waveform is called ASK.

The ASK waveform for one pulse (i.e. binary '1') is given as:

$$s_{ask}(t) = \sum_{n} a_n f(t - nT),$$

(3)

where $f(t) = A_c \sin \omega_c t$ is the basic carrier signal, $\{a_n\}$ is the data sequence, and $T$ is the data bit duration. For $a_n = 1$ or 0, the expression for ASK can be shown as:

$$s_{ask}(t) = \begin{cases} A_c \sin \omega_c t & 0 < t \leq T \\ 0 & \text{otherwise} \end{cases}$$

For binary "1"

(4)

The transmission bandwidth of ASK is given as: $B_f = 2B$, where, $B$ is the baseband (data) bandwidth. The transmission channel is assumed to be ideal with response given as:

$$C(f) = \begin{cases} 1 & |f| \leq B_{ask} \\ 0 & \text{otherwise} \end{cases}$$

(5)

where $B_{ask}$ is the ASK bandwidth.
2.2.2. Matlab Implementation: In this case study you will carry out full simulation of ASK system as outlined below:

➢ Transmitter:

\[
\begin{align*}
\text{Transmitter:} & \\
\text{Fig. 3} & \\
\text{Baseband data (NRZ)} & \quad \text{ASK } s_{\text{ask}}(t) \\
S(t) & \quad \text{Amp. (A)} \\
\text{Carrier } c(t) & \quad \text{Antenna } (G_t)
\end{align*}
\]

\(P_t\) is the transmitted average power, and \(G_t\) is the transmitter antenna gain.

➢ Channel: The channel is assumed to be free space, where there is little inherent filtering or distortion other than frequency dependent absorption in the atmosphere, provide there are only one direct line of sight path between the transmitter and receiver. However, in situations where there are more than one path, the signal will experience significant amount of distortion when observed at the receiver. Different path delay means different phase delay between the rays. If the phase difference between two rays is 0°, the rays will reinforce each other, whereas if the phase difference is 180°, the rays in fact partially cancel each other. A band-limited channel can be modelled as a linear filter whose frequency response characteristic matches the frequency response of the channel. A two path radio channel could be modelled as:

\[
\begin{align*}
\text{Fig. 4 Two path channel model.} & \\
\text{ASK}_1 & \quad \text{Delay} \\
\text{Delayed ASK} & \quad \text{Resultant ASK}
\end{align*}
\]

➢ Receiver: During transmission the ASK signal is contaminated with white Gaussian noise \(n(t)\) as outlined above. Since ASK is a form of AM signal, then the following methods may be used as the demodulators: (i) Envelop detector (no-coherent detection) and (ii) Product detector (Coherent detection). Here you will be implementing the product detector as shown in Fig. 5. For product detection the carrier reference signal \(c(t)\) must be provide, which should have the same phase and frequency as the transmitter carrier signal. Also shown is the channel where noise is added to the received ASK signal.

The received ASK signal power at the receiving antenna is:

\[
P_r = P_t \frac{G_t G_r \lambda^2}{(4\pi d)^2 L}
\]

(6)

where, \(G_r\) is the receiver antenna gain, \(d\) is the separation distance (m) between transmitter and receiver, and \(L\) is the path loss.

The RMS voltage generated at the input of the receiver is:
\[ E_{\text{rms}} = (P_r R_{an})^{0.5} \]  

where \( R_{an} \) is the antenna impedance.

\[ e_r(t) = s_{\text{ask}}(t) + n(t) \]  

This is then passed through a unity gain band-pass filter, with centre frequency and bandwidth of \( \omega_c \) and \( 2n\omega_m \), respectively, in order to band limit the noise, but not the ASK signal, where \( n = 1, 2, \ldots \). The output of the band-pass filter is given by:

\[ e_{\text{bp}}(t) = s_{\text{ask}}(t) + n_c(t) \cos \omega_c t + n_s(t) \cos \omega_m t \]  

where \( n_c(t) \) and \( n_s(t) \) are the quadrature components of band limited noise.

The output of the band-pass filter is then passed through a multiplier, the output which contains a DC, high frequency (related to the carrier) and low frequency (related to the modulating frequency) components. The output of the multiplier is passed through a low pass filter in order to reject the entire high frequency component. Finally a comparator is used to convert the output of the low pass filter which has should have the same frequency as the modulating (information) signal (data) into a data signal.

By comparing the transmitted data with the received data, the number of errors can be measured for a given signal to noise ratio. By changing signal-to-noise ratio a set of errors could be measured.

**Task 1:** Drive an expression for the signal at the output of the multiplier and identify all the frequency components.

**Task 2:** Drive an expression for the signal at the output of the low pass filter.

**Task 3:** Decide on the specification for the band pass filter.

### 3 System Specifications

#### 3.1 Transmitter:
- Modulating signal - Squarewave: Unipolar NRZ, data rate = 5 kbits/s, and peak amplitude = 5 V
- Carrier signal: Sinewave: amplitude = 20 V and frequency 1.0 MHz. (Low frequency for simulation purpose)
- Amplifier gain = 10 dB
- Antenna gain = 0 dB
3.2 **Channel:**
- Line of sight
- Channel path length = 20 km
- Path loss is = 95 dB/20 km Free space

3.3 **Receiver:**
- Antenna gain = 0 dB
- Antenna impedance = 50 Ohms
- Amplifier gain 20 dB
- Noise: White Gaussian
- Band-pass filter: You need to select its centre frequency and bandwidth. Once selected discuss it with the Laboratory Supervisor, before proceeding with the implementation.
- Multiplier:
  - Low pass filter: - Butterworth order: \(6^{th}\), bandwidth (you select), and gain (optional).
- Comparator (slicer): You need to select the correct threshold (reference) level.

4 **Simulation work to carry out:** You are required to investigate the following two cases.

**Case 1:** Free Space Propagation Model - No multipath

**Case 2:** Multipath Propagation (two paths)

First start with cases 1, finish it off them move on to case 2.

For a given set of parameters: noise level, band-pass filter, low pass filter, carry out the complete system and show the followings:

(i) **Time domain waveforms and frequency spectra for:**
- **Transmitter:** Modulating signal, Carrier signal, and Modulated carrier.
- **Channel:** Noise signal, ASK + noise signal.
- **Receiver:** Filtered ASK + Noise signal, Output of the non-linear device, and Recovered modulating signal.

(ii) **Compare the input and output data, and count the number of errors**

5 **Marking Scheme:** You will working on your own, but will submit a group report (3 per group).

The report should include: contents, abstract, background information, analysis, development of simulation software code, simulation results, comparison of simulated and calculated results, conclusions, references, and a copy of your Matlab codes on a disk.

The case study will be assessed by the following:-

i) Report: One per group, (3 per group): 70%

ii) Log book: 15%. **Your log book needs to be signed at the end of each session. Log books with no signatures will result in no marks being awarded.**

iii) Attendence: 15%.

**Submission date:** Friday Week 21, and will form 100% of the assessment. All group members will receive the same mark unless the report contains a statement to the contrary. Please hand it in to the School Office.

7 **References**

8 **Report Format**
The purpose of this Matlab case study is to illustrate concepts of signal transmission using computer simulation. Your reports should document the rationale for the case study, how it was executed, all the results, and important of all an interpretation of the results, and lesson learned. The reports should show an understanding of the transmission system concepts under investigation.

Follow the outline given below for report:
1. Title, Content, Statement of objectives, Background information, Simulation strategy for achieving the objectives, Presentation and interpretation of the results, Comparison with theoretical results, Conclusions and lessons learned, Further work, References

The length of each of these sections may vary. Be concise and neat, the goal is to communicate the results of your efforts. Think about the reader as you are constructing your reports. The total length of your report should not exceed 12 pages types single-spaced excluding figures.

Format for figures and tables.
1) Each figure/table should be placed as close to the first reference to it in the text as possible. Placing the figure/table on a separate page following the first reference to it in the text is permissible.
2) Each figure/table must have a title.
3) All axis on graphs must be labelled.
4) Each figure/table should be self-contained, that is, the title, axis labels, and other information in the figure/table should provide the reader enough information to interpret the item.

Prof. Z Ghassemlooy, Oct. 2003

Appendix A: MATLAB Codes for single path system. You need to upgrade it for multipath system.

```
%echo on
clear
cle
close all
fs=6.0e+6;   %sampling frequency;
fm=20.0e+3;   %modulating frequency;
am=1;   %peak amplitude of the modulating )data) signal;
fcs=2.0e+5;   %carrier frequency;
ac=1; %peak carrier amplitude;
n=2*(6*fs/fm);  %Maximum number of points wrt the 5 cycles of fc
fmax=(2*n*fc/fs) %Maximum frequency range
ts=1/fs;   %Sampling Time;
final=(1/fs)*(n-1); % maximum time
t=0:(1/fs):(final); %time vector
fup=fm; %Bandpass filter bandwidth (half)
fve=0:1.6666e+3:fs; %Frequency vector
fve=0:1.6666e+3:fs; %Frequency vector
vn=0.01;   %Noise amplitude (0 to 1 volts)
High=2.5; % set the upper limit of the slicer output
Low=-2.5; % set the lower limit of the slicer output
```
vt=0;  %Slicer threshold level
error=0;  %Error is equal to zero

sq=am*SQUARE(2*pi*fm*t);  %square wave modulating signal
s=1==sq;  %form a 0 to 1 signal instead of -1 to 1

figure
subplot(321);  %plotting the modulating signals;
plot (t,s), xlabel('Time in secs'), ylabel('Signal (Volts)')
title('Modulating Signal')

car=ac*sin(2*pi*fc*t);  %Sinewave carrier waveform;
subplot(322);  %Plotting the carrier waveform;
plot (t,car), xlabel('Time (s)'), ylabel('Amplitude (V)')
title('Carrier Waveform')

ASK=s.*car;  %ASK signal;
subplot(323);  %plot ASK waveform;
plot (t,ASK), xlabel('Time (s)'), ylabel('Amplitude (V)')
title('ASK Waveform')

ASKfrq = fft(ASK,n);  %now convert to frequency domain
mag = abs(ASKfrq)./(n-1);
subplot(324);  %plot ASK spectrum;
plot(fvect,mag),
subplot(325);
plot(fvect(1:fmax),mag(1:fmax)),  %To zoom in
xlabel('Frequency (Hz)'), ylabel('Amplitude (V)'), title('Frequency domain for ASK')

noise=vn*(randn(size(t)));  %Noise signal
subplot(326);
plot(t,noise), title('Filtered noise')

ASKN = (ASK + noise);  %ASK + noise
figure  %plot new set of figures
subplot(321);  %plot ASK + Noise;
plot (t,ASKN), title('Signal+Noise'), grid,
xlabel('Time (s)'), ylabel('Magnitude(V)'),
grid on
subplot(322);  %ASK + Noise Spectrum;
S Pamnoise=spectrum(ASKN,n);
specplot(S Pamnoise, fs), title('Power spectral density: signal+noise')

ASKNfrq = fft(ASKN);  %now convert to frequency domain
magr = abs(ASKNfrq)./(final/ts);
subplot(323);
plot(fvect,magr)
xlabel('Frequency in Hz'), ylabel('Amplitude (Volts)'), title('Frequency domain for ASK + Noise')
axis;

%Bandpass-filtering signal + noise, using Butterworth filter;
%fup=20.0e+3;
lc=(fc-fup);  %lowe 3dB point;
uc = (fc + fup); % upper 3dB point;
Wp = [lc uc] / (fs/2); % Passband frequency points normalising wrt the half sampling frequency
Ws = [(fc/2) (fc + (fc/2))] / (fs/2); % stopband;
Rp = 3; % attenuation at passband;
Rs = 50; % attenuation at stopband;
[nf, Wn] = buttord(Wp, Ws, Rp, Rs); [b, a] = butter(nf, Wn);

% RECEIVER;
bASKN = filter(b, a, ASKN); % Band-pass filter ASK + Noise;
subplot(324); % Bandlimited ASK + Noise
plot(t, bASKN), title('Bandlimited Signal + Noise'), grid,
xlabel('Time (s)'), ylabel('Magnitude (V)'), axis;

bASKNfrq = fft(bASKN); % now convert to frequency domain
magrb = abs(bASKNfrq) ./ (final / ts);
subplot(325);
plot(fvect, magrb)
xlabel('Frequency in Hz'), ylabel('Amplitude (Volts)'), title('Bandlimited ASK + Noise Frequency spectrum')
axis;

cm = (ASKN) .* car; % Multiplying the ASK + Noise with the carrier
subplot(326); % plot Multiplier output;
pnorm = filter(b, a, cm);
plot(t, cnorm), title('Multiplier output'), grid,
xlabel('Time (s)'), ylabel('Magnitude (V)'),
cmfrq = fft(cm); % now convert to frequency domain
magcm = abs(cmfrq) ./ (final / fs);
figure
subplot(321); % plot Multiplier output
plot(fvect, magcm),
xlabel('Frequency in Hz'), ylabel('Amplitude (Volts)'), title('Frequency domain at the output of the multiplier')
axis;
grid on

% low pass filter;

fco = 3 * fm; % cutoff frequency;
[b, a] = butter(6, (fco) / 2 / fs);
so = filter(b, a, cm);
so = so / 10;
so = so - mean(so);
subplot(322); % plot recovered signal
plot(t, so), title('Output signal'),
axis([0.9e-4 3e-4 -8 8])
xlabel('Time (s)'), ylabel('Magnitude (V)'),
sorfq = fft(so); % now convert to frequency domain
magso = abs(sorfq) ./ (final * fs);
subplot(323); %plot(fvect, magr(1:((final/ts)/2)+1))
plot(fvect, magso); grid,
xlabel('Frequency in Hz'), ylabel('Amplitude (Volts)'), title('Frequency domain for recovered signal')
axis;

% calculate the number of elements in matrix so, the output of the filter
l=length(so); % number of elements in so
for i=1:l
    if so(i) >= vt
        Vs(i)=High;
    else
        Vs(i)=Low;
    end
end
Vo=Vs+2.5;

subplot(324); % plot recovered signal;
plot (t,Vo), title('Regenerated signal'),
axis([0.9e-4 3e-4 -8 8])
xlabel('Time (s)'), ylabel('Magnitude(V)')
sqr=am*SQUARE(2*pi*fm*(t-32e-6)); % square wave modulating signal
sr=1==(sqr); % form a 0 to 1 signal instead of -1 to 1
sf=5*sr;
subplot(325)
plot (t,sf), title('Input signal'),
axis([0.9e-4 3e-4 -8 8])
xlabel('Time (s)'), ylabel('Magnitude(V)')
a=5*xor(sf,Vo); % show where errors occur against time.
subplot(326)
plot (t,a), title('Error pulses'),
axis([0.9e-4 3e-4 -8 8])
xlabel('Time (s)'), ylabel('Magnitude(V)')
for i=1:l
    if sf(i) ~= Vo(i) % if output and input are not the same then increment error
        error=error+1;
    end
end
error % display number of errors