# SOFTWARE SIMULATION TECHNIQUES FOR TEACHING COMMUNICATION SYSTEMS

*Z. GHASSEMLOOY and R. SAATCHI*

*School of Engineering, Sheffield Hallam University, Pond St., Sheffield, UK.*

**Abstract**    Theoretical concepts learnt during lectures are usually reinforced by experimental work. Computer simulations in many cases are an attractive alternative to hardware implementation. This is due to factors such as ease of system implementation, interactiveness, availability and growing use of fast personal computers.

In this article a programming language known as Matlab is used to demonstrate the ease that many communication systems can be simulated and their behaviour analysed. To illustrate this the essential codes for the Matlab simulations together with sample results for both analogue modulation and pulse modulation schemes are provided.


*Keywords: teaching methods, computer simulations, modulation, telecommunications engineering*.

## 1    INTRODUCTION

The increasing availability of fast personal computers is making simulation techniques effective for teaching in many areas. Many systems can easily be computer simulated and their behaviour analysed under different working conditions. The results may then be collected and displayed rapidly. The accuracy of results may increase and compared with hardware approach where the students need to read and document information from various instruments.


The aim of this article is to demonstrate ways in which communication systems can be simulated using a programming language called Matlab [1]. A brief description of Matlab language is provided. Two typical case studies are used to demonstrate the effectiveness of Matlab in simulating communication systems. A typical communication system consists of a transmitter, channel and receiver is shown in Fig. 1. Also shown is the noise

source which is added to the modulated signal. The first case study demonstrates amplitude modulation  where as the second cases study introduces pulse modulation technique.
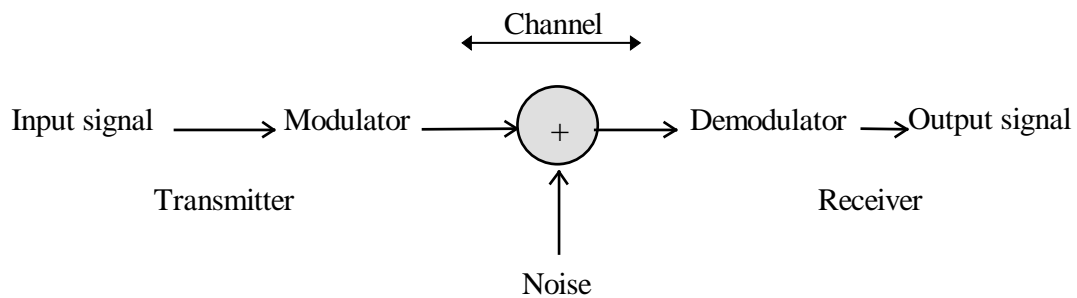


Fig. 1 A typical communication system block diagram.

## 1.1 Brief Introduction to Matlab

The Matlab software was initially developed to be a "matrix laboratory". It is gradually evolved to be an interactive programming language for scientific and engineering computation. The basic units of operation in Matlab are matrices. Matlab enables matrices to be easily manipulated, for example they can be added, subtracted, multiplied, divided, transposed, etc. Matlab has numerous tool boxes which facilitate complex scientific and engineering mathematical operations to be carried out with a minimum amount of programming.  Matlab programming language is developed in such a way that it can be learnt with ease [2]. The order in which the statements are executed can be controlled by the use of flow control statements such as IF, FOR and WHILE.

## 2 MODULATION

Modulation is an essential process in communication since it enables multiple signals to be transmitted simultaneously over a common medium or communication channel. The process involves transferring the spectrum of the signal to be transmitted (i.e. the modulating signal) to a higher frequency. The process involves using a signal known as the carrier. For a sinusoidal carrier, either its amplitude, frequency or phase can be varied by the modulating signal. When its amplitude is varied in accordance with the modulating

signal, the form of modulation is known as amplitude modulation. On the other hand if its frequency or phase are varied, the result is frequency or phase modulation respectively.

## 2.1    Case Study 1: Amplitude Modulation (AM)

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

$$e(t) = E_c[1+m(t)]\cos\omega_c t \tag{1}$$

where, $E_c$ and $\omega_c$ are the amplitude and frequency of the carrier signal, respectively and $m(t)= E_m \cos \omega_m$ is the modulating signal. The modulation index $M = E_C/E_M$.

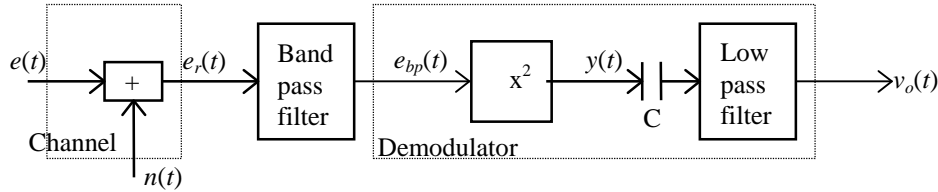During transmission the transmitted AM signal is contaminated with white Gaussain noise $n(t)$, see Fig. 2



Fig. 2   AM receiver block diagram.

The received signal plus noise is given by:

$$e_r(t) = E_c[1+m(t)]\cos\omega_c + n(t) \tag{2}$$

This is then passed through a unity gain band-pass filter, with centre frequency and bandwidth of $\omega_c$ and $2\omega_m$, respectively, in order to bandlimit the noise, the output of which is given by:

$$e_{bp}(t) = E_c[1+m(t)]\cos\omega_c t + n_c(t)\cos\omega_c t + n_s(t)\cos\omega_c t \tag{3}$$

where $n_c(t)$ and $n_s(t)$ are the quadratuer components of bandlimited noise.

To demodulation the received signal it is passed through a no-linear device the output which is given as:

$$y(t) = e_{bp}{}^2(t) = E_c{}^2/2\Big[1+2m(t)+m^2(t)+2m(t)\cos2\omega_c t+\cos2\omega_c t+m^2(t)\cos2\omega_c t\Big]$$
$$+ E_c n_c(t) + 2E_c\cos2\omega_c t - E_c n_s(t)\sin2\omega_c t$$

$$\tag{3}$$

Finally, after removing the dc components, the modulating signal $m(t)$ is recovered by simply employing a low pass filter with the output given as:

$$v_O(t) = E_c{}^2 m(t) + E_c n_c(t) \qquad\qquad (4)$$

where the first term is the information and the second term is the noise signal.

The essential Matlab code segments for simulating the above AM system are shown in the following sub-sections.

### 2.1.1 AM MATLAB codes:

```
n=3600; %Maximum number of points,
fs=6.0e+6; %sampling frequency;
fm=20.0e+3; %modulating frequency;
fc=2.0e+5; %carrier frequency;
t=0:(1/fs):(1/fs)*(n-1);     %time vector,
fvect=0: (1/fs)*(n-1):fs;   %Frequency vector
tmax=(6*(fs/fm))+1;
fmax= 2*n*(fc/fs);
M=0.5; %modulation index;
sig=sin(2*pi*fm*t); %Modulating signal,
car=sin(2*pi*fc*t); %Carrier signal;
AM=(1+M*sig).*car;% AM signal,
AMfrq = fft(AM,n);
mag = abs(AMfrq)./(n-1); %AM frequency spectrum
```

The resulting AM signal, with 50% modulation index, and a carrier frequency of 200 kHz, and its frequency spectrum are shown in Fig. 2.a and b, respectively. The spectrum consists of the carrier and a pair of side bands, at 200 kHz ± 20 kHz, components.

During transmission noise signal is added to the modulated AM signal and the effects both in time and frequency domains are shown in Fig. 3.c and d.

### 2.1.2 MATLAB codes for AM plus noise and bandlimited received signal:

```
noise=0.7*(randn(size(t)));  %Noise signal
AMN = (AM + noise); %signa + noise,
AMNfrq = fft(AMN);%now convert to frequency domain,
magr = abs(AMNfrq)./(final/ts);
%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); %Normalised passband freqency points,
Ws=[(fc/2) (fc+(fc/2))]/(fs/2); %stopband;
Rp=3;%attenuation at passband;
Rs=50; %attenuation at stopband;
```

```
[n,Wn]=buttord(Wp,Ws, Rp,Rs),
[b,a]=butter(n,Wn);
bAMN=filter(b,a,AMN);%Band-pass filter AM + Noise;
bAMNfrq = fft(bAMN);%now convert to frequency domain,
magrb = abs(bAMNfrq)./(final/ts);
```
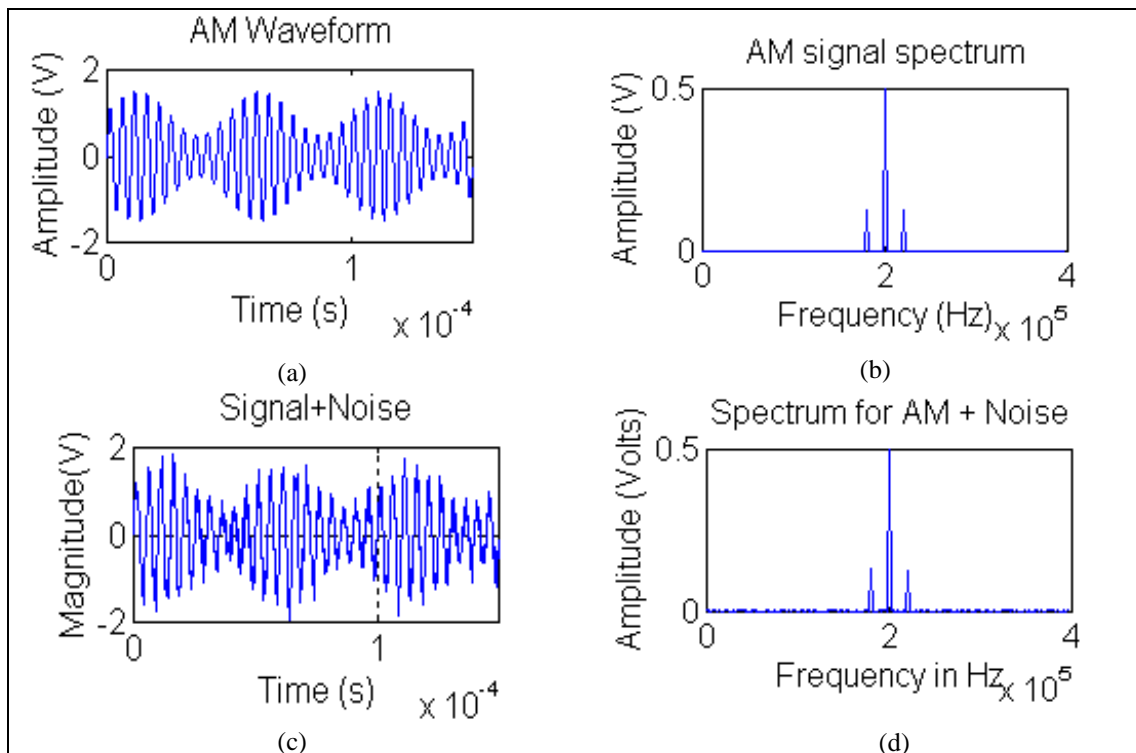


Fig. 3 AM and AM plus noise waveforms.

At the receiver a predetection bandlimiting filter is employed in order to reduce the noise power without effecting the signal, and the results are shown in Fig. 4.a and b.
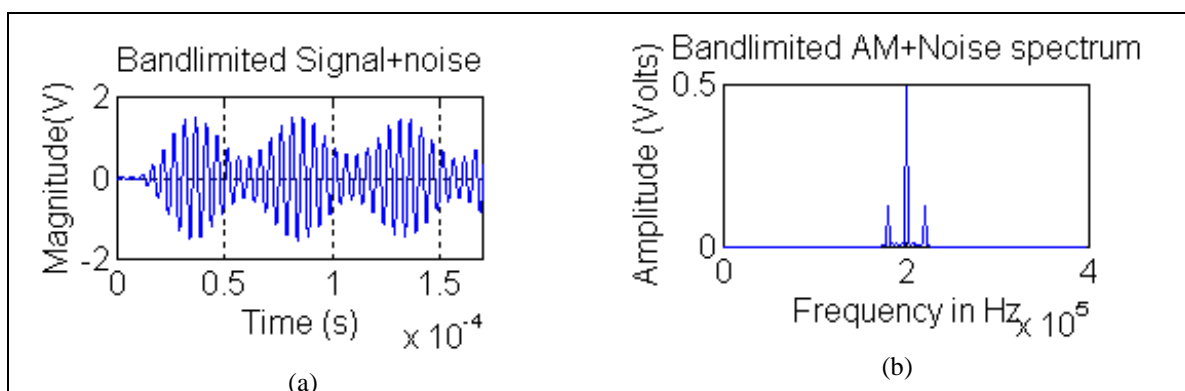


Fig. 4 Bandlimited AM plus noise waveforms.

*2.1.3 MATLAB codes for the AM demodulator:*

```
SQ=(bAMN).*bAMN; %Squaring the received AM + noise signal
SQfrq = fft(SQ);%now convert to frequency domain,
magsq = abs(SQfrq)./(final/ts);
%low pass filter;fco=2*fm;%cutoff frequency;
[b,a] = butter(4,(fco)*2/fs);
so=filter(b,a,SQ);
so=so*10;
so = so - mean(so);
tv=0.5e-4:(1/fs):(1/s)*(n-1);
sofrq=fft(so);%now convert to frequency domain,
magso=abs(sofrq)./(final*fs);
```

The waveform at the output of the squarer is shown in Fig. 5.a and its corresponding spectrum is illustrated in Fig. 5.b, showing the presence of the modulating signal at the baseband region, frequency component at twice the carrier frequency and a set of sidebands around the carrier components. The modulating signal recovered using a simple 4th order low pass filter, see Fig. 5.c and d. The distortion on the recovered signal, both in time and frequency domain waveforms, is due to addition of high noise voltage (i.e. low signal-to-noise ratio).
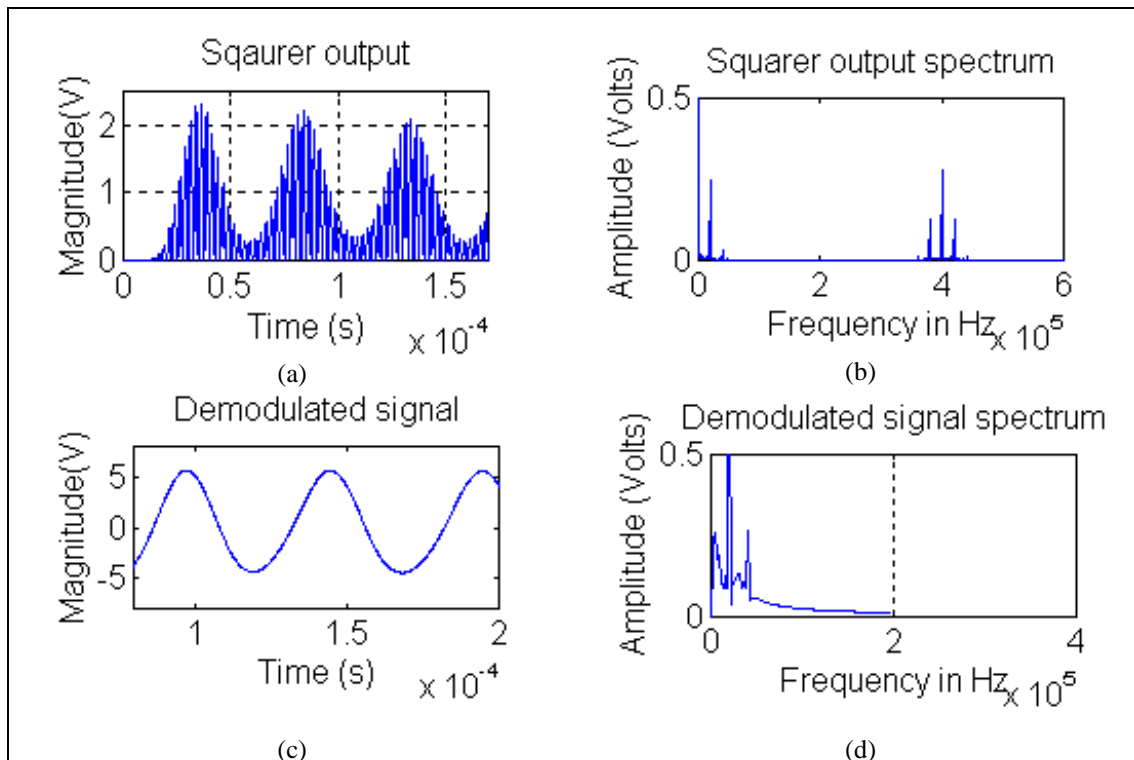


Fig. 5 Demodulated waveform:

## 2.2    Case Study 2: Pulse Width Modulation (PWM)

Digital communication systems can be implemented using a variety of different modulation schemes one of which is based on pulse time modulation (PTM) methods. In PTM schemes one of a range of time dependent features of a constant amplitude squarewave carrier waveform is used to convey information (see Table 1).

Table 1  PTM schemes.

| Possible schemes | Variable |
|---|---|
| Pulse width modulation (PWM) | Width (duration) |
| Pulse position modulation(PPM) | Position |
| Pulse interval modulation(PIM) | Interval (space) |
| Pulse interval and width modulation (PIWM) | Interval and width |
| Pulse frequency modulation (PFM) | Frequency |
| Squarewave FM (SWFM) | Frequency |

In this case-study a PTM scheme known as pulse width modulation (PWM) was selected. PWM is an attractive scheme for transmission of analogue and data signals compared with purely digital modulation techniques [3-6]. In PWM the width of a constant amplitude pulse carrier is changed according to the sample values of the modulating signal. A block diagram and waveforms associated with a simple PWM system are shown in Fig. 6.

At the transmitter the analogue signal and a sawtooth ramp are compared directly at the input of a comparator. The output of the comparator is the PWM pulse train with a constant amplitude and modulated duration. The duration of the $k$th plus is given as:

$$\tau_k = \tau_0[1 + Mm(kT_c)] \tag{5}$$

where $\tau_0$ is the unmodulated pulse width representing $Mm(T_c) = 0$, and $M$ is the modulation index $= 2\Delta\tau/T_c$, $T_c$ is the sampling interval, and $m(t) = \sin \omega_m t$ is the single tone modulating signal .

At the receiver, the PWM signal, contaminated with noise during transmission, is passed through a slicer in order to regenerate the original transmitted PWM pulse train. Finally, a second order lowpass filter is employed to recover the modulating signal located at the baseband region of the PWM frequency spectrum, see Fig. 7.b.
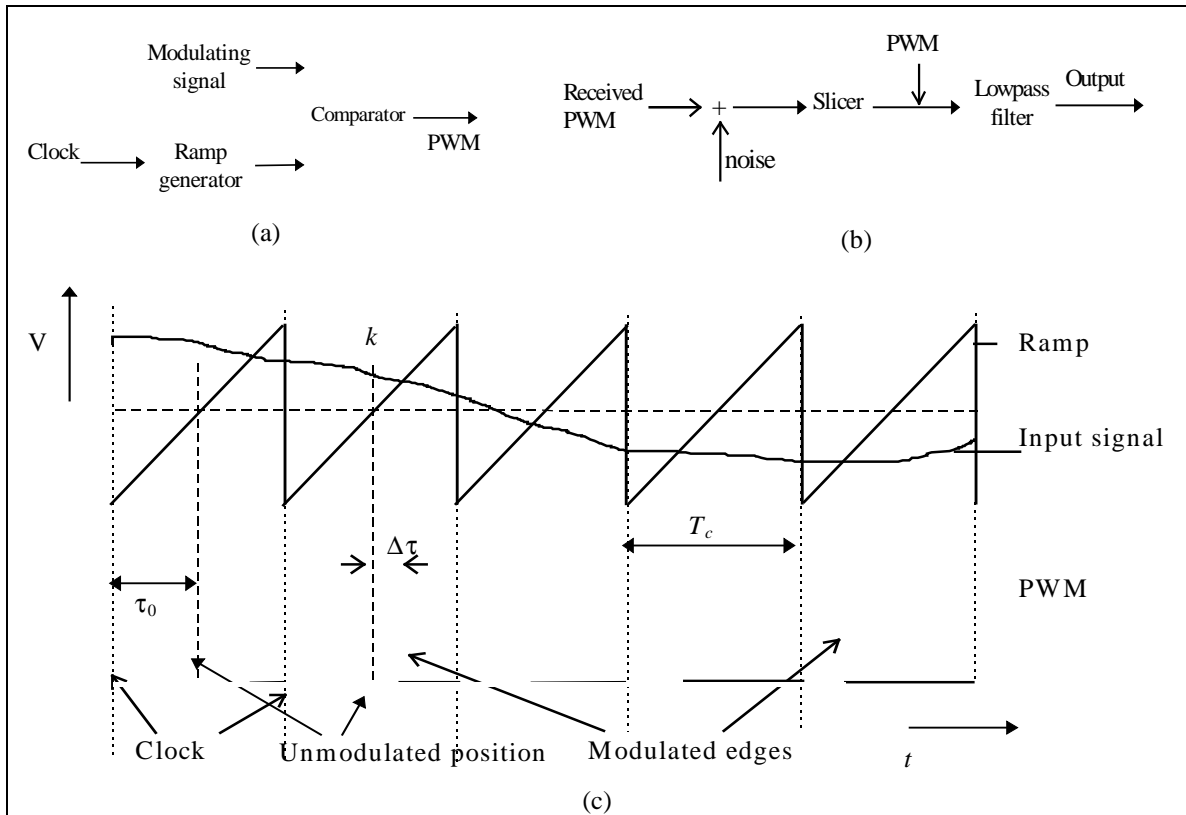
Fig. 6 PWM system: (a) transmitter, (b) receiver and (c) waveforms.

The essential Matlab codes for implementing PWM transmission systemare shown in the following sections. The codes with minor modifications can be applied to the other PTM schemes.

*2.2.1 PWM transmitter MATLAB codes:*

```
n=4096;% total number of points
fc= 200.0e+3;%carrier frequency,
fs = 20*fc;%sampling frequency, ts= 1/fs; %sampling interval
fm = 10.0e+3;%modulating frequency,
t = 0:(ts):( ts*(n-1)); %time vector,
fvect = 0:1/ (ts*(n-1)):1/(ts*2); %frequency axis vector
sig = 0.5*sin(t*fm*2*pi);  %modulating signal
t2 = rem(fc*(2*pi)*t,(2*pi));
tri = (((1.0*t2)/(pi))-vpp); %ramp waveform
pwmsig = tri >= sig;pwmsig1 = pwmsig - 0.5;%bipolar PWM signal
pwmfrq = fft(pwmsig);
mag = abs(pwmfrq)./( (ts*(n-1))/ts); % PWMfrequency spectrum
```

The frequencies used for the modulating signal $f_m$ and the ramp (carrier) signal $f_c$ were

30 kHz and 200 kHz, respectively. The trailing edge modulated PWM pulse train, the

ramp and modulating waveforms are shown in Fig. 7.a. The frequency spectrum of the PWM signal is displayed in Fig. 7.b, which contains the modulating signal, the carrier frequency and its harmonics and a set of side tones $f_c \pm f_m$ around the carrier and its harmonics.



(a)                                                                  (b)
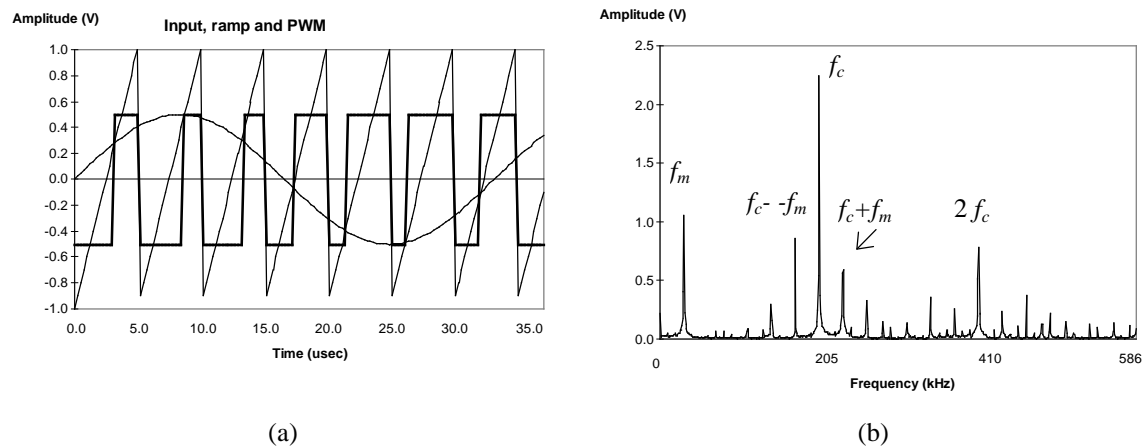
Fig. 7 (a) Transmitter waveforms and (b) PWM frequency spectrum.

To illustrate distortion during signal transmission a white Gaussian noise signal is added to the PWM signal. The noise is bandlimited and the channel is modelled as a simple first order low first order pass filter. The Matlab codes for implementing these are listed below.

### 2.2.2 Channel and noise Matlab codes:

```
noise=(0.1)*(randn(size(t)));%generating noise signal
[b,a]=butter(3,(4.0e+5)*2/fs);%generating filter (third order) polynom
bnoise=filter(b,a,noise);%bandlimitted noise signal
%adding PWM signal and noise
signoi=pwmsig1+noise;
%low pass filtering signal + noise
fnc=400.0e+3;%channel bandwidth (Hz)
[b1,a1]=butter(2,(fnc)*2/fs); %generating filter (second order) polynom
bsignoi=filter(b1,a1,signoi):
%PWM + noise frequency spectrum
SPsignoi=spectrum(signoi,n);specplot1(SPsignoi,fs),
```

For an infinite channel bandwidth, the PWM signal with noise and its regenerated version are shown in Fig. 8.a. A simulation was also carried out to show the effect of

bandlimiting the channel on transmitted PWM signal, and the results together with regenerated PWM waveform are shown in Fig. 8.b.
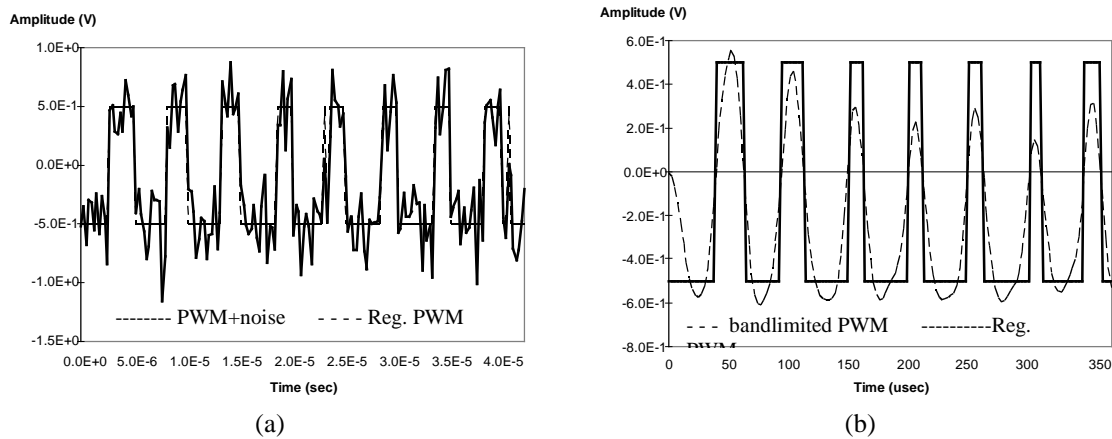


(a)                                                           (b)

Fig. 8 PWM waveforms (a) with noise and its regenerated, and (b) bandlimited and regenerated.

*2.2.3 PWM receiver Matlab codes:*

```
%Slicer
cax=[0 0.01 -0.5 0.5];
axis(cax);
slicer=sign(signoi);
slicer=(sign(slicer+0.1));
slicer=slicer./2;% regenerator PWM signal
pwmrfrq = fft(slicer);
magr = abs(pwmrfrq)./(n-1);% regenerated PWM frequency spectrum
%receiver low pass filter
foc=10.0e+3;%cut off frequency (Hz)
[b,a]=butter(2,(foc)*2/fs);%generating filter (second order) polynom
signal=filter(b,a,comppre);
signal=signal*10;% amplifying the recovered signal
%Recovered signal frequency spectrum
fv=0:1/(ts*(n-1)):1/(40*ts);%frequency vector
signalfrq = fft(signal);
mago = abs(signalfrq)./( (n-1)/ts);
```

The received PWM signal is regenerated using a slicer and demodulated using a second order low pass filter, see Fig. 9.a. Also shown for comparison is the input waveform. Reducing the order of the low pass filter results in also passing through high frequency components (lower side tones) superimposed on the desired recovered output signal as shown in Fig. 9.b.
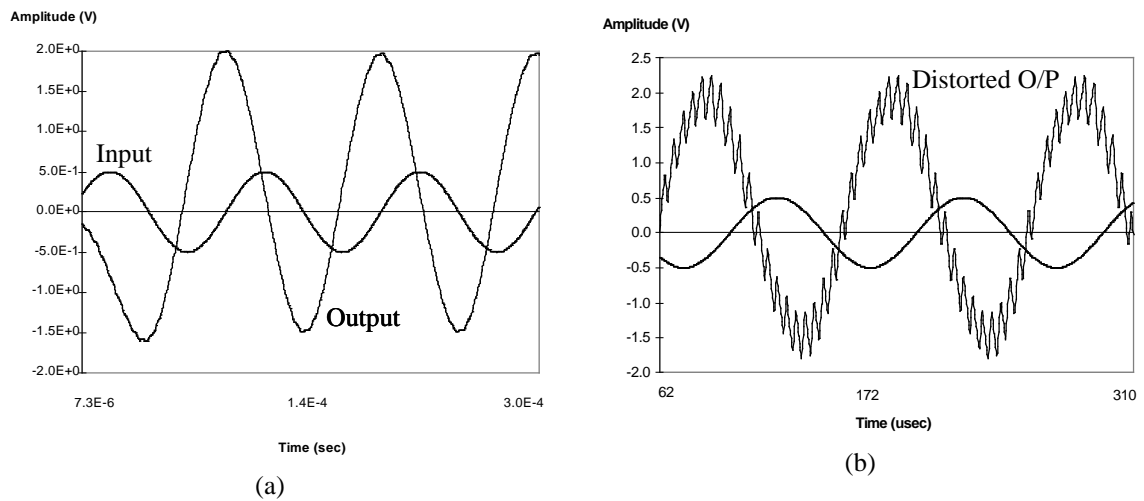
Fig. 9 (a) Input and output waveforms and (b) output waveform and its distorted version.

## Discussion

Computer simulation enables properties of systems to be investigated and complex concepts to be demonstrated. However, system simulation require thorough understanding of concepts. In situations were hardware development of systems can prove costly, system simulation provides an attractive alternative approach. The advent of fast and low cost personal computers gives simulation approach extra importance. System simulation for teaching purposes is very valuable as students can carryout the investigations without the constraints associated with hardware system development (for example cost, safety, etc.). Mistakes in setting up of a hardware may cause serious damage to a system while when simulating it the students can recover from the effect without any problem.

The following codes are used for ploting time and frequency domain waveforms.

```
plot (t(1:(tmax/2)),AM(1:(tmax/2))),
xlabel('Time (s)'), ylabel('Amplitude (V)'), title('')
plot(fvect(1:fmax),mag(1:fmax)),
xlabel('Frequency (Hz)'), ylabel('Amplitude (V)'), title(''),
```

## Conclusions

11

In this article a programming language called Matlab was used to illustrate how typical communication systems, both analogue and digital, can be simulated and analysed. The techniques presented are valuable for teaching communication engineering or related areas. The case-studies carried out clearly demonstrated the effectiveness of computer simulation for teaching complex concepts to undergraduate engineering students.

## Acknowledgements

## References

1. Matlab,: 'The MathWorks, Inc.' (1993).

2. Pärt-Enander, E., Sjöberrg, A., Melin, B., and Isaksson, P.,: 'The Matlab handbook' (Addison-Wesley,
1996).

3. Young, P.H.,: 'Electronic communication techniques' (Macmillan Publishing Company 1990, 3rd Edition, pp.137-174).

4. Arnold, J. M., and Berry, M.: 'Pulse width modulation for optical fibre transmission of video signals', IEE International Conference on impact of VLSI technology on communication systems, 1983, London.

5. Suh, S. Y.:"Pulse width modulation for analog fiber-optic communications", J. lightwave Technol., 1987, LT-5, pp. 102-112

6. Wilson, B., Ghassemlooy, Z.: "Optical pwm data link for high quality analogue and video signals', J. Physics E, 1987, 20, pp. 841-845