Aim ↬ To study the concept and fundamentals of Pulse Width Modulation and Demodulation.

Software Required ↬ MATLAB

Theory ↬

Pulse Width Modulation (PWM) is a technique where the width of a series of pulses is varied according to the instantaneous amplitude of the message signal. This modulation technique is widely used in applications such as motor control, light dimming, and communication systems. PWM can be viewed as a method for converting an analog signal into a digital signal by varying the duty cycle of the pulse train.

The modulated signal in PWM can be expressed as:

$$s(t) = \sum a_n \cdot p(t - nT_s)$$

where:

- a_n is the amplitude of the nth pulse,
- p(t) is the pulse shape,
- T_s is the sampling period.

In PWM, the width of each pulse is directly proportional to the instantaneous amplitude of the message signal. Unlike continuous modulation techniques like AM or FM, PWM deals with discrete pulse widths, making it effective for various applications, including signal processing and power delivery.

Generation of PWM Signal ↴

The PWM signal is generated by sampling the message signal at regular intervals and varying the width of the pulses according to the sampled values. The pulses can take various shapes, depending on system requirements. PWM can be classified as analog or digital based on the application.

Demodulation of PWM Signal ↴

PWM demodulation involves converting the variable-width pulses back into the original message signal. A low-pass filter can be used to smooth the PWM signal and reconstruct the continuous signal. Additionally, sample-and-hold circuits may be employed to hold the amplitude of incoming pulses to accurately recreate the original message signal.

Frequency Spectrum of PWM Signal ↴

The frequency spectrum of a PWM signal consists of a fundamental frequency component along with harmonics due to pulse width variation. The bandwidth of a PWM signal depends on the bandwidth of the message signal and the pulse width.

Applications of PWM ↴

PWM is used in motor control for speed regulation, LED lighting for dimming, and audio modulation in communication systems to ensure high-quality signal transmission.

Method ↴

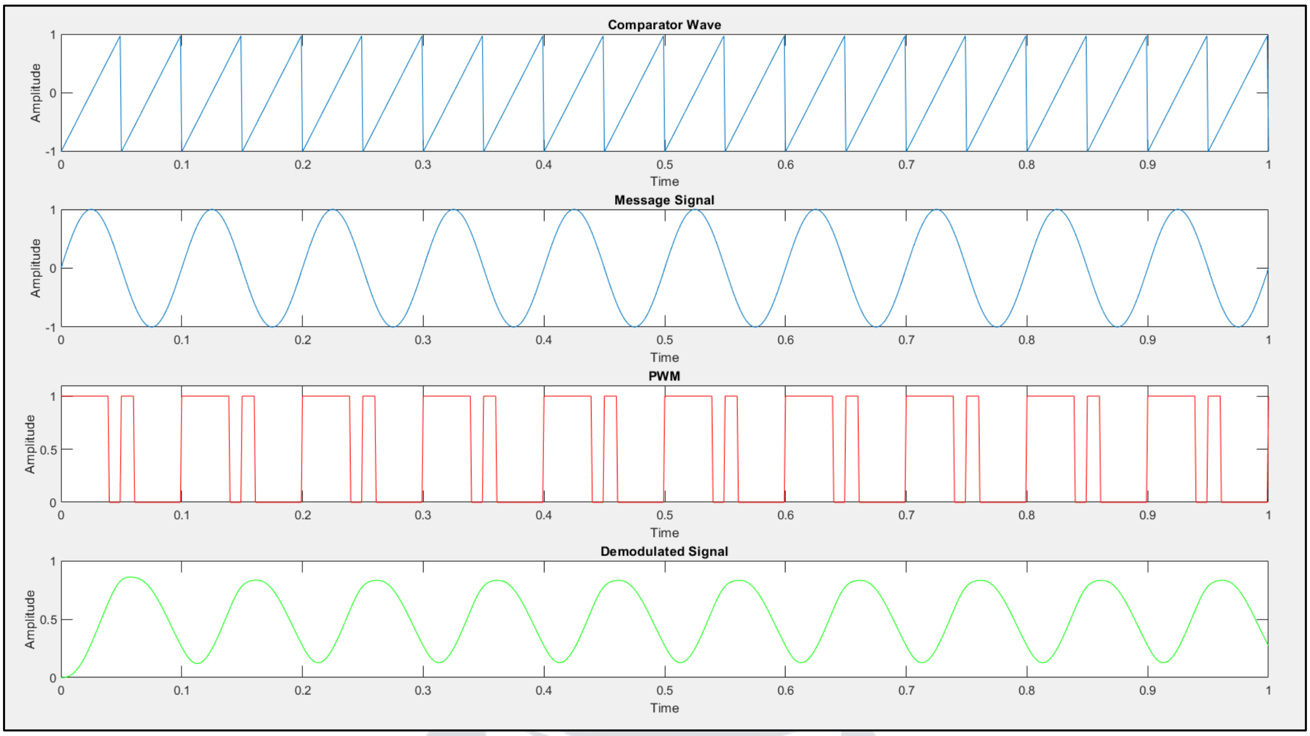
The code implements PWM by defining parameters for the message and comparator signals, generating a sawtooth wave as the comparator, and creating a sine wave for the message signal. The PWM waveform is formed by comparing the message signal values to the comparator signal and adjusting the pulse width accordingly. To recover the original message, a low-pass filter is applied to the modulated PWM signal, allowing visualization of the comparator wave, message signal, modulated PWM, and demodulated output.

Code ↬

```
% Pulse Width Modulation
close all;
clear all;
clc;

fs = 1000;
fm = 10;
a = 1;

t = 0:0.001:1;

stooth = 1.01*a.*sawtooth(2*pi*20*t);

msg = a.*sin(2*pi*fm*t);

pwm = zeros(1, length(stooth));
for i = 1:length(stooth)
   if (msg(i) >= stooth(i))
      pwm(i) = 1;
```

```
    else
        pwm(i) = 0;
    end
end

[b, a] = butter(3, fm / (fs / 2));
demodulated_pwm = filter(b, a, pwm);

figure;

subplot(4, 1, 1);
plot(t, stooth);
title('Comparator Wave');
xlabel('Time');
ylabel('Amplitude');

subplot(4, 1, 2);
plot(t, msg);
title('Message Signal');
xlabel('Time');
ylabel('Amplitude');

subplot(4, 1, 3);
plot(t, pwm, 'r');
title('PWM');
axis([0 1 0 1.1]);
xlabel('Time');
ylabel('Amplitude');

subplot(4, 1, 4);
plot(t, demodulated_pwm, 'g');
title('Demodulated Signal');
xlabel('Time');
ylabel('Amplitude');
```

Output ↬



Result ↬

The experiment demonstrated the generation and demodulation of a Pulse Width Modulation (PWM) signal. The message signal was sampled to create a pulse-width modulated signal, and a low-pass filter was used to recover the original signal.

Conclusion ↬

PWM is a robust modulation technique for transmitting data by varying pulse widths according to the message signal. The recovered signal accurately reflects the original message, demonstrating PWM's utility in various application.

Precautions ↬

- Ensure accurate sampling to avoid aliasing.
- Use an appropriate low-pass filter to recover the message signal without distortion.
- Select the correct sampling frequency for efficient modulation and demodulation.