Analog and Digital Modulation Techniques

Firas Dimashki

ECCE Department, Faculty of Engineering
Notre Dame University – Louaize, Deir El Qamar, Lebanon
Email: firasdimashki@gmail.com

***Abstract*— This research studies different analog and digital communications modulation techniques. It includes an implementation for AM, FM, PCM, BASK, and QPSK using MATLAB while analysing each technique according to different metrics.**

***Keywords*— Communication Systems, Modulation, AM, DSB-LC, FM, PCM, BASK, QPSK**

I. Introduction

Communication systems play a pivotal role in connecting individuals, devices, and networks across the globe. These systems facilitate the exchange of information, enabling seamless communication in various forms such as voice, data, and multimedia. The efficiency and reliability of communication systems are critical for modern society, impacting fields ranging from telecommunications and broadcasting to the internet and satellite communication.

Modulation, a fundamental concept in communication systems, is the process of varying a carrier signal's characteristics to encode information. This manipulation allows the transmission of information over a communication channel, overcoming challenges such as noise, interference, and signal degradation. Modulation is a key aspect of communication system design, influencing the system's performance, bandwidth utilization, and overall robustness. For most applications, communication, especially through wireless channels, would not be possible if not for the use of modulation technology, and the following points illustrate why.

Bandwidth Utilization: Modulation enables the efficient use of available bandwidth by allowing multiple signals to coexist without interference. Through techniques like frequency division multiplexing (FDM) and time division multiplexing (TDM), modulation contributes to optimizing the utilization of the communication medium.

Noise and Interference Mitigation: Communication channels are susceptible to various sources of noise and interference. Modulation techniques help in combatting these challenges by encoding information in a way that allows for the reconstruction of the original signal even in the presence of disturbances.

Reducing Antenna Size: As communication devices become smaller and more portable, the conventional size constraints of antennas pose challenges. Advanced techniques, such as metamaterial-based miniaturization, phased array technologies, and innovative antenna designs, are being researched and implemented to achieve reduced antenna sizes while maintaining or even enhancing performance.

Long-Distance Transmission: Modulation is crucial for long-distance communication, where attenuation and signal degradation can occur. By modulating signals onto higher frequency carrier waves, communication systems can transmit information over extended distances with minimal loss of signal quality.

Data Rate Enhancement: Modulation techniques influence the data rate at which information can be transmitted. Higher-order modulation schemes allow for increased data rates, supporting the growing demand for faster and more efficient communication services.

As technology advances and communication requirements evolve, the selection and implementation of appropriate modulation schemes become essential for designing robust and high-performance communication systems. In the context of the upcoming research, the exploration of both analog and digital modulation schemes will offer valuable insights into the practical aspects of modulation and its impact on system performance.

II. Amplitude Modulation

Amplitude Modulation (AM) stands as the most straightforward analog modulation technique for implementation. A signal is employed to gradually alter the amplitude of the carrier in accordance with the modulating signal's level. With AM, the carrier's amplitude modulation broadens the spectrum of the carrier, encompassing the original carrier component and upper (U) and lower (L) sidebands. The sidebands, carrying identical information, enable the conveyance of the entire baseband signal even if only one sideband is transmitted.

The basic AM signal $s(t)$, also known as the Double Side Band Large Carrier (DSB-LC) signal, takes the form:

$$s(t) = [1 + k_a m(t)]c(t)$$

Here k_a is the modulation index, which is used to prevent overmodulation, $m(t)$ is the baseband information-bearing signal with frequency components lower than the carrier frequency f_c, and $c(t)$ is the carrier signal such that:

$$c(t) = A_c \cos(2\pi f_c t)$$

The concept of the envelope is crucial in understanding the characteristics of modulated RF signals. The envelope represents the outline of the modulated carrier and is identical to the baseband signal in AM. The envelope's peak signifies maximum short-term power of the RF signal, particularly noticeable with 100% AM. Demodulating the basic AM signal involves bandpass filtering, rectifying, and lowpass filtering, with a simple diode sufficing for demodulation.

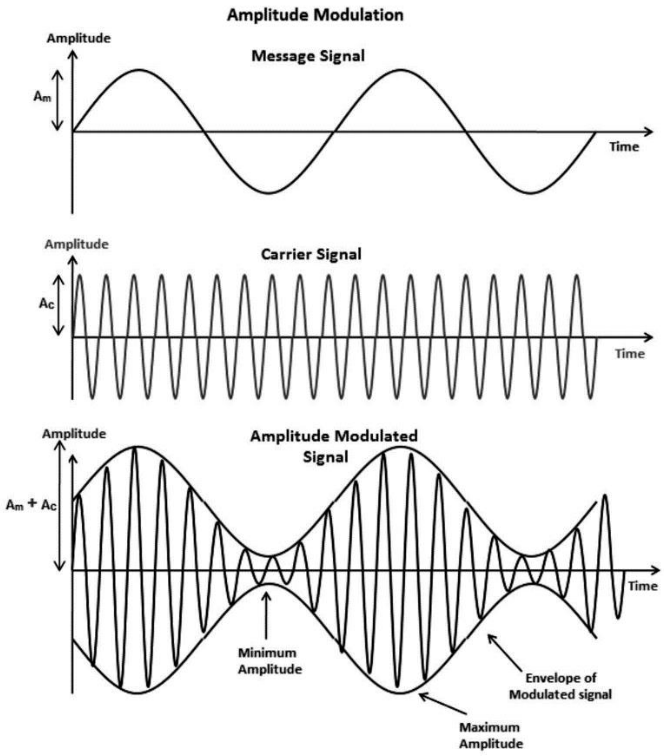

Fig. 1 Amplitude Modulation

Analysing modulation characteristics often involves simplifying baseband signals to one-tone or two-tone signals for ease of derivation. For instance, a single-tone baseband signal $m(t) = cos(\omega_m t + \phi)$ results in a modulated signal with three frequency components—carrier frequency, one just below, and one just above—each with bandwidth equivalent to the baseband signal. [1]

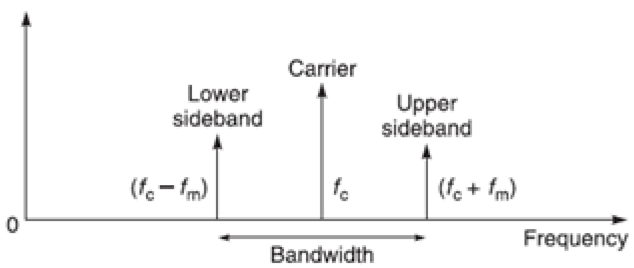

Fig. 2 AM Spectrum

Although the DSB-LC modulation and demodulation techniques are very simple, this technique has many disadvantages. For instance, the transmitted modulated signal is vulnerable to noise interference and has double the bandwidth of the message signal, and the carrier signal is transmitted along with the baseband signal, which requires most of the transmission power. In fact, the maximum efficiency of the DSB-LC is about 67%. For these reasons, other amplitude modulation techniques have been developed, such as the Double Side Band Suppressed Carrier (DSB-SC) modulation technique which solves the power issue, and the Single Side Band (SSB) modulation technique which solves both the power and bandwidth issues.

III. Frequency Modulation

Another commonly used analog modulation scheme is Frequency Modulation (FM). In FM, the frequency of the modulated carrier is determined by the amplitude of the baseband signal.

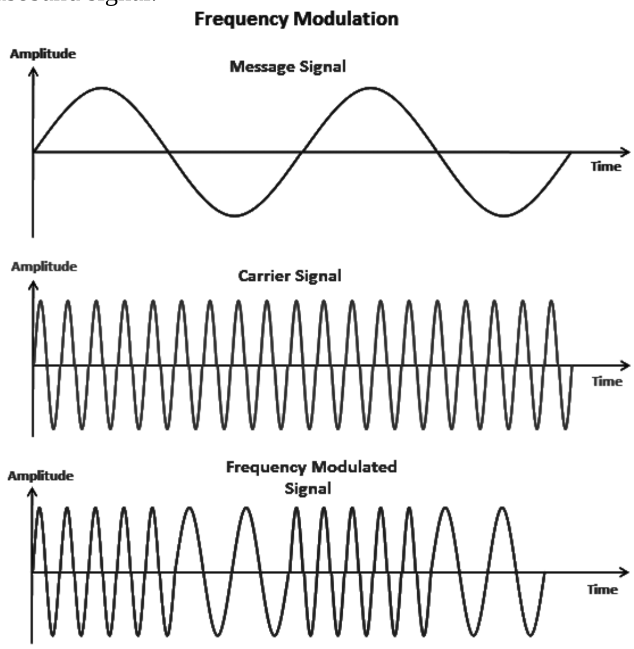

Fig. 3 Frequency Modulation

Examining the FM waveform in figure above, when the baseband signal reaches its peak value, the modulated carrier operates at its maximum frequency, and vice versa. Consequently, the time-varying signal exhibits a spread-out bandwidth. [1] The idea behind this modulation technique can be illustrated by the following formulae:

$$s(t) = A \cdot \cos\,(2\pi f_i(t) \cdot t)$$

$$fi(t) = f_c + k_f m(t)$$

$$\rightarrow s(t) = A \cdot \cos\,(2\pi f_c t + 2\pi k_f m(t) \cdot t)$$

$$\rightarrow s(t) = A \cdot \cos\left(2\pi f_c t + 2\pi k_f \int_0^t m(\tau) d\tau\right)$$

Here, $s(t)$ represents the modulated signal, $m(t)$ represents the baseband signal, f_c represents the carrier frequency, and k_f represents the frequency sensitivity.

One approach to implementing an FM modulator is to utilize a voltage-controlled oscillator (VCO) with the baseband signal controlling the frequency of the oscillator. Another approach implies using a Varactor modulator circuit, which uses a Varactor diode to map the baseband amplitude into capacitance, which can be used to change the oscillation frequency of the circuit.

To reconstruct the original narrower bandwidth baseband signal, an FM receiver compresses the transmitted signal in frequency. FM demodulation can be conceptualized as providing signal enhancement, or equivalently, noise suppression, introducing a process akin to analog processing gain. During demodulation, only the components of the original FM signals are coherently collapsed to a narrower bandwidth baseband signal, while noise, being uncorrelated, is still spread out (though rearranged). Consequently, the signal-to-noise power ratio increases, as after demodulation, only the power of the noise in the smaller bandwidth of the baseband signal is significant. Therefore, compared to AM, FM significantly enhances tolerance to noise that may be introduced during signal transmission. [1]

In an effort to study the bandwidth of the modulated signal, we can use Carson's rule, which relates the modulation frequency to the baseband frequency and the frequency variation:

$$BW \cong 2(f_m + \Delta f)$$

$$\Delta f = k_f |m(t)|$$

IV. Pulse Code Modulation

Pulse Code Modulation (PCM) is a digital modulation technique used in telecommunications and audio signal processing. It is employed for the digital representation of analog signals by converting continuous-time analog signals into discrete-time digital signals. PCM is widely used in various applications, including voice communication, audio recording, and digital transmission systems. It consists of three stages:

Sampling: The first step in PCM involves the sampling of the continuous-time analog signal. The analog signal is sampled at regular intervals in time, resulting in a series of discrete samples.

Quantization: The sampled amplitude values are then quantized, meaning that each sample is assigned a discrete numerical value. The quantization process involves mapping the continuous amplitude range of the analog signal into a finite set of discrete levels or codes. The number of bits used for quantization, known as the quantization level, determines the precision of the representation. Higher quantization levels allow for more accurate representation but require more bits per sample.

Encoding: The quantized values are then encoded into binary code, usually represented in the form of binary numbers. Each quantized sample is assigned a unique binary code, and the length of the binary code is determined by the number of quantization levels.

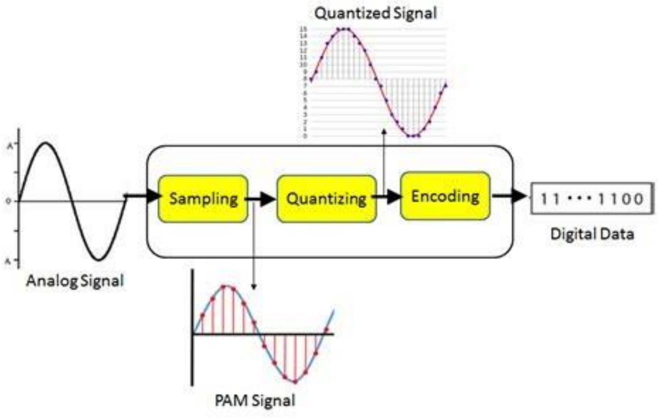

Fig. 4 Pulse Code Modulation

The output digital data can have multiple forms, two of which are the on-off signaling and nonreturn-to-zero (NRZ) signaling.

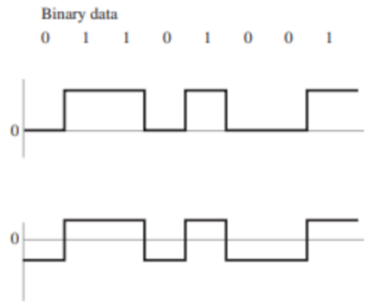

Fig. 4 On-Off Signaling vs Nonreturn-to-Zero Signaling

Pulse code modulation is in itself the first stage of shift keying bandpass modulation techniques, which are used to transmit digital data.

V. Binary Amplitude Shift Keying

Binary Amplitude Shift Keying (BASK) is a digital modulation scheme where the amplitude of a carrier signal is varied between two levels to convey digital information. BASK is a type of amplitude shift keying (ASK) that specifically deals with binary digital signals. It is a straightforward modulation technique, commonly used in various communication systems, including simple wireless communication and certain types of digital data transmission.

In general, the BASK modulation process is very simple and can be illustrated as follows:

$$s(t) = \begin{cases} A_0 \cdot \cos(2\pi f_c t) \text{ for binary 1} \\ A_1 \cdot \cos(2\pi f_c t) \text{ for binary 0} \end{cases}$$

Here, A_0 and A_1 are constants (A_1 is sometimes considered 0) and f_c is the carrier frequency. The BASK modulation technique, though very simple, has many shortcomings. For instance, it only transmits one bit every period T_b. In addition, it is very vulnerable to noise which can lead to bit errors during transmission.

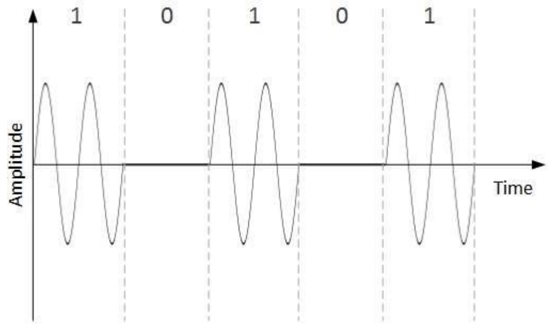

Fig. 5 BASK Modulation

VI. Quadra-Phase Shift Keying

Quadrature Phase Shift Keying (QPSK) is a digital modulation scheme that extends the concept of Binary Phase Shift Keying (BPSK) to transmit two bits per symbol instead of one. QPSK is widely used in digital communication systems, particularly in scenarios where bandwidth efficiency is critical. The baseband bitstream entering the system modulates the phase of the carrier, generating the modulated signal. However, the sharp phase changes in the resulting modulated waveform contribute to excessive bandwidth. Nevertheless, an effective phase shift keying modulator addresses this issue by initially applying a lowpass filter to the binary data before modulating the carrier. This filtering process eliminates the abrupt phase changes in the modulated signal, thereby minimizing the necessary bandwidth.

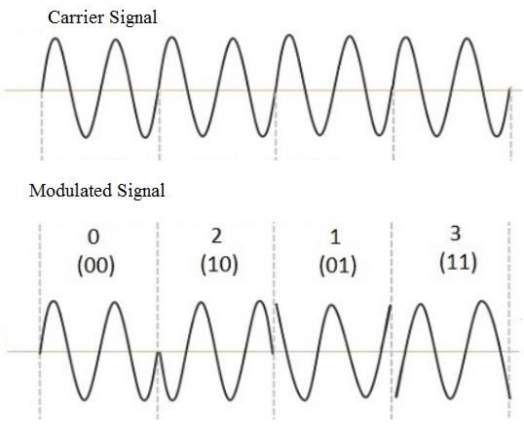

Fig. 6 QPSK Modulation

QPSK modulation can be illustrated using the following formula:

$$s_i(t) = A \cdot \cos\left(2\pi f_c t + (2i-1)\frac{\pi}{4}\right), \qquad i = 1,2,3,4$$

Here, A is a constant, f_c is the carrier frequency, and i stands for the number assigned to the di-bit that is being transmitted. The di-bits can be mapped to values of i as follows (other mappings can suffice):

$$10 \rightarrow 1$$
$$00 \rightarrow 2$$
$$01 \rightarrow 3$$
$$11 \rightarrow 4$$

As such, the following constellation diagram can represent this modulation scheme:

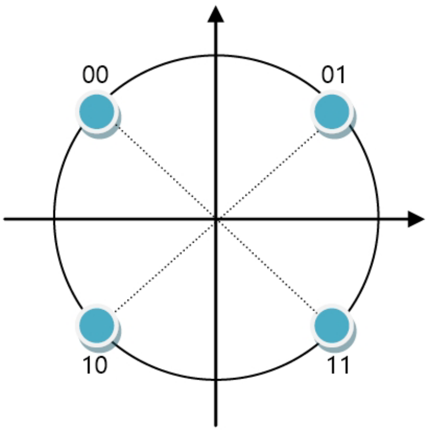

Fig. 7 QPSK Constellation Diagram

VII. MATLAB Simulation

Each of the modulation schemes that were explained previously was implemented using MATLAB and tested on single tones and on different audio files (by changing a few lines of the code). No noise was applied to the transmission channels. The carrier frequency used for all implementations is 40 kHz. In reality, a higher frequency would be selected, but for simulations, increasing this frequency results in very large system memory requirements.

AM was implemented for a single tone as follows:

```
clc
clear

%% Modulation Parameters

% Carrier amplitude
Ac = 1;
% Carrier frequency
fc = 40000; % 40 kHz
% Message tone amplitude
Am = 1;
% Message tone frequency
fm = 1000;
% Modulation index
ka = 0.75 / Am;
% Simulation sampling frequency
fs = 8 * fc;

%% Modulation

% Creating message signal for tone
t = 0 : 1 / fs : 0.005;
message = Am * cos(2*pi*fm*t);

% Amplitude Modulation
carrier = Ac * cos(2*pi*fc*t);
modulated = (1 + ka * message) .* carrier;

% Plotting message, carrier, and modulated signals
figure(1)
subplot(3, 1, 1);
plot(t, message);
```

```
title("Message signal")
ylabel("m(t)")
subplot(3, 1, 2);
plot(t, carrier);
title("Carrier signal")
ylabel("c(t)")
subplot(3, 1, 3)
plot(t, modulated);
title("Modulated signal")
ylabel("s(t)")
xlabel ("time (s)")

%% Demodulation (Envelope Detector)

% Rectification
x1 = abs(modulated);

% Low pass filter
f0 = 20000;
[bl, al] = butter(5, f0 / (fs / 2), "low");
x2 = filter(bl, al, x1);

% Remove DC component
x3 = detrend(x2);

% Restore amplitude
output = x3 * 2;

% Plotting modulated, rectified, and demodulated signals
figure(2)
subplot(3, 1, 1)
plot(t, modulated)
title("Modulated signal")
ylabel("s(t)")
subplot(3, 1, 2)
plot(t, x1)
title("Rectified signal")
subplot(3, 1, 3)
plot(t, output)
title("Demodulated signal")
ylabel("m'(t)")
xlabel("time (s)")
```

FM was implemented for a single tone as follows:

```
clc
clear

%% Modulation parameters

% Carrier amplitude
Ac = 1;
% Carrier frequency
fc = 40000; % 40 kHz
% Message tone amplitude
Am = 1;
% Message tone frequency
fm = 1000;
% Frequency sensitivity
kf = 10000; % 100 Hz/V
% Simulation sampling frequency
fs = 8 * fc;

%% Modulation

% Creating message signal for tone
t = 0 : 1 / fs : 0.005;
message = Am * cos(2*pi*fm*t);
% Frequency modulation
carrier = cos(2*pi*fc*t);
integralm = cumsum(message) / fs;
modulated = Ac * cos(2*pi*fc*t + 2*pi*kf*integralm);

% Plotting message, carrier, and modualted signals
figure(1)
subplot(3, 1, 1);
plot(t, message);
title("Message signal")
ylabel("m(t)")
subplot(3, 1, 2);
plot(t, carrier);
title("Carrier signal")
ylabel("c(t)")
subplot(3, 1, 3)
plot(t, modulated);
title("Modulated signal")
ylabel("s(t)")
xlabel("time (s)")

%% Demodulation

% Obtaining instantaneous frequency
x = hilbert(modulated);
% unwrapping angle for differentiation
phase = unwrap(angle(x));
freq = [0, diff(phase)];

% Amplifying and normalizing
normalized = fc/kf * freq;
normalized = detrend(normalized);
normalized(normalized > 1) = 1;
normalized(normalized < -1) = -1;

% Removing DC component
output = normalized;

% Plotting modulated and demodulated signals
figure(2)
subplot(2, 1, 1)
plot(t, modulated);
title("Modulated signal")
ylabel("s(t)")
subplot(2, 1, 2)
plot(t, output);
title("Demodulated signal")
ylabel("m'(t)")
xlabel("time (s)")
```

BASK was implemented for a single tone as follows:

```
clc
clear

%% Modulation parameters

% Sampler frequency
fsampler = 200; % 20 kHz
% Quantizing bits
b = 6;
% Bit period
Tb = 0.0001; % 100 microseconds
% Carrier frequency
fc = 40000; % 40 kHz
% Message tone amplitude
Am = 1;
% Message tone frequency
fm = 1000;
```

```
% Simulation sampling frequency
fs = 8 * fc;

%% Pulse code modulation

% Creating message signal for tone
txlimit = 0.005;
t1 = 0 : 1 / fs : txlimit;
message = Am * cos(2*pi*fm*t1);

% Sampling
sampled = resample(message, fsampler, fm);
t2 = (0 : 1 / fsampler : (length(sampled) - 1) /
fsampler) * fm / fs;

% Quantizing
quantized = floor((0.99 * normalize(sampled, "range")) *
2 ^ b);

% Encoding
encoded = split(strjoin(string(dec2bin(quantized, b)),
""), "");
encoded = encoded(2 : length(encoded) - 1);
encoded = transpose(str2double(encoded));

% Plot audio, sampled audio, and quantized audio
figure(1)
subplot(3, 1, 1)
plot(t1, message)
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Message signal")
ylabel("m(t)")
subplot(3, 1, 2)
stem(t2, sampled, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Sampled signal")
subplot(3, 1, 3)
stem(t2, quantized, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Quantized signal")
xlabel("time (s)")

% Clearing memory for better performance
clear sampled
clear quantized

%% Amplitude shift keying

t3 = 0 : 1 / fs : length(encoded) * Tb;
t3 = t3(1:length(t3)-1);

% Serializing encoded signal (on-off signaling)
serial = resample(encoded, fs, ceil(1 / Tb));
serial(serial < 0.5) = 0;
serial(serial >= 0.5) = 1;

% Modulating the serialized signal
carrier = cos(2*pi*fc*t3);
modulated = serial .* carrier;

% Plotting serialized signal and modulated signal
figure(2)
subplot(2, 1, 1)
plot(t3, serial)
title("Serialized signal")
subplot(2, 1, 2)
plot(t3, modulated)
title("Modulated signal")
ylabel("s(t)")
xlabel("time (s)")
```

```
% Clearing memory for better performance
clear encoded
clear carrier
clear serial

%% Demodulation

% Envelope detector
x1 = abs(modulated);

% Low pass filter and normalizer
f02 = 1 / Tb;
[bl, al] = butter(1, f02 / (fs / 2), "low");
x1 = filter(bl, al, x1);
x1 = normalize(x1, "range");

% Sampling
x2 = x1(fs * Tb / 4 : fs * Tb : length(x1));

% Decision making
decoded = x2;
decoded(decoded < 0.5) = 0;
decoded(decoded >= 0.5) = 1;

% Plotting modulated signal and output of envelope
detector and sampler
figure(3)
subplot(2, 1, 1)
plot(t3, modulated);
title("Modulated signal")
ylabel("s(t)")
subplot(2, 1, 2)
plot(t3, x1);
title("Detected signal")
xlabel("time (s)")

% Clearing memory for better performance
clear modulated
clear x1
clear x2

% Deserializing
deserialized = string(decoded);
deserialized = [deserialized{:}];
deserialized = mat2cell(deserialized, 1, b * ones(1,
length(deserialized) / b));

% Deserializing decoded signal
deserialized = transpose(bin2dec(deserialized));

% Dequantizing deserialized signal to obtain original
message
output = (deserialized - (2 ^ (b-1) - 0.5)) / (2 ^ (b-
1));

% Plotting decoded signal, demodulated signal, and
original message
figure(4)
subplot(2, 1, 1)
stem(t2, deserialized, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Deserialized signal")
subplot(2, 1, 2)
plot(t2, output, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Demodulated signal")
ylabel("m'(t)")
xlabel("time (s)")
```

QPSK was implemented for a single tone as follows:

```
clc
clear

%% Modulation parameters

% Sampler frequency
fsampler = 200;
% Quantizing bits
b = 6;
% Bit period
Tb = 0.0001; % 100 microseconds
% Carrier frequency
fc = 40000; % 40 kHz
% Message tone amplitude
Am = 1;
% Message tone frequency
fm = 1000;
% Simulation sampling frequency
fs = 8 * fc;

%% Pulse code modulation

% Creating message signal for audio
txlimit = 0.005;
t1 = 0 : 1 / fs : txlimit;
message = Am * cos(2*pi*fm*t1);

% Sampling
sampled = resample(message, fsampler, fm);
t2 = (0 : 1 / fsampler : (length(sampled) - 1) /
fsampler) * fm / fs;

% Quantizing
quantized = floor((0.99 * normalize(sampled, "range")) *
2 ^ b);

% Encoding
encoded = split(strjoin(string(dec2bin(quantized, b)),
""), "");
encoded = encoded(2 : length(encoded) - 1); % remove
empty strings
encoded = transpose(str2double(encoded));

% Plot audio, sampled audio, and quantized audio
figure(1)
subplot(3, 1, 1)
plot(t1, message)
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Message signal")
ylabel("m(t)")
subplot(3, 1, 2)
stem(t2, sampled, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Sampled signal")
subplot(3, 1, 3)
stem(t2, quantized, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Quantized signal")
xlabel("time (s)")

% Clearing memory for better performance
clear sampled
clear quantized

%% QuadriPhase shift keying

% Symbolizing encoded signal
symbolized = reshape(encoded, [2, length(encoded) / 2]);
```

```
symbolized = string(symbolized);
symbolized = symbolized(1, :) + symbolized(2, :);
symbolized = bin2dec(symbolized);

% Changing the map causes distortion
map = [2, 3, 1, 4];
symbolized = map(symbolized + 1);

T = 2 * Tb;
t3 = 0 : 1 / fs : length(symbolized) * T;
t3 = t3(1:length(t3)-1);

% Serializing symbolized signal (non-return to zero)
serial = round(resample(symbolized, fs, ceil(1 / T), 0));

% Modulating the serialized signal
carrier = cos(2*pi*fc*t3);
% Note: filter bitstream first
modulated = cos(2*pi*fc*t3 + (2*serial - 1) * pi/4);

% Plotting serialized signal and modulated signal
figure(2)
subplot(3, 1, 1)
plot(t3, serial)
title("Serialized signal")
subplot(3, 1, 2)
plot(t3, carrier)
title("Carrier signal")
ylabel("c(t)")
subplot(3, 1, 3)
plot(t3, modulated)
title("Modulated signal")
ylabel("s(t)")
xlabel("time (s)")

% Clearing memory for better performance
clear encoded
clear serial

%% Demodulation

% Coherent detector
detected1 = modulated .* carrier;
detected2 = modulated .* imag(hilbert(carrier));

% Low pass filter and normalizer
f0 = 1 / T;
[bl, al] = butter(1, f0 / (fs / 2), "low");
filtered1 = filter(bl, al, detected1);
filtered1 = normalize(filtered1, "range");
filtered2 = filter(bl, al, detected2);
filtered2 = normalize(filtered2, "range");

% Sampling
sampled1 = filtered1(fs * T / 4 : fs * T :
length(filtered1));
sampled2 = filtered2(fs * T / 4 : fs * T :
length(filtered2));

% Decision making
decoded1 = sampled1;
decoded1(decoded1 < 0.5) = 0;
decoded1(decoded1 >= 0.5) = 1;
decoded2 = sampled2;
decoded2(decoded2 < 0.5) = 0;
decoded2(decoded2 >= 0.5) = 1;

% Plotting modulated signal and filtered output of
coherent detector
figure(3)
subplot(3, 1, 1)
```

```
plot(t3, modulated);
title("Modulated signal")
ylabel("s(t)")
subplot(3, 1, 2)
plot(t3, filtered1);
title("First sampled signal")
subplot(3, 1, 3)
plot(t3, filtered2);
title("Second sampled signal")
xlabel("time (s)")

% Clearing memory for better performance
clear carrier
clear modulated
clear detected1
clear detected2
clear filtered1
clear filtered2
clear sampled1
clear sampled2

% Combining and deserializing decoded signals
deserialized = string(decoded1) + string(decoded2);
deserialized = [deserialized{:}];
deserialized = mat2cell(deserialized, 1, b * ones(1,
length(deserialized) / b));
deserialized = transpose(bin2dec(deserialized));

% Dequantizing deserialized signal to obtain original
message
output = (deserialized - (2 ^ (b-1) - 0.5)) / (2 ^ (b-
1));

% Plotting decoded signal and demodulated signal
figure(4)
subplot(2, 1, 1)
stem(t2, deserialized, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Combined deserialized signal")
subplot(2, 1, 2)
plot(t2, output, 'Marker', 'None')
xlim([-0.01 * txlimit, 1.01 * txlimit])
title("Demodulated signal")
ylabel("m'(t)")
xlabel("time (s)")
```

VIII. MATLAB Simulation Results

The AM code produces two plots:

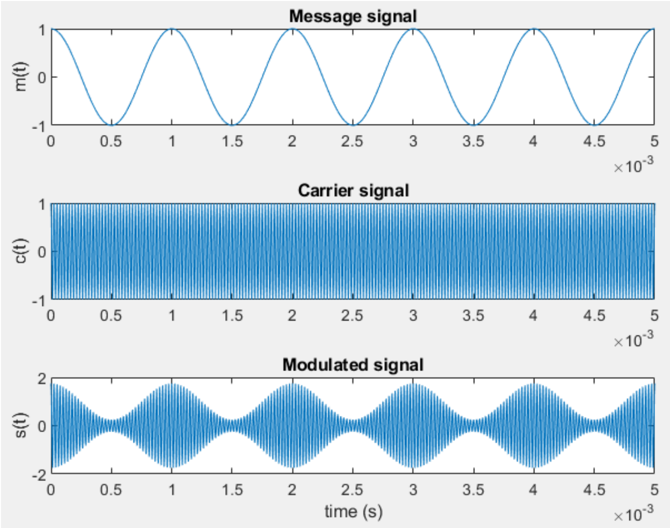

Fig. 8 MATLAB AM Modulation

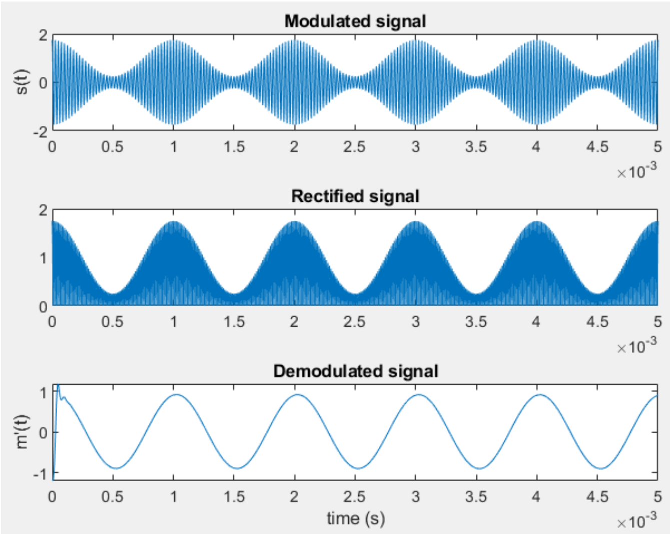

Fig. 9 MATLAB AM Demodulation

As seen in the figures, the modulated signal represents a modulation of the carrier signal according to the amplitude of the message signal. The envelope of the modulated signal is identical to the message signal. When the signal is demodulated, it is first rectified then filtered with a low pass filter to obtain the original frequency. Finally, the DC offset is removed, and the original signal is obtained. This modulation scheme is one of the simplest, but as stated previously, it has power and bandwidth issues, and that can be shown in the following spectrum plot of the modulated signal:

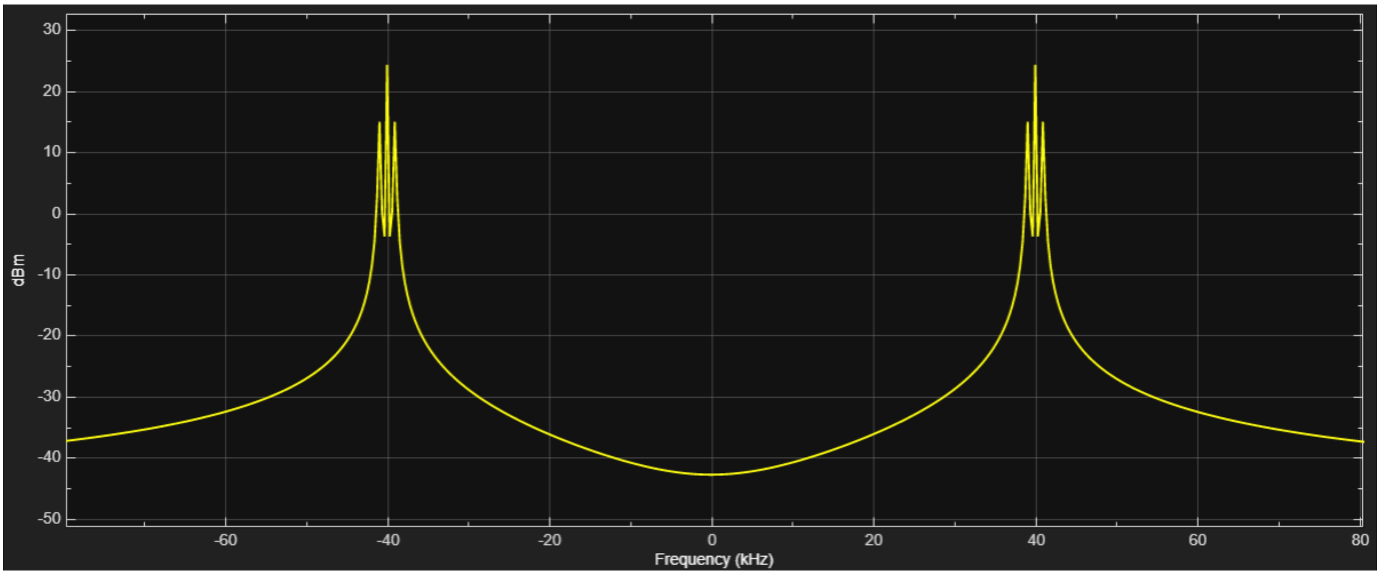

Fig. 10 MATLAB AM Spectrum

As seen in the output of the spectrum analyser, the middle spike in the positive and negative frequencies represents the carrier, which is transmitted alongside the message, which means more power is needed for transmission with no implication for the actual message. Moreover, both the upper and lower side bands are transmitted, which means double the bandwidth of the message signal is needed for transmission.

After verifying that the AM modulation scheme works with single tones, it was tested with an audio file. The following plots show the results. Note that because of the high carrier frequency and the large number of samples, some signals might not be completely visible in the plotting window.

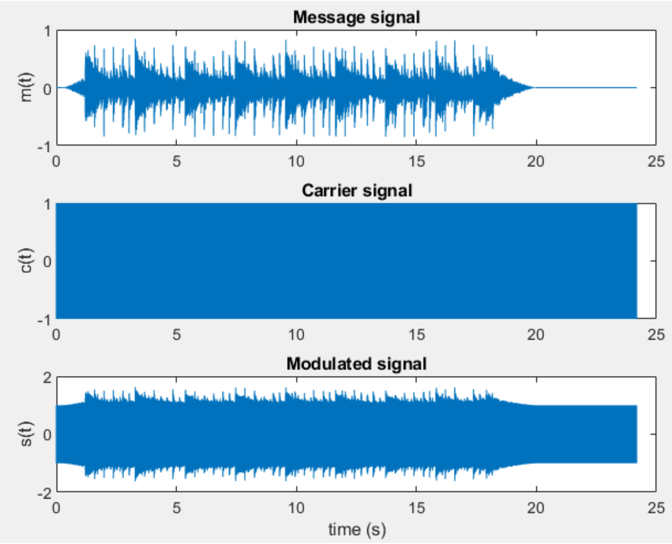

Fig. 11 MATLAB AM Audio Modulation

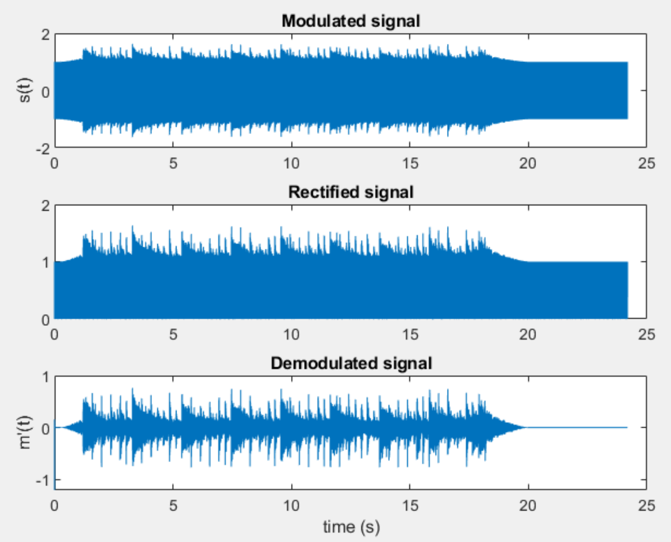

Fig. 12 MATLAB AM Audio Demodulation

It is clear that this AM system works well with audio signals as well, since the demodulated signal is very similar to the original message signal.

Next, the FM system is tested with a single tone, and the results are shown in the following figures:

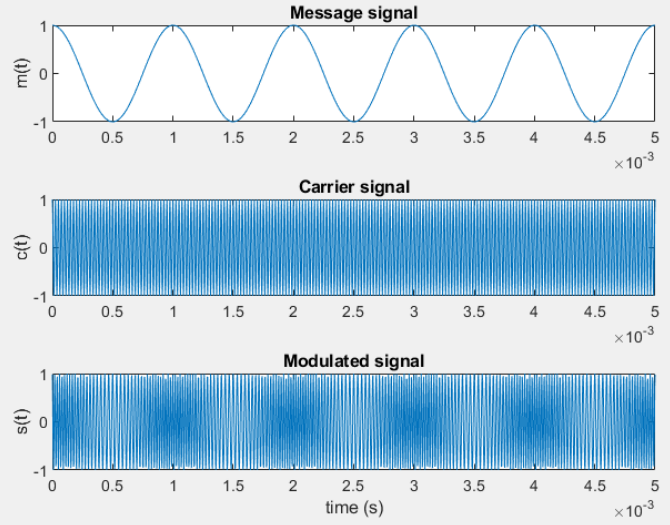

Fig. 13 MATLAB FM Modulation

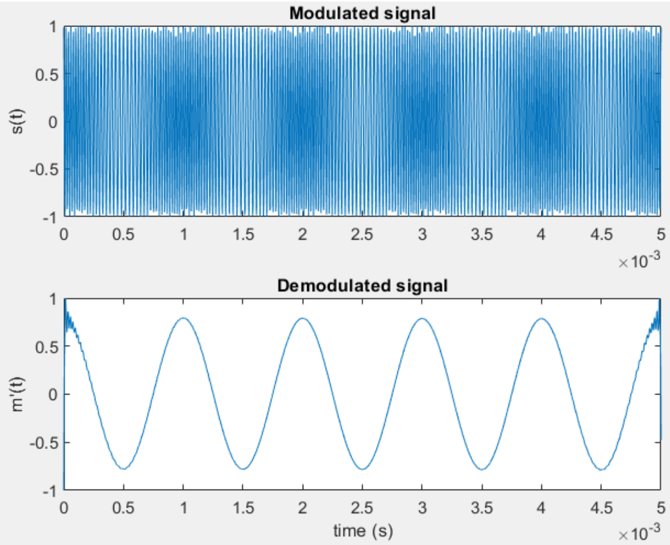

Fig. 14 MATLAB FM Demodulation

As seen in the plots, the higher the amplitude of the message signal, the higher the frequency of the modulated signal, and the original signal is recovered well after demodulation, with small distortions towards the boundaries of the signal. The same system is tested with an audio file:

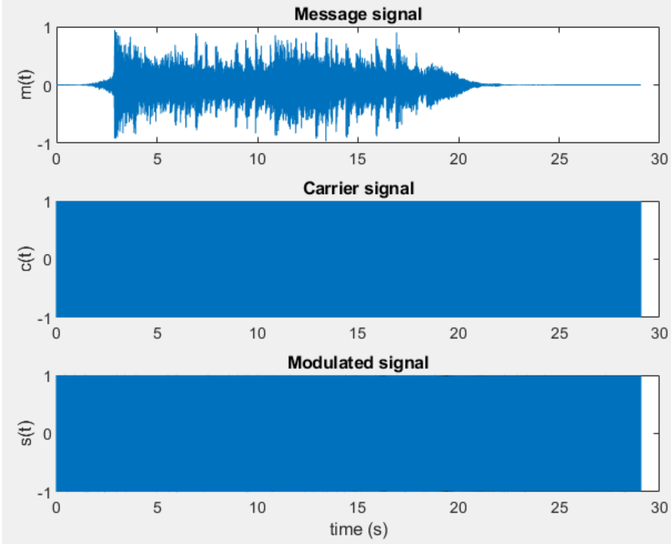

Fig. 14 MATLAB FM Audio Modulation

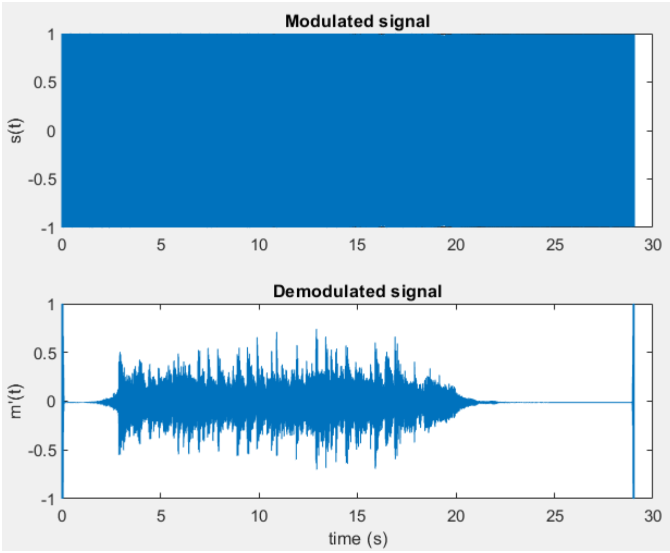

Fig. 15 MATLAB FM Audio Demodulation

The FM system works just as well for audio files.

Next, The BASK system is tested for single tones. Here, because every sample is converted to a 6-bit number, this means that for every sample, there are 6 bit periods in the serialized and modulated signals. For this reason, the actual signals may not be completely visible.

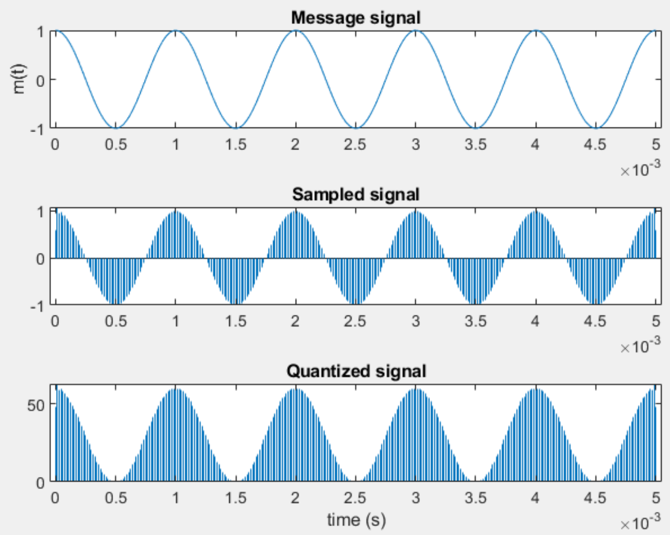

Fig. 16 MATLAB BASK Sampling and Quantization

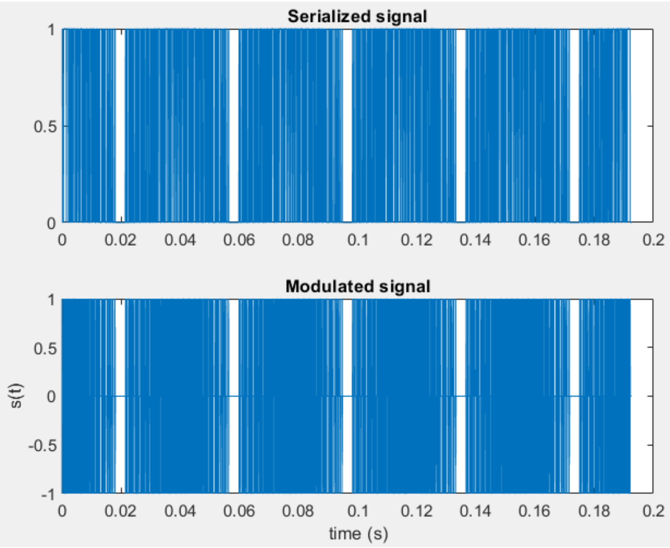

Fig. 17 MATLAB BASK Modulation

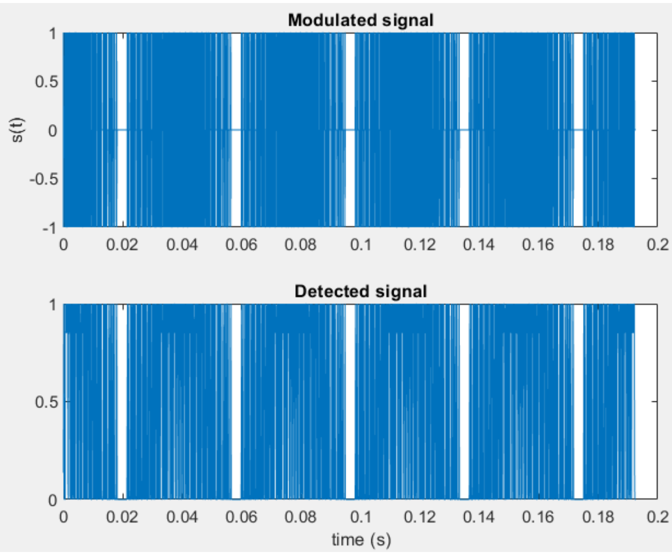

Fig. 18 MATLAB BASK Signal Detection

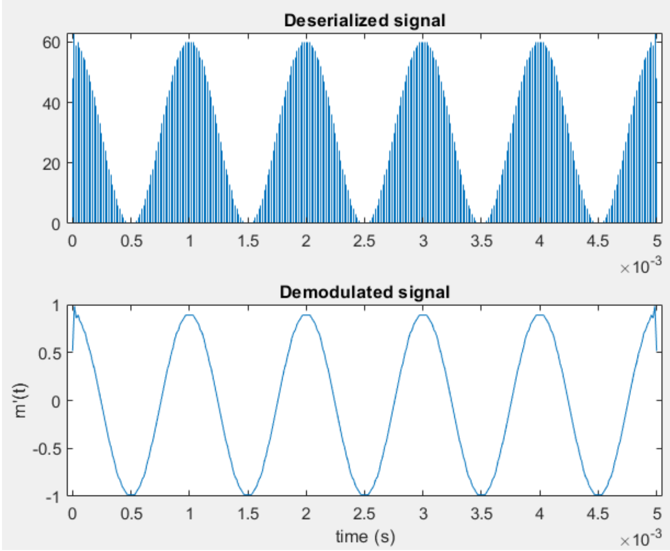

Fig. 19 MATLAB BASK Demodulation

First, the message signal undergoes PCM, where it is sampled, quantized, and encoded. The encoded part cannot be shown in the plot because it is an array of zeros and ones, not a signal. Next, the encoded part is serialized into on-off signaling and multiplied with the carrier to produce the modulated signal. For demodulation, the modulated signal is passed through an envelope detector, sampled, and passed through a decision-making system to obtain a decoded version of the signal. This decoded version is then deserialized into samples and dequantized into the demodulated signal which is very similar to the original message. The same system is then used with an audio file, and the following are the results:

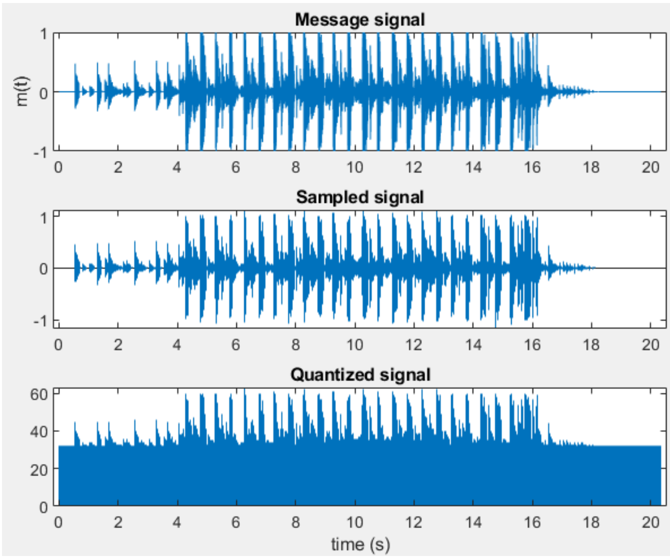

Fig. 20 MATLAB BASK Audio Sampling and Quantization

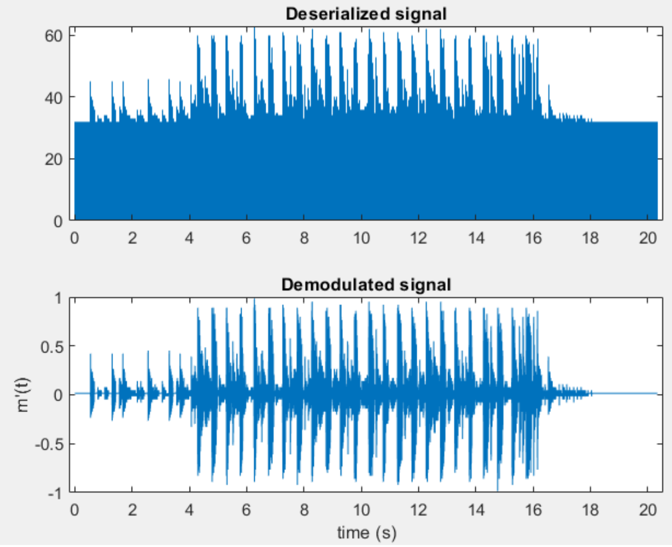

Fig. 21 MATLAB BASK Audio Demodulation

The other plots of the system are skipped in this paper because they do not convey any information with so many samples being handled. Nonetheless, the system works very well for audio signals, except for the fact that, because of the sampling and relatively low bit count for quantization (6 bits) the obtained audio signal is not completely pure.

Finally, the QPSK is tested, first with a single tone:

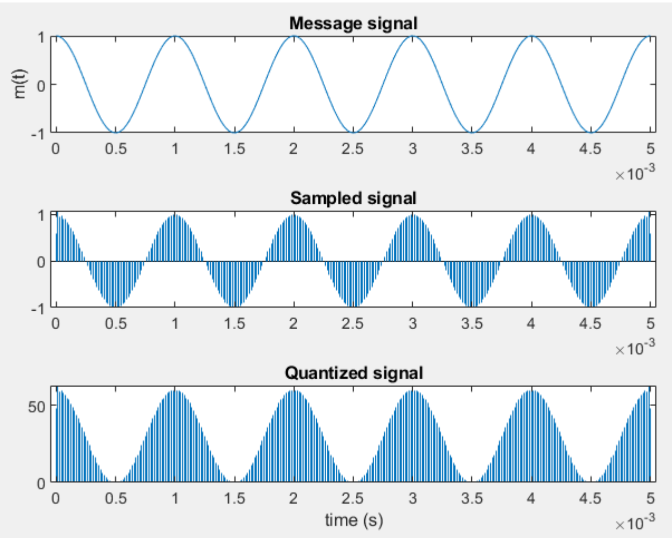

Fig. 22 MATLAB QPSK Sampling and Quantization

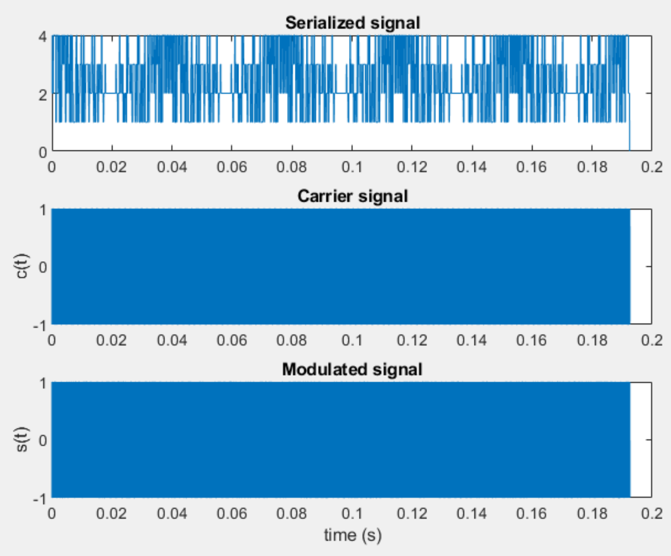

Fig. 23 MATLAB QPSK Modulation

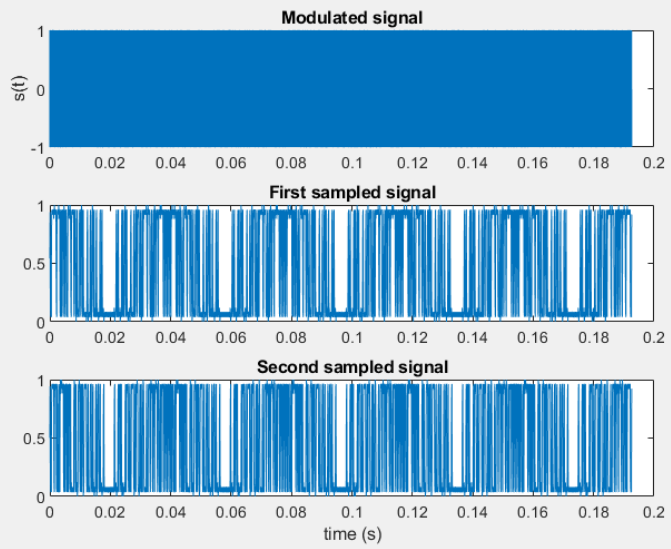

Fig. 24 MATLAB QPSK Signal Detection

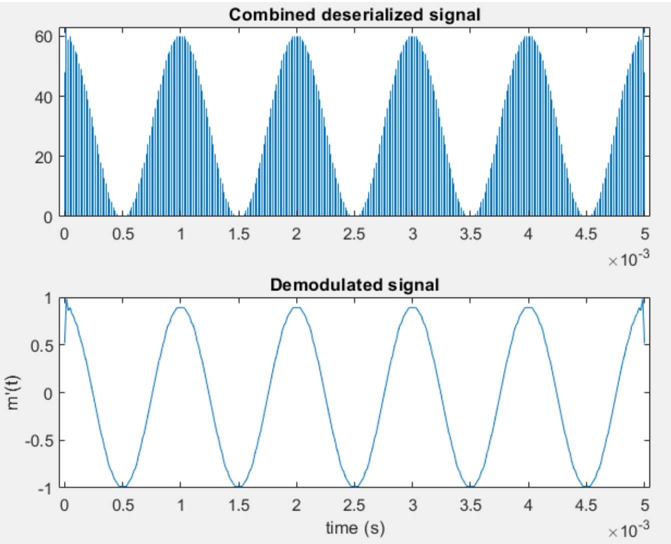

Fig. 25 MATLAB QPSK Demodulation

The original message is sampled, quantized, and encoded, then, each two bits are taken together to form a 2-bit number (between 0 and 3) and mapped as follows:

$$0 \rightarrow 2$$
$$1 \rightarrow 3$$
$$2 \rightarrow 1$$
$$3 \rightarrow 4$$

These number decide the phase shift of the modulated signal according to the QPSK formula.

For demodulation, the modulated signal is multiplied by the carrier and its Hilbert transform to obtain two signals, which are filtered, sampled, and passed through a decision-making system to obtain the decoded version of each. Next, the two signals are combined, deserialized, and dequantized to obtain the demodulated signal, which is very similar to the original signal.

The system is also tested with an audio file:

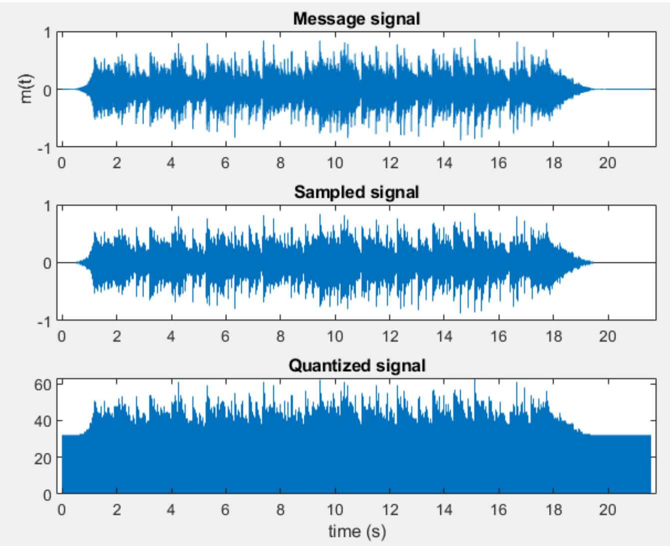

Fig. 26 MATLAB QPSK Audio Sampling and Quantization

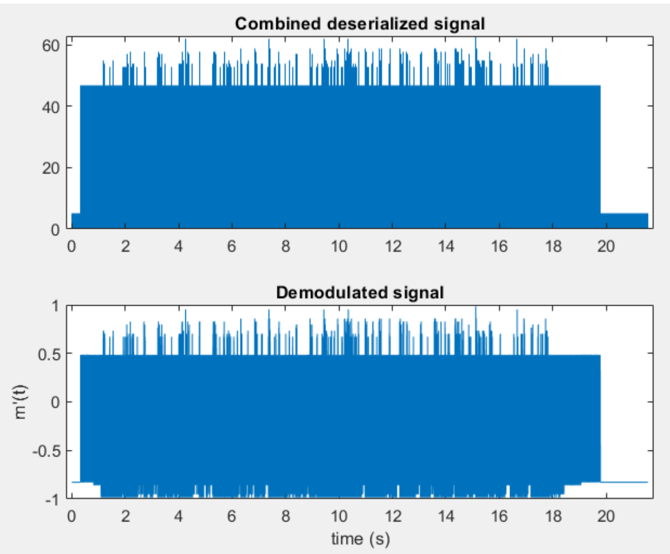

Fig. 27 MATLAB QPSK Audio Demodulation

Again, the other plots of the system are skipped because they do not convey any information with so many samples being handled. The test with an audio file proved to be successful as well.

IX. Conclusion

In conclusion, this research paper has delved into the intricacies of communication systems and modulation techniques, with a particular focus on analog and digital methods. We explored the fundamental principles of Amplitude Modulation (AM) and Frequency Modulation (FM) as analog modulation schemes, understanding their advantages and disadvantages. Additionally, the paper provided an in-depth examination of Binary Amplitude Shift Keying (BASK) and Quadrature Phase Shift Keying (QPSK) as digital modulation techniques, highlighting their operational principles and strengths.

The study emphasized the importance of modulation in efficiently transmitting information over various communication channels. AM and FM, being prominent analog techniques, have found enduring utility, especially in broadcast and audio applications. On the other hand, the digital techniques of BASK and QPSK demonstrated their significance in modern digital communication systems, offering improved bandwidth efficiency and robustness against noise.

Furthermore, the investigation touched upon practical considerations such as reducing antenna size and addressed the impact of modulation techniques on bandwidth requirements. The exploration of these topics contributes to the broader understanding of communication engineering and its practical implications.

As we navigate the evolving landscape of communication technologies, the insights gained from this research underscore the ongoing relevance of modulation techniques in shaping the efficiency, reliability, and adaptability of communication systems. Future advancements in this field are likely to leverage these foundational principles as we continue to push the boundaries of information transmission and connectivity. In essence, this research lays a groundwork for both understanding the current state of communication systems and inspiring further exploration into innovative solutions that will define the future of communication technology.

References

[1] Michael Steer, North Carolina State University, libretexts.org “Analog Modulation” [Online]. Available: https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Electronics/Microwave_and_RF_Design_I_-_Radio_Systems_(Steer)/02%3A_Modulation/2.04%3A_Analog_Modulation

[2] Michael Steer, North Carolina State University, libretexts.org “Phase Shift Keying Modulation” [Online]. Available: https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Electronics/Microwave_and_RF_Design_I_-_Radio_Systems_(Steer)/02%3A_Modulation/2.13%3A_Phase_Shift_Keying_Modulation

[3] (2023) Marc Lichtman, “Digital Modulation” [Online]. Available: https://pysdr.org/content/digital_modulation.html

[4] (2007) Simon Haykin, Michael Moher, “An Introduction to Analog and Digital Communications,” 2nd edition