

QAM 解調之 MATLAB 專題實作 結報

通訊三 407430025 蘇沛錦

QAM(Quadrature amplitude modulation)是由兩個相位相差九十度的正交調幅載波所組成，假設每次傳送區塊中含有 k 個位元，其可以分割為兩組分別為 $(k/2)$ 位元的子區塊(假設 k 是偶數)，這兩個區塊都使用 $(k/2)$ 位元的 D/A 轉換器，以提供載波所需的調變電壓。

下面描述了一個產生16-QAM訊號的數學模型。

The 16-QAM signal is represented by $s(t) = A_R \cos(2\pi f_c t) + A_I \sin(2\pi f_c t)$ where A_R and A_I are real numbers. The mapping from the information vector to (A_R, A_I) is given in Figure 2.

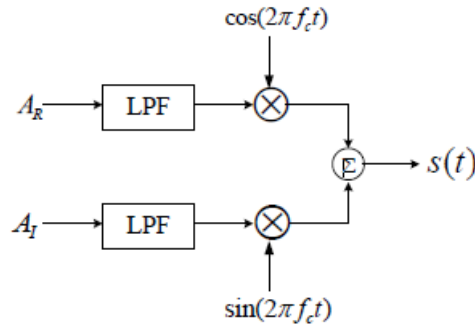


Figure 1: QAM modulator

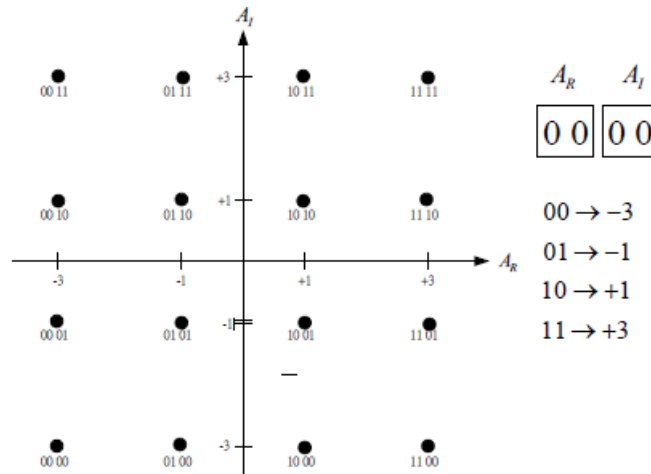


Figure 2: 16-QAM constellation

16-QAM 解調

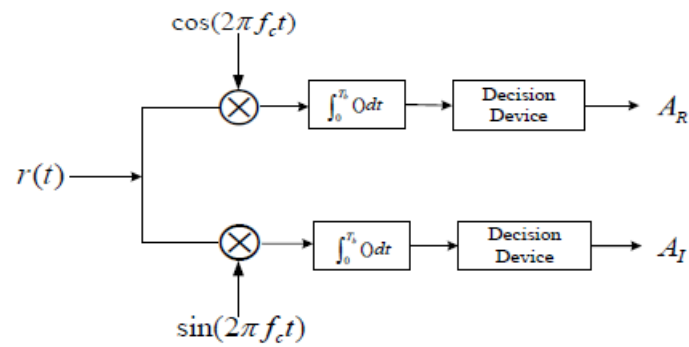


Figure 3: QAM demodulator

音訊取樣進行調變：

A voice transmission system with digital communication is shown in Figure 4.

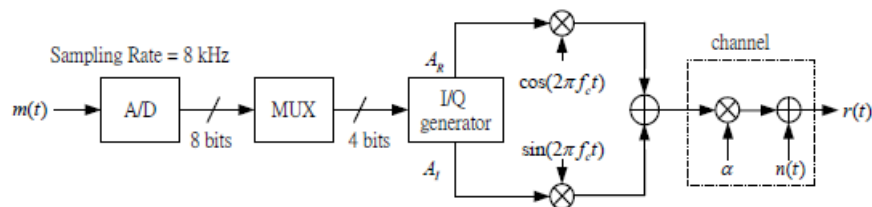


Figure 4: The voice transmission system with 16-QAM.

The voice signal is sampled with sampling rate of , and Each sampled signal is then quantized to 8 bits (0~255) 。

The total bit rate for the voice stream = 64kbps(8bit × 8k)

The range and levels of the quantizer is shown in Figure 5.

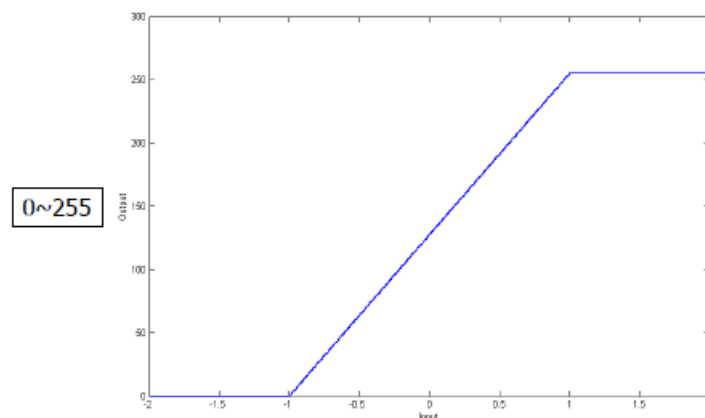


Figure 5: Input/Output relation of the quantizer.

The voice stream is then modulated by a 16-QAM modulator.

Let $([b_7, b_6, b_5, b_4], [b_3, b_2, b_1, b_0])$ be a voice sample. Each voice sample is then divided into two vectors, (b_3, b_2, b_1, b_0) and (b_7, b_6, b_5, b_4) .

The first vector (b_3, b_2, b_1, b_0) is modulated into the first 16-QAM signal and the second vector (b_7, b_6, b_5, b_4) is modulated into the second 16-QAM signal.

In our system, the carrier frequency is assumed to be $f_c = 64$ kHz, the symbol interval is $\frac{1}{(64\text{kbps}/4\text{bits})} = 62.5\mu\text{s}$ which corresponds to a baud rate of 16kHz (64kbps/4bits).

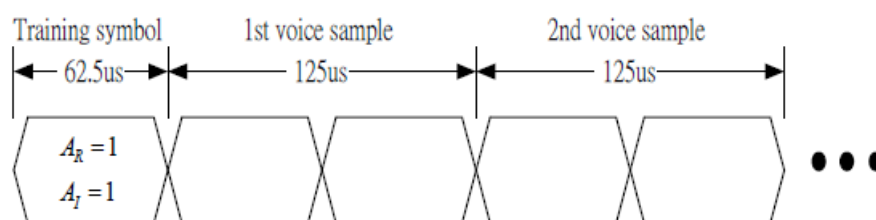
The signal $s(t)$ is then transmitted over the channel.

In the channel, the signal is attenuated by a factor α .

In addition, an additive white Gaussian noise is added resulting the final received signal $r(t)$ expressed as $r(t) = \alpha(A_R \cos(2\pi f_c t) + A_I \sin(2\pi f_c t)) + n(t)$

In order to estimate the attenuation factor α , a training symbol with $A_R = 1$ and $A_I = 1$ is transmitted prior to the actual data.

Figure 7: Signal format for voice transmission



以下是解調過程的MATLAB程式:

```
clc;
clear all;
load rxQAM16n; %載入QAM訊號

rx11 = rxQAM16(1:32).*phi1;
```

```

%phi1 = sin(2*pi*fc*[0:31]*Tsample);
rx12 = rxQAM16(1:32).*phi2;
%phi2 = cos(2*pi*fc*[0:31]*Tsample);

q1 = zeros(length(rx11),1);
q1(1) = rx11(1);
for i = 2:32
    q1(i) = q1(i-1) + rx11(i);
end

q2 = zeros(length(rx12),1);
q2(1) = rx12(1);
for i = 2:32
    q2(i) = q2(i-1) + rx12(i);
end

alpha = (q1(32) + q2(32))/2;

A_R1 = zeros(1,length(z));
A_I1 = zeros(1,length(z));

A_R2 = zeros(1,length(z));
A_I2 = zeros(1,length(z));

A_R = zeros(1,2*length(z));
A_I = zeros(1,2*length(z));

for i = 1: length(z)
    rx1 = rxQAM16N((i-1)*64+1+32:(i-
1)*64+32+32).*phi1;
    rx2 = rxQAM16N((i-1)*64+1+32:(i-
1)*64+32+32).*phi2;

    q1 = zeros(length(rx1),1);
    q1(1) = rx1(1);
    for j = 2:32
        q1(j) = q1(j-1) + rx1(j);
    end

```

```

    q2 = zeros(length(rx2),1);
    q2(1) = rx2(1);
    for k = 2:32
        q2(k) = q2(k-1) + rx2(k);
    end

    A_R1(1,i) = q1(32)/alpha; %AR_1
    A_I1(1,i) = q2(32)/alpha; %AI_1
end

for i = 1: length(z)
    rx21 = rxQAM16N((i-1)*64+33+32:(i-
1)*64+64+32).*phi1;
    rx22 = rxQAM16N((i-1)*64+33+32:(i-
1)*64+64+32).*phi2;

    q1 = zeros(length(rx21),1);
    q1(1) = rx21(1);
    for j = 2:32
        q1(j) = q1(j-1) + rx21(j);
    end

    q2 = zeros(length(rx22),1);
    q2(1) = rx22(1);
    for k = 2:32
        q2(k) = q2(k-1) + rx22(k);
    end

    A_R2(1,i) = q1(32)/alpha; %AR_2
    A_I2(1,i) = q2(32)/alpha; %AI_2
end

for i = 1: 2*length(z)
    if i <= length(z)
        A_R(1,i) = A_R1(1,i); %AR_1
        A_I(1,i) = A_I1(1,i); %AI_1
    else

```

```

        A_R(1,i) = A_R2(1,i-80000); %AR_2
        A_I(1,i) = A_I2(1,i-80000); %AI_2
    end
end

%%畫出 16-QAM Constellation
figure(1);
clf;
plot(A_I, A_R, 'x');
axis([-4, 4, -4, 4]);
grid on

xlabel('A_R');
ylabel('A_I');
title('16-QAM constellation');

m = zeros(length(z),1);

for i = 1:length(z)
    temp = round(A_R1(1,i));
    switch temp
        case -3
            m(i,1) = m(i,1);
        case -1
            m(i,1) = bitset(m(i,1),1);
        case 1
            m(i,1) = bitset(m(i,1),2);
        case 3
            m(i,1) = bitset(m(i,1),1);
            m(i,1) = bitset(m(i,1),2);
    end

    temp2 = round(A_I1(1,i));
    switch temp2
        case -3
            m(i,1) = m(i,1);
        case -1
            m(i,1) = bitset(m(i,1),3);

```

```

        case 1
            m(i,1) = bitset(m(i,1),4);
        case 3
            m(i,1) = bitset(m(i,1),3);
            m(i,1) = bitset(m(i,1),4);
    end

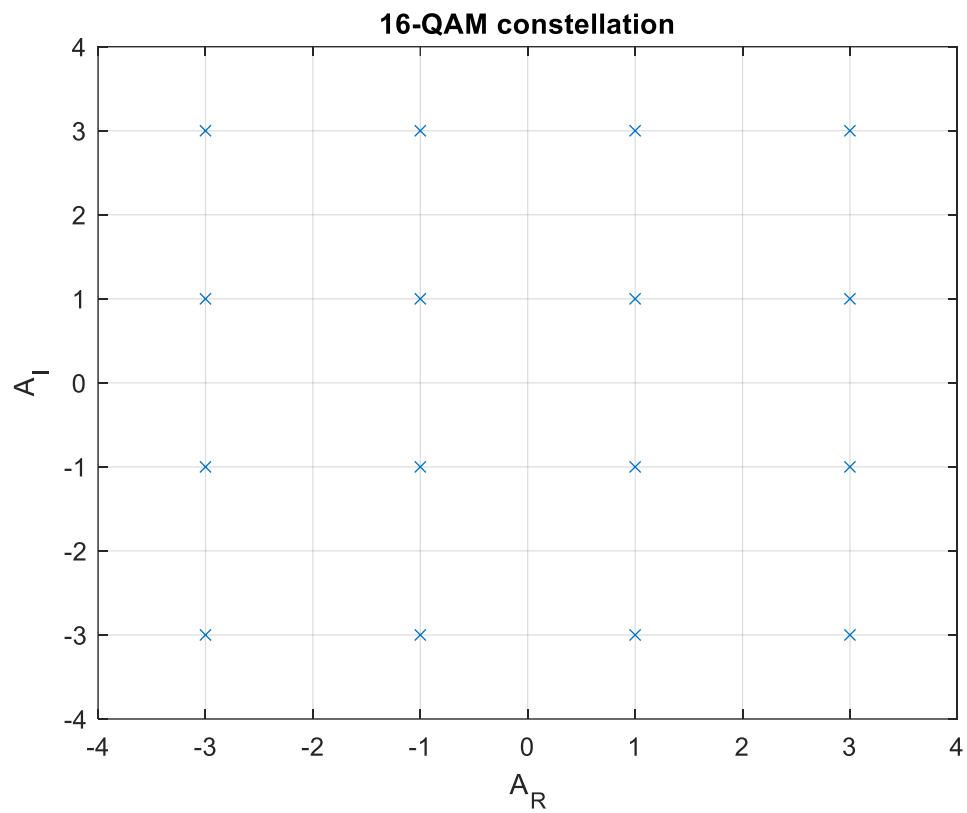
    temp3 = round(A_R2(1,i));
    switch temp3
        case -3
            m(i,1) = m(i,1);
        case -1
            m(i,1) = bitset(m(i,1),5);
        case 1
            m(i,1) = bitset(m(i,1),6);
        case 3
            m(i,1) = bitset(m(i,1),5);
            m(i,1) = bitset(m(i,1),6);
    end

    temp4 =round(A_I2(1,i));
    switch temp4
        case -3
            m(i,1) = m(i,1);
        case -1
            m(i,1) = bitset(m(i,1),7);
        case 1
            m(i,1) = bitset(m(i,1),8);
        case 3
            m(i,1) = bitset(m(i,1),7);
            m(i,1) = bitset(m(i,1),8);
    end
end

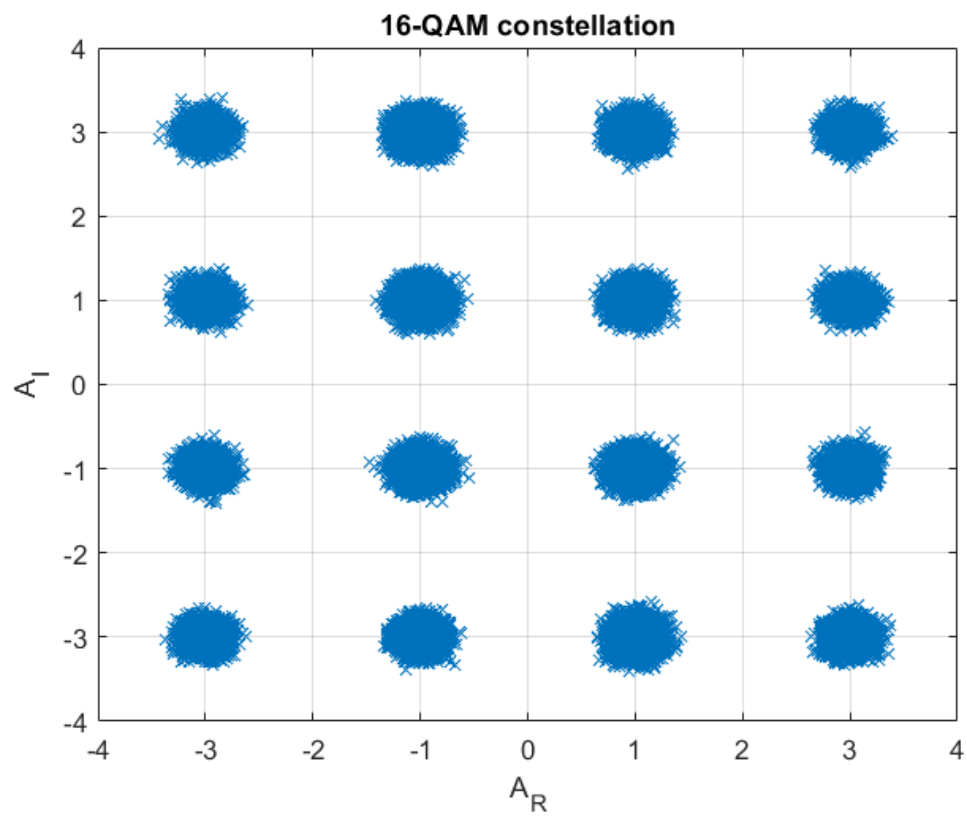
%result = (m-127)/120*max_y + mean_y;

%播放聲音
soundsc(m,FS);

```



圖一、QAM 星座圖 (針對 rxQAM16.m 之接收訊號)



圖二、QAM 星座圖 (針對 rxQAM16N.m 之接收訊號)

實驗心得

這週的 MATLAB 實作的實驗內容為 QAM 解調，首先 QAM(Quadrature Amplitude Modulation)的調變是由兩個正交調幅載波訊號所組成，也就是將 A_R 和 A_I ，分別乘上 $\cos(2\pi f_c t)$ 以及 $\sin(2\pi f_c t)$ 來合成訊號 $s(t)$ 。當中 A_R 與 A_I 都是實數，而 (A_R, A_I) 則是構成 information vector，例如：「00」代表位置為「-3」、「01」代表位置為「-1」、「10」代表位置為「+1」，而「11」則代表位置為「+3」。關於 QAM 的解調，是將接收到的訊息 $r(t)$ 分別各自乘上 $\cos(2\pi f_c t)$ 和 $\sin(2\pi f_c t)$ ，並透過積分與 Decision Device 以得到 A_R 與 A_I 。而實驗一開始是將音訊進行取樣和做 QAM 調變，我們先將音訊以 8kHz 的取樣頻率做取樣，並將這些取樣訊號以 8bits 做量化處理(也就是 0~255)，最終可知總位元率為 64kbps。接下來將這 8bits 的訊號進行 16-QAM 調變，由於 16-QAM 每次可調變 4bits 的訊號，因此我們將 8 個位元分成二個 16-QAM 進行調變。也就是將 $(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$ 這 8 個位元， (b_3, b_2, b_1, b_0) 後 4bits 調變成第一組 16-QAM 訊號； (b_7, b_6, b_5, b_4) 前 4bits 調變成第二組 16-QAM 的訊號，因此最終 $s(t)$ 為兩組的 16-QAM 訊號所構成。要注意的是，因為調變後的訊號經過通道除了會有雜訊 $n(t)$ 的干擾，還有衰減因子 α 的影響。所以在解調的過程中，要先得知通道衰減因子 α 的值，才不會讓解調出來的訊號失真。而為了瞭解 α 在通道當中的實際數值，我們會透過在調變訊號中加入已知訊號，也就是 training symbol，以便我們在解調一開始就計算出通道的 α 數值。

在藉由 MATLAB 撰寫 QAM 解調程式碼的過程中，自己試著從調變的程式碼以及解調變的方塊圖理解、推導解調的步驟，首先是將已知的 training symbol 進行解調，以便得知通道內 α 的值。而由於 training symbol 為 $A_R = 1$ 和 $A_I = 1$ ，所以在計算 α 值時，是將兩個分別解調出來的 A_R 與 A_I 相加取平均。透過程式碼編寫並實際測試，我們算出通道中 α 的值為 16。在得知 α 的實際值後，接著便是將剩下的接收訊號 $r(t)$ 進行解調，最後再除上 α 才算得到正確的訊號。運用前面針對 training symbol 的解調過程，我們照著步驟一一解出 A_R 和 A_I ，並要記得除上 α ，才算得到正確的訊號。自己也將所計算出的訊號，畫成課堂中用來討論 QAM 調變的星座圖(Constellation)。而我們有兩種 QAM 調變的訊號，一是等同傳送訊號 $s(t)$ 的 rxQAM.m 接收訊號；另一個則是考慮通道雜訊 $n(t)$ 的 rxQAMN.m 接收訊號。藉由將兩種訊號解調記錄在星座圖上，我們發現 rxQAM.m 接收訊號的星座圖上每個點的位置都精準地位於這 16 個點上；而考慮通道雜訊 $n(t)$ 的 rxQAMN.m 接收訊號，其星座圖中點的分布就大部分而言，都散布在這 16 個點的附近。不過這 16 個點中每點的分布狀況都呈現像高斯分布，均勻地集中在每個中心點周圍。最後我們將解調出的訊號 A_R 和 A_I ，轉換成原本 $(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$ 的 8bits 音訊訊號。在這邊助教也十分熱心提醒我，在轉換過程中要注意這 8bits 的解調順序，才不會有問題產生。非常感謝助教的提醒，最後自己也順利還原成原始的音訊訊號，並成功得播放出來。透過實際寫程式碼對 QAM 訊號進行解調，並將解調出的訊號透過座標標示在星座圖。除了感到非常有成就感之外，也能幫助自己更加瞭解數位通訊課程中所教的各種數位調變的內容，非常充實！