

頻率解調(FM)之 MATLAB 專題實作(相位差分法) 結報

通訊三 407430025 蘇沛錦

在非同步解調中，經由斜率檢測器，我們可將調頻訊號還原。此外，進一步我們也可以使用平衡式鑑頻器的調頻解調器。

在本次實驗中，我們將探討另一種類型的同步解調器。

FM訊號

$$s[n] = A \cos(2\pi f_c n T_s + \phi[n])$$

將FM訊號分別乘上 $\cos(2\pi f_c n T_s)$ 及 $\sin(2\pi f_c n T_s)$ ，並經由低通濾波器得

$$\begin{aligned} x_R[n] &= \text{LPF}[s[n]\cos(2\pi f_c n T_s)] = A \cos(\phi[n]) \\ x_I[n] &= \text{LPF}[s[n]\sin(2\pi f_c n T_s)] = -A \sin(\phi[n]) \end{aligned}$$

將上述訊號表示成複數

$$x[n] = x_R[n] + jx_I[n] = A \exp(-j\phi[n])$$

由以上關係得

$$\phi[n] = -\tan^{-1}\left(\frac{x_I[n]}{x_R[n]}\right)$$

我們的訊息訊號藏 $\phi[n]$ 在，其關係式如下

$$\phi[n] = 2\pi k_f \sum_{k=0}^n m[k]$$

由上式我們得到

$$m[n] = \frac{1}{2\pi k_f} (\phi[n] - \phi[n-1])$$

現在我們將接收訊號的前一點乘上後一點的共軛得

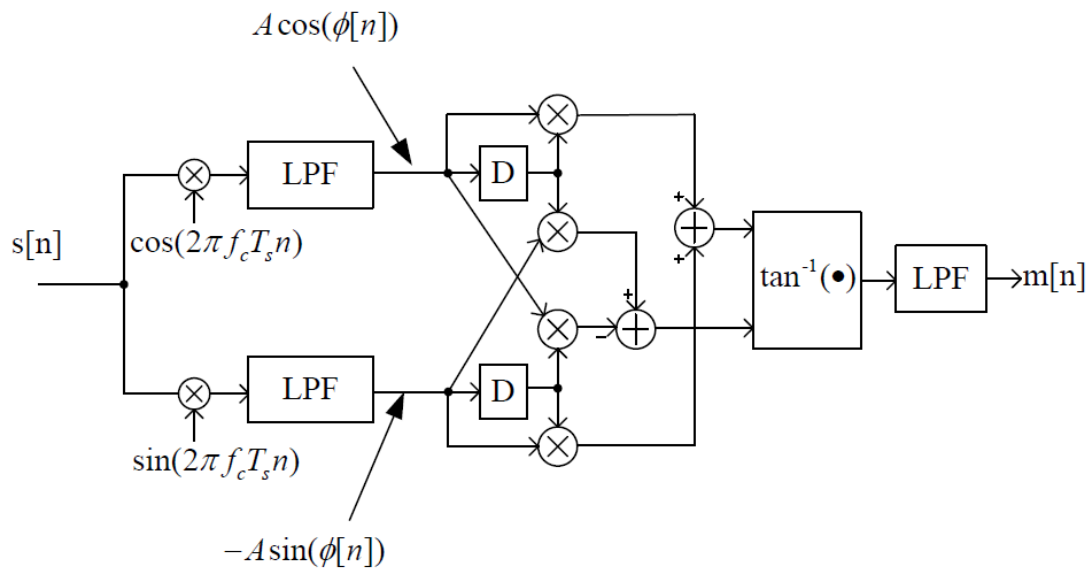
$$y[n] = x[n-1]x^*[n] = A^2 \exp(j\phi[n] - \phi[n-1]) = A^2 \exp(2\pi k_f m[n])$$

因此訊息訊號 $m[n]$ 就藏在 $y[n]$ 的相位中。

最後一個步驟是將訊息訊號由 $y[n]$ 取出得

$$m[n] = \frac{1}{2\pi k_f} \tan^{-1}\left(\frac{y_I[n]}{y_R[n]}\right)$$

由以上分析得到解調器如下



實驗結果

- 載入FM訊號
- 設定
 - 低通路波器長度(這裡設定300)
 - 訊號取樣頻率(64k)
- 建立低通濾波器1&2 第一低通濾波截止頻率20K，第二低通濾波截止頻率5K
- 設定時間index t
- 將FM訊號分別乘上 $\cos(2\pi f_c t / \text{取樣頻率})$ 及 $\sin(2\pi f_c t / \text{取樣頻率})$ ，PS. 注意向量維度。
- 分別經過低通濾波器1
- 設一複數將實部項與虛部項相加 hint: $y = x1 + j*x2$;
- 複數訊號的前一點(n)乘上後一點(n-1)的共軛
hint: 1. $z = y(1:\text{length}(y)-1) .* \text{conj}(y(2:\text{length}(y)))$
2. 注意維度
- 把聲音從複數相位中取出 hint: $\text{angle}()$;
- 把取出的訊號經過低通濾波器2
- 播放聲音

以下是解調過程的MATLAB程式:

```
%%%畫出波形
load rxFM; %載入FM訊號
figure(1);
clf;
offset = 100000;
subplot(2,1,1);
plot(z(offset+1:offset+100)); %畫出訊息訊號
xlabel('Samples');
ylabel('Amplitude');

subplot(2,1,2);
plot(rxFM(offset+1:offset+100)); %畫出FM訊號
xlabel('Samples');
ylabel('Amplitude');

%設定
%A.低通濾波器長度(這裡設定300)
%B.訊號取樣頻率(64k)

%第一低通濾波器截止頻率20K
for i = 1:300
    a1(i) = sinc(20e3 * (i-300/2)/64e3);
end

%第二低通濾波器截止頻率5K
for i = 1:300
    a2(i) = sinc(5e3 * (i-300/2)/64e3);
end

%將FM訊號分別乘上cos(2pi*fc*t/取樣頻率) 及 sin(2pi*fc*t/取樣頻率)
b1 = rxFM.*cos(2*pi*fc*t'/64e3);
b2 = rxFM.*sin(2*pi*fc*t'/64e3);

%分別經過低通濾波器1
x1 = conv(a1,b1);
```

```

x2 = conv(a1,b2);

%設一複數 將實部項與虛部項相加
y = x1 + j*x2;

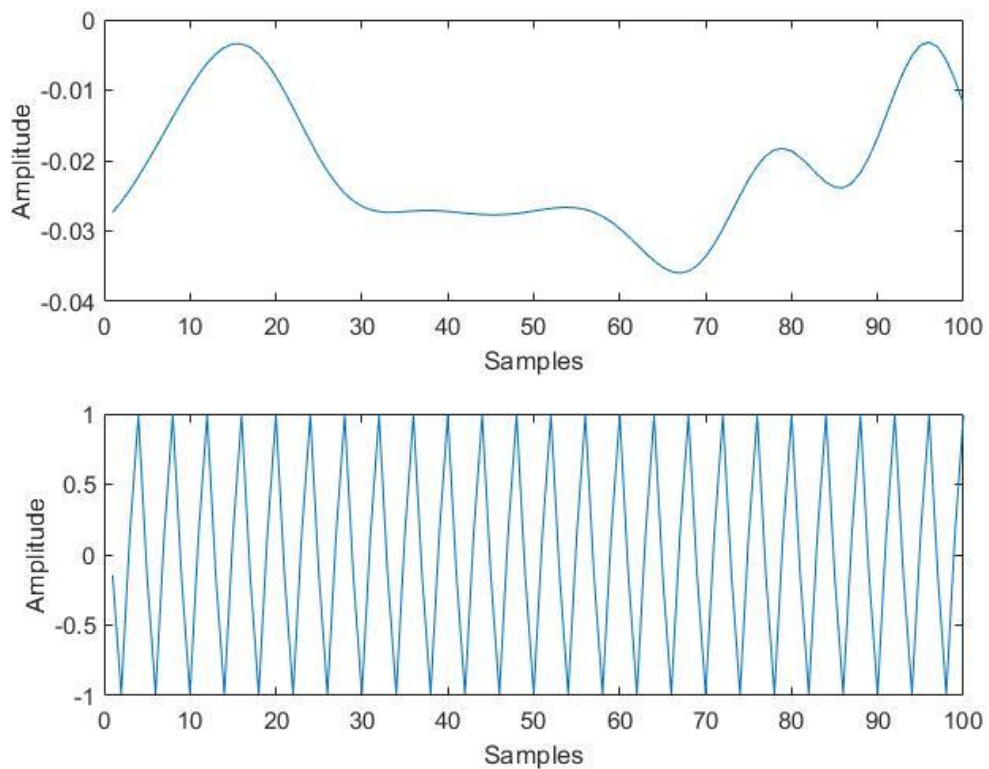
%複數訊號的前一點(n) 乘上後一點(n-1)的共軛
z = y(1:length(y)-1) .* conj( y(2:length(y)) );

%把聲音從複數相位中取出
c = angle(z);

%把取出的訊號經過低通濾波器2
result = conv(c,a2);

%播放聲音
soundsc(result,64000);

```



圖一、調變前原始訊息訊號與調變後 FM 訊號之波形

實驗心得

在本次實驗工作中，我們進一步使用平衡式鑑頻器的調頻解調器對 FM 訊號進行解調。首先將 FM 訊號 $s[n]$ 分別乘上 $\cos(2\pi f_c n T_s)$ 及 $\sin(2\pi f_c n T_s)$ ，並經過低通濾波器得到 $x_R[n]$ 和 $x_I[n]$ 。我們再將兩個訊號： $x_R[n]$ 和 $x_I[n]$ 表示成複數形式 $x[n] = x_R[n] + jx_I[n]$ ，並改寫成 $A \exp(-j\phi[n])$ 以得到 $\phi[n]$ 。而我們的訊息訊號 $m[n]$ 就藏在 $\phi[n]$ 內，所以我們透過將接收到的訊號的前一項 $x[n-1]$ 乘上後一項的共軛 $x^*[n]$ ，便可得到 $y[n]$ 。最後我們將 $y[n]$ 的實部「 $y_I[n]$ 」除上虛部「 $y_R[n]$ 」並做反正切 (\tan^{-1})，再乘上 $1/(2\pi k_f)$ ，就能解調出訊息訊號 $m[n]$ 。透過相位差分法，我們發現解調出來的聲音訊號，比起上個工作的「微分法」解調，相位差分法的聲音更加清楚且較為宏亮，失真的情況大幅改善。

在編寫程式碼的過程中，自己不太了解的地方就是如何在 MATLAB 實作將 FM 訊號乘上 $\cos(2\pi f_c n T_s)$ 和 $\sin(2\pi f_c n T_s)$ 。由於我們現在做的部分是離散的訊號，因此自己原本是想藉由迴圈將二種不同的離散訊號一一相乘。但在助教的熱心提醒下我才知道 MATLAB 有一個很方便的運算子：「點乘 ($.*$)」，透過點乘我們可以輕鬆將二種離散的訊號分別一個一個相乘，不用再藉由迴圈執行。這也讓我體會到難怪老師一開始在介紹 MATLAB 時便說到：如果很會寫 MATLAB 的人，應該會盡量避免使用迴圈的部分。這是因為利用大量迴圈執行其實並不會提升程式的效率，反而會增加程式的複雜度。而藉由本次 FM 的 MATLAB 模擬解調變的實作課程，除了再次複習 FM 的調變原理外，更重要的是讓自己學習到 MATLAB 的語法及使用技巧，也讓自己體認到 MATLAB 許多強大的數學計算函式以及多種工具箱 (Toolbox)，都可以善加利用在數位通訊、訊號處理等領域上！