

数字调制按方法可以分为多进制幅度键控(M-ASK)、正交幅度键控(QASK)、多进制频率键控(M-FSK)以及多进制相位键控(M-PSK)。数字调制包括数模转换和模拟调制两部分,如图 3-1 所示。



图 3-1 数字调制过程

在数字信号通信快速发展以前主要是模拟通信。为了合理使用频带资源,提高通信质量,需要使用模拟调制技术。通常,连续波的模拟调制是以正弦信号为载波的调制方式,分为线性调制和非线性调制。线性调制是指调制后的信号频谱为调制信号的频谱的平移或线性变换,而非线性调制则没有这个性质。

每一种调制都通过以下几个特点来表征。

- (1) 调制信号的时域表达式。
- (2) 调制信号的频域表达式。
- (3) 调制信号的带宽。
- (4) 调制信号的功率分布。
- (5) 调制信号的信噪比。

3.1 双边带调幅与解调

1. 双边带调幅

在双边带调幅(DSB-AM)中,已调信号的时域表示为

$$u(t) = m(t)c(t) = A_c m(t) \cos(2\pi f_c t + \phi_c) \quad (3-1)$$

式中: $m(t)$ 是消息信号; $c(t) = A_c \cos(2\pi f_c t + \phi_c)$ 为载波; f_c 是载波的频率; ϕ_c 是初始相位。为了讨论方便,取初相 $\phi_c = 0$ 。

对 $u(t)$ 作傅里叶变换,即可得到信号的频域表示

$$U(f) = \frac{A_c}{2} M(f - f_c) + \frac{A_c}{2} M(f + f_c) \quad (3-2)$$

传输带宽 B_T 是消息信号带宽 W 的两倍,即 $B_T=2W$ 。

【例 3-1】 某消息信号 $m(t)=\begin{cases} 1, & 0 \leq t \leq t_0/3 \\ -2, & t_0/3 < t \leq 2t_0/3 \\ 0, & \text{其他} \end{cases}$,用信号 $m(t)$ 以 DSB-AM 方式

调制载波 $c(t)=\cos(2\pi f_c t)$, 所得到的已调信号记为 $u(t)$ 。设 $t_0=0.15\text{s}$, $f_c=250\text{Hz}$ 。试比较消息信号与已调信号,并绘制它们的频谱。

其实现的 MATLAB 程序代码如下:

```
>> clear all;
t = 0.15; % 信号保持时间
ts = 0.001; % 采样时间间隔
fc = 250; % 载波频率
fs = 1/ts; % 采样频率
df = 0.3; % 频率分辨率
t1 = [0:ts:t]; % 时间向量
m = [ones(1,t/(3*ts)), -2*ones(1,t/(3*ts)), zeros(1,t/(3*ts)+1)]; % 定义信号序列
y = cos(2*pi*fc.*t1); % 载波信号
u = m.*y; % 调制信号
[n,m,df1] = fftseq(m,ts,df); % 傅里叶变换
n = n/fs;
[ub,u,df1] = fftseq(u,ts,df);
ub = ub/fs;
[Y,y,df1] = fftseq(y,ts,df);
f = [0:df1:df1*(length(m)-1)] - fs/2; % 频率向量
subplot(221);
plot(t1,m(1:length(t1))); % 未解调信号
title('未解调信号');
subplot(222);
plot(t1,u(1:length(t1))); % 解调信号
title('解调信号');
subplot(223);
plot(f,abs(fftshift(n))); % 未解调信号频谱
title('未解调信号频谱');
subplot(224);
plot(f,abs(fftshift(ub))); % 解调信号频谱
title('解调信号频谱');
```

该程序运行后得到的信号和调制信号及信号调制前后的频谱对比如图 3-2 所示。

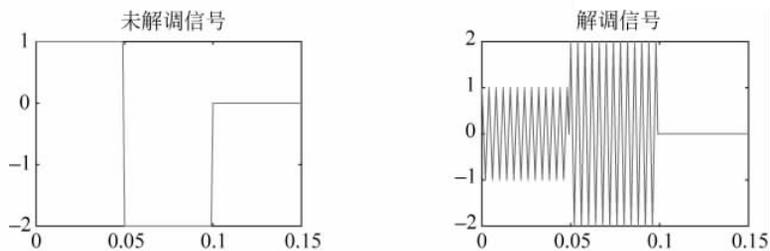


图 3-2 DSB-AM 得到的信号和调制信号及信号调制前后的频谱图

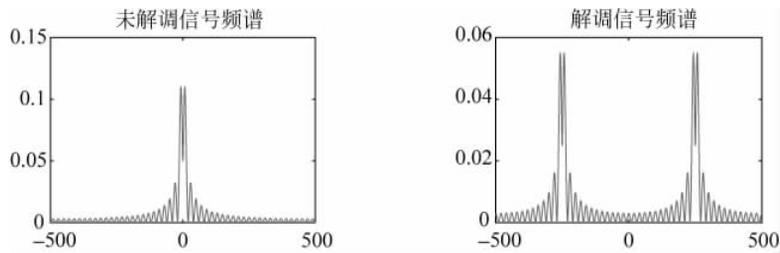


图 3-2 (续)

在以上代码中调用的自定义函数的代码如下：

```
function [M,m,df] = fftseq(m,tz,df)
fz = 1/tz;
if nargin == 2 % 判断输入参数的个数是否符合要求
    n1 = 0;
else
    n1 = fz/df; % 根据参数个数决定是否使用频率缩放
end
n2 = length(m);
n = 2^(max(nextpow2(n1),nextpow2(n2)));
M = fft(m,n); % 进行离散傅里叶变换
m = [m,zeros(1,n-n2)];
df = fz/n;
function p = ampower(x)
% 此函数用作计算信号功率
p = (norm(x)^2)/length(x); % 计算出信号能量
t0 = 0.15;
tz = 0.001;
m = zeros(1,501);
for i = 1:1:125 % 计算第 1 段信号值的功率
    m(i) = i;
end
for i = 1:126:1:375 % 计算第 2 段信号值的功率
    m(i) = m(125) - i + 125;
end
for i = 376:1:501 % 计算第 3 段信号值的功率
    m(i) = m(375) + i - 375;
end
m = m/1000; % 功率归一化
n_hat = imag(hilbert(m));
```

DSB-AM 调制信号的解调过程如图 3-3 所示。

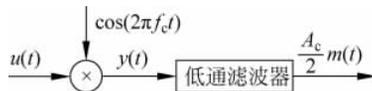


图 3-3 DSB-AM 调制信号的解调

调制信号 $u(t) = A_c m(t) \cos(2\pi f_c t)$ 与接收机本地振荡器所产生的正弦信号 $\cos(2\pi f_c t)$ 相乘, 可得混频器输出为

$$y(t) = A_c m(t) \cos^2(2\pi f_c t) = \frac{A_c}{2} m(t) + \frac{A_c}{2} m(t) \cos(4\pi f_c t) \quad (3-3)$$

它的傅里叶变换为

$$Y(f) = \frac{A_c}{2} M(f) + \frac{A_c}{2} M(f - 2f_c) + \frac{A_c}{2} M(f + 2f_c) \quad (3-4)$$

可见, 混频器输出由一个低频分量 $\frac{A_c}{2} M(f)$ 和 $\pm f_c$ 处的两个高频分量组成。

2. 双边带解调

将 $y(t)$ 通过带宽为 W 的低通滤波器, 高频分量被滤除, 而与消息信号成正比的低通分量 $\frac{A_c}{2} m(t)$ 被解调。如果调制相位 ϕ_c 未知, 则需使用 Costas 环解调方法来恢复接收信号的相位信息。Costas 环解调法如图 3-4 所示。

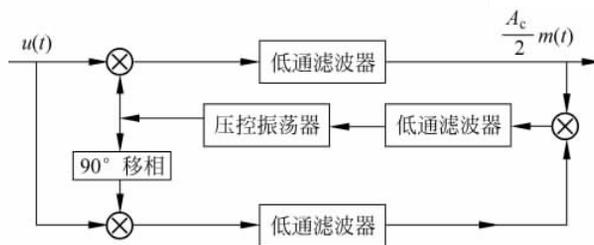


图 3-4 Costas 环解调法

【例 3-2】 对例 3-1 的单边带调制信号进行双边带解调, 并绘制消息信号的时频域曲线。

其实现的 MATLAB 代码如下:

```
>> clear all;
t = 0.15; % 信号保持时间
ts = 1/1500; % 采样时间间隔
fc = 250; % 载波频率
fs = 1/ts; % 采样频率
df = 0.3; % 频率分辨率
t1 = [0:ts:t]; % 时间向量
m = [ones(1, t/(3 * ts)), -2 * ones(1, t/(3 * ts)), zeros(1, t/(3 * ts) + 1)]; % 定义信号序列
c = cos(2 * pi * fc * t1); % 载波信号
u = m * c; % 调制信号
y = u * c; % 缩放
[n, m, df1] = fftseq(m, ts, df); % 傅里叶变换
n = n/fs;
[ub, u, df1] = fftseq(u, ts, df);
ub = ub/fs;
[Y, y, df1] = fftseq(y, ts, df);
```

```

Y = Y/fs;
f_c_off = 150; % 滤波器的截止频率
n_c_off = floor(150/df1); % 设计滤波器
f = [0:df1:df1 * (length(m) - 1)] - fs/2; % 频率向量
h = zeros(size(f));
h(1:n_c_off) = 2 * ones(1,n_c_off);
h(length(f) - n_c_off + 1:length(f)) = 2 * ones(1,n_c_off);
dem1 = h. * Y; % 滤波器输出的频率
dem = real(fftshift(dem1)) * fs; % 滤波器的输出
subplot(221);
plot(t1,m(1:length(t1))); % 未解调信号
title('未解调信号');
subplot(222);
plot(t1,dem(1:length(t1))); % 解调信号
title('解调信号');
subplot(223);
plot(f,abs(fftshift(n))); % 未解调信号频谱
title('未解调信号频谱');
subplot(224);
plot(f,abs(fftshift(dem1))); % 解调信号频谱
title('解调信号频谱');

```

运行程序,效果如图 3-5 所示。

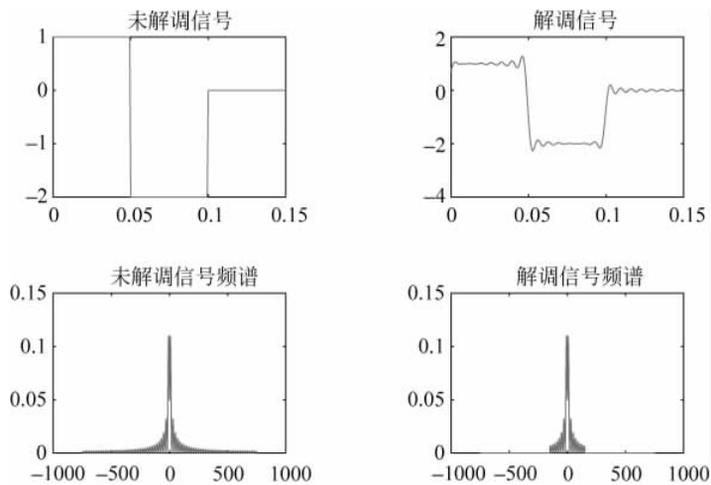


图 3-5 未调制信号、解调信号及其相应的频谱比较

为了恢复消息信号 $m(t)$, 将混频信号 $y(t)$ 通过一个带宽为 150Hz 的低通滤波器。这里, 滤波器带宽的选择可以具有一定的任意性, 这是因为被调信号没有严格的带限。对于有严格带限的被调信号, 低通滤波器带宽的最佳选择为 W , 即被调信号的带宽。因此, 本例所用的理想低通滤波器为

$$H(f) = \begin{cases} 1, & |f| \leq 150 \\ 0, & \text{其他} \end{cases}$$

3.2 常规双边带调幅

常规双边带调幅(AM)在很多方面与双边带幅度调制类似。不同的是,用 $1+am_n(t)$ 代替 $m(t)$ 。在此, a 是调制指数, $m_n(t)$ 是经过归一化处理的消息信号。

在常规 AM 中,调制信号的时域表示为

$$u(t) = A_c[1 + am_n(t)]\cos(2\pi f_c t) \quad (3-5)$$

对 $u(t)$ 作傅里叶变换,即可得到信号的频域表示

$$U(f) = \frac{A_c}{2}[\delta(f - f_c) + aM(f - f_c) + \delta(f + f_c) + aM(f + f_c)] \quad (3-6)$$

传输带宽 B_T 是消息信号带宽的 2 倍,即 $B_T=2W$ 。

【例 3-3】 对例 3-1 中提供的信号进行常规 AM 调制,给定调制指数 $a=0.6$,试绘制信号和调制信号的频谱。

其实现的 MATLAB 程序代码如下:

```
>> clear all;
t = 0.15; % 信号保持时间
ts = 0.001;
fc = 250; % 载波频率
fs = 1/ts; % 采样频率
df = 0.3; % 频率分辨率
a = 0.6; % 调制系数
t1 = [0:ts:t]; % 时间向量
m = [ones(1,t/(3*ts)), -2*ones(1,t/(3*ts)), zeros(1,t/(3*ts)+1)]; % 定义信号序列
c = cos(2*pi*fc.*t1); % 载波信号
m1 = m/max(abs(m)); % 调制信号
u = (1+a*m1).*c; % 调制信号载波
[n,m,df1] = fftseq(m,ts,df); % 傅里叶变换
n = n/fs;
[ub,u,df1] = fftseq(u,ts,df);
ub = ub/fs;
f = [0:df1:df1*(length(m)-1)]-fs/2; % 频率向量
subplot(221);
plot(t1,m(1:length(t1))); % 未解调信号
title('未解调信号');
subplot(222);
plot(t1,u(1:length(t1))); % 解调信号
title('解调信号');
subplot(223);
plot(f,abs(fftshift(n))); % 未解调信号频谱
title('未解调信号频谱');
subplot(224);
plot(f,abs(fftshift(ub))); % 解调信号频谱
title('解调信号频谱');
```

运行程序,效果如图 3-6 所示。

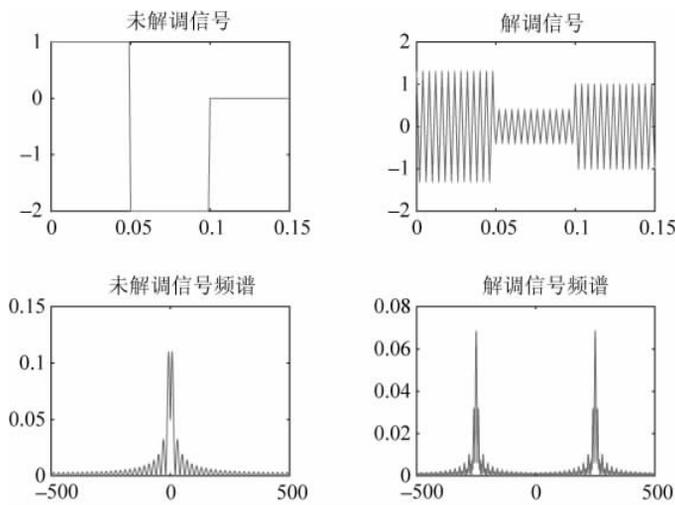


图 3-6 常规幅度调制信号的频谱

3.3 抑制载波双边带调幅

由于常规 AM 调制的效率太低,耗用了大量功率,在小功率场合很不方便,而抑制载波双边带调幅(DSB-SC)就克服了效率低的缺点,它的特点是直接将未调信号与载波相乘,而不是先叠加一个直流在未调信号上再相乘。时域表达式为

$$S_{\text{DSB}}(t) = Af(t)\cos(\omega_c t + \theta_c) \quad (3-7)$$

抑制载波双边带调幅的频谱与常规调幅类似,但没有载频的冲激分量。如果记 $F(f)$ 为调制信号的频域表达式,则已调信号的频域表达式为

$$S_{\text{DSB}}(f) = \frac{A}{2}F(f - f_c) + \frac{A}{2}F(f + f_c) \quad (3-8)$$

从频域表达式可看出,已调信号的频带宽度仍是调制信号的频带的 2 倍: $B_T = 2W$, 如图 3-7 所示。

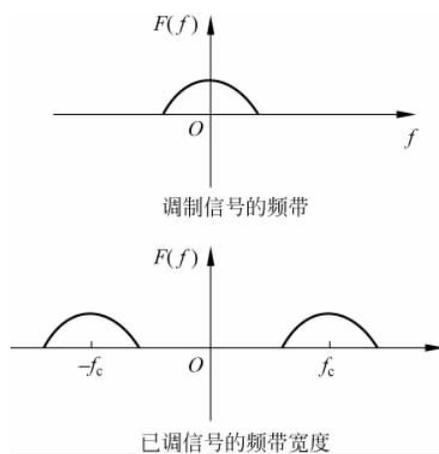


图 3-7 抑制载波调幅的频谱图

【例 3-4】 已知未调制信号为 $S(t) = \begin{cases} \text{sinc}(200t), & |t| \leq t_0 \\ 0, & \text{其他} \end{cases}$, 其中, t_0 取 2s; 载波为

$C(t) = \cos 2\pi f_c t, f_c = 100\text{Hz}$, 用抑制载波调幅来调制信号, 给出调制信号 $M(t)$ 的波形, 画出 $S(t)$ 与 $M(t)$ 的频谱。

其中, $M(t) = S(t)C(t)$, 即

$$M(t) = \begin{cases} 3\text{sinc}(10t)\cos(400\pi t), & |t| \leq 0.1 \\ 0, & \text{其他} \end{cases}$$

其实现的 MATLAB 代码如下：

```
>> clear all;
t0 = 2; % 信号持续时间
ts = 0.001; % 采样时间间隔
fc = 100; % 载波频率
fs = 1/ts;
df = 0.3; % 频率分辨率
t = [-t0/2:ts:t0/2]; % 定义时间序列
% 以下三解为定义信号序列
x = sin(200 * t);
m = x. / (200 * t);
m(1001) = 1; % 避免产生无穷大的值
c = cos(2 * pi * fc. * t); % 载波
u = m. * c; % 抑制载波调制
[M,m,df1] = fftseq(m,ts,df); % 傅里叶变换
M = M/fs;
[U,u,df1] = fftseq(u,ts,df); % 傅里叶变换
U = U/fs; % 频率压缩
f = [0:df1:df1 * (length(m) - 1)] - fs/2;
subplot(2,2,1);plot(t,m(1:length(t))); % 作出未调信号的波形
axis([-0.4,0.4,-0.5,1.1]);
xlabel('时间');title('未调信号');
subplot(2,2,3);plot(t,c(1:length(t)));
axis([-0.1,0.1,-1.5,1.5]);
xlabel('时间');title('载波');
subplot(2,2,2);plot(t,u(1:length(t)));
axis([-0.2,0.2,-1,1.2]);
xlabel('时间');title('已调信号');
figure;
subplot(2,1,1);plot(f,abs(fftshift(M)));
xlabel('频率');title('未调信号的频谱');
subplot(2,1,2);plot(f,abs(fftshift(U)));
xlabel('频率');title('已调信号的频谱');
```

运行程序,得到抑制载波调幅波形如图 3-8 所示,得到的抑制载波调幅频谱图如图 3-9 所示。

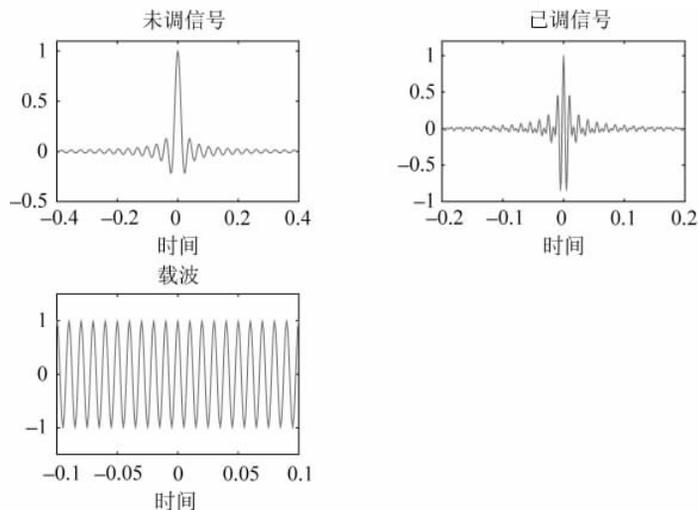


图 3-8 抑制载波调幅波形图

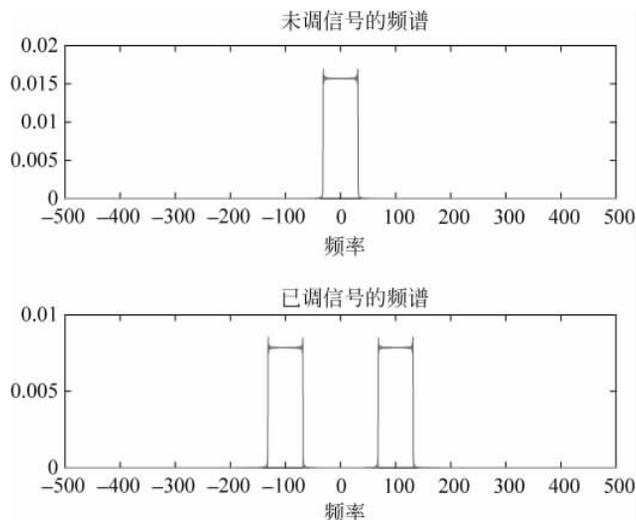


图 3-9 抑制载波调幅频谱图

3.4 单边带调幅与解调

1. 希尔伯特变换

实信号 $x(t)$ 的希尔伯特变换就是将该信号中所有频率成分的信号分量移相 $-\pi/2$ 而得到的新信号, 记为 $\hat{x}(t)$ 。对于单频率正弦信号, 设 $m(t) = A\cos(2\pi ft + \phi)$, 则其希尔伯特变换为

$$\hat{m}(t) = A\cos\left(2\pi ft + \phi - \frac{\pi}{2}\right) = A\sin(2\pi ft + \phi) \quad (3-9)$$

对于任意实周期信号 $x(t)$, 可用周期傅里叶级数展开表示为

$$x(t) = \sum_{n=0}^{\infty} a_n \cos(2\pi nft + \phi_n) \quad (3-10)$$

其希尔伯特变换为

$$\begin{aligned} \hat{x}(t) &= \sum_{n=0}^{\infty} a_n \cos\left(2\pi nft + \phi_n - \frac{\pi}{2}\right) \\ &= \sum_{n=0}^{\infty} a_n \sin(2\pi nft + \phi_n) \end{aligned} \quad (3-11)$$

实信号 $x(t)$ 的解析信号 $y(t)$ 是一个复信号, 其实部为信号 $x(t)$ 本身, 虚部为 $x(t)$ 的希尔伯特变换 $\hat{x}(t)$, 即

$$y(t) = x(t) + j\hat{x}(t) \quad (3-12)$$

MATLAB 中提供了希尔伯特变换函数 `hilbert` 利用 FFT 来计算任意离散时间序列的解析信号序列。函数的调用格式为

`x=hilbert(xr)`: `xr` 是实信号序列; 返回参数 `x` 是一个复数信号序列, `x` 的实部就是 `xr`, `x` 的虚部则是 `xr` 的希尔伯特变换序列。

`x=hilbert(xr, n)`: `n` 作为 FFT 的点数。

【例 3-5】 对 $x(t) = \sin(t)$ 进行希尔伯特变换。

其实现的 MATLAB 代码如下：

```
>> clear all;
t = 0:0.1:30;
y = sin(t);
s_y = hilbert(y);           % 希尔伯特变换
plot(t, real(s_y), t, imag(s_y), 'r:');
legend('原始信号', '希尔伯特变换结果');
```

程序执行后得出的原始信号和希尔伯特变换信号如图 3-10 所示。

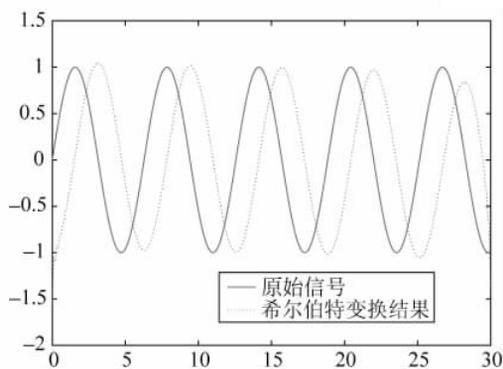


图 3-10 信号 $x(t) = \sin(t)$ 及其希尔伯特变换结果

2. 单边带调幅

去掉双边带幅度调制 (DSB-AM) 的一边就得到 SSB-AM。依据所保留的边带是上边还是下边, 可以分为 USSB 和 LSSB 两种不同的方式, 此时信号的时域表示为

$$u(t) = A_c m(t) \cos(2\pi f_c t) / 2 \mp A_c \hat{m}(t) \sin(2\pi f_c t) / 2 \quad (3-13)$$

在频域表示为

$$U_{\text{USSB}}(f) = \begin{cases} M(f - f_c) + M(f + f_c), & f_c \leq |f| \\ 0, & \text{其他} \end{cases} \quad (3-14)$$

$$U_{\text{LSSB}}(f) = \begin{cases} M(f - f_c) - M(f + f_c), & f_c \leq |f| \\ 0, & \text{其他} \end{cases} \quad (3-15)$$

这里 $\hat{m}(t)$ 是 $m(t)$ 的希尔伯特变换, 定义为 $\hat{m}(t) = m(t) * (1/\pi t)$, 频域表示为 $\hat{m}(f) = -j \text{sgn}(f) M(f)$ 。SSB 幅度调制占 DSB-AM 一半的带宽, 即等于信号带宽: $B_T = W$ 。

【例 3-6】 设基带信号为一个在 150~400Hz 内、幅度随频率逐渐递减的音频信号, 载波信号为 1000Hz 的正弦波, 幅度为 1, 仿真采样频率设为 10 000Hz, 仿真时间为 1s。求 SSB 调制输出信号波形和频谱。

其实现的 MATLAB 程序代码如下：

```
>> clear all;
Fs = 10000;           % 仿真的采样频率
t = 1/Fs:1/Fs:1;     % 仿真时间点
```

```

m_t(Fs * 1) = 0; % 基带信号变量初始化
for f = 150:400 % 基带信号发生: 频率 150~400Hz
    m_t = m_t + 0.01 * sin(2 * pi * f * t) * (400 - f); % 幅度随线性递减
end
m_t90shift = imag(hilbert(m_t)); % 基带信号的希尔伯特变换
carriercos = cos(2 * pi * 1000 * t); % 1000Hz 载波 cos
carriersin = sin(2 * pi * 1000 * t); % 1000Hz 正交载波 sin
S_SSB1 = m_t .* carriercos - m_t90shift .* carriersin; % 上边带 SSB
S_SSB2 = m_t .* carriercos + m_t90shift .* carriersin; % 下边带 SSB
% 下面作出各波形以及频谱
figure;
subplot(421);
plot(t(1:100), carriercos(1:100), t(1:100), carriersin(1:100), 'm'); % 载波
subplot(422);
plot([0:9999], abs(fft(carriercos))); % 载波频谱
axis([0 2000 - 500 12000]);
subplot(423);
plot(t(1:100), m_t(1:100)); % 基带信号
subplot(424);
plot([0:9999], abs(fft(m_t))); % 载波频谱
axis([0 2000 - 500 12000]);
subplot(425);
plot(t(1:100), S_SSB1(1:100)); % SSB 波形上边带
subplot(426);
plot([0:9999], abs(fft(S_SSB1))); % SSB 波形上边带
axis([0 2000 - 500 12000]);
subplot(427);
plot(t(1:100), S_SSB2(1:100)); % SSB 波形下边带
subplot(428);
plot([0:9999], abs(fft(S_SSB2))); % SSB 波形下边带
axis([0 2000 - 500 12000]);

```

运行程序,效果如图 3-11 所示,其中作出了 0~0.01s 内的信号时域波形和 0~2000Hz 内的幅度频谱。由图可知,单边带调制是对基带信号的线性频谱搬移,调制前后频谱仅仅是位置发生变化,频谱形状没有改变。但是,基带信号和单边带调制输出信号在时域波形上没有简单的对应关系。

3. 单边带解调

单边带信号的解调方法是双边带解调,设接收机中本地载波为

$$c(t) = \cos[2\pi(f_c + \Delta f)t + \Delta\phi] \quad (3-16)$$

式中: Δf 和 $\Delta\phi$ 分别为本地载波和发送端调制载波之间的频率误差和相位误差。双边带解调器的相乘输出信号为

$$\begin{aligned}
 s_{\text{DSB}}(t)c(t) &= \frac{A}{2} \sum_{n=0}^{\infty} a_n \cos[2\pi(f_c + nf)t + \phi_n] \cos[2\pi(f_c + \Delta f)t + \Delta\phi] \\
 &= \frac{A}{2} \sum_{n=0}^{\infty} a_n \cos\{[2\pi(nf - \Delta f)t + (\phi_n - \Delta\phi)]\} + \text{高频分量} \quad (3-17)
 \end{aligned}$$

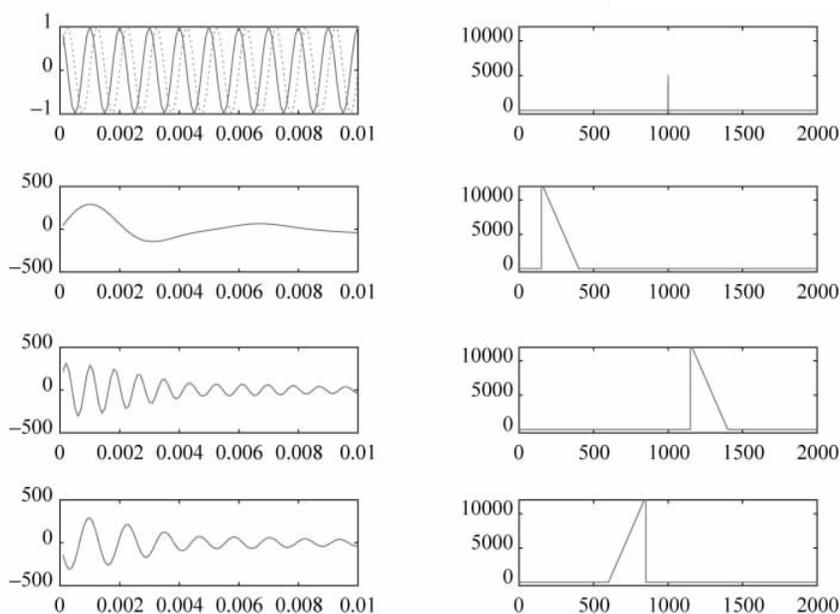


图 3-11 利用希尔伯特变换进行单边带调制的信号波形及对应幅度频谱

经过低通滤波器后,高频分量被滤除,最后得到解调输出为

$$\hat{m}(t) = \frac{A}{2} \sum_{n=0}^{\infty} a_n \cos[2\pi(nf - \Delta f)t + (\phi_n - \Delta\phi)] \quad (3-18)$$

对比发送基带信号 $m(t)$,解调输出信号中的频率分量存在一定的频率偏移和相位偏移。人耳对于语音波形的相位失真是不敏感的,频率失真会影响语音音色,但若频率偏移较小(几赫兹到几十赫兹内),对语音的可懂度就不会造成大的影响。在实际的语音单边带通信机中,一般采用一个高稳定度的晶体振荡器或频率合成器来产生本地解调载波,而不像双边带的解调那样需要用锁相环(PLL)来恢复载波,这就大大降低了单边带接收机的技术复杂度和成本。

【例 3-7】 对例 3-6 产生的单边带(上边带)信号进行相干解调,仿真其解调波形和幅度频谱。

其仿真程序代码如下:

```
>> clear all;
FS = 10000;
t = 1/FS:1/FS:1;
m_t(FS * 1) = 0; %基带信号变量初始化
for f = 150:400 %基带信号发生: 频率 150~400Hz
    m_t = m_t + 0.01 * sin(2 * pi * f * t) * (400 - f); %幅度随线性递减
end
m_t90shift = imag(hilbert(m_t)); %基带信号的希尔伯特变换
carriercos = cos(2 * pi * 1000 * t); %1000Hz 载波 cos
carriersin = sin(2 * pi * 1000 * t); %1000Hz 正交载波 sin
S_SSB1 = m_t * carriercos - m_t90shift * carriersin; %上边带 SSB
out = S_SSB1 * carriercos; %相干解调
[a,b] = buffer(4,500/(FS/2)); %低通滤波设计为 4 阶,截止频率为 500Hz
demsig = filter(a,b,out); %解调输出
```

```

% 下面作出各滤波波形以及频谱
figure(1);
subplot(321);
plot(t(1:100),S_SSB1(1:100));           % SSB 波形
subplot(322);
plot([0:9999],abs(fft(S_BBS1)));       % SSB 频谱
axis([0 2000 - 500 12000]);
subplot(323);
plot(t(1:100),out(1:100));           % 相干解调波形
subplot(324);
plot([0:9999],abs(fft(out)));         % 相干解调频谱
axis([0 2000 - 500 12000]);
subplot(325);
plot(t(1:100),demsig(1:100));        % 低通输出信号
subplot(326);
plot([0:9999],abs(fft(demsig)));      % 低通输出频谱
axis([0 2000 - 500 12000]);

```

单边带信号的相干解调中的低通滤波器用于将相关乘法器输出的载波二次谐波分量滤除,程序中滤波器设计为4阶巴特沃斯低通,截止频率为500Hz。程序执行后输出的解调波形和幅度频谱如图3-11所示。对比图3-12中的发送基带信号,可见解调输出时域波形是发送基带信号波形的近似。如果单边带解调时使用的本地载波与发送调制载波之间存在频差和相位差,那么解调输出的时域波形将产生严重失真。但是,解调信号的幅度谱与发送基带信号幅度谱之间失真不大。对于语音信号,实验表明,单边带解调相干载波频差和相位差引起的解调波形失真对话音信号的可懂度影响较小。下面的程序仿真了单边带解调时本地载波与发送调制载波之间存在频差和相位差的情况,仿真结果如图3-13所示。

```

>> clear all;
FS = 10000;
t = 1/FS:1/FS:1;
m_t(FS * 1) = 0;           % 基带信号变量初始化
for f = 150:400           % 基带信号发生: 频率 150~400Hz
    m_t = m_t + 0.01 * sin(2 * pi * f * t) * (400 - f); % 幅度随线性递减
end
m_t90shift = imag(hilbert(m_t)); % 基带信号的希尔伯特变换
carriercos = cos(2 * pi * 1000 * t); % 1000Hz 载波 cos
carriersin = sin(2 * pi * 1000 * t); % 1000Hz 正交载波 sin
S_SSB1 = m_t. * carriercos - m_t90shift. * carriersin; % 上边带 SSB
out = S_SSB1. * cos(2 * pi * 2018 * t + 1); % 存在频率误差的相位误差时间的相干解调
[a,b] = buffer(4,500/(FS/2)); % 低通滤波设计 4 阶,截止频率为 500Hz
demsig = filter(a,b,out); % 解调输出
% 下面作出各滤波波形以及频谱
figure(1);
subplot(321);
plot(t(1:100),S_SSB1(1:100));           % SSB 波形
subplot(322);
plot([0:9999],abs(fft(S_SSB1)));       % SSB 频谱
axis([0 2000 - 500 12000]);
subplot(323);
plot(t(1:100),out(1:100));           % 相干解调波形
subplot(324);
plot([0:9999],abs(fft(out)));         % 相干解调频谱

```

```
axis([0 2000 - 500 12000]);
subplot(325);
plot(t(1:100),demsig(1:100)); % 低通输出信号
subplot(326);
plot([0:9999],abs(fft(demsig))); % 低通输出频谱
axis([0 2000 - 500 12000]);
```

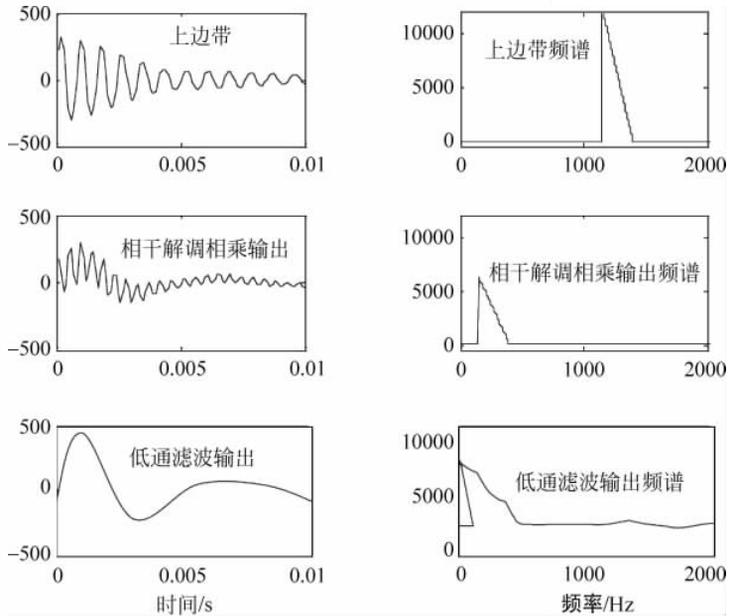


图 3-12 单边带信号相干解调波形及对应幅度频谱

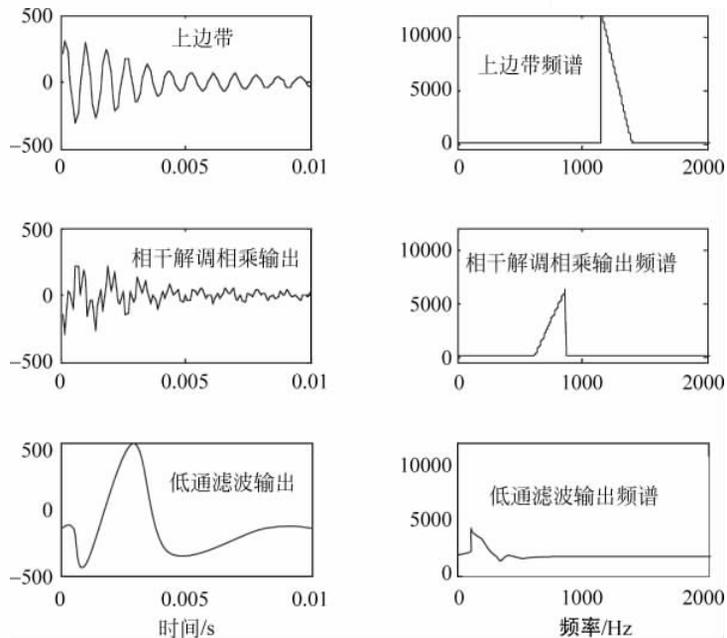


图 3-13 存在频差和相位差情况下的单边带信号相干解调波形及对应幅度频谱