

# 1 AM waves

## 1.1 Generation

```
clearvars; close all;
global t;
fm = 1;
fc = 10;
Ac = 5;
t = -2:.001:2;
msg = sin(2*fm*pi.*t);
carrier = Ac * cos(2*pi*fc.*t);

m = .1; Aml = (1 + m.*msg) .* carrier;
m = 1; Ame = (1 + m.*msg) .* carrier;
m = 3; Amg = (1 + m.*msg) .* carrier;

plotwave(msg, 1, "Message signal");
plotwave(carrier, 2, "Carrier signal");
plotwave(Aml, 3, "Modulated wave m = 0.1");
plotwave(Ame, 4, "Modulated wave m = 1");
plotwave(Amg, 5, "Modulated wave m = 3");

function plotwave(fx, index, heading)
    global t;
    subplot(5, 1, index);
    plot(t, fx);
    title(heading);
    xlabel("t");
end
```

## 1.2 Various Modulations

```
clearvars; close all;
global t;
global f;

fs = 500;
t = 0:1/fs:1-1/fs;
f = fs/2* linspace(-1, 1, fs);

fm = 5;
fc = 20;
Ac = 5;

msg = sin(2*fm*pi.*t);
carrier = Ac * cos(2*pi*fc.*t);

dsbsc = msg .* carrier;
```

```

dsbfc = (1 + 2*msg) .* carrier;
m2 = cos(2*fm*pi.*t);
ssbsc = msg .* carrier + Ac * m2 .* sin(2*pi*fc.*t);

plotsg(msg, 1, "message signal");
plotsg(carrier, 3, "Carrier signal");
plotsg(dsbfc, 5, "DSB SC");
plotsg(dsbfc, 7, "DSB FC");
plotsg(ssbsc, 9, "SSB SC");

function plotsg(fx, index, heading)
    global t;
    global f;

    subplot(5, 2, index)
    plot(t, fx);
    title(heading)
    xlabel("t");

    fx_fft = abs(fftshift(fft(fx))) / length(t);
    subplot(5, 2, index + 1)
    plot(f, fx_fft);
    title(heading + " - Spectrum");
    xlabel("f (Hz)")
    xlim([0 50]);
end

```

### 1.3 Demodulation

```

%% Square law demodulator
% AM wave -> square it -> LPF -> demodulation

clearvars; close all;
global t;
global f;

fs = 800;
t = 0:1/fs:1-1/fs;
f = fs/2* linspace(-1, 1, fs);

fm = 5;
fc = 20;
Ac = 5;

msg = sin(2 * fm * pi .* t);
carrier = Ac * cos(2 * pi * fc .* t);

am = (1 + 1 * msg) .* carrier;

```

```

[num, den] = butter(4, fm/fs*2, 'low');
sq = am .* am;
y = filter(num, den, sq);
y = y - mean(y);

plotsg(msg, 1, "Message signal");
plotsg(carrier, 3, "Carrier wave");
plotsg(am, 5, "AM wave");
plotsg(sq, 7, "Square of AM");
plotsg(y, 9, "Demodulated wave");

function plotsg(fx, index, heading)
    global t;
    global f;

    subplot(5, 2, index)
    plot(t, fx);
    title(heading)
    xlabel("t");

    fx_fft = abs(fftshift(fft(fx))) / length(t);
    subplot(5, 2, index + 1)
    plot(f, fx_fft);
    title(heading + " - Spectrum");
    xlabel("f (Hz)")
    xlim([0 50]);
end

```

## 2 FM modulation

### 2.1 modulation

```
clearvars; close all;
global t; global f; global index;
fs = 5000;
index = 1;
t = linspace(0, 1, fs);
f = fs/2 * linspace(-1, 1, fs);
fm = 2;
fc = 50;
Ac = 5;
fv = 20;

msg = sin(2*fm*pi.*t);
carrier = Ac * cos(2*pi*fc.*t);
msg_int = cumsum(msg) / fs;
fm = Ac * cos(2*pi*fc.*t + 2*pi*f*fv*msg_int);

plotFxSpec(msg, "message");
plotFxSpec(carrier, "Carrier");
plotFxSpec(fm, "FM wave");

function plotFxSpec(fx, heading)
    global index; global t; global f;
    subplot(3, 2, index);
    plot(t, fx);
    ylim([min(fx), max(fx)] * 1.1);
    title(heading);

    fx_fft = abs(fftshift(fft(fx))) / length(t);
    subplot(3, 2, index + 1);
    plot(f, fx_fft);
    title(heading + " - Spectrum");
    xlabel("f (Hz)");
    xlim([0 100]);
    index = index + 2;
end
```

### 2.2 demodulation

```
clearvars; close all;
global t; global f; global index;

fs = 5000;
index = 1;
t = linspace(0, 1, fs);
f = fs/2 * linspace(-1, 1, fs);
```

```

fm = 2;
fc = 50;
Ac = 2;
fv = 20;
msg = sin(2 * fm * pi .* t);
carrier = Ac * cos(2 * pi * fc .* t);
msg_int = cumsum(msg) / fs;
fm = Ac * cos(2*pi*fc.*t + 2*pi*fv*msg_int);

z= hilbert(fm);
inst_phase = unwrap(angle(z));
offsetTerm = 2 * pi * fc * t;
demodulated = diff(inst_phase - offsetTerm);
demodulated = [0, demodulated];

plotFxSpec(msg, "Message");
plotFxSpec(fm, "FM wave");
plotFxSpec(demodulated, "Demodulated wave");
function plotFxSpec(fx, heading)
    global index; global t; global f;
    subplot(3, 2, index);
    plot(t, fx);
    ylim([min(fx), max(fx)] * 1.1);
    title(heading);

    fx_fft = abs(fftshift(fft(fx))) / length(t);
    subplot(3, 2, index + 1)
    plot(f, fx_fft);
    title(heading + " - Spectrum");
    xlabel("f (Hz)")
    xlim([0 100]);
    index = index + 2;
end

```

## 3 PAM

### 3.1 PAM modulation

```
clearvars; close all;
global t; global index;

fs = 5000;
fm = 1;
fp = 10;
t = linspace(0, 2, fs);
index = 1;

msg = sin(2 * pi * fm .* t);
sq = square(2 * pi * fp * t) / 2 + .5;
pam = msg .* sq;

plotFx(msg, "Message");
plotFx(sq, "Pulse train");
plotFx(pam, "Modulated wave");

function plotFx(fx, heading)
    global index; global t;
    subplot(3, 1, index);
    plot(t, fx);
    ylim([min(fx), max(fx)] * 1.1);
    title(heading);
    index = index + 1;
end
```

### 3.2 PAM demodulation

```
clearvars; close all;
global t; global index;

fs = 5000;
fm = 1;
fp = 10;
t = linspace(0, 3, fs);
index = 1;

msg = sin(2 * pi * fm .* t);
sq = square(2 * pi * fp * t) / 2 + .5;
pam = msg .* sq;

rectWin = rectpuls(t, 2/fp);
windowed = conv(pam, rectWin);
windowed = windowed(1:ceil(length(windowed) / 2));
[num, den] = butter(1, (fm+fp)/2/fs, 'low');
```

```
dem = filter(num, den, windowed);

plotFx(msg, "Message");
plotFx(pam, "PAM wave");
plotFx(windowed, "Windowed wave");
plotFx(dem, "Demodulated wave");

function plotFx(fx, heading)
    global index; global t;
    subplot(4, 1, index);
    plot(t, fx);
    ylim([min(fx), max(fx)] * 1.1);
    title(heading);
    index = index + 1;
end
```

## 4 BSK

### 4.1 BASK

```
clearvars; close all;
fc = 5;
Eb = 1; Tb = 1;
fs = 100;
msgbits = [1 0 1 1 0 1 0];

t = linspace(0, length(msgbits), fs * length(msgbits));
msgsampled = kron(msgbits, ones(1, fs));

s1 = sqrt(2*Eb/Tb) * cos(2*pi*fc.*t);

bask = s1 .* msgsampled;
N = length(msgsampled);
f = fs * (-N/2:N/2-1) / N;
fx_fft = fftshift(abs(fft(bask))) / N;

tiledlayout(4, 1);
plot_(t, msgsampled, "Message");
plot_(t, s1, "Carrier (5 Hz)");
plot_(t, bask, "BASK");
plot_(f, fx_fft, "BASK - Spectrum");
xlabel("f (Hz)")
xlim([0 10]);

function plot_(t, fx, title_)
    nexttile;
    plot(t, fx);
    k = (max(fx) - min(fx)) * 0.1;
    ylim([min(fx) - k, max(fx) + k]);
    title(title_);
end
```

### 4.2 BPSK

```
clearvars; close all;
Eb = 1; Tb = 1;
fs = 100; fc = 5;
msgbits = [1 0 1 0 1];

t = linspace(0, length(msgbits), fs * length(msgbits));
msgsampled = kron(msgbits, ones(1, fs));

s1 = sqrt(2*Eb/Tb) * cos(2*pi*fc.*t);
s2 = sqrt(2*Eb/Tb) * cos(2*pi*fc.*t+pi);
```



```

bpsk = s1 .* msgsampled + s2 .* ~msgsampled;
N = length(msgsampled);
f = fs * (-N/2:N/2-1) / N;
fx_fft = fftshift(abs(fft(bpsk))) / N;

tiledlayout(5, 1);
plot_(t, msgsampled, "Message");
plot_(t, s1, "Carrier 1");
plot_(t, s2, "Carrier 2");
plot_(t, bpsk, "BPSK");
plot_(f, fx_fft, "BPSK - Spectrum");
xlabel("f (Hz)")
xlim([0 15]);

function plot_(t, fx, title_)
    nexttile;
    plot(t, fx);
    k = (max(fx) - min(fx)) * 0.1;
    ylim([min(fx) - k, max(fx) + k]);
    title(title_);
end

```

### 4.3 BFSK

```

clearvars; close all;
fc1 = 5; fc2 = 8;
Eb = 1; Tb = 1;
fs = 100;
msgbits = [1 0 1 1 0 1 0];
t = linspace(0, length(msgbits), ...
    fs * length(msgbits));
msgsampled = kron(msgbits, ones(1, fs));

s1 = sqrt(2*Eb/Tb) * cos(2*pi*fc1.*t);
s2 = sqrt(2*Eb/Tb) * cos(2*pi*fc2.*t);
bfsk = s1 .* msgsampled + s2 .* ~msgsampled;
N = length(msgsampled);
f = fs * (-N/2:N/2-1) / N;
fx_fft = fftshift(abs(fft(bfsk))) / N;

tiledlayout(5, 1);
plot_(t, msgsampled, "Message");
plot_(t, s1, "Carrier1 (5 Hz)");
plot_(t, s2, "Carrier2 (8 Hz)");
plot_(t, bfsk, "BFSK");
plot_(f, fx_fft, "BFSK - Spectrum");
xlabel("f (Hz)")
xlim([0 15]);

```

```
function plot_(t, fx, title_)
    nexttile;
    plot(t, fx);
    k = (max(fx) - min(fx)) * 0.1;
    ylim([min(fx) - k, max(fx) + k]);
    title(title_);
end
```

## 5 Performance measure of PSK

```
clearvars;

msg2 = randi([0,1],100000,1);
msg4 = randi([0,3],100000,1);
snrs = 0:100;

[er2, psk2, demod2] = modsys(msg2, 2, snrs);
[er4, psk4, demod4] = modsys(msg4, 4, snrs);

subplot(2, 2, 1); plot(snrs, er2, snrs, er4);
legend("BPSK", "QPSK");
title("Symbol error rate vs output SNR");
xlabel("Output SNR"); ylabel("Symbol error rate (log)");
subplot(2, 2, 2); periodogram(psk2); hold on;
periodogram(psk4); hold off;

subplot(2, 2, 3); scatter(real(demod2), imag(demod2), ".");
axis([-2, 2, -2, 2]); title("BPSK constellation")
subplot(2, 2, 4); scatter(real(demod4), imag(demod4), ".");
axis([-2, 2, -2, 2]); title("QPSK constellation")

function [errs, psk, demod] = modsys(msg, M, snrs)
    psk = pskmod(msg, M, pi/M);
    errs = [];
    for i = snrs
        rxSig = awgn(psk, i);
        demod = pskdemod(rxSig, M, pi/M);
        [~, er] = symerr(msg, demod);
        errs = [errs, log10(er)];
    end
    demod = awgn(psk, 20);
end
```

## 6 Line Codes

```
clearvars;

bits = [0 1 0 0 1 1 0 1];
Tb = 100;
nprz = kron(bits, ones(1, Tb));
nprz = kron(bits, [ones(1, Tb/2), zeros(1, Tb/2)]);
prz = kron(bits * 2 - 1, ones(1, Tb));
prz = kron(bits * 2 - 1, [ones(1, Tb/2), zeros(1, Tb/2)]);
manch = kron(bits * 2 - 1, [-ones(1, Tb/2), ones(1, Tb/2)]);
bipolar = [];
n = 1;
for b = bits
    bipolar = [bipolar, b * n];
    if (b == 1)
        n = n * -1;
    end
end

bitst = [0, bits];
diffnt = ones(1, length(bitst));
for i = 2:length(bitst)
    diffnt(i) = ~xor(bitst(i), diffnt(i-1));
end

tiledlayout(8, 1);
stairz([bits, 0], "Input");
for i = 1:length(bits)
    text(i+0.5, 0.5, string(bits(i)), 'FontSize', 15);
end
stairz(nprz, "NPNRZ");
stairz(nprz, "NPRZ");
stairz(pnrz, "PNRZ");
stairz(prz, "PRZ");
stairz(manch, "Manchester");
stairz([bipolar, 0], "bipolar");
stairz([diffnt(2:end), 0], "Differential");

function stairz(bits, title_)
    nexttile;
    stairs(bits);
    title(title_);
    d = (max(bits) - min(bits)) * 0.2;
    ylim([min(bits)-d, max(bits)+d]);
end
```