**基于亮点模型的潜艇回波仿真（直线阵）**

***参考文献:***

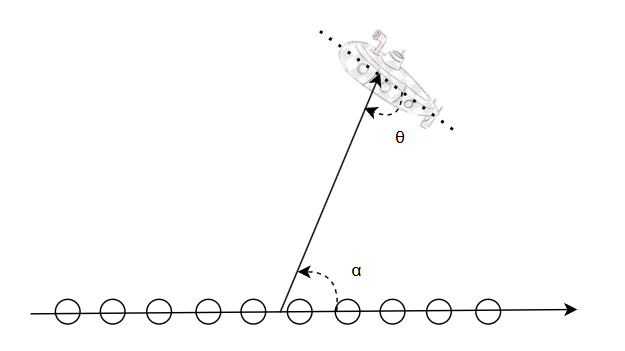
***Simulation of Echoes from Submarine in Shallow Waters Rongguang Sun, Xin Ma, XiangIan Shu***

***Submarine Echo Simulation Method Based on the Highlight Model Xiao Chen Ya-an Li***

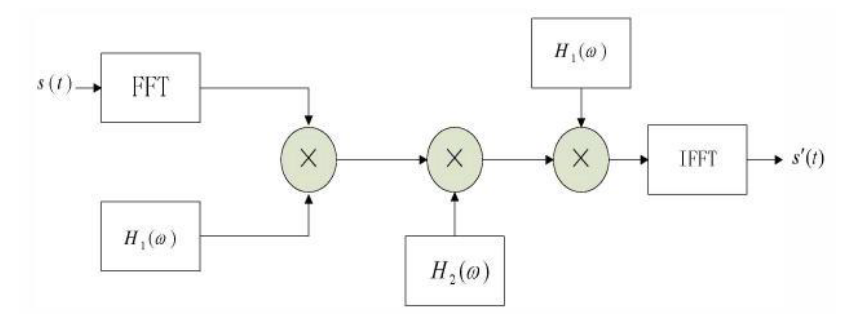
***基于亮点模型的潜艇回波仿真 何心怡 蒋兴舟 林建域***

***一种基于亮点模型的潜艇回波仿真 董仲臣 ,李亚安,陈 晓***

clc;clear;close all;



和信道决定回波形状，为设定的波束角度



H1为信道转移函数，H2为潜艇亮点模型转移函数

# 1.仿真发射信号

%入射潜艇的角度

theta = 10;

fs = 20e3;

fc = 5e3;

% 定义时间参数

t = 0:1/fs:20 - 1/fs; % 时间向量，这里生成0到20秒的时间序列，采样间隔为1/fs

N = length(t);

f = -fs/2:fs/N:fs/2-fs/N;

Ts = 1/fs;

生成包络

% 定义梯形脉冲参数

Am = 1; % 幅度为1V

%总脉宽120ms，上升10ms，下降10ms

width = 100e-3; % 宽度为100ms

rise\_time = 10e-3; % 上升时间设为10ms

fall\_time = 10e-3; % 下降时间设为10ms

a = genTrapSig(Am,t,width,rise\_time,fall\_time);

%绘制梯形脉冲信号

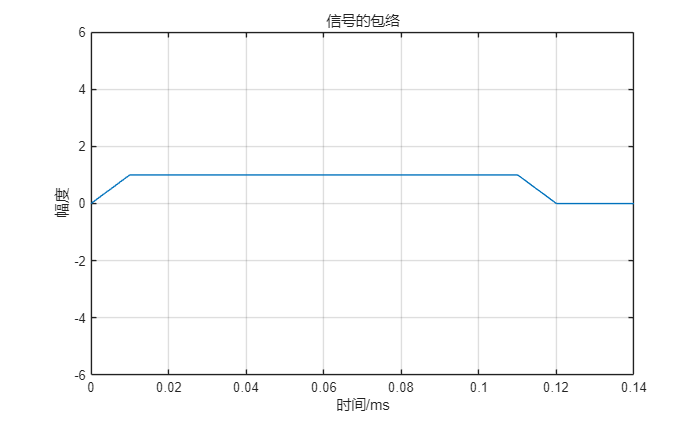
figure;

plot(t, a);

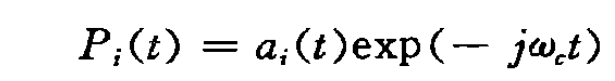
xlabel('时间/ms');ylabel('幅度');title('信号的包络');

grid on;

axis([0,0.14,-6,6]);



发射的窄带脉冲



wc = 2\*pi\*fc;

P = a.\*exp(j\*wc\*t);

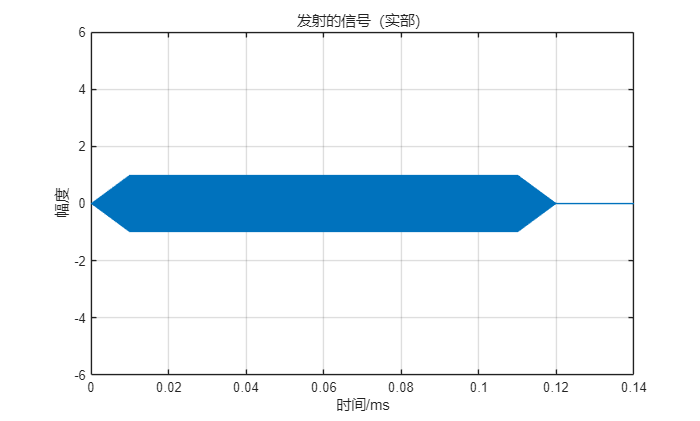
figure;

plot(t, real(P));

xlabel('时间/ms');ylabel('幅度');title('发射的信号（实部）');

grid on;

axis([0,0.14,-6,6]);



# 2.多普勒频移

潜艇运动产生的多普勒频移：



v = 4.112;%潜艇运动速度8海里/h 0.514\*8

c = 1500;

wd = 2\*wc\*v\*cosd(theta)/c;%多普勒角频率

PWithDoplar = P.\*exp(1j\*wd\*t);

figure

subplot(2,1,1)

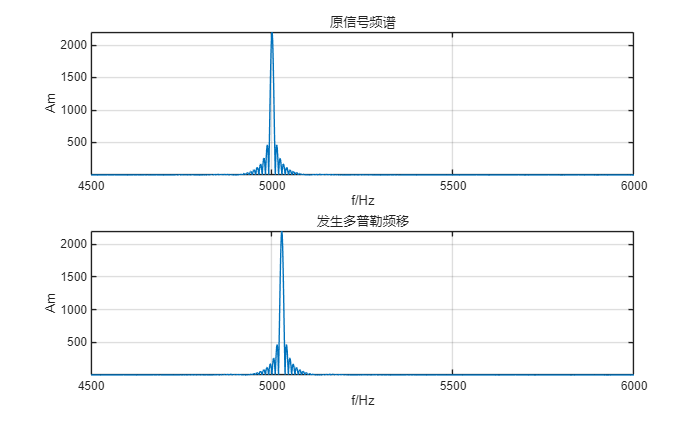
plot(f,abs(fftshift(fft(P))));xlabel('f/Hz');ylabel('Am');title('原信号频谱');grid on;

axis([4500,6000,-inf,inf]);

subplot(2,1,2)

plot(f,abs(fftshift(fft(PWithDoplar))));xlabel('f/Hz');ylabel('Am');title('发生多普勒频移');grid on;

axis([4500,6000,-inf,inf]);



P = PWithDoplar;%变量替换

# 3.使用Bellhop计算脉冲响应（修改中）

## 仿真深海场景

## 声速梯度

figure;

bellhop Env\MunkB\_Coh;clc;

plotssp('Env\MunkB\_Coh');

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Munk profile, coherent

Frequency = 5000 Hz

Number of media = 1

Spline approximation to SSP

Attenuation units: dB/wavelength

VACUUM

z (m) alphaR (m/s) betaR rho (g/cm^3) alphaI betaI

( Number of pts = 51 Roughness = 0.00 Depth = 5000.00 )

0.00 1548.52 0.00 1.00 0.0000 0.0000

200.00 1530.29 0.00 1.00 0.0000 0.0000

250.00 1526.69 0.00 1.00 0.0000 0.0000

400.00 1517.78 0.00 1.00 0.0000 0.0000

600.00 1509.49 0.00 1.00 0.0000 0.0000

800.00 1504.30 0.00 1.00 0.0000 0.0000

1000.00 1501.38 0.00 1.00 0.0000 0.0000

1200.00 1500.14 0.00 1.00 0.0000 0.0000

1400.00 1500.12 0.00 1.00 0.0000 0.0000

1600.00 1501.02 0.00 1.00 0.0000 0.0000

1800.00 1502.57 0.00 1.00 0.0000 0.0000

2000.00 1504.62 0.00 1.00 0.0000 0.0000

2200.00 1507.02 0.00 1.00 0.0000 0.0000

2400.00 1509.69 0.00 1.00 0.0000 0.0000

2600.00 1512.55 0.00 1.00 0.0000 0.0000

2800.00 1515.56 0.00 1.00 0.0000 0.0000

3000.00 1518.67 0.00 1.00 0.0000 0.0000

3200.00 1521.85 0.00 1.00 0.0000 0.0000

3400.00 1525.10 0.00 1.00 0.0000 0.0000

3600.00 1528.38 0.00 1.00 0.0000 0.0000

3800.00 1531.70 0.00 1.00 0.0000 0.0000

4000.00 1535.04 0.00 1.00 0.0000 0.0000

4200.00 1538.39 0.00 1.00 0.0000 0.0000

4400.00 1541.76 0.00 1.00 0.0000 0.0000

4600.00 1545.14 0.00 1.00 0.0000 0.0000

4800.00 1548.52 0.00 1.00 0.0000 0.0000

5000.00 1551.91 0.00 1.00 0.0000 0.0000

Number of pts = 51

ACOUSTO-ELASTIC half-space

5000.00 1600.00 0.00 1.80 0.8000 0.0000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Number of source depths, NSz = 1

Source depths, Sz (m)

1000.00

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Number of receiver depths, NRz = 501

Receiver depths, Rz (m)

0.00 ... 5000.00

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Number of receiver ranges, NRr = 100

Receiver ranges, Rr (km)

0.00 ... 10.00

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Coherent TL calculation

Geometric hat beams

Number of beams = 0

Nbeams calculated automatically, Nbeams = 10000

Beam take-off angles (degrees)

-20.300000 20.300000

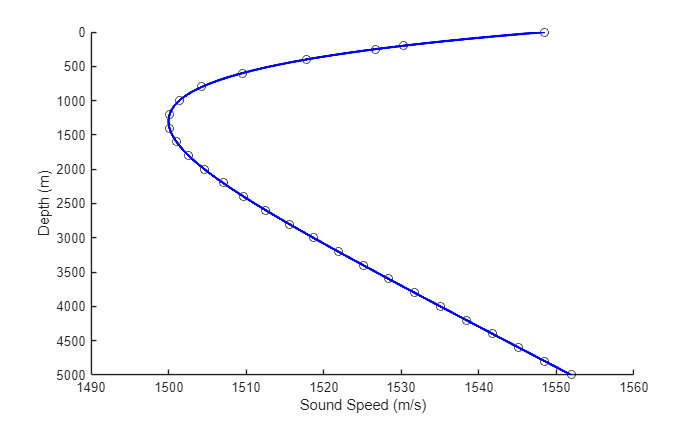
Step length, deltas = 50 m

Maximum ray depth, zBox = 5500 m

Maximum ray range, rBox = 10 km

No beam shift in effect

Point source (cylindrical coordinates)



根据声速梯度图，当深度为1000m时，声速为1500m/s

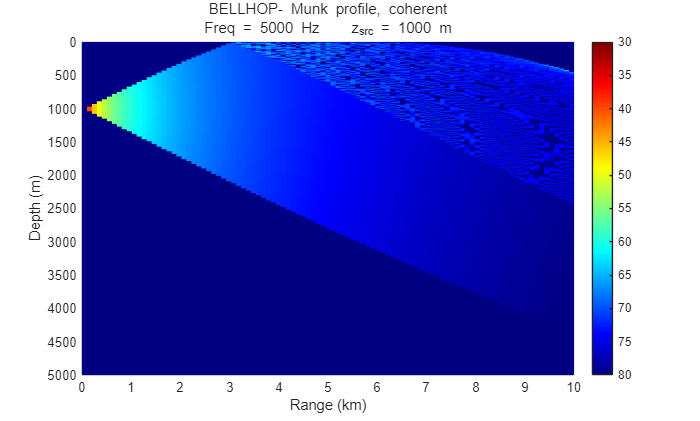
## 传播损失

%计算传播损失

bellhop('Env\MunkB\_Coh');

figure

plotshd('Env\MunkB\_Coh.shd');



## 声线传播轨迹

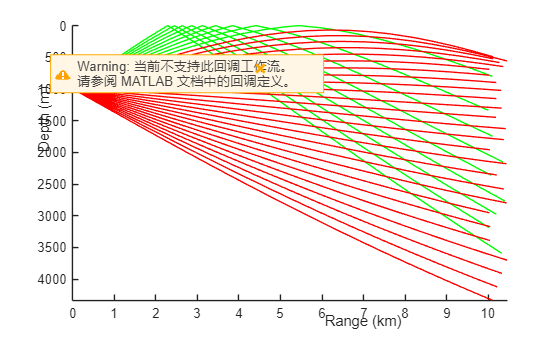
bellhop('Env\MunkB\_ray');

figure

global units;

units = 'km';%这样输出单位为km

plotray('Env\MunkB\_ray.ray');



## 脉冲响应

bellhop计算脉冲响应

bellhop ('Env\MunkB\_Arr');

figure

%plotarr('testEnvs\MunkB\_Arr.arr',1,1,1);grid on;

[ Arr, Pos ] = read\_arrivals\_asc( 'Env\MunkB\_Arr.arr' );

Amp = real(Arr.A);%到达幅度

delay = Arr.delay;%到达时延

Amp\_Delay = [delay;Amp];

Amp\_Delay(:,all(Amp\_Delay==0,1))=[]; %去掉0值

Amp\_Delay=sortrows(Amp\_Delay',1); %按照时延从小到大排序

% figure

% stem(Amp\_Delay(:,1),Amp\_Delay(:,2));

t = 0:1/fs:20 - 1/fs; % 时间向量，这里生成0到0.14秒的时间序列，采样间隔为1/fs

IR = zeros(1,length(t));

%将bellhop得到的序列插入进单位脉冲响应中

delayIndex = round(Amp\_Delay(:,1)./Ts);

for i = 1:length(delayIndex)

IR(delayIndex(i)) = Amp\_Delay(i,2);

end

TF1 = fftshift(fft(IR));

f = -fs/2:fs/N:fs/2-fs/N;

figure

subplot(2,1,1)

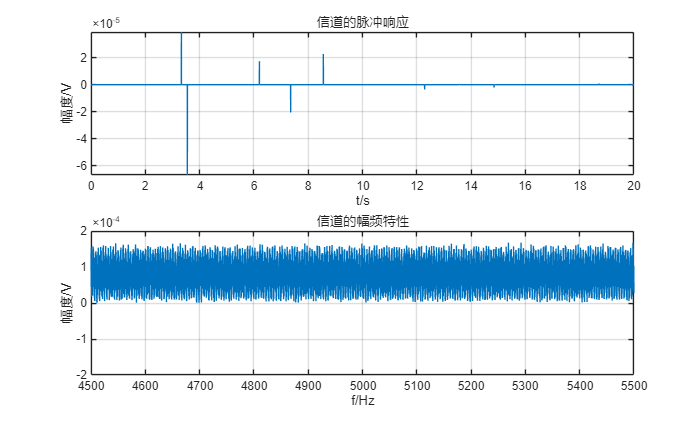
plot(t,IR);xlabel('t/s');ylabel('幅度/V');grid on;

axis([0,20,-inf,inf]);title('信道的脉冲响应');

subplot(2,1,2)

plot(f,abs(TF1));xlabel('f/Hz');ylabel('幅度/V');grid on;

axis([4500,5500,-20e-5,20e-5]);title('信道的幅频特性');



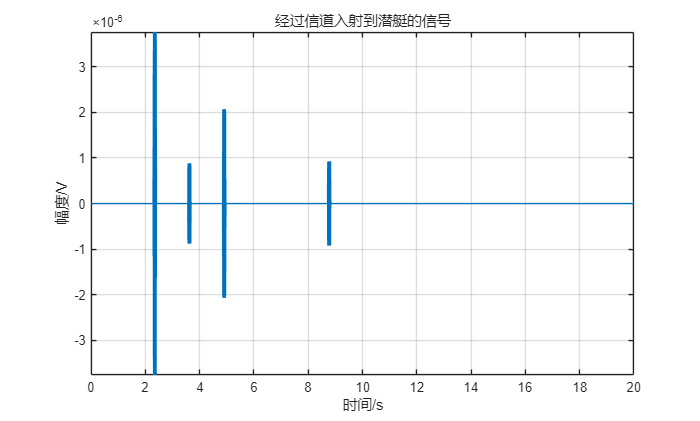
信号经过信道

PAfterChan = conv(real(IR),P,'same');

figure

plot(t,real(PAfterChan));xlabel('时间/s');ylabel('幅度/V');grid on;

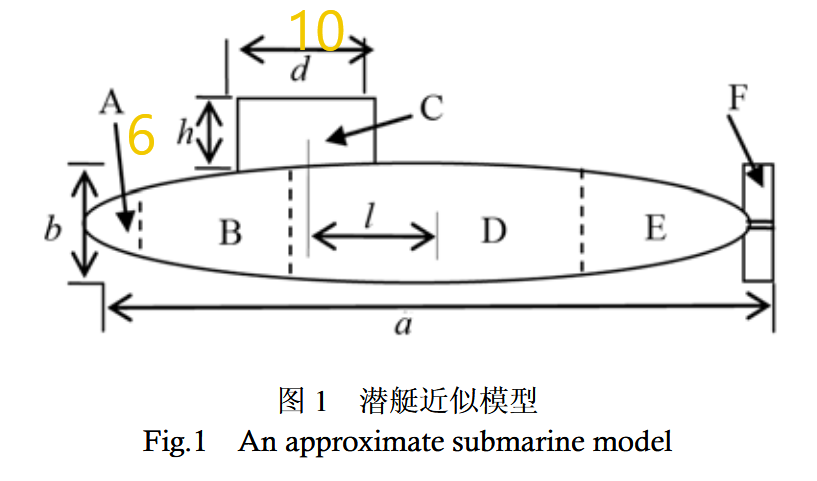
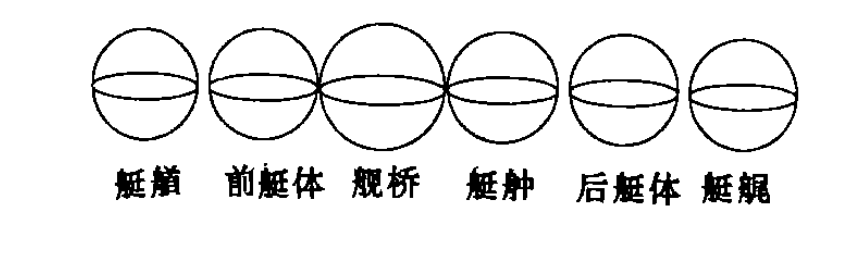
title('经过信道入射到潜艇的信号');axis([0,20,-inf,inf]);

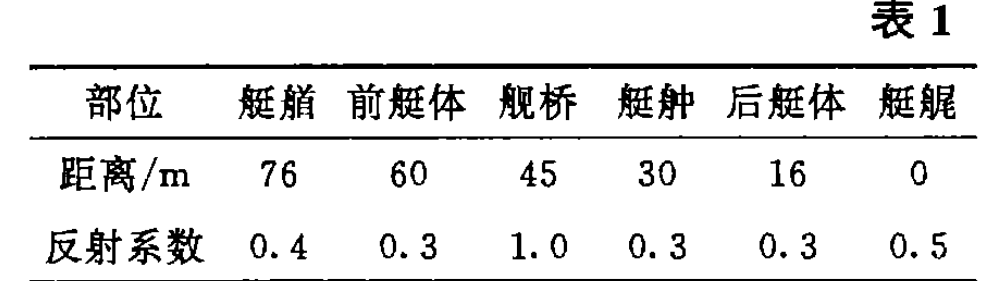


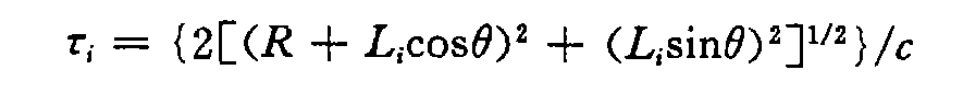
P = PAfterChan;

# 4.每个亮点的时延

R为鱼雷与潜艇距离,即鱼雷与代表艇尾的亮点距离;0为照射角;表示该亮点与艇艉的距离







tauHighLight = zeros(6,1);

L = zeros(6,1); %L存放亮点与艇艇的距离

L = [76, 60, 45, 30, 16, 0];

R = 5000; %两者的距离5km

c = 1500; %深度1000m 声速1500m/s

% for j = 1:6

% term1 = (R + L(j)\*cos(theta\*pi/180)).^2;

% term2 = (L(j)\*sin(theta\*pi/180))^2;

% numerator = 2\*sqrt(term1 + term2);

% tau(j) = numerator/c;

% end

for j = 1:6

term1 = (L(j)\*cosd(theta)).^2;

term2 = (L(j)\*sind(theta))^2;

numerator = 2\*sqrt(term1 + term2);

tauHighLight(j) = numerator/c; %此时c为1500m/s 深度1000m

end

亮点的时延

for i = 1:6

fprintf('%f s\n', tauHighLight(i));

end

0.101333 s

0.080000 s

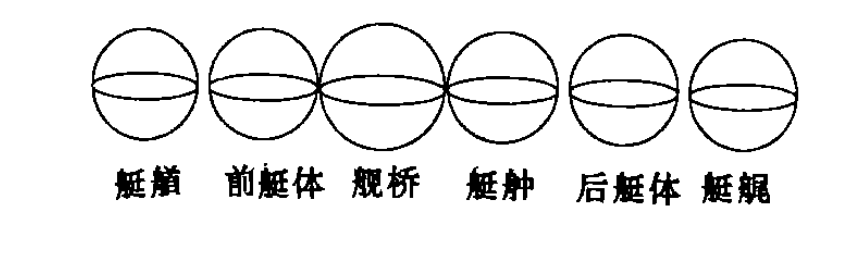
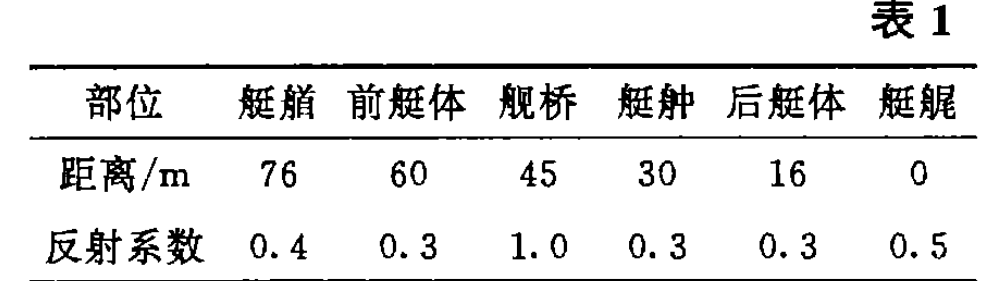
0.060000 s

0.040000 s

0.021333 s

0.000000 s

# 5.计算亮点的子回波幅度



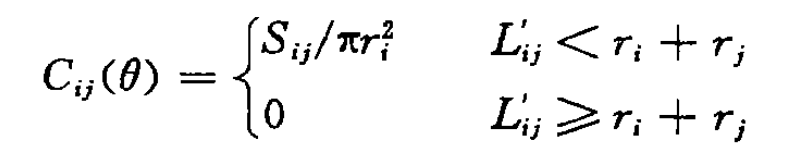
计算每个亮点的半径（这里数据不全）

% 定义距离和反射系数

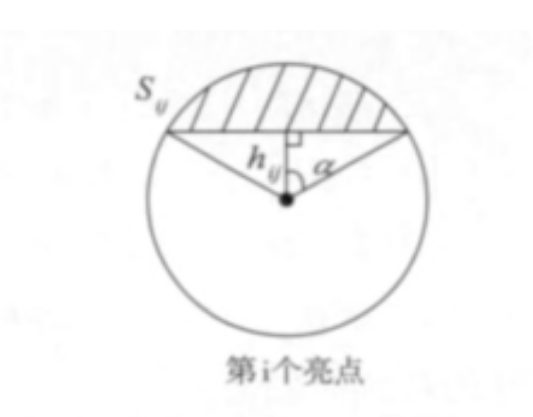
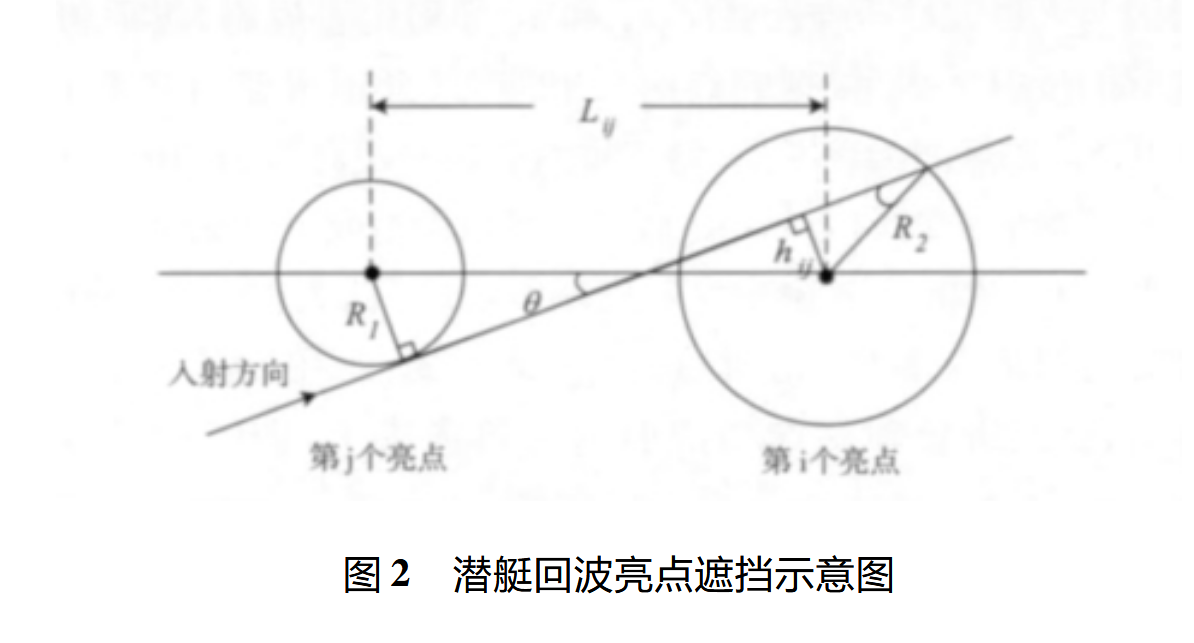
b = [0.4, 0.3, 1.0, 0.3, 0.3, 0.5];

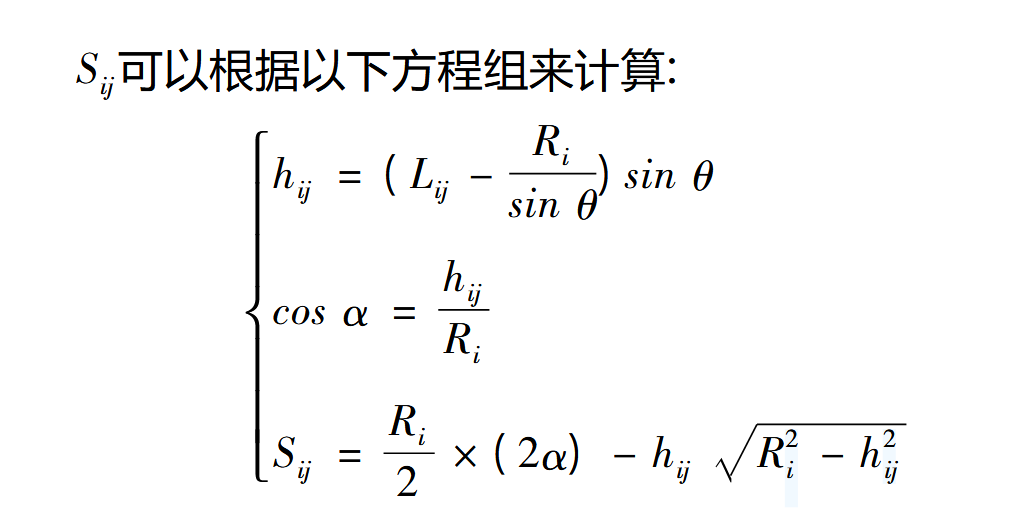
r = [7,7,8,7,7,7];

## 计算隐蔽系数



计算Sij





L\_ij = zeros(6,6);

h\_ij = zeros(6,6);

S\_ij = zeros(6,6);

for i =1:6

for j = 1:6

L\_ij(i,j) =abs( L(j) - L(i) );

h\_ij(i,j) = (L\_ij(i,j) - r(i)/(sind(theta)+eps) ).\*sind(theta);

gamma = acos(h\_ij(i,j)/r(i));

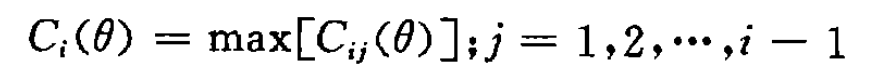
S\_ij(i,j) = r(i).^2 /2.\*(2\*gamma) - h\_ij(i,j)\*sqrt(r(i).^2 - h\_ij(i,j).^2);

% C\_ij(i,j) = S\_ij(i,j)./(pi\*r(i).^2);

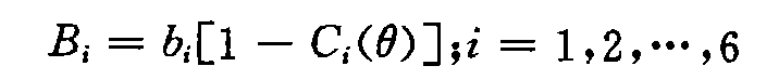
end

end

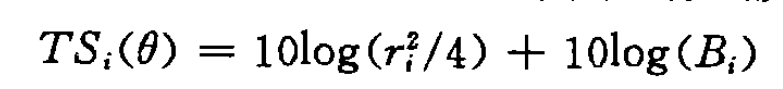
任何一个亮点最多可能被5个亮点所遮挡，最多也就有5个隐蔽系数,那么亮点i的隐蔽系数等于其中最大的隐蔽系数:

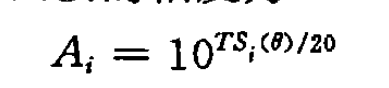


一个亮点的实际反射系数为



则一个亮点在考虑有隐蔽情况下其目标强度为



该亮点子回波的幅度为

% 计算隐蔽系数

C\_ij = zeros(6,6);

for i = 1:6

for j = 1:6

if abs( L\_ij(i,j)\*sind(theta) ) > r(i) + r(j) %传播方向上的投影长度小于亮点半径之和

C\_ij(i, j) = S\_ij(i,j)/(pi \* (r(i).^2) ); % 由于没有S\_ij数据，这里设为0

else

C\_ij(i, j) = 0;

end

end

end

% 计算每个亮点的实际反射系数

B = zeros(1, 6);

for i = 1:6

C(i) = max(C\_ij(i, :));

B(i) = b(i) .\* (1 - C(i));

% 计算目标强度

TS(i) = 10 \* log10(r(i).^2 / 4) + 10 \* log10(B(i));

% 计算亮点子回波幅度

A(i) = 10.^(TS(i) / 20);

end

亮点的回波幅值

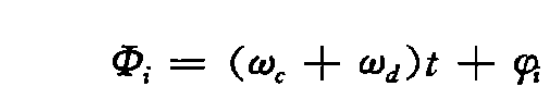
% 输出结果

disp(A);

2.2136 1.9170 4.0000 1.9170 1.9170 2.4749

# 6.入射产生相位跳变

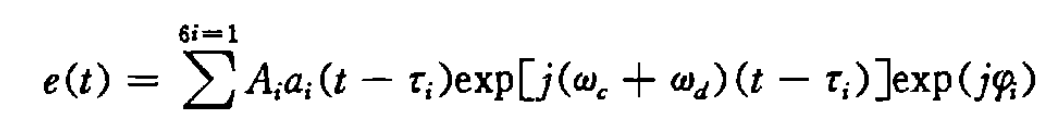
跳变相位应该是一个0~2Π的随机变量，在这里设定成180°

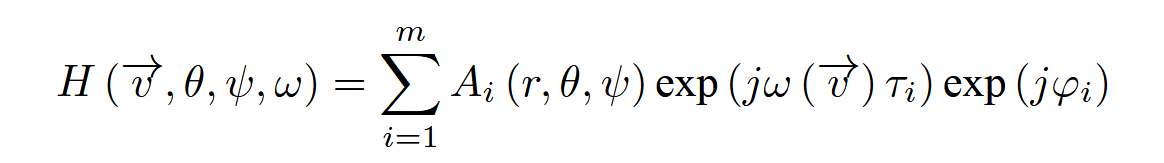


phi = pi;%假设相位跳变180°

# 7.计算回波

## 计算亮点模型的转移函数





此公式为亮点模型的转移函数，将之前的计算带入

%计算亮点模型的转移函数

highlightTF = zeros(6,N);

temp = zeros(1,N);

parfor i = 1:6%计算每个部位的转移函数

highlightTF(i,:) = A(i)\*exp(-1j\*(2\*pi\*f)\*tauHighLight(i)).\*exp(1j\*phi);

end

正在使用 'Processes' 配置文件启动并行池(parpool)...

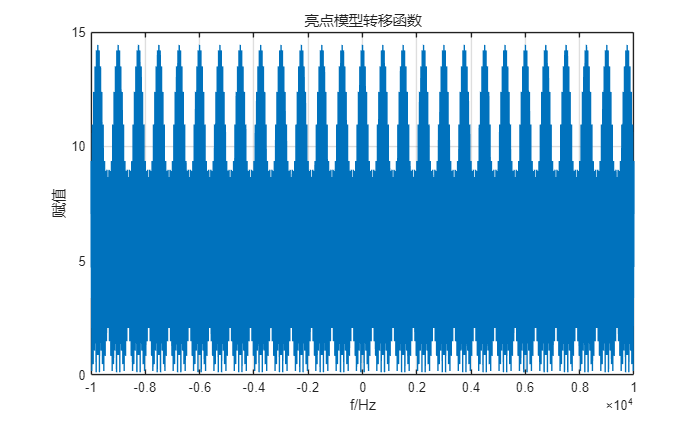
已连接到具有 8 个工作进程的并行池。

TF2 = sum(highlightTF,1);%将转移函数相加

figure

plot(f,abs(TF2));xlabel('f/Hz');ylabel('赋值');grid on;title('亮点模型转移函数');

axis(-fs/2,fs/2,-inf,inf);



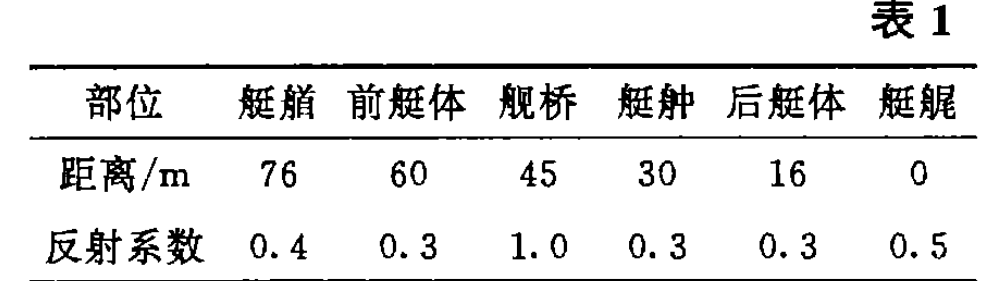
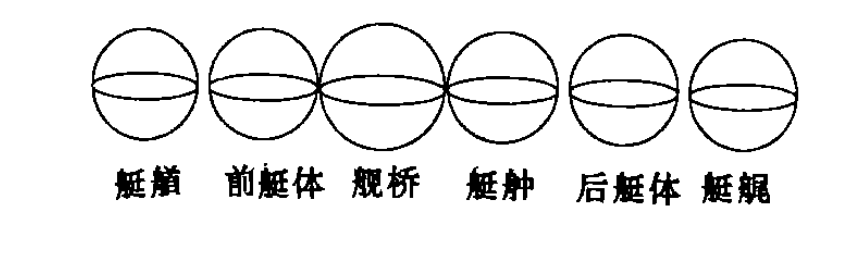
PAfterRef = ifft(fftshift(fft(P)).\*TF2);%逆变换得到时域的波形

parfor i = 1:6

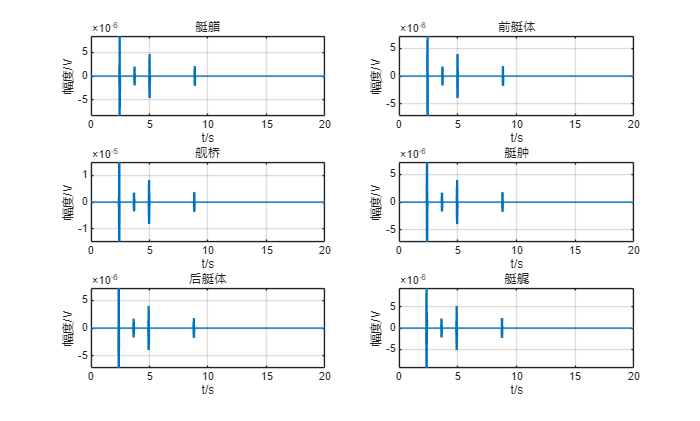
PRef(i,:) = ifft(fftshift(fft(P)).\*highlightTF(i,:));%每个部位的反射信号

end

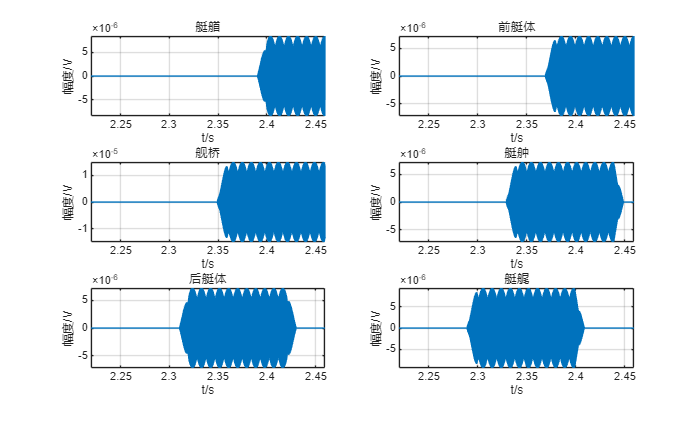
各个部位亮点的回波



helpViewEveryPart(PRef,t,[0,20,-inf,inf]);



helpViewEveryPart(PRef,t,[2.2197,2.4597,-inf,inf]);

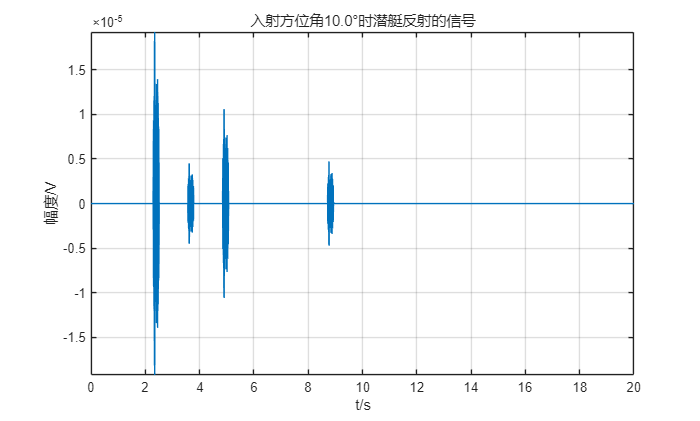


潜艇的回波

figure

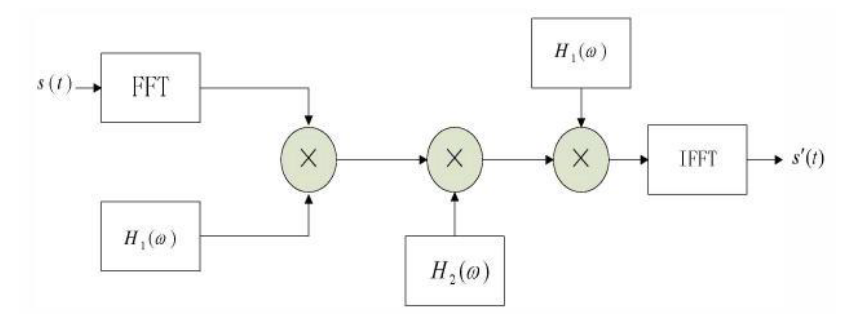
plot(t,real(PAfterRef));xlabel('t/s');ylabel('幅度/V');grid on;

title(sprintf('入射方位角%.1f°时潜艇反射的信号', theta));axis([0,20,-inf,inf]);



e = PAfterRef;

# 8.信号返回发射点：

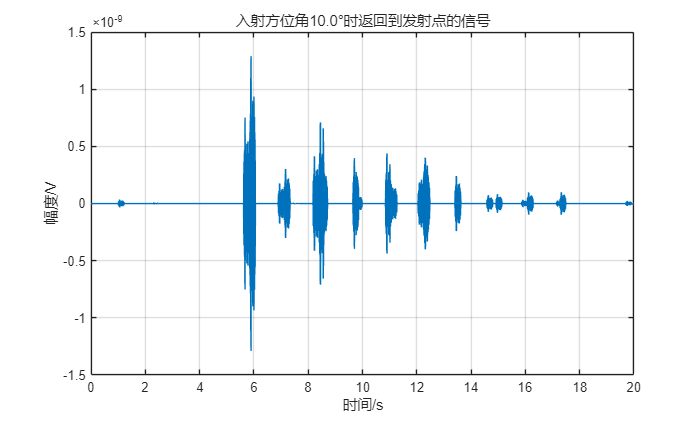


eAfterChan = ifft((fftshift(fft(e)).\*TF1));

figure

plot(t,real(eAfterChan));xlabel('时间/s');ylabel('幅度/V');grid on;

title(sprintf('入射方位角%.1f°时返回到发射点的信号', theta));



s = eAfterChan;%交换变量

# 9.波束形成

接收到的信号波束形成

考虑阵列情况

d = 1; % 阵元间距（10m长的线阵，共10个元素）

N = 10; % 阵元数量

angles = -90:10:90; % 角度范围 -90° 到 90°

alpha = 10;%阵列的接收角度（0°~180°）

%% 仿真阵元域数据

t\_total = t; % 总时间向量

L = length(t\_total);

fft\_len = 2^nextpow2(L); % FFT长度

element\_signals = zeros(N, length(t\_total)); % 初始化阵元信号矩阵

alpha = alpha-90;

for i = 1:N

delay = -(i-1)\*d\*sind(alpha)/c; % 计算时延

delay\_samples = round(delay \* fs); % 时延转换为样本数

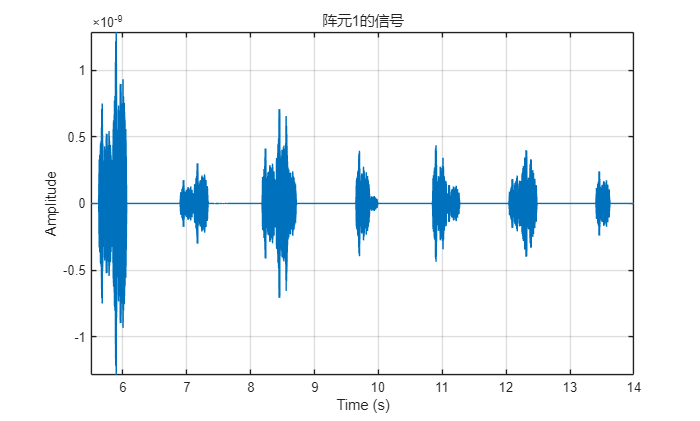
element\_signals(i,:) = apply\_delay(s, delay\_samples);

end

figure;

plot(t,real(element\_signals(1,:)));

title('阵元1的信号');axis([5.5,14,-inf,inf]);xlabel('Time (s)');ylabel('Amplitude');grid on;

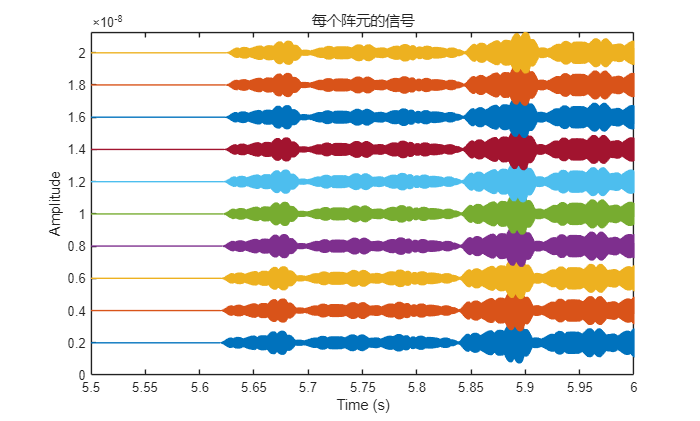


每个阵元接收的信号

figure;

plot(t,real(element\_signals).'+[1:N]\*2e-9);

title('每个阵元的信号');axis([5.5,6,0,inf]);xlabel('Time (s)');ylabel('Amplitude');



波束形成

%% 时延波束形成

beamformed\_signals\_time\_domain = zeros(length(angles), L); % 初始化时域波束形成信号矩阵

for j = 1:length(angles)

alpha = angles(j);

% beamformed\_signal\_freq\_domain = zeros(1, fft\_len); % 初始化频域波束形成信号

beamformed\_signal\_time\_domain = zeros(1, L);

for i = 1:N

delay = (i-1)\*d\*sind(alpha)/c; % 计算时延

delay\_samples = round(delay \* fs); % 时延转换为样本数

element\_signal = apply\_delay(element\_signals(i,:), delay\_samples);

% element\_signal\_fft = fft(element\_signal, fft\_len); % 计算FFT

% beamformed\_signal\_freq\_domain = beamformed\_signal\_freq\_domain + element\_signal;

beamformed\_signal\_time\_domain = beamformed\_signal\_time\_domain + element\_signal;

end

% beamformed\_signal\_time\_domain = ifft(beamformed\_signal\_freq\_domain, fft\_len); % 计算IFFT

beamformed\_signals\_time\_domain(j,:) = beamformed\_signal\_time\_domain(1:L); % 截取有效部分

end

% 输出波束域数据

figure;

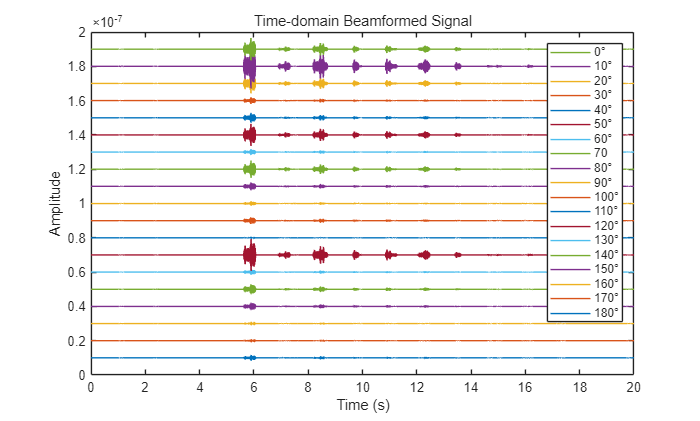
% plot(t\_total, real(beamformed\_signals\_time\_domain).'+angles);

helpViewBeamInEveryDegree(t\_total,beamformed\_signals\_time\_domain,angles,5e-9);

xlabel('Time (s)');

ylabel('Amplitude');

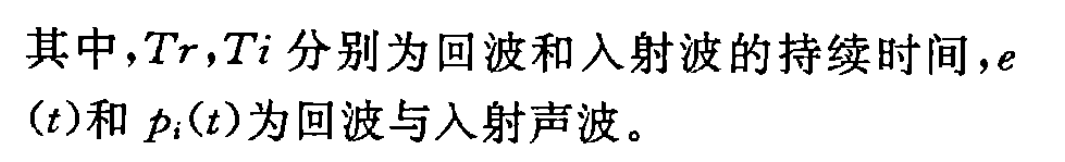
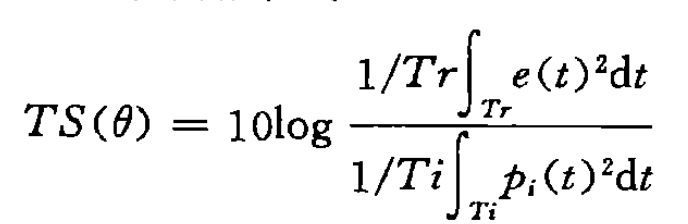
title('Time-domain Beamformed Signal');



save('TD.mat', 't\_total', 'beamformed\_signals\_time\_domain');

%保存

# 10.计算目标强度（修改中）：



%根据回波的包络计算脉宽

figure

%回波包络

subplot(2,1,1)

plot(t, abs(e));

xlabel('时间/s');ylabel('幅度');title(sprintf('信号与目标夹角为%.1f°时的回波包络', theta));

axis([0,10,-inf,inf]);grid on;

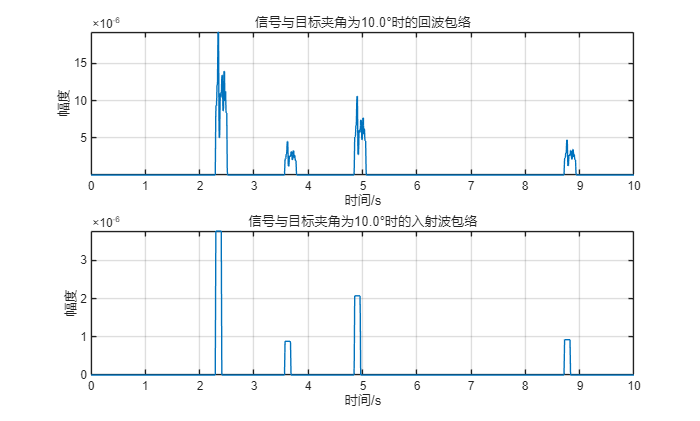
%入射波包络

subplot(2,1,2)

plot(t, abs(P));

xlabel('时间/s');ylabel('幅度');title(sprintf('信号与目标夹角为%.1f°时的入射波包络', theta));

axis([0,10,-inf,inf]);grid on;



% %计算回波脉宽

% start\_index\_echo = find(abs(e) > 0, 1, 'first');%应该设置一个阈值

% end\_index\_echo = find(abs(e) > 0, 1, 'last');

% W\_echo = (t(end\_index\_echo) - t(start\_index\_echo));

% fprintf('回波脉宽为：%.10f s\n', W\_echo);

% %入射波为32ms

% fprintf('入射波波脉宽为：%.3f s\n', width);

% %分子项

% TS\_submarine = 10\*log10( ( sum(real(e).^2)/W\_echo ) / ( sum(real(P).^2)/width ) );

% fprintf('入射信号为%d° 时目标强度为：%.3f dB\n', theta,TS\_submarine);