WES\_268A\_preLab3

**Theory Problems**

**Text, letter

Description automatically generated**

**Text, letter

Description automatically generated**

Mean of y(T) = E{

, =0

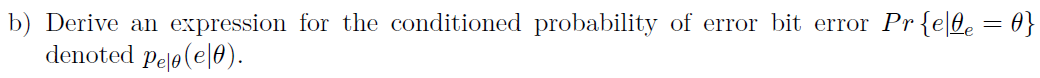
Variance of y(T) = E{y(T)

=

=

= =

1)**]**

****

*No/2*

*=*

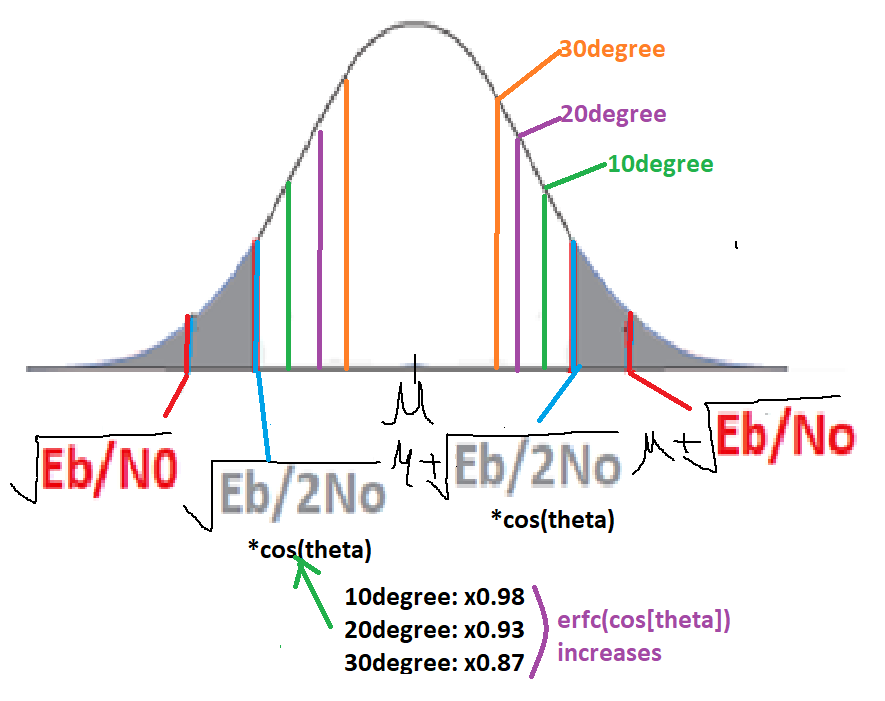
So,*pe*|*θ*(*e*|*θ*) =

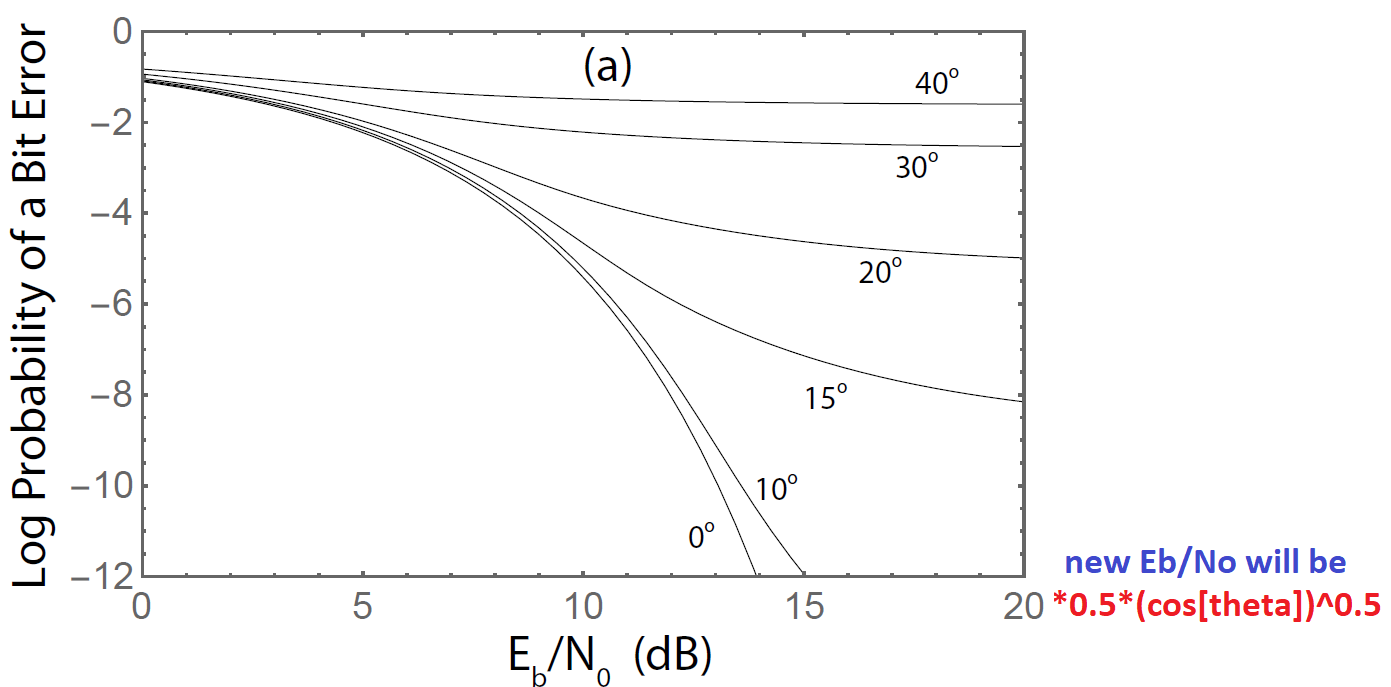
Where Eb =

**Text

Description automatically generated**

*Pe = =*

****

****

**Text

Description automatically generated**

**Chart, diagram, histogram

Description automatically generated**

=

as T 🡪 8/9T, amplitude will be 8/9 of correctly received signal.

Then P = = 64/81(79.01%) of original signal power(21% energy loss ).

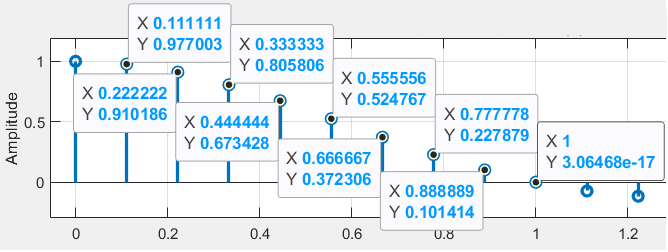
1. Raised cosine function

Text

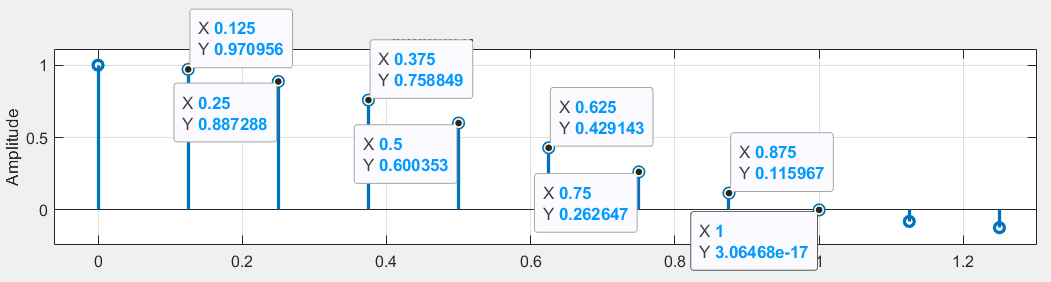
Description automatically generated with low confidence

With q(t) = sin(

< 9 sample case > voltage sum = 4.592, P = v^2 = 21.093



< 8 sample case > voltage sum = 4.025, P = v^2 = 16.202(76.8%, 23.2% energy loss)



1. 1x10^-9 🡺 7.59x10^-8

**Chart, line chart

Description automatically generated**

**Text

Description automatically generated**

X(k) = (k)+j Acos *+jAsin*

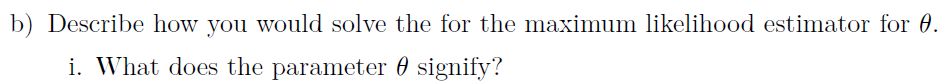
*\*Assumes*

L( (1/2 exp[-1/2], s(k;

-1/2/

-1/No , as /(fs/2) = No

L( (1/2 exp[-1/No]

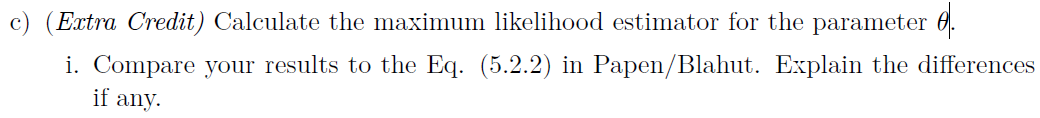


1. Find the joint-pdf of observations
2. Find the likelihood function L( or log-likelihood function could be found if it helps.
3. Optimize likelihood function or log-likelihood function.

: set the derivative of it to zero and second order derivation to be negative to find the max point.

1. estimated θ is arctangent of I-channel integration/ Q-channel integration .

So, the balance between I/Q should be taken care of for the wanted phase.

Log( L(

To maximize it it`s derivative should be zero as follows.

Let set J() =

=

So,

It`s different from the 5.5.2 below. If imaginary terms are taken with out ‘j’ it`s the same but don`t know why is that.

(5.5.2)

Graphical user interface, text, application

Description automatically generated

=

=

+

=+

=

]

Matlab problem 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* PulseShape = **Rectangular**

**–** SamplesPerSymbol = **8**

**–** SyncSequence = **1000010010110011111000110111010**

**–** SyncSequenceFormat = *BPSK*

**–** SyncSeqLength = **31 bits**

**–** NumDataBits = **96 bits**

**–** PacketFormat =[SYNC SEQ (31bits)]**[ DATA BITS (96 bits) ]**[SYNC SEQ

(31bits)]

**–** DataFile = *msg*.*mat*

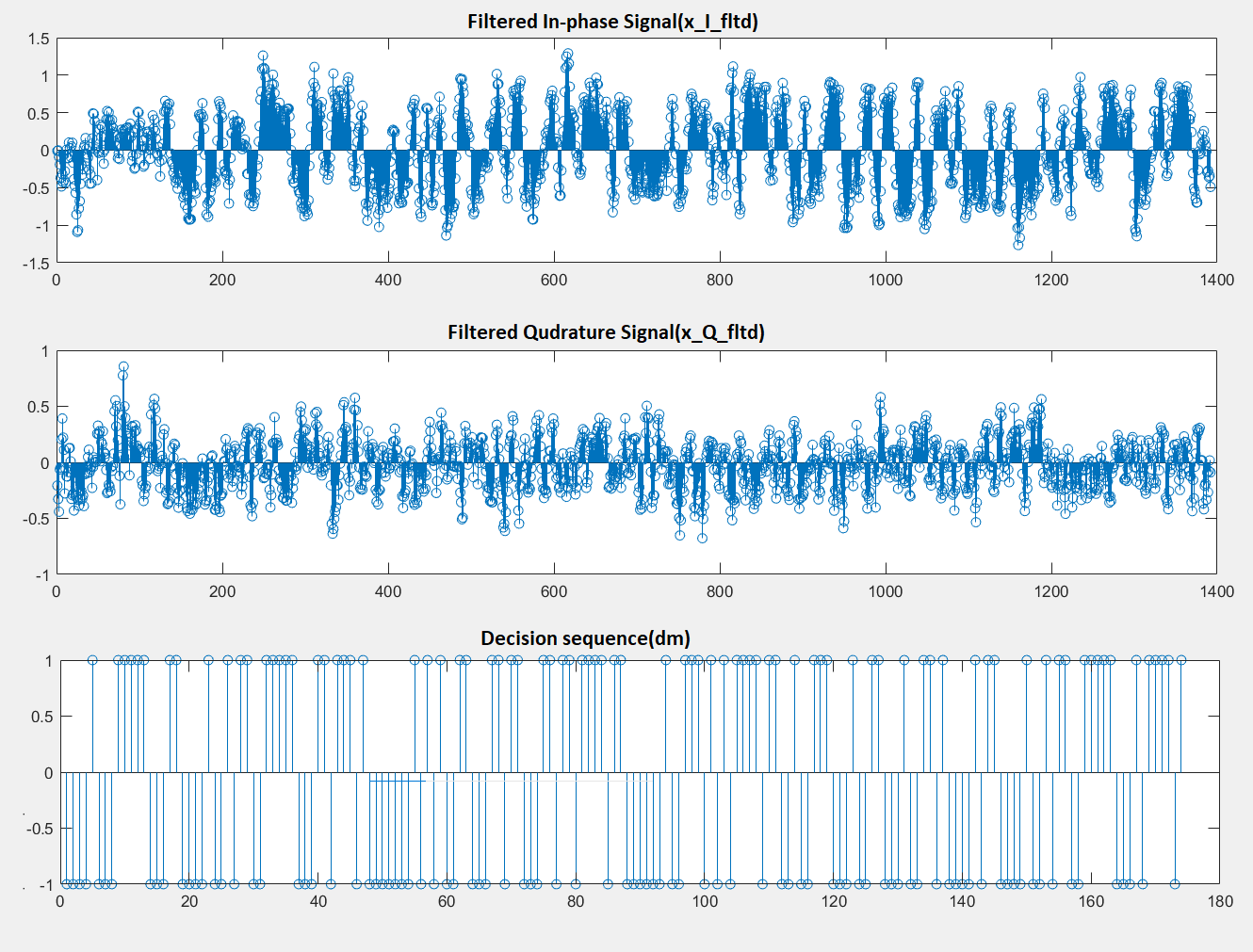
**–** DataFileFormat = Complex Baseband Samples

1. Reading Samples x(n) from file

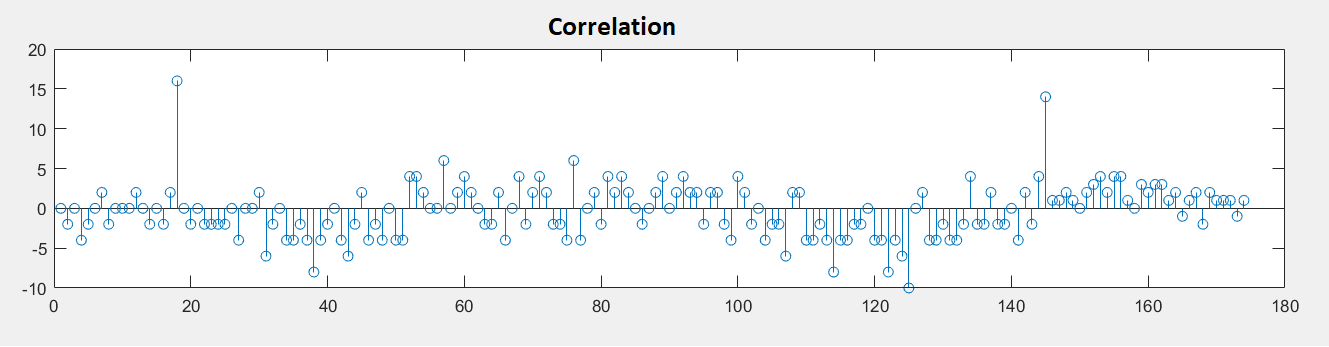
A picture containing graphical user interface

Description automatically generated

1. Matched Filter h(n) and Decision d(m)



1. Correlation



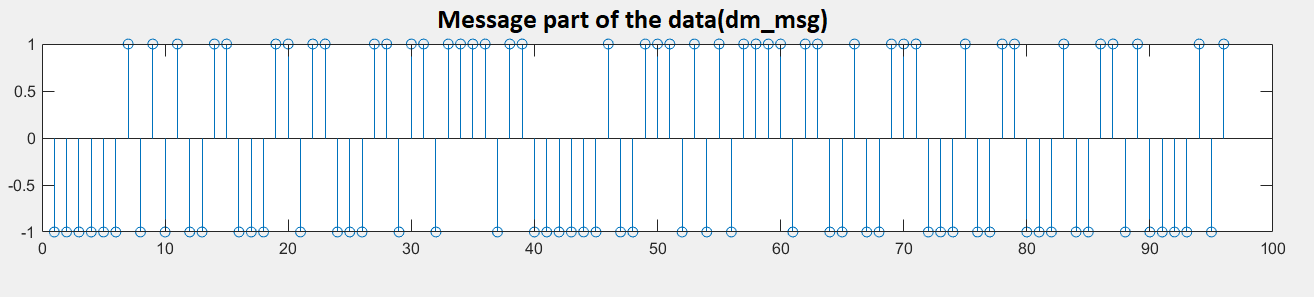
1. Examining the Correlation Peaks

Timeline

Description automatically generated with medium confidence

m1 = 18, m2 = 145

1. Removing the Headers and Extracting the Data Bits



1. (By MSB first for each 8 bits)

00000010: @

10100110: e

00110110: l

00110110: l

11110110: o

00000100: space

11101010: W

11110110: o

01001110: r

00100110: d

00100110: d

10000101: ¡

( I can tell it should e Hello World[¡](https://en.wiktionary.org/wiki/%C2%A1#Catalan) **but was not able to get it maybe due to ‘dm’ was not correct.)**

2. Implementation of a PLL

< Standard PLL>

Chart

Description automatically generated

< Costas PLL >Chart, line chart

Description automatically generated

**<code for Matlab P1>**

x=load("msg.mat");

x\_I = real(x.x);

x\_Q = imag(x.x);

figure(1)

subplot(211)

plot(x\_I)

title(['msg.mat I data'])

ylabel('Amplitude')

subplot(212)

plot(x\_Q)

title(['msg.mat Q data'])

ylabel('Amplitude')

figure(2)

fs=8;N=128/fs; B =0.999;NN=1392/2; fftlen = 1024;

ndx=(-N:1/fs:N); % Impulse Response Sample Span

h=(sin(pi\*ndx).\*cos(B\*pi\*ndx))./((pi\*ndx).\*(1-(2\*B\*ndx).^2));

h(round(fs\*N+1))=1.0; % Insert value 1.0 at Midpoint

hc=zeros(1,NN\*2); % Array of zeros

hc(NN+1+(-N\*fs:N\*fs))=h; % Center impulse response in array

hc=fftshift(hc)/fs; % Shift center to Origin, now Non Causal, and scale

subplot(3,1,1)

stem(ndx,h,'linewidth',2); % Plot non causal impulse response

grid on;

axis([-5 5 -0.3 1.2])

title(['Impulse Response, h(n)',num2str(20\*N+1),' Tap Filter'])

text(-0.5,0,'Time Index');

ylabel('Amplitude')

subplot(3,1,2)

Hc = fftshift(real(fft(hc,fftlen)))/max(fftshift(real(fft(hc,fftlen))));%/3.3027;

plot((-0.5:1/fftlen:0.5-1/fftlen)\*fs,Hc,'linewidth',2) % Mag Freq Resp

hold on

plot([-0.5\*fs -0.5 -0.5 0.5 0.5 0.5\*fs],[0 0 1 1 0 0],'r','linewidth',2)

hold off;

grid on;

axis([-0.5\*fs 0.5\*fs -0.2 1.2])

set(gca,'XTick',[-5:0.5:5])

title(['Magnitude Frequency Response, ',num2str(20\*N+1),' Tap Filter'])

ylabel('Magnitude')

subplot(3,1,3)

plot((-0.5:1/fftlen:0.5-1/fftlen)\*fs,20\*log10(Hc),'linewidth',2) % Log Mag Freq Resp

hold on

plot([-0.5\*fs -0.5 -0.5 0.5 0.5 0.5\*fs],[-20 -20 0 0 -20 -20],'r','linewidth',2)

hold off

grid on

axis([-0.5\*fs 0.5\*fs -20 10])

set(gca,'XTick',[-0.5\*fs:0.5:-0.5\*fs])

title(['Log Magnitude Frequency Response, ',num2str(20\*N+1),' Tap Filter'])

xlabel('Frequency')

ylabel('Log Mag (dB)')

%%

x\_I\_ip=zeros(1,NN\*2); % Array of zeros

x\_I\_ip(1:1:1392)=x\_I; % Center impulse response in array

x\_Q\_ip=zeros(1,NN\*2); % Array of zeros

x\_Q\_ip(1:1:1392)=x\_Q; % Center impulse response in array

y\_I\_fltd = filter(hc,1,x\_I\_ip);

y\_Q\_fltd = filter(hc,1,x\_Q\_ip);

figure(3)

subplot(511)

stem(y\_I\_fltd)

subplot(512)

stem(y\_Q\_fltd)

subplot(513)

dm = zeros(1,NN\*2);

dm = ceil(y\_I\_fltd(1:8:end))\*2-1;

dm = dm./abs(dm);

stem(dm)

%% correlate

subplot(514)

x = [ 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0];

z = correlate(x,dm);

stem(z)

subplot(515)

dm\_msg = zeros(1,96);

dm\_msg =dm(49:1:144)

stem(dm\_msg)

dm\_msg\_out = zeros(1,96);

dm\_msg\_out = dm\_msg(144:1:49)

**<code for Matlab P2>**

clear all; close all;

nn = 2000;

n = 1:1:nn;

%n\_1 = 2:1:nn+1;

fs = 2000; % samples per second

Ts = 1/fs; % period

delF = 20; % Hz

fn = 10; % Hz

zeta = 0.707 ; %0.707 ;

n0 = 200; % samples

n1 = 2000; % samples

%%Initilize PLL Loop

% (Setting up a Simulation to test the step response)

%st = ones(1,nn);

for i = 1:nn

if i < n0

st(i) = 0;

elseif i> n0-1

st(i) = delF;

end

end

%% xc definition

xc0 = zeros(1,nn);

xc = zeros(2,nn);

for j = 1:nn

if j< n0+1

xc0(j) = 0;

elseif j > n0-1

xc0(j) = exp(1i\*2\*pi\*delF\*(j-n0)\*Ts); % n0 <= n < n

end

xI(j) = real(xc0(j)); xQ(j) = imag(xc0(j));

xc(1,j) = xI(j);

xc(2,j) = xQ(j);

end

e = zeros(1,length(xc));

fi = zeros(1,length(xc)+1);

fint = zeros(1,length(xc)+1)

fo = zeros(1,length(xc));

vi = zeros(1,length(xc));

vo = zeros(1,length(xc));

th = zeros(1,length(xc));

%% Kt

Kt = 4\*pi\*zeta\*fn;

%% Ka

Ka = pi\*fn/zeta;

%% PLL implementation

for l = 2:length(xc)

%% phase rotator

%ro(1) = [1 0; 0 1];

ro= [ cos(th(l-1)) -sin(th(l-1)) ; sin(th(l-1)) cos(th(l-1))];

%xc(l) = ro\*xc(1,l)

s(1,l) = ro(1,1)\*xc(1,l)+ro(1,2)\*xc(2,l);

s(2,l) = ro(2,1)\*xc(1,l)+ro(2,2)\*xc(2,l);

sI(l) = s(1,l) ; sQ(l) =s(2,l);

sI(l) =s(1,l);

sQ(l) =s(2,l);

%% phase comparator

%standard PLL

e(l) = s(2,l);%atan(abs(s(1,l)/abs(s(2,l))))-atan(abs(xc(1,l)/abs(xc(2,l))));%

%costas PLL

%e(l) = s(2,l)\*s(1,l);%sI(n)sQ(n;

fi(l) = e(l)\*Kt;

fint(l) =fint(l-1) + Ka\*Ts/2\*(fi(l)+fi(l-1));

fo(l) = fi(l) + fint(l);

%% Numerically Controlled Oscillator

vi(l) = fo(l);

vo(l) = vo(l-1) + Ts/2.\*(vi(l)+vi(l-1));

th(l+1) = -vo(l);

%% loop filter

fint(l+1) =fint(l) + Ka\*Ts/2\*(fi(l+1)+fi(l));

end

figure(1)

% plot(1:length(xc), xc)

% hold on

plot(1:length(st), st)

hold on

plot(1:length(fo), fo/2/pi)

% hold on

% plot(1:length(th), th)