% In-Lecture Assignment #5 on April 17, 2019

% Consider the Costas phase locked loop described in Section 10.4
% of Johnson, Sethares and Klein (JSK) on pages 206-209.

% The Matlab code for the Costas loop is available by first running
% pulrecsig.m and then costasloop.m from the JSK Matlab code at
% http://users.ece.utexas.edu/~bevans/courses/realtime/homework/SRD-MatlabFiles.zip

% Please complete JSK Exercise 10.19 parts (a), (b) and (c):
% Use the preceding code in to "play with" the Costas loop algorithm.

% a. How does the stepsize mu affect the convergence rate?

% For an initial guess of zero for the phase offset, the Costas loop converges
% to a phase offset value of
% -0.98 at time 0.21s for mu = 0.003
% -0.98 at time 0.77s for mu = 0.0016
% -0.98 at time 3.75s for mu = 0.0003
% Note that -0.98 is within 2% of the optimum value of the phase offset of -1.0.
% The smaller the mu, the larger the iterations it takes for the algorithm to converge.

% b. What happens if mu is too large (say mu=1)?

% Algorithm diverges for mu = 0.02. Same goes for mu values larger than 0.02.

% c. Does the convergence speed depend on the value of the phase offset?

% For mu = 0.0003, the Costas loop converges to a phase offset value of
% -0.98 at time 2.39s for an initial guess for the phase offset of -0.75
% -0.98 at time 3.17s for an initial guess for the phase offset of -0.50
% -0.98 at time 3.66s for an initial guess for the phase offset of -0.25
% -0.98 at time 3.75s for an initial guess for the phase offset of 0.00
% Note that -0.98 is within 2% of the optimum value of the phase offset of -1.0.
% The closer the initial guess is to the optimum value, the faster the convergence

```matlab
% pulrecsig.m: create pulse shaped received signal

N=10000; M=20; Ts=.0001;   % # symbols, oversampling factor
time=Ts*N*M; t=Ts:Ts:time; % sampling interval & time vector
m=pam(N,4,5);              % 4-level signal of length N
mup=zeros(1,N*M);
mup(1:M:N*M)=m;            % oversample by integer length M
ps=hamming(M);             % blip pulse of width M
s=filter(ps,1,mup);        % convolve pulse shape with data
fc=1000; phoff=-1.0;       % carrier freq. and phase
c=cos(2*pi*fc*t+phoff);    % construct carrier
rsc=s.*c;                  % modulated signal (small carrier)
rlc=(s+1).*c;              % modulated signal (large carrier)

fftrlc=fft(rlc);                     % spectrum of rlc
[m,imax]=max(abs(fftrlc(1:end/2)));  % index of max peak
```

```
ssf=(0:length(t)-1)/(Ts*length(t));  % frequency vector
freqL=ssf(imax)                      % freq at the peak
phaseL=angle(fftrlc(imax))           % phase at the peak

fftrsc=fft(rsc);                     % spectrum of rsc
[m,imax]=max(abs(fftrsc(1:end/2)));  % find frequency of max peak
freqS=ssf(imax)                      % freq at the peak
phaseS=angle(fftrsc(imax))           % find phase at the peak

% costasloop.m simulate costas loop with input from pulrecsig.m
r=rsc;                                % rsc from pulrecsig.m
fl=500; ff=[0 .01 .02 1]; fa=[1 1 0 0];
h=firpm(fl,ff,fa);                    % LPF design
mu=.003;                              % algorithm stepsize
f0=1000;                              % freq. at receiver
theta=zeros(1,length(t)); theta(1)=0; % estimate vector
zs=zeros(1,fl+1); zc=zeros(1,fl+1);   % buffers for LPFs
for k=1:length(t)-1                   % z contains past inputs
  zs=[zs(2:fl+1), 2*r(k)*sin(2*pi*f0*t(k)+theta(k))];
  zc=[zc(2:fl+1), 2*r(k)*cos(2*pi*f0*t(k)+theta(k))];
  lpfs=fliplr(h)*zs'; lpfc=fliplr(h)*zc'; % output of filters
  theta(k+1)=theta(k)-mu*lpfs*lpfc;   % algorithm update
end

plot(t,theta),
title('Phase Tracking via the Costas Loop')
xlabel('time'); ylabel('phase offset')
```