ME 581 – H03 **Name** _____

1) Download and view the web cutter video "wc.mov" from the class web page. Run MATLAB video analysis code "wc_dig.m" provided below. It will save the first video frame as image "wc_first_frame.jpg" and digitize x-y pixel locations for red and green dots on revolutes B and C respectively. Output will be provided in text file "wc_keep.txt".

Use Microsoft Paint to manually digitize column-row locations for the centers of revolutes A and D from "wc_first_frame.jpg". Remember to convert column-row locations to x-y pixel locations as shown at the bottom of the MATLAB code.

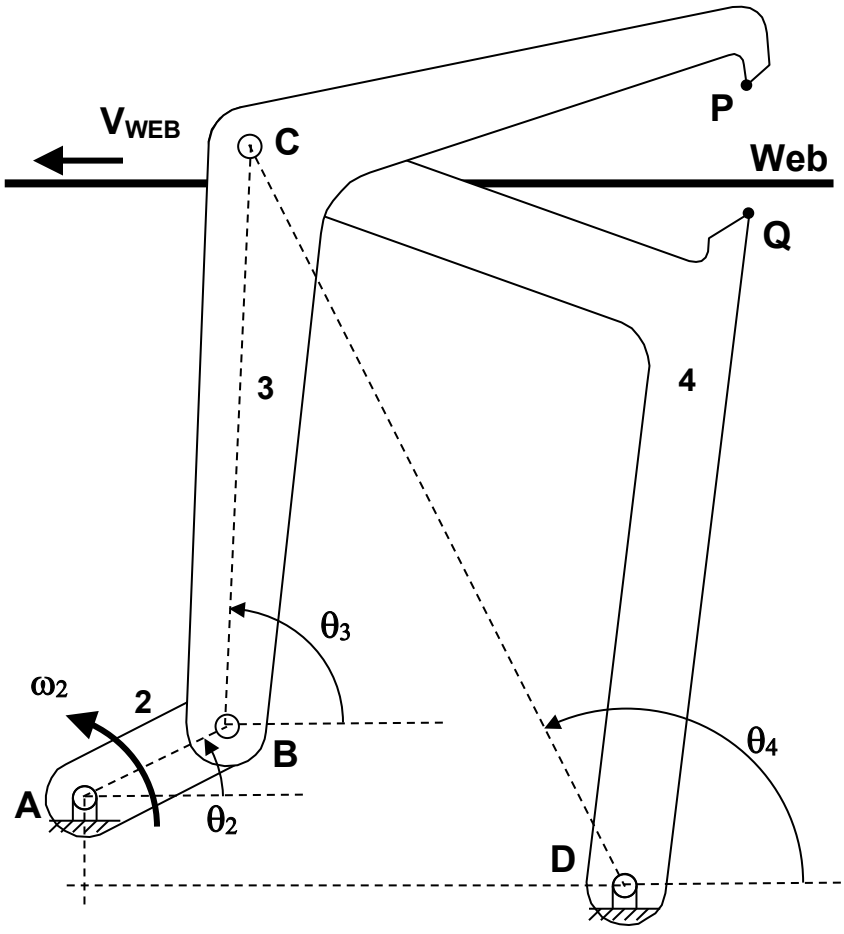
2) Plot angles θ_3 and θ_4 as functions of angle θ_2.

3) Calculate and plot ω_2 as a function of time using simple finite difference and using Savitsky-Golay floating interpolants. Be certain to use units of rad/sec. Determine the mean value of ω_2 and root-mean-square (RMS) **about the mean value.**

mean ω_2 3.55 rad/sec RMS ω_2 about mean ω_2 0.423 rad/sec

EXTRA CREDIT
Fit a circle to x-y data for revolute B and compare the center to your manual measurement.

Ax,Ay manual 289, 774 pix Ax,Ay circle 283.4940, 762.1857 pix
mean = 3.546 rad/sec, RMS = 0.291 rad/sec



```
% wc_dig.m - digitize red/green dots on web cutter from MOV video
% HJSIII, 19.12.05

clear

% video file name
fn_input = [ 'wc.mov' ];

% create video file reader object
VR_obj = VideoReader( fn_input );

% get video information
video_fps = VR_obj.FrameRate;          % frames per second
video_duration = VR_obj.Duration;      % sec
%video_frames = VR_obj.NumberOfFrames;  % must recreate object to rewind after using
NumberofFrames
%video_width = VR_obj.Width;
%video_height = VR_obj.Height;

% expected number of frames
nframe = round( video_duration * video_fps );
fps = round( video_fps );
disp( ' ' )
disp( [ 'reading ' num2str(nframe) ' frames recorded at ' num2str(fps) ' frames per second' ] )
disp( ' ' )
disp( 'see Figure 1' )
disp( ' ' )

% step through video
iframe = 0;
keep = [];
while hasFrame( VR_obj )
  a_rgb = readFrame( VR_obj );  % "readFrame" returns class uint8
  [ nr, nc, nk ] = size( a_rgb );
  iframe = iframe + 1;

% save first image
  if iframe==1,
    imwrite( a_rgb, 'wc_first_frame.jpg' )
  end

% convert to CIE L*a*b*
% L* intensity 0=dark, 100=bright - a_lab(:,:,1)
% a* green<0, red>0 - a_lab(:,:,2)
% b* blue<0, yellow>0 - a_lab(:,:,3)
a_lab = rgb2lab( a_rgb );  % size (nr,nc,3) - class double

% find red pixels
bw_r = ( a_lab(:,:,2) > 40 );  % size (nr,nc) - class logical

% find green pixels
bw_g = ( a_lab(:,:,2) < -30 );  % size (nr,nc) - class logical

% find centroid of one object in each black/white image
s_r = regionprops( bw_r, 'Centroid' );  % class structure
s_g = regionprops( bw_g, 'Centroid' );

% column and row stored in structure.Centroid
cr_r = s_r.Centroid;  % size (1,2) - class double
cr_g = s_g.Centroid;

% new figure
figure( 1 )
  clf
  warning( 'OFF', 'images:initSize:adjustingMag' )  % disbale warning for large images

% RGB image in UL
  subplot( 2, 2, 1 )
  imshow( a_rgb )
  title( [ 'frame ' num2str(iframe) ] )

% BW image for red in LL
  subplot( 2, 2, 3 )
  imshow( bw_r )
  title( 'red a*>40' )
```

```
  hold on
  plot( [ 0 cr_r(1) ], [ 0 cr_r(2) ], 'r' )  % line from origin to centroid

% BW image for green in LR
  subplot( 2, 2, 4 )
  imshow( bw_g )
  title( 'green a*<-30' )
  hold on
  plot( [ 0 cr_g(1) ], [ 0 cr_g(2) ], 'g' )  % line from origin to centroid

% update graphics
  drawnow

% save centroids
  keep = [ keep ; [ cr_r  cr_g ] ];

end  % bottom - while hasFrame

% row number increases in negative y direction
keep(:,2) = nr - keep(:,2);
keep(:,4) = nr - keep(:,4);

% show x-y results
figure( 2 )
  clf
  plot( keep(:,1),keep(:,2),'r', keep(:,3),keep(:,4),'g' )
  axis equal

% save to TXT file - x_red  y_red  x_green  y_green
save( 'wc_keep.txt', 'keep', '-ascii' )

% bottom - wc_dig
```

++

circle fit for unknown radius

$$(x-a)^2+(y-b)^2=r^2 \qquad (x^2+y^2)=\begin{bmatrix}1 & x & y\end{bmatrix}\begin{Bmatrix}r^2-a^2-b^2\\ 2a\\ 2b\end{Bmatrix}$$

$$\underset{\text{nobs x 1}}{\{Y\}}=\begin{Bmatrix}(x_1^2+y_1^2)\\ (x_2^2+y_2^2)\\ \vdots\\ (x_{nobs}^2+y_{nobs}^2)\end{Bmatrix} \qquad \underset{\text{nobs x 3}}{[X]}=\begin{bmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ \vdots & \vdots & \vdots\\ 1 & x_{nobs} & y_{nobs}\end{bmatrix} \qquad \underset{\text{3 x 1}}{\{\beta\}}=\begin{Bmatrix}r^2-a^2-b^2\\ 2a\\ 2b\end{Bmatrix}$$

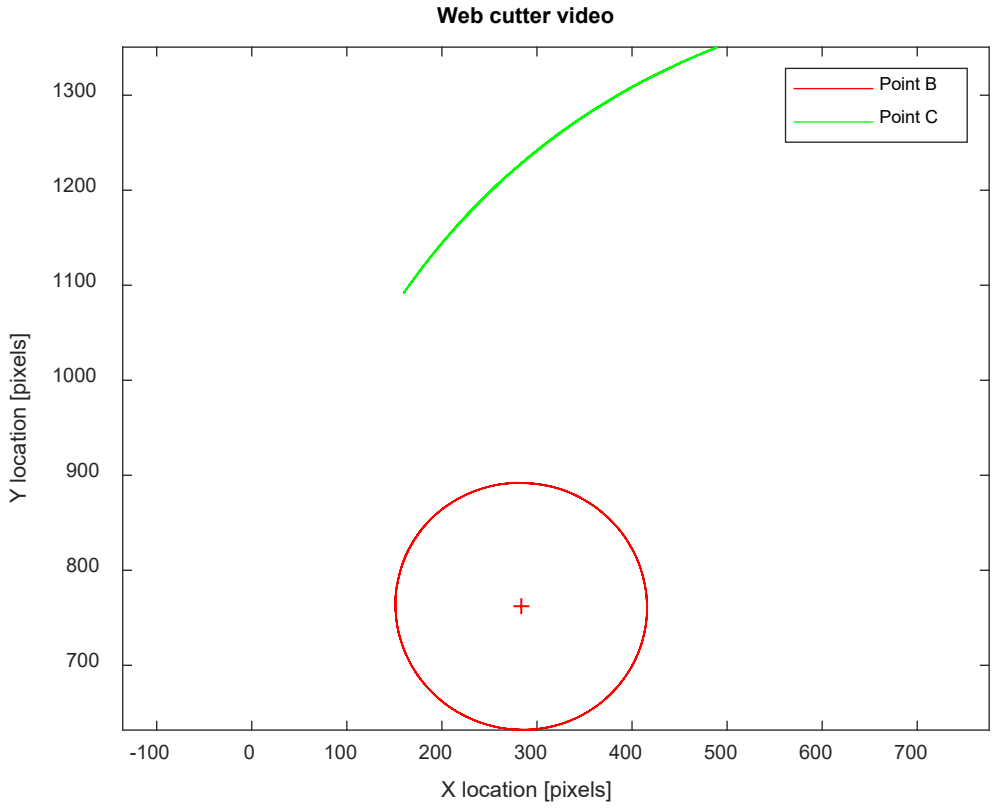
$$\{\beta\}=\left([X]^T[X]\right)^{-1}[X]^T\{Y\}$$

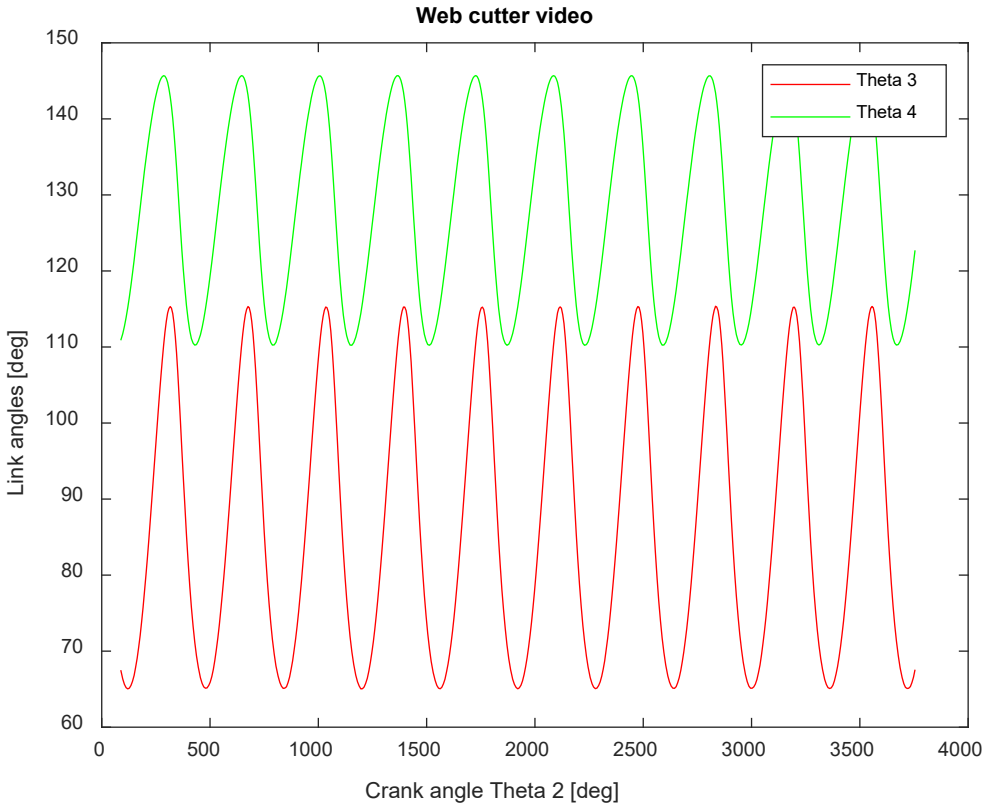
circle fit for known radius

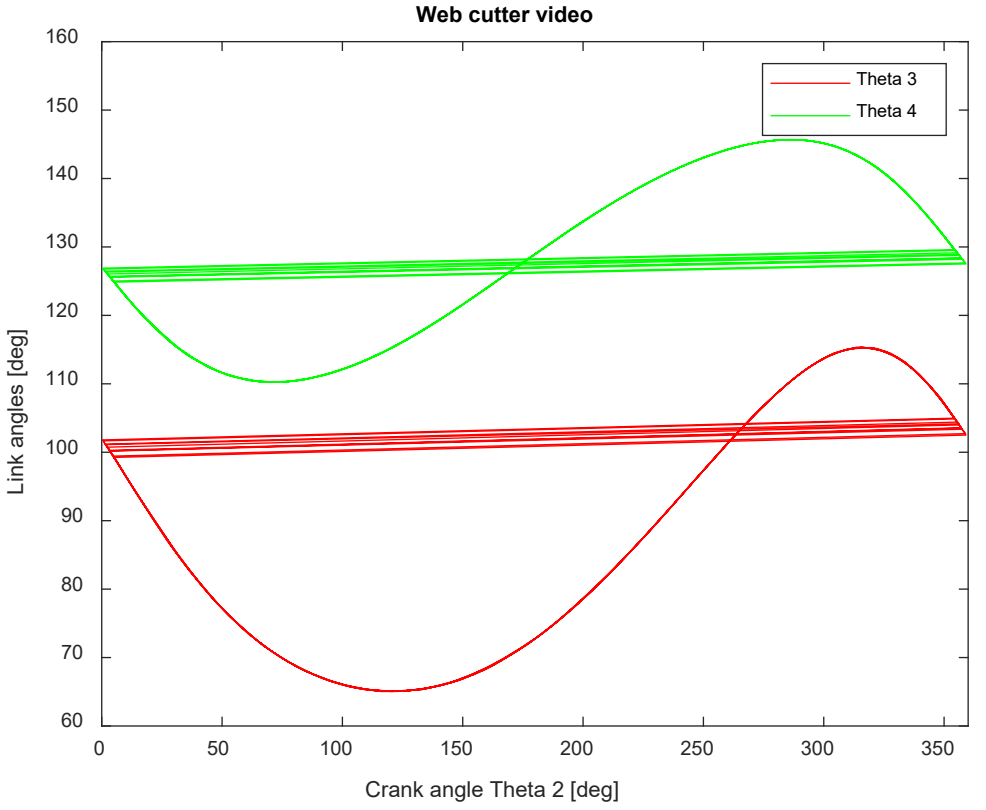
$$\underset{\text{nobs x 1}}{\{Y\}}=\begin{Bmatrix}(x_1^2+y_1^2-r^2)\\ (x_2^2+y_2^2-r^2)\\ \vdots\\ (x_{nobs}^2+y_{nobs}^2-r^2)\end{Bmatrix} \qquad \underset{\text{nobs x 3}}{[X]}=\begin{bmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ \vdots & \vdots & \vdots\\ 1 & x_{nobs} & y_{nobs}\end{bmatrix} \qquad \underset{\text{3 x 1}}{\{\beta\}}=\begin{Bmatrix}-a^2-b^2\\ 2a\\ 2b\end{Bmatrix}$$

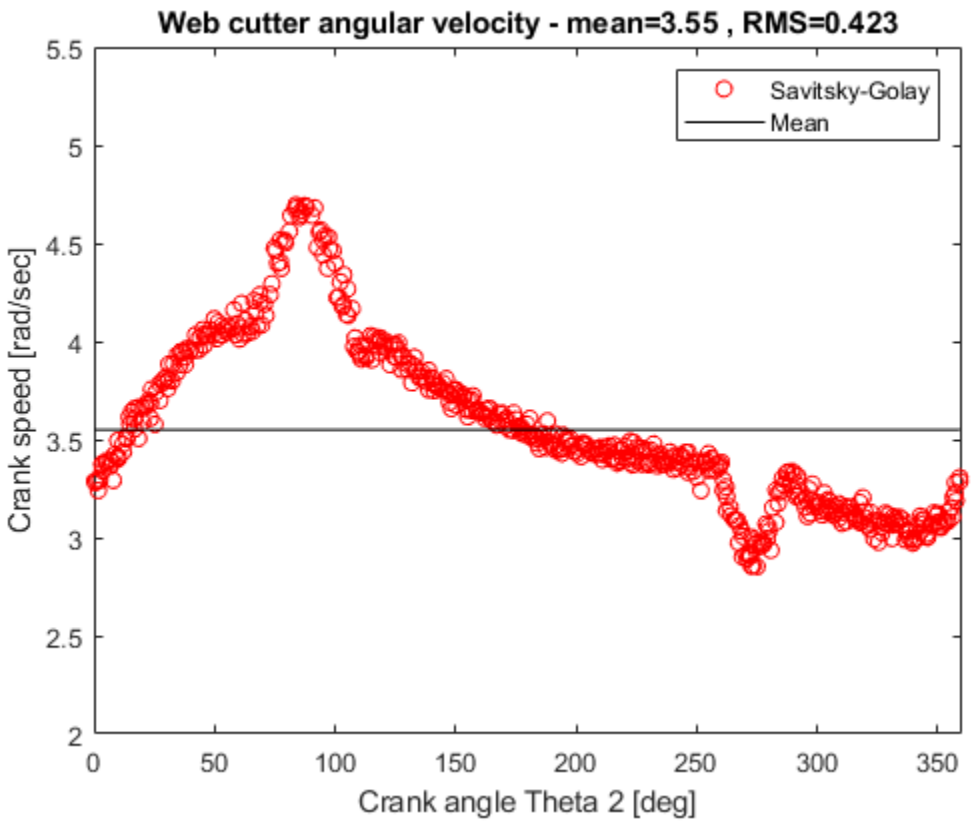
ME 581 – H03 **Name** _____

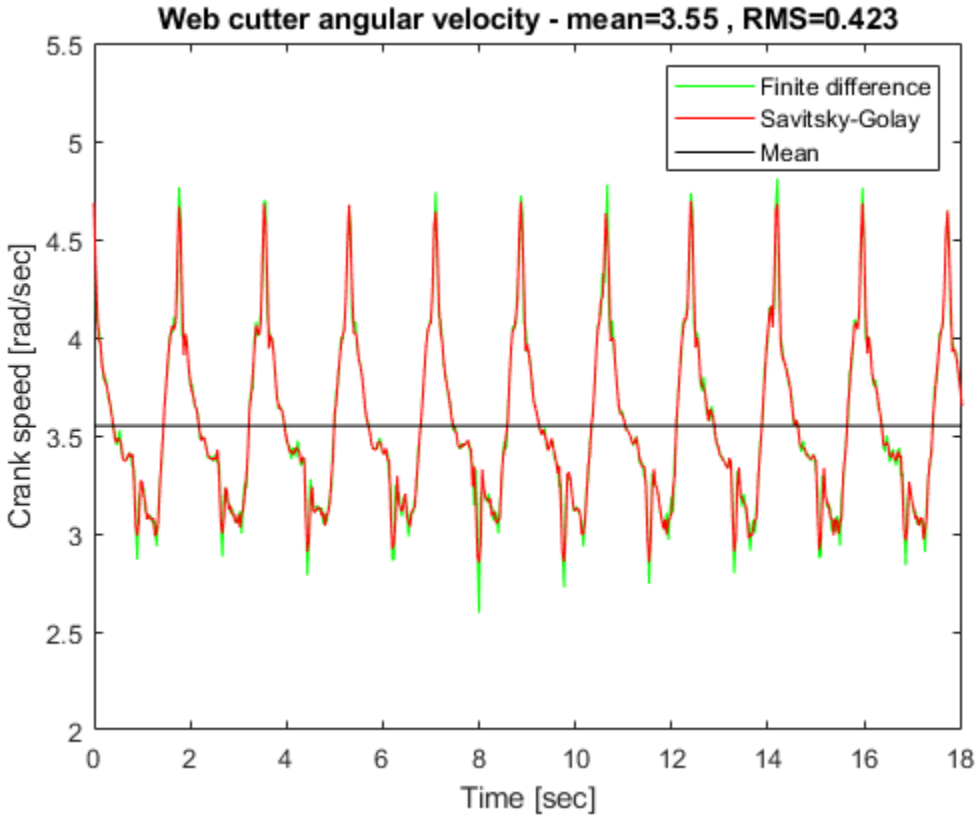
```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

using Ax,Ay from circle fit

harmonic amplitudes
    0.5272    0.2004    0.0350    0.0272    0.0940    0.0231    0.0573    0.0098    0.0593    0.0236    0.0364

Ax_fit =  283.4940
Ay_fit =  762.1857
```

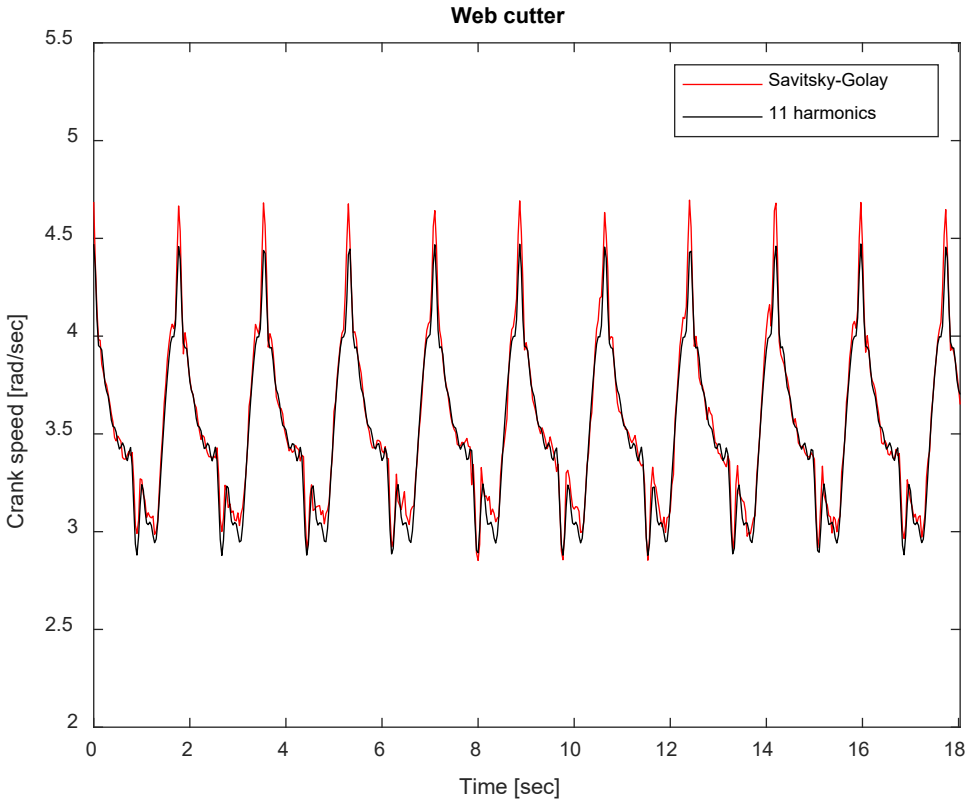


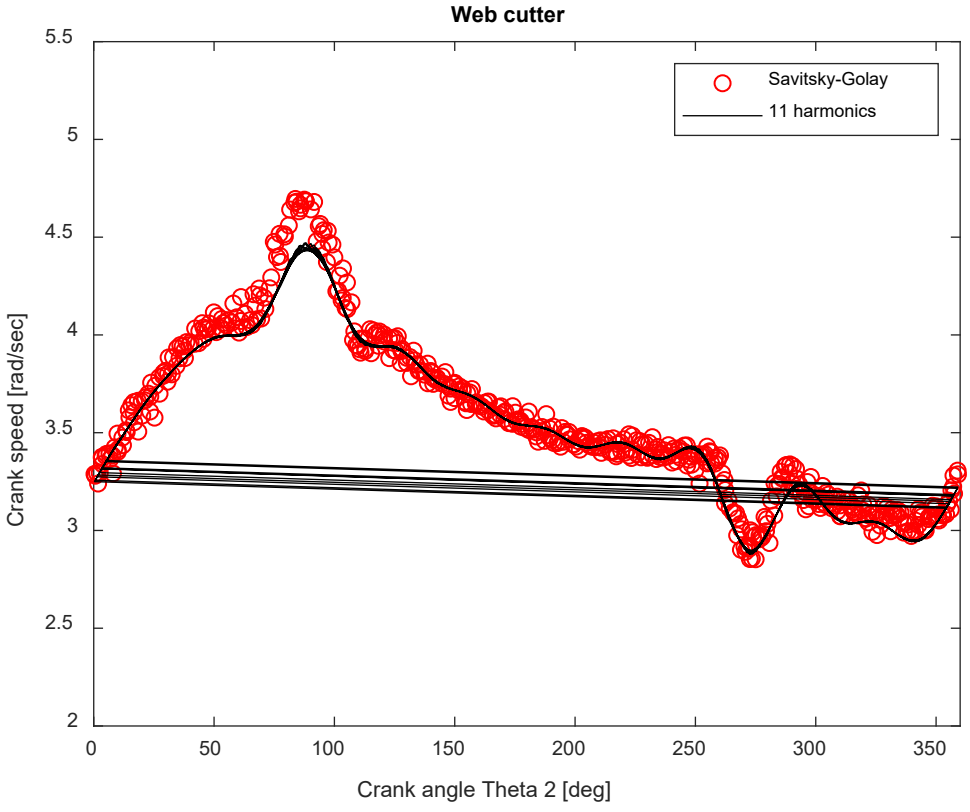

Web cutter video
Theta 3
Theta 4
Link angles [deg]
Crank angle Theta 2 [deg]
150
140
130
120
110
100
90
80
70
60
0
500
1000
1500
2000
2500
3000
3500
4000

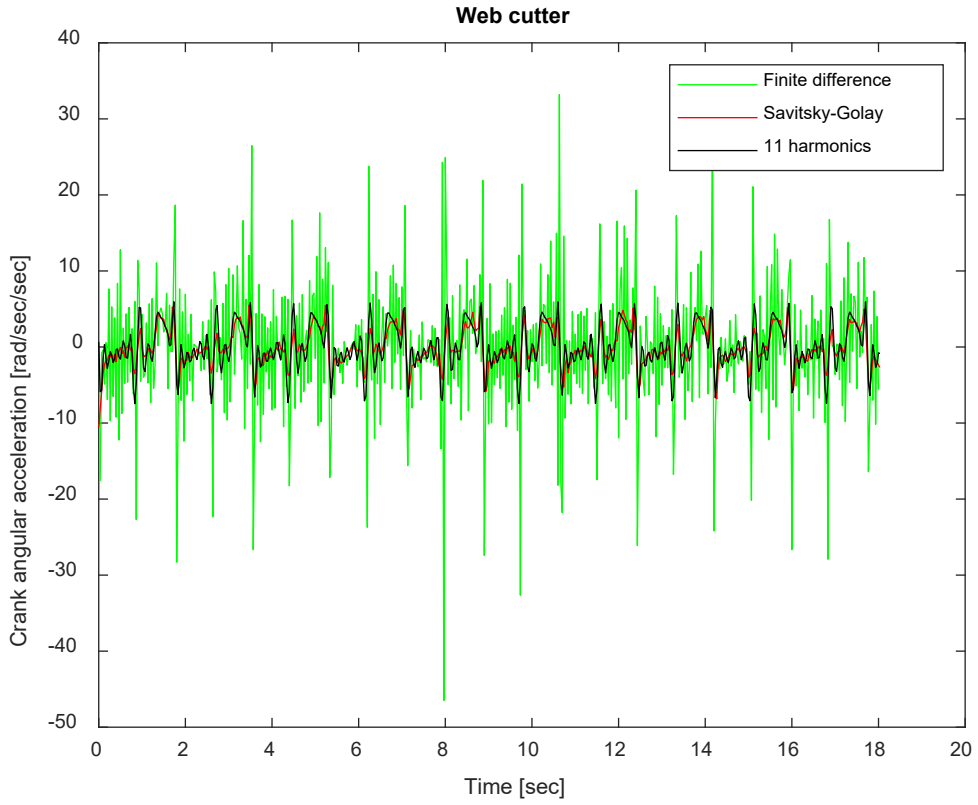

Web cutter video
Theta 3
Theta 4
Link angles [deg]
Crank angle Theta 2 [deg]
160
150
140
130
120
110
100
90
80
70
60
0
50
100
150
200
250
300
350

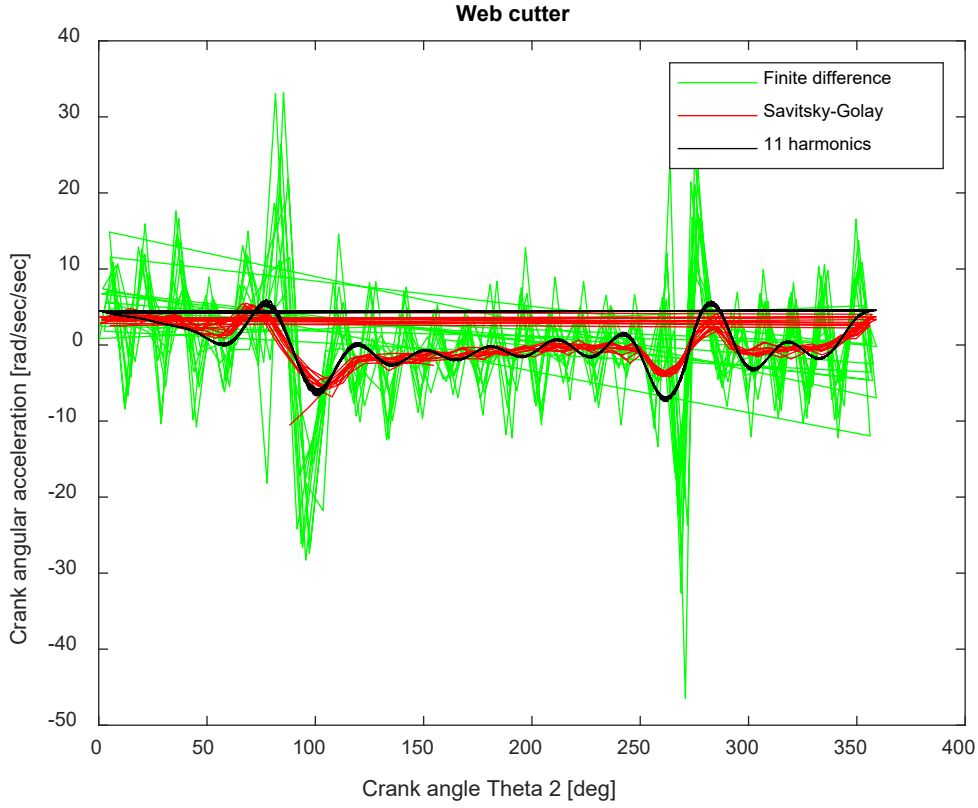

Web cutter angular velocity - mean=3.55 , RMS=0.423
Savitsky-Golay
Mean
Crank speed [rad/sec]
Crank angle Theta 2 [deg]
5.5
5
4.5
4
3.5
3
2.5
2
0
50
100
150
200
250
300
350


Web cutter angular velocity - mean=3.55 , RMS=0.423
Finite difference
Savitsky-Golay
Mean
Crank speed [rad/sec]
Time [sec]
5.5
5
4.5
4
3.5
3
2.5
2
0
2
4
6
8
10
12
14
16
18

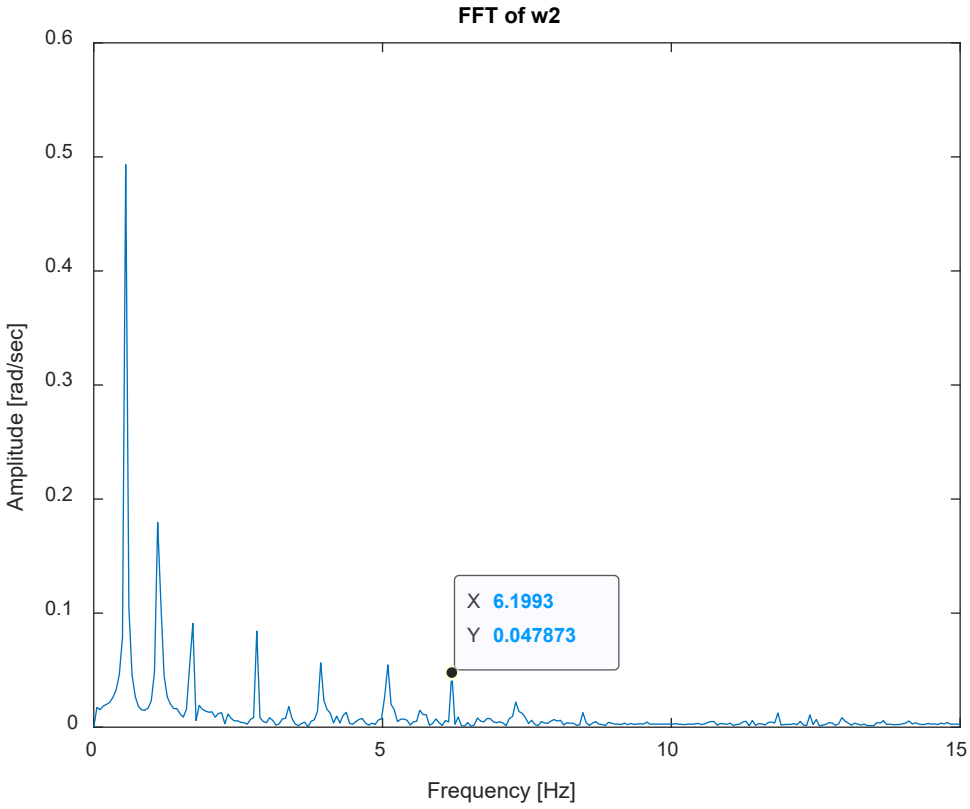

Web cutter
Savitsky-Golay
11 harmonics
Crank speed [rad/sec]
Time [sec]
5.5
5
4.5
4
3.5
3
2.5
2
0
2
4
6
8
10
12
14
16
18


Web cutter
Savitsky-Golay
11 harmonics
Crank speed [rad/sec]
Crank angle Theta 2 [deg]
5.5
5
4.5
4
3.5
3
2.5
2
0
50
100
150
200
250
300
350


Web cutter
Finite difference
Savitsky-Golay
11 harmonics
Crank angular acceleration [rad/sec/sec]
Time [sec]
40
30
20
10
0
-10
-20
-30
-40
-50
0
2
4
6
8
10
12
14
16
18
20


Web cutter
Finite difference
Savitsky-Golay
11 harmonics
Crank angular acceleration [rad/sec/sec]
Crank angle Theta 2 [deg]
40
30
20
10
0
-10
-20
-30
-40
-50
0
50
100
150
200
250
300
350
400

ME 581 – H03 **Name** _____



ME 581 – H03 **Name** _____

```
% h03.m - web cutter four-bar video analysis for ME 581
%   Homework 3
% HJSIII, 19.12.03

clear

% constants
d2r = pi / 180;

% column-row locations for A and D from "wc_first_frame.jpg", units = [pixels]
Ax = 289;
Ay_row = 1146;

Dx = 727;
Dy_row = 1215;

% convert column-row to x-y
nr = 1920;
Ay = nr - Ay_row;
Dy = nr - Dy_row;

% center from circle fit
%Ax = 283.4940;
%Ay = 762.1857;

% load raw locations for B and C, units = [pixels]
raw = load( 'wc_keep.txt' );
Bx = raw(:,1);
By = raw(:,2);
Cx = raw(:,3);
Cy = raw(:,4);

% time
fs = 30;                    % 30 frames per second
h = 1 / fs;
npts = length( Bx );
time = ( 0 : (npts-1) )' * h;

% angles
th2 = atan2( (By-Ay) , (Bx-Ax) );
th2 = unwrap( th2 );                  % unwrap th2

th3 = atan2( (Cy-By) , (Cx-Bx) );
th4 = atan2( (Cy-Dy) , (Cx-Dx) );

% angles in degrees
th2_deg = th2 / d2r;
th3_deg = th3 / d2r;
th4_deg = th4 / d2r;

% Savitsky-Golay filter
[ p2, v2, a2, j2 ] = filt_7pt_mat( th2', h );
[ p3, v3, a3, j3 ] = filt_7pt_mat( th3', h );
[ p4, v4, a4, j4 ] = filt_7pt_mat( th4', h );
p2 = p2';
w2 = v2';
a2 = a2';

% angular velocity and angular acceleration by finite difference - second order - zap end points
w2_fd = ( [diff(th2);NaN] + [NaN;diff(th2)] ) /2/h;
a2_fd = ( [diff(th2);NaN] - [NaN;diff(th2)] ) /h/h;

% mean and standard dev
w2_mean = mean( w2 );
w2_cen = w2 - w2_mean;
w2_RMS = std( w2_cen );

% harmonics per revolution
w2_har = w2_mean * ones( npts,1 );
a2_har = zeros( npts,1 );
for har = 1 : 11,
 A(har) = 2 * w2_cen' * cos(har*p2) /npts;
 B(har) = 2 * w2_cen' * sin(har*p2) /npts;
 amp(har) = sqrt( A(har)^2 + B(har)^2 );
 w2_har = w2_har  +A(har)*cos(har*p2)  +B(har)*sin(har*p2);
```

```
 a2_har = a2_har  -A(har)*har*w2_mean*sin(har*p2)  +B(har)*har*w2_mean*cos(har*p2);
end

disp( 'harmonic amplitudes' )
disp( amp )

% circle fit for revolute B
Y = Bx.*Bx + By.*By;
X = [ ones(npts,1)  Bx  By ];
beta = inv(X'*X) * X'*Y;
Ax_fit = beta(2) / 2
Ay_fit = beta(3) / 2

% check raw data and circle fit
figure( 1 )
  clf
  plot( Bx,By,'r', Cx,Cy,'g', Ax_fit,Ay_fit,'r+' )
  legend( 'Point B', 'Point C' )
  axis equal
  xlabel( 'X location [pixels]' )
  ylabel( 'Y location [pixels]' )
  title( 'Web cutter video' )

% plot raw angles versus time
figure( 2 )
  clf
  plot( th2_deg,th3_deg,'r', th2_deg,th4_deg,'g' )
  legend( 'Theta 3', 'Theta 4' )
  xlabel( 'Crank angle Theta 2 [deg]' )
  ylabel( 'Link angles [deg]' )
  title( 'Web cutter video' )

% plot raw angles versus crank angle
figure( 3 )
  clf
  plot( mod(th2_deg,360),th3_deg,'r', mod(th2_deg,360),th4_deg,'g' )
  legend( 'Theta 3', 'Theta 4' )
  xlabel( 'Crank angle Theta 2 [deg]' )
  ylabel( 'Link angles [deg]' )
  title( 'Web cutter video' )
  axis( [ 0 360  60 160 ] )

% plot angular velocity versus crank angle
figure( 4 )
  clf
  plot( mod(th2_deg,360),w2,'ro', mod(th2_deg,360),w2_mean*ones(npts),'k'  )
  xlabel( 'Crank angle Theta 2 [deg]' )
  ylabel( 'Crank speed [rad/sec]' )
  legend( 'Savitsky-Golay', 'Mean' )
  title( [ 'Web cutter angular velocity - mean=' num2str(w2_mean,4) ' , RMS=' num2str(w2_RMS,3) ]
)
  axis( [ 0 360  2 5.5 ] )

% plot angular velocity versus time
figure( 5 )
  clf
  plot( time,w2_fd,'g', time,w2,'r', time,w2_mean*ones(npts),'k' )
  xlabel( 'Time [sec]' )
  ylabel( 'Crank speed [rad/sec]' )
  legend( 'Finite difference', 'Savitsky-Golay', 'Mean' )
  title( [ 'Web cutter angular velocity - mean=' num2str(w2_mean,4) ' , RMS=' num2str(w2_RMS,3) ]
)
  axis( [ 0 time(end)  2 5.5 ] )

% plot angular velocity versus time with harmonics
figure( 6 )
  clf
  plot( time,w2,'r', time,w2_har,'k' )
  xlabel( 'Time [sec]' )
  ylabel( 'Crank speed [rad/sec]' )
  legend( 'Savitsky-Golay', '11 harmonics' )
  title( 'Web cutter' )
  axis( [ 0 time(end)  2 5.5 ] )

% plot angular velocity versus crank angle
```

```
figure( 7 )
  clf
  plot( mod(th2_deg,360),w2,'ro', mod(th2_deg,360),w2_har,'k'  )
  xlabel( 'Crank angle Theta 2 [deg]' )
  ylabel( 'Crank speed [rad/sec]' )
  legend( 'Savitsky-Golay', '11 harmonics' )
  title( 'Web cutter' )
  axis( [ 0 360  2 5.5 ] )

% plot angular acceleration versus time
figure( 8 )
  clf
  plot( time,a2_fd,'g', time,a2,'r', time,a2_har,'k' )
  xlabel( 'Time [sec]' )
  ylabel( 'Crank angular acceleration [rad/sec/sec]' )
  legend( 'Finite difference', 'Savitsky-Golay', '11 harmonics' )
  title( 'Web cutter' )

% plot angular velocity versus crank angle
figure( 9 )
  clf
  plot( mod(th2_deg,360),a2_fd,'g', mod(th2_deg,360),a2,'r', mod(th2_deg,360),a2_har,'k'  )
  xlabel( 'Crank angle Theta 2 [deg]' )
  ylabel( 'Crank angular acceleration [rad/sec/sec]' )
  legend( 'Finite difference', 'Savitsky-Golay', '11 harmonics' )
  title( 'Web cutter' )

% FFT
a_fft = fft(w2_cen) *2 /npts;     % complex number - units [rad/sec]
a_fft(1) = a_fft(1) /2;           % scale DC term
amp_fft = abs( a_fft );
phase = angle( a_fft ) *180 /pi;  % units [deg]
df = fs /npts;                    % units [Hz]
freq = [ 0:(npts-1) ]' * df;      % units [Hz]

% plot amplitude
figure( 10 )
  clf
  plot( freq,amp_fft )
  axis( [ 0 15  0 0.6 ] )
  xlabel( 'Frequency [Hz]' )
  ylabel( 'Amplitude [rad/sec]' )
  title( 'FFT of w2' )

%{
% plot phase - very noisy
subplot( 2,1,2 )
  plot( freq,phase )
  xlabel( 'Frequency [Hz]' )
  ylabel( 'Phase [deg]' )
%}

% bottom - h03
```