ME 481 - H07 **Name** ________________________

1) Download and view the howitzer cart video "howitzer_cart.mp4" from the class web page. The howitzer cart slides on an air track and fires a ball vertically.

Run video analysis MATLAB code "hc_dig.m" provided below. It will save the first video frame as a JPG image and digitize x-y pixel locations for the ball and cart. Output will be provided in text file "hc_keep.txt".

Use Microsoft Paint to determine frame rate and pixels per meter from image "frame001.jpg".

2) Use finite difference derivatives to provide the following seven MATLAB graphs.

- a) horizontal position of the ball versus time
- b) vertical position of the ball versus time
- c) vertical versus horizontal position of the ball
- d) horizontal velocity of the ball versus time
- e) vertical velocity of the ball versus time
- f) horizontal acceleration of the ball versus time
- g) vertical acceleration of the ball versus time

3) Repeat part 2) above using Savitsky-Golay interpolants and plot **on the same MATLAB graphs.** Provide hard copy of your code.

4) Calculate acceleration of gravity using the slope of vertical velocity of the ball and using the mean vertical acceleration of the ball.

g_{slope_v} = -9.8417 mps^2 g_{mean_acc} = -9.8063 mps^2

5) Calculate coefficient of Coulomb friction drag on the cart and describe how you performed this calculation.

μ = ignore last three data points for cart based on plots and difference between linear fit and mean results
slope of linear fit to horizontal velocity of cart = -0.0533 mps^2 = 0.0054 G $\mu = 0.0054$
mean horizontal acceleration of cart = -0.0552 mps^2 = 0.0056 G $\mu = 0.0056$

6) Calculate mass of the ball and describe how you performed this calculation.

Horizontal velocity of ball decreases almost linearly. Find V_{X_MEAN} for horizontal velocity and
m_{BALL} = use linear fit to find A_{X_FIT}. Aerodynaic drag $F_{DRAG} = 0.5 * C_d * \rho_{AIR} * (V_{X\_MEAN})^2 * A_{FRONTAL}$.
Modify hc_dig.m to save frame002.jpg and measure diameter of ball ~ 10 pixels. Then compute frontal area $A_{FRONTAL}$ and convert to meters^2. Use $C_d = 0.47$ for sphere and $\rho_{AIR} = 1.225$ kg/m^3.
Mass of ball $m_{BALL} = F_{DRAG} / A_{X\_FIT}$ = 0.0041 kg = 4.1 g . It is probably a foam ball.

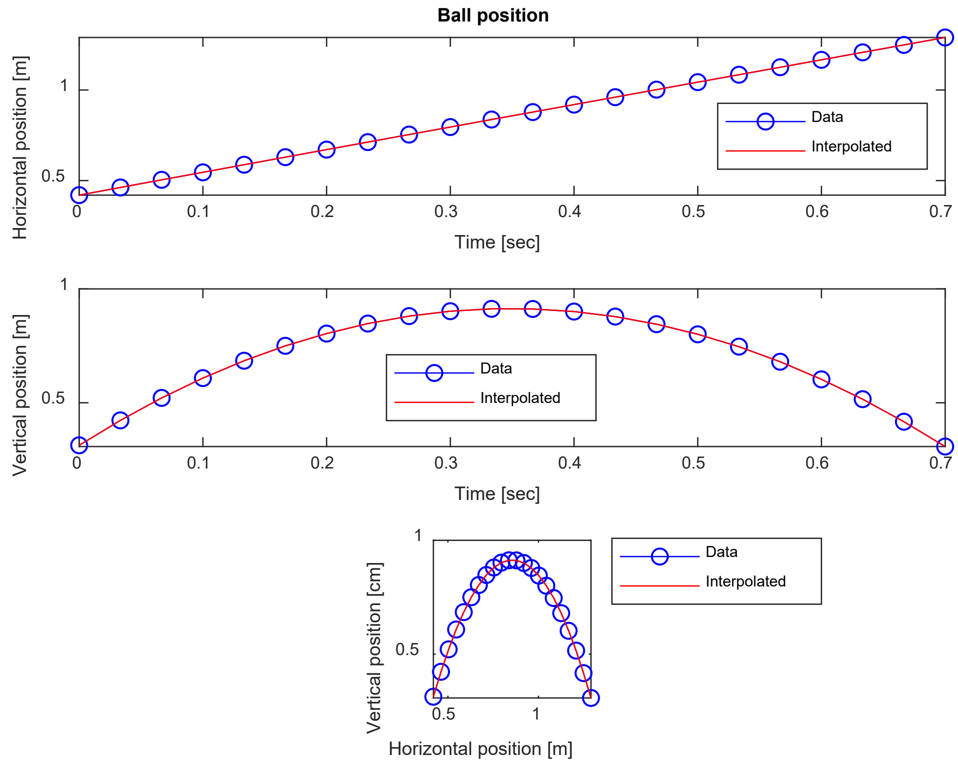
EXTRA CREDIT

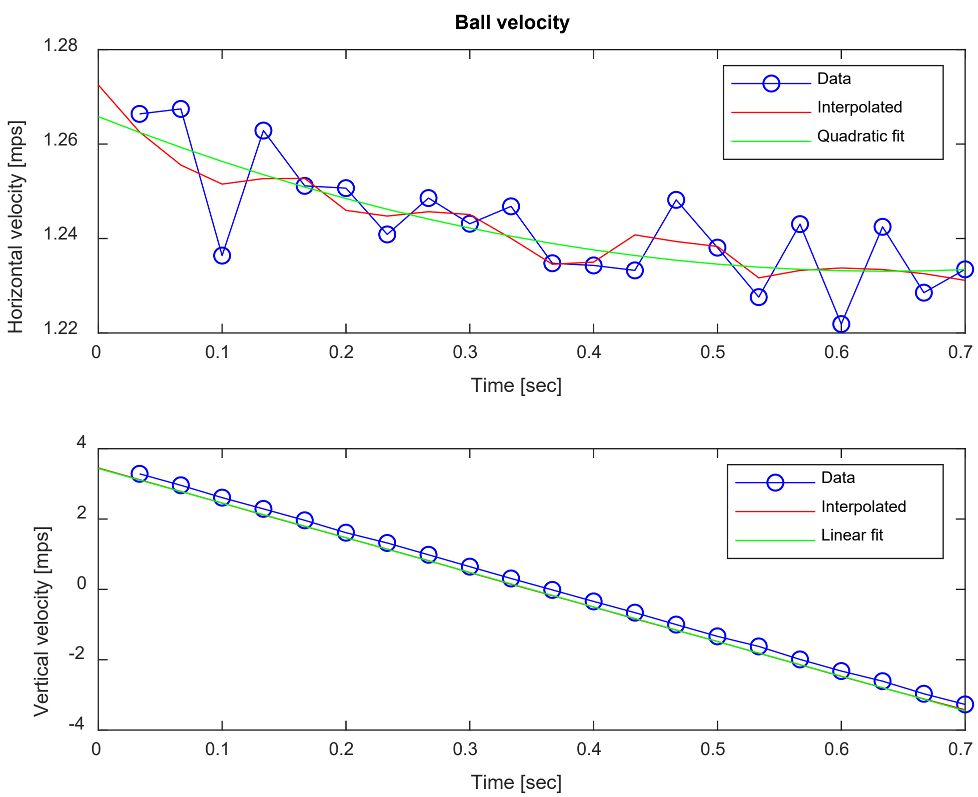
Provide plots for needle position, velocity and acceleration as functions of crank angle from the video "wanzer.mov". Use a*>30 for the red dot and a*<-30 for the green dot. Diameter of the driver disk is 100 mm.

```
% hc_dig.m - digitize ball launched from howitzer cart in MP4 video
% HJSIII, 20.04.24

clear

% video file name
fn_input = [ 'howitzer_cart.mp4' ];

% 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;

% 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 frame as JPG
  s_frame = [ '000' num2str(iframe) ];
  fn_frame = [ 'frame' s_frame( end-2 : end ) '.jpg' ];
  if iframe==1,
    imwrite( a_rgb, fn_frame )
  end

% only analyze frame 2 through 23
  if ( iframe > 1 ) & (iframe< 24 ),

% 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 dark pixels for ball
    bw_dark = ( a_lab(:,:,1) < 40 );  % size (nr,nc) - class logical

% find yellow pixels for dot on cart
    bw_yellow = ( a_lab(:,:,3) > 30 );  % size (nr,nc) - class logical

% find centroid of one object in each black/white image
% use reduced AOI for ball - columns 11 to 640  - rows 51 to 355
    s_ball = regionprops( bw_dark( 51:355, 11:640 ), 'Centroid' );  % class structure
    s_cart = regionprops( bw_yellow, 'Centroid' );

% column and row stored in structure.Centroid
    cr_ball = s_ball.Centroid;  % size (1,2) - class double
    cr_cart = s_cart.Centroid;

% add offsets for ball AOI
    cr_ball(1) = cr_ball(1) + 10;
    cr_ball(2) = cr_ball(2) + 50;

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

% RGB image in UL
  subplot( 2, 2, 1 )
  imshow( a_rgb )
  title( fn_frame )

% BW image for ball in LL
  subplot( 2, 2, 3 )
  imshow( bw_dark )
```

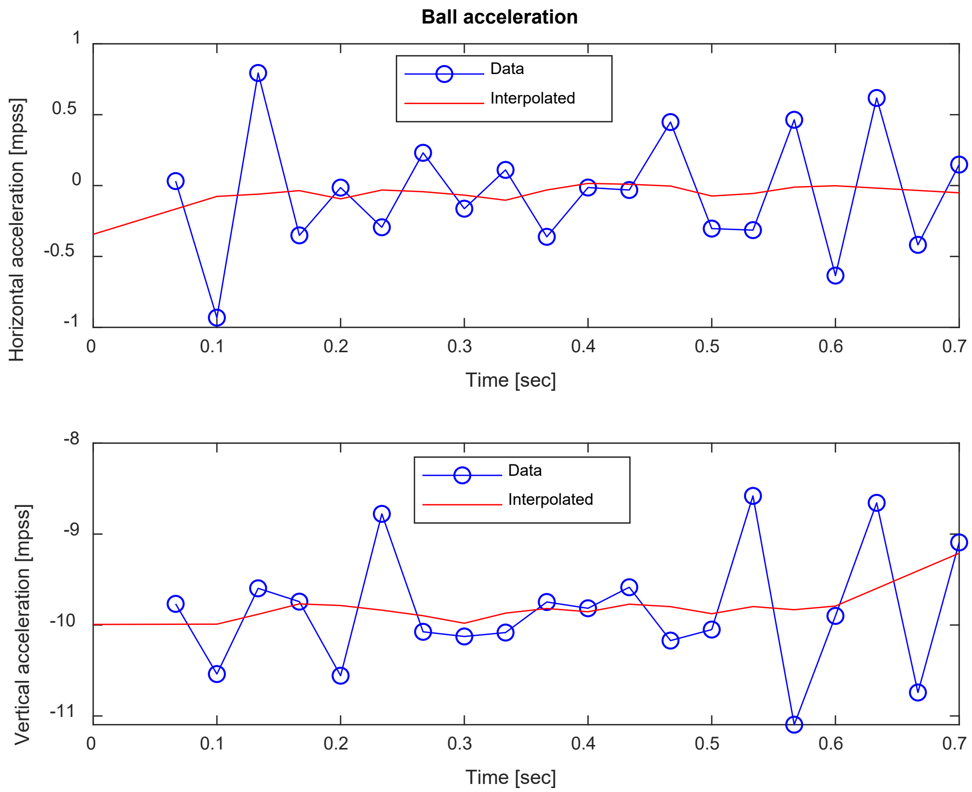
```
  title( 'dark L*<40' )
  hold on
  plot( [ 0 cr_ball(1) ], [ 0 cr_ball(2) ], 'r' )  % line from origin to centroid

% BW image for cart in LR
  subplot( 2, 2, 4 )
  imshow( bw_yellow )
  title( 'yellow b*>30' )
  hold on
  plot( [ 0 cr_cart(1) ], [ 0 cr_cart(2) ], 'y' )  % line from origin to centroid

% update graphics
    drawnow

% save centroids
    keep = [ keep ; [ cr_ball  cr_cart ] ];

  end  % bottom - if iframe
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_ball  y_ball  x_cart  y_cart
save( 'hc_keep.txt', 'keep', '-ascii' )

% bottom - hc_dig
```

ME 481 - H07 **Name** ________________________

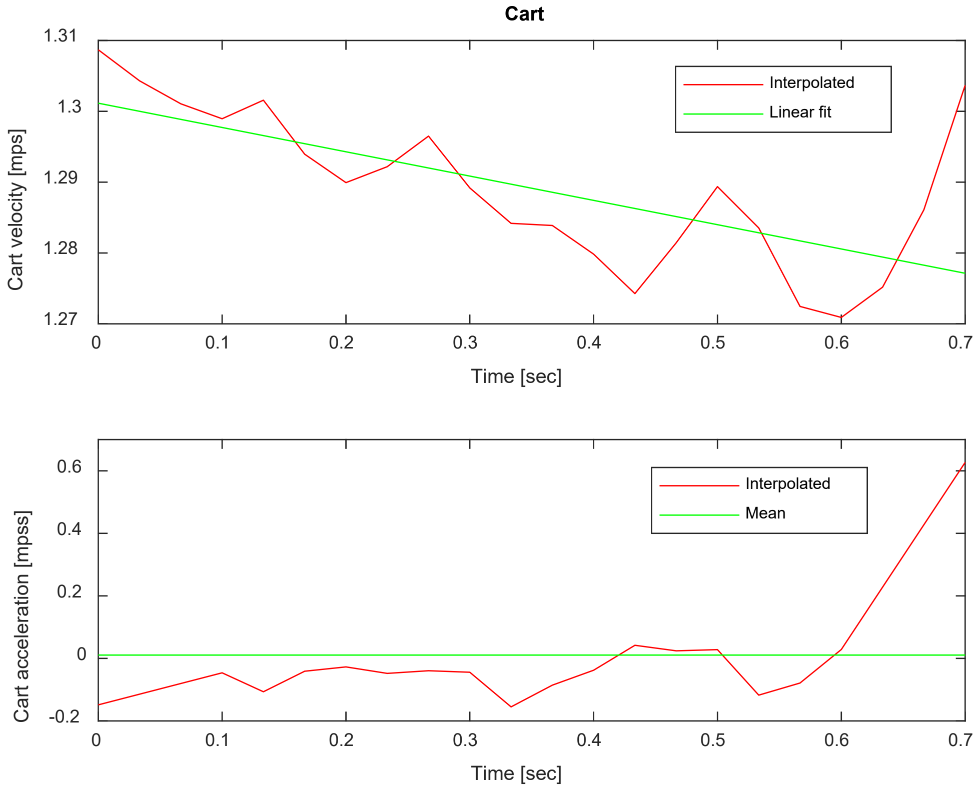


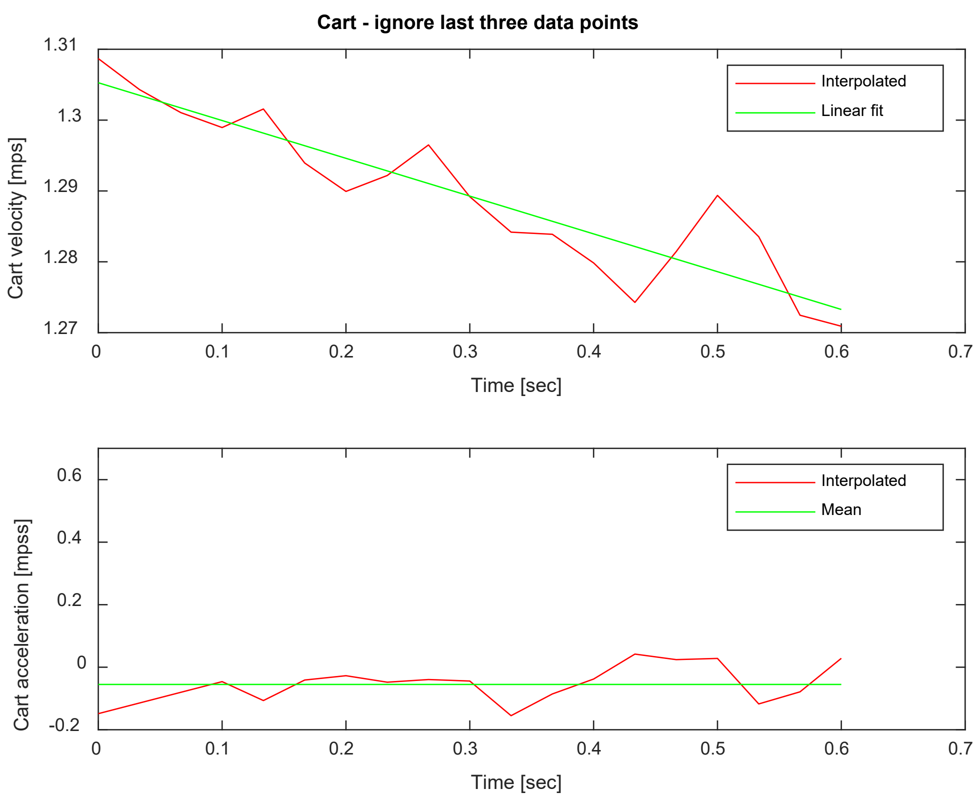


ME 481 - H07 Name ________________________



ME 481 - H07 Name ________________________





```
>> h07

vx_mean =    1.2433

ax_fit =   -0.0463

mass_ball =    0.0041

ay_fit =   -9.8417

ay_mean =   -9.8063

ac_mean =    0.0212

ac_fit =   -0.0286

ac_19_mean =   -0.0554

ac_19_fit =   -0.0542

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% h07.m - analyze howitzer cart data
% HJSIII, 20.04.24

clear

% meters per pixel, frames per second
mpp = 1 / (533-103);
fps = 30;
h = 1 / fps;

% read raw data
% frame, top x y, bottom x y, right x y, left x y, cart x
raw = load( 'hc_keep.txt' );
x = mpp * raw(:,1)';   % 1xn vector
y = mpp * raw(:,2)';
x_cart = mpp * raw(:,3)';

xy = [ x ; y ];        % 2xn matrix
[ k, n ] = size( xy );
nm1 = n - 1;
t = h * ( 0: (n-1) );  % 1xn vector

% first order finite differences
vx = [ NaN  diff(xy(1,:)) ] / h;
vy = [ NaN  diff(xy(2,:)) ] / h;

ax = [ NaN  diff(vx) ] /h;
ay = [ NaN  diff(vy) ] /h;

% Savitsky-Golay interpolant and derivatives
[ pxy, vxy, axy, jxy ] = filt_7pt_mat( xy, h );
[ pc, vc, ac, jc ]     = filt_7pt_mat( x_cart, h );

% horizontal motion of ball
vx_mean = mean( vxy(1,:) )
p = polyfit( t, vxy(1,:), 1 );
vx_fit = polyval( p, t );
ax_fit = p(1)

% aerodynamic drag for sphere
D_pix = 10;         % ball diameter [pixels] digitized from frame002
D_m = D_pix * mpp;  % ball diameter [m]
A = pi*D_m*D_m/4;   % ball frontal area [m.m]
Cd = 0.47;          % drag coefficient [dimensionless]
rho_air = 1.225;    % [kg/m/m/m]
F_drag = 0.5 * Cd * rho_air * vx_mean * vx_mean * A;
mass_ball = F_drag / abs(ax_fit)

% vertical motion of ball
p = polyfit( t, vxy(2,:), 1 );
vy_fit = polyval( p, t );
ay_fit = p(1)
ay_mean = mean( axy(2,:) )
```

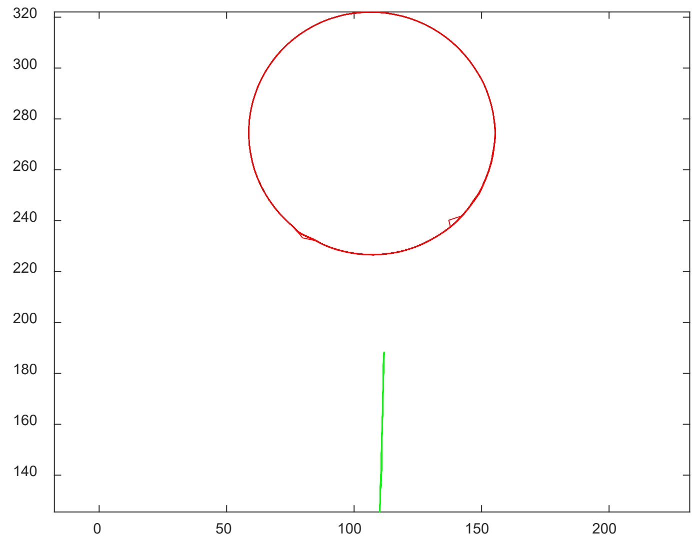
```
% horizontal motion of cart
ac_mean = mean( ac )
p = polyfit( t, vc, 1 );
vc_fit = polyval( p, t );
ac_fit = p(1)

% ignore last three data points for cart
t_19 = t(1:19);
vc_19 = vc(1:19);
ac_19 = ac(1:19);
ac_19_mean = mean( ac_19 )
p = polyfit( t_19, vc_19, 1 );
vc_19_fit = polyval( p, t_19 );
ac_19_fit = p(1)

% plot ball position
figure( 1 )
  clf
  subplot( 3,1,1 )
  plot( t,xy(1,:),'bo-', t,pxy(1,:),'r' )
  xlabel( 'Time [sec]' )
  ylabel( 'Horizontal position [m]' )
  title( 'Ball position' )
  legend( 'Data', 'Interpolated' )

  subplot( 3,1,2 )
  plot( t,xy(2,:),'bo-', t,pxy(2,:),'r' )
  xlabel( 'Time [sec]' )
  ylabel( 'Vertical position [m]' )
  legend( 'Data', 'Interpolated' )

  subplot( 3,1,3 )
  plot( xy(1,:),xy(2,:),'bo-', pxy(1,:),pxy(2,:),'r' )
  axis square
  xlabel( 'Horizontal position [m]' )
  ylabel( 'Vertical position [cm]' )
  legend( 'Data', 'Interpolated' )

% plot ball velocity
figure( 2 )
  clf
  subplot( 2,1,1 )
  plot( t,vx,'bo-', t,vxy(1,:),'r', t,vx_fit,'g' )
  xlabel( 'Time [sec]' )
  ylabel( 'Horizontal velocity [mps]' )
  title( 'Ball velocity' )
  legend( 'Data', 'Interpolated', 'Linear fit' )

  subplot( 2,1,2 )
  plot( t,vy,'bo-', t,vxy(2,:),'r', t,vy_fit,'g' )
  xlabel( 'Time [sec]' )
  ylabel( 'Vertical velocity [mps]' )
  legend( 'Data', 'Interpolated', 'Linear fit' )

% plot ball acceleration
figure( 3 )
  clf
  subplot( 2,1,1 )
  plot( t,ax,'bo-', t,axy(1,:),'r' )
  xlabel( 'Time [sec]' )
  ylabel( 'Horizontal acceleration [mpss]' )
  title( 'Ball acceleration' )
  legend( 'Data', 'Interpolated' )

  subplot( 2,1,2 )
  plot( t,ay,'bo-', t,axy(2,:),'r' )
  xlabel( 'Time [sec]' )
  ylabel( 'Vertical acceleration [mpss]' )
  legend( 'Data', 'Interpolated' )

% plot cart velocity and acceleration - all data points
figure( 4 )
  clf
  subplot( 2,1,1 )
```

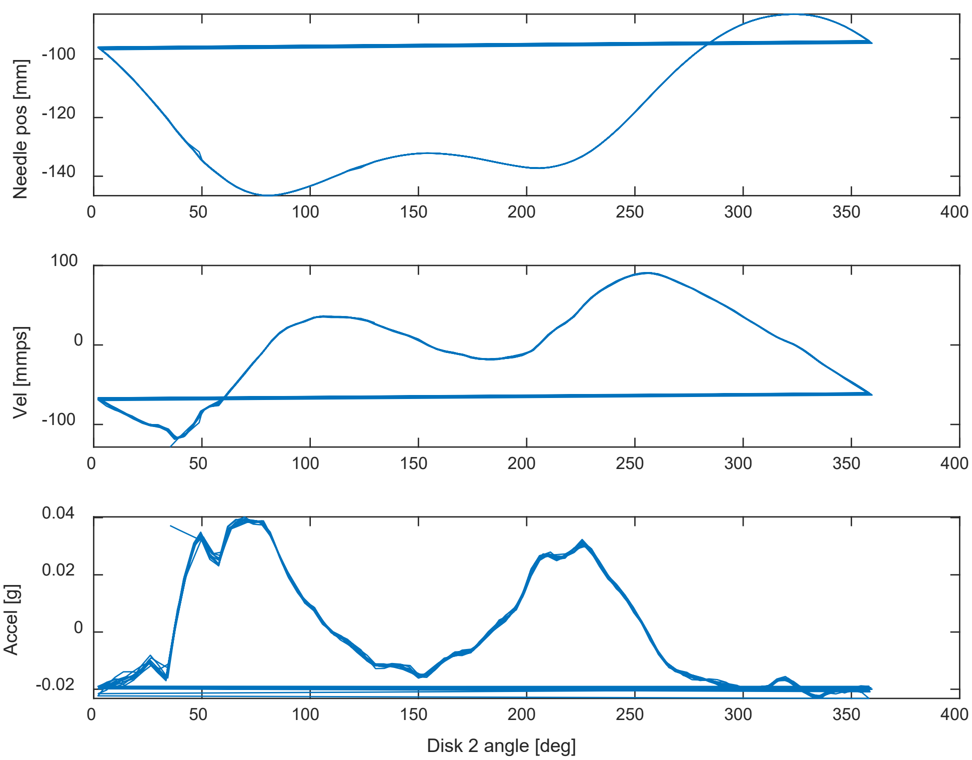
```
  plot( t,vc,'r', t,vc_fit,'g' )
  xlabel( 'Time [sec]' )
  ylabel( 'Cart velocity [mps]' )
  title( 'Cart' )
  legend( 'Interpolated', 'Linear fit' )
  axis( [ 0 0.7  1.27 1.31 ] )

  subplot( 2,1,2 )
  plot( t,ac,'r', [ t(1) t(end) ],[ ac_mean ac_mean ],'g' )
  xlabel( 'Time [sec]' )
  ylabel( 'Cart acceleration [mpss]' )
  legend( 'Interpolated', 'Mean' )
  axis( [ 0 0.7  -0.2 0.7 ] )

% plot cart velocity and acceleration - ignore last three data points
figure( 5 )
  clf
  subplot( 2,1,1 )
  plot( t_19,vc_19,'r', t_19,vc_19_fit,'g' )
  xlabel( 'Time [sec]' )
  ylabel( 'Cart velocity [mps]' )
  title( 'Cart - ignore last three data points' )
  legend( 'Interpolated', 'Linear fit' )
  axis( [ 0 0.7  1.27 1.31 ] )

  subplot( 2,1,2 )
  plot( t_19,ac_19,'r', [ t_19(1) t_19(end) ],[ ac_19_mean ac_19_mean ],'g' )
  xlabel( 'Time [sec]' )
  ylabel( 'Cart acceleration [mpss]' )
  legend( 'Interpolated', 'Mean' )
  axis( [ 0 0.7  -0.2 0.7 ] )

% bottom - h07
```

ME 481 - H07 **Name** ________________________

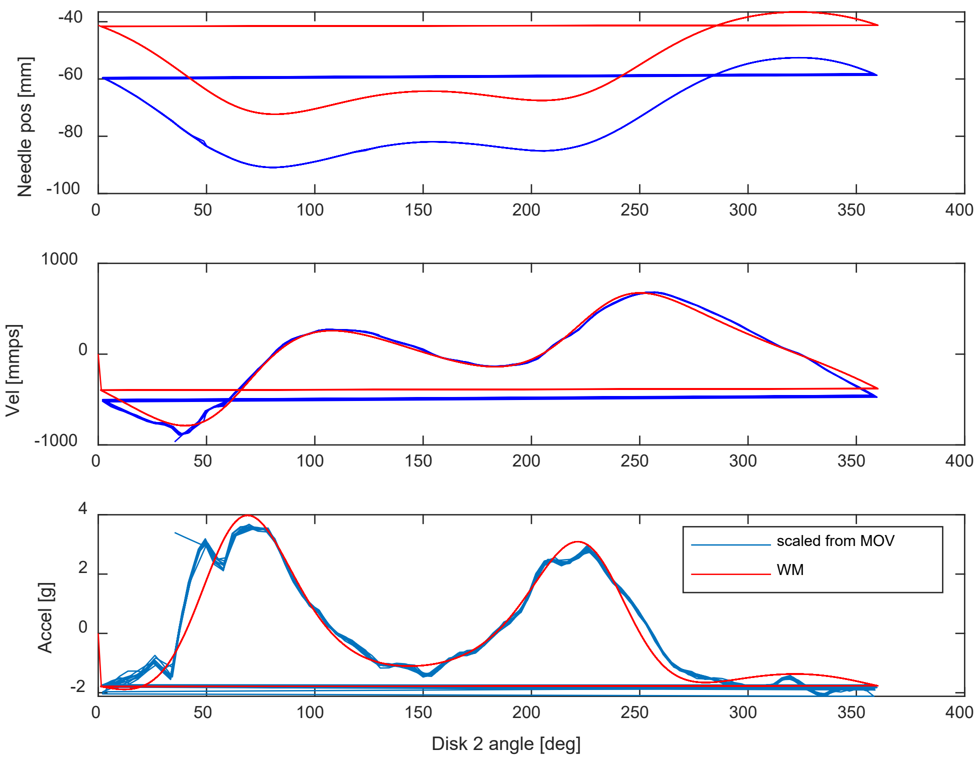
EXTRA CREDIT

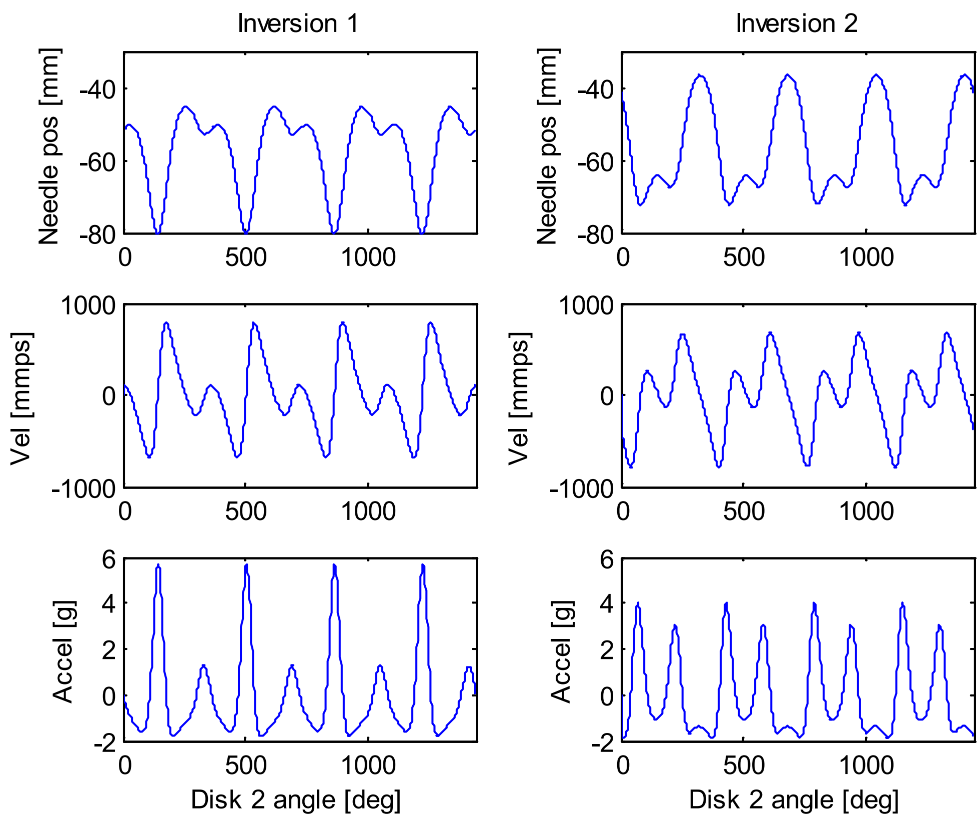
Provide plots for needle position, velocity and acceleration as functions of crank angle from the video "wanzer.mov". Use a*>40 for the red dot and a*<-30 for the green dot. Diameter of the driver disk is 100 mm.





ME 481 - H07 Name ________________________





```
% wanzer_dig.m - extra credit - digitize Wanzer needle bar mechanism in MOV video
% HJSIII, 19.12.04

clear

% video file name
fn_input = [ 'wanzer.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;

% 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 frame as JPG
  s_frame = [ '000' num2str(iframe) ];
  fn_frame = [ 'frame' s_frame( end-2 : end ) '.jpg' ];
  if iframe==1,
    imwrite( a_rgb, fn_frame )
  end

% only analyze frame 2 through 23
%  if ( iframe > 1 ) & (iframe< 24 ),

% 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 for crank disk
    bw_red = ( a_lab(:,:,2) > 30 );  % size (nr,nc) - class logical

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

% find centroid of one object in each black/white image
% use reduced AOI for red dot on crank
    s_crank = regionprops( bw_red( 301:1000, : ), 'Centroid' );  % class structure
    s_needle = regionprops( bw_green, 'Centroid' );

% column and row stored in structure.Centroid
    cr_crank = s_crank.Centroid;  % size (1,2) - class double
    cr_needle = s_needle.Centroid;

% add offsets for crank AOI
    cr_crank(2) = cr_crank(2) + 300;

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

% RGB image in UL
  subplot( 2, 2, 1 )
  imshow( a_rgb )
  title( fn_frame )

% BW image for red in LL
  subplot( 2, 2, 3 )
  imshow( bw_red )
  title( 'red a*>30' )
```

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

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

% update graphics
    drawnow

% save centroids
    keep = [ keep ; [ cr_crank  cr_needle ] ];

%  end  % bottom - if iframe
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_crank  y_crank  x_needle  y_needle
save( 'wanzer_keep.txt', 'keep', '-ascii' )

% bottom - wanzer_dig
```

ME 481 - H07 Name ________________________

```
% wanzer_plot - plot Wanzer data digitized from wanzer.mov
% HJSIII - 20.01.24

% constants
d2r = pi / 180;

% data from frame001.jpg using MS-Paint
disk_pix = 486;
col_c = 512;
row_c = 607;
nrow = 1920;

% actual DIA of disk
disk_dia = 100; % [mm]

% pixels per mm
ppmm = disk_pix /disk_dia;

% load raw data
raw = load( 'wanzer_keep.txt' );
[ n, nc ] = size( raw );

% scale pixel to mm
raw = raw /ppmm;

% check plot for raw data
figure( 10 )
  clf
  plot( raw(:,1),raw(:,2),'r', raw(:,3),raw(:,4),'g' )
  axis equal

% time step - 30 fps
h = 1 / 30;

% crank angle
xc = col_c /ppmm;
yc = (nrow - row_c) /ppmm;
th = -unwrap( atan2( raw(:,2)-yc, raw(:,1)-xc ) );
th_deg = mod( th/d2r, 360 );

% time and angular velocity of crank
t = h * ( 0 : (n-1) )';
p = polyfit( t, th, 1 );
w_disk = p(1);             % [rad/sec]

% needle position
xy = raw(:,3:4)';

% Savitsky-Golay smoothing
[ p, v, a, j ] = filt_7pt_mat( xy, h );

% plot vertical motion
figure( 1 )
  clf
  subplot( 3, 1, 1 )
    plot( th_deg, p(2,:)-yc )
    ylabel( 'Needle pos [mm]' )

% plot vertical velocity
  subplot( 3, 1, 2 )
    plot( th_deg, v(2,:) )
    ylabel( 'Vel [mmps]' )

% plot vertical acceleration
  subplot( 3, 1, 3 )
    plot( th_deg, a(2,:)/9810 )
    ylabel( 'Accel [g]' )
    xlabel( 'Disk 2 angle [deg]' )

% scale to match H05
disk_dia_H05 = 62;   % [mm]
pos_scale = disk_dia_H05 / disk_dia;

w_H05 = 2*pi * 240 /60;  % [rad/sec]  for 240 rpm
w_scale = w_H05 / w_disk;
```

```
vel_scale = pos_scale * w_scale;
acc_scale = pos_scale * w_scale * w_scale;

% load WM vel and accel
keep_wm = load( 'h05_keep_wm.txt' );
th_wm = keep_wm(:,1);
pos_wm = keep_wm(:,2);
vel_wm = keep_wm(:,3);
acc_wm = keep_wm(:,4);

% plot vertical motion scaled to match H05
figure( 2 )
  clf
  subplot( 3, 1, 1 )
    plot( th_deg,(p(2,:)-yc)*pos_scale,'b', th_wm,pos_wm,'r'  )
    ylabel( 'Needle pos [mm]' )

% plot vertical velocity
  subplot( 3, 1, 2 )
    plot( th_deg,v(2,:)*vel_scale,'b', th_wm,vel_wm,'r' )
    ylabel( 'Vel [mmps]' )

% plot vertical acceleration
  subplot( 3, 1, 3 )
    plot( th_deg,a(2,:)/9810*acc_scale, th_wm,acc_wm,'r' )
    ylabel( 'Accel [g]' )
    xlabel( 'Disk 2 angle [deg]' )
    legend( 'scaled from MOV', 'WM' )

% bottom - wanzer_plot
```