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 as well as area for the ball in pixels squared. 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.

gslope\_v = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ gmean\_acc = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

 = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

mBALL = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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, 21.03.12

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', ‘Area’ ); % 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;

area\_ball = s\_ball.Area; % scalar - class double

% 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 area\_ball ] ];

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