ME 581: Simulation of Mechanical Systems
Inverse Kinematics of a 3-Degree of Freedom Solar Tracker

Greg Brulo

Abstract

ARPA-E’s MOSAIC (Micro-scale Optimized Solar-cell Arrays with Integrated Concentration) program’s goal is to “lower the cost of solar systems by using integrated concentration to significantly increase PV module efficiency without increasing manufacturing costs.” The Pennsylvania State University’s solution to this problem requires actuators to move the PV (photovoltaic) cells throughout the day to keep them in the path of the concentrated light. This report covers the kinematic analysis of a concept actuation system. The goal of the analysis is to determine if the current actuator concept is feasible.

2. Methodology

2.1. 2D Model Development and Definitions

The notation used in the kinematic analysis was the E. J. Haug method. [1]

{r_i} - the global position of the origin of the reference frame attached to body i
{r_i}^P - global position of point P attached to body i
{s_i}^P - position of point P on body i relative to the reference frame for body i measured in local body-fixed directions.
{d_{ij}} – relative location between two points on bodies i and j measured in global directions
[\mathbf{A}_i] – orthonormal rotation matrix that describes attitude of body i

[2]

Figure 1 illustrates the link locations on the mechanism.

Link 1 – Solar panel frame
Link 3 – Middle sheet containing photovoltaic cells
Links 2, 4, 5 – Actuation wires (variable length)
Links 6, 7, 8 – Springs (variable length)

The links connect from points on the frame (link 1) to points on the middle sheet (link 3). For example link 2 connects point A on the frame to point A on the middle sheet. The diagram shows point A and point J to be at the same location on the middle sheet. This is only true in this configuration. Different design iterations may position points A and J at a place along the edge other than at the corner. The location of these points are controlled in the MATLAB code found in Appendix 2.

The 2D model representation allows the middle sheet to move in a planar method. It is allowed to translate in two directions and rotate on the frame.

When the middle sheet is centered the local origin for link 3 is coincident with the global origin on link 1 as seen in Figure 1.

2.2. Inverse Kinematics

The goal of the analysis is to determine if the current actuator concept is feasible. An inverse kinematics analysis was chosen since to determine the lengths and forces for each link for a given location of the middle sheet (link 3). A feasible concept requires the control wires (links 2, 4, 5) to maintain tension for all positions of the middle sheet.

Global locations for each point were solved using the transformation equation (1.1).
Vectors \( \{d_{ij}\} \) describe the length and orientation of each links.

\[
\{d_{ij}\} = \{r_j\}^P - \{r_i\}^P
\]  \hspace{1cm} \text{(1.2)}

The link length and angle were derived from equation (1.2).

Spring forces were solved once the length and angles of links 6, 7 and 8 were known.

\[
F_i = (L_i - L_{i, \text{initial}}) \cdot k_i + \text{initial tension}_i
\]  \hspace{1cm} \text{(1.3)}

where:
- \( L_i \) – initial length of link \( i \)
- \( L_{i, \text{initial}} \) – initial length (length when \( r_3 = \{0,0\}^T \))
- \( k_i \) – spring rate for link \( i \)
- \( \text{initial tension}_i \) – tension when \( L_i = L_{i, \text{initial}} \)

Three equations were used to solve for the forces from links 2, 4 and 5. These three equations were the sum of forces in the global x direction on link 3, sum of forces in the global y direction on link 3 and the sum of moments about the local origin on link 3. The local origin of link 3 was at the center of gravity location. These equations can be found in the MATLAB code. The forces from links 2, 4, 5, 6, 7 and 8 can be found in Figure 2.

### 2.3. Inverse Kinematics and MATLAB

#### Execution

After the equations for finding the link lengths, angles, locations and forces were derived, they were executed in MATLAB. The MATLAB code loads two files near the beginning which load the configurable blue print geometry for links 1 and 3. Spring parameters for links 6, 7 and 8 can be configured in the main code for design iterations.

The path for link 3 must be chosen by the operator. The operator can load a predetermined path or manual place link 3 in the positions the operator is interested in.

Once all the parameters are loaded, the MATLAB code uses a for loop to solve the kinematics at each position of link 3 in the path. The code uses the MATLAB linear system of equations solver to solve the 3 equations for the forces from links 2, 4 and 5 inside the for loop.

Once the for loop solves the kinematics at each position of the path, the MATLAB code loads the next path if required. This allows the user to execute the kinematic solution for multiple paths.

### 3. Simulation and Results

The feasibility of the design was evaluated by positioning the link 3 in different positions about its maximum envelope of travel. The middle sheet must be actuated +/- 10 mm in both the global x and global y direction from a centered position. Link 3 is considered centered when the global position of its origin is \( \{0,0\}^T \).

After either link 1 or link 3 geometry changed, the initial lengths of links 6, 7 and 8 needed calculated. This was done by loading a manual path for link 3 into the MATLAB simulation. The manual path placed link 3 in its centered position. These initial lengths were then included in the parameters for the springs.

The first design concept tested did not include links 7 and 8. This design only used one spring,
link 6. Unfortunately this design was not feasible since link 2 was in compression when the global location of link 3 was \((10,0)^T\).

After the first design concept did not work springs 7 and 8 were added. This concept worked at most positions of link 3. Some positions of link 3 required either link 2, 4 or 5 to be in compression. This compression could be corrected at the given position by changing the spring attachment points on link 1. However this practice resulted in a link having a compressive force at a different location of link 3.

A hypothesis was made describing the phenomenon of not being able to have a single set of spring geometry which results in tension of links 2, 4 and 5 at every position of link 3 was caused by the dramatic angle changes of the springs. The change in transmission angle of link 8 can be seen in Figure 3. The transmission angle changes from 63° to 90° to 153° in the given example. The hypothesis describes the reason for the links changing from tension to compression is a direct result of the larger than ideal transmission angle changes of the spring links such as link 8.

The blue print information for link 1 was updated in MATLAB and the analysis was repeated. Now links 2, 4 and 5 maintain tension at every position of link 3.

A sample path for how the solar tracker would move link 3 throughout the day was loaded into the MATLAB model. This path represents an implementation of the solar tracker in State College Pennsylvania. Appendix 1 shares the position data and force data for this application.

4. Conclusion

This model was successfully used to evaluate the feasibility of the PSU MOSAIC design concept. The model showed the first concept was not feasible. A second concept was created based on these results. The second concept was proven to be feasible based on an analysis using this model. This tool will be used to study further iterations of the design concepts after publishing this report.

References


Appendix 1

The kinematic and force results simulate the latest design concept in State College Pennsylvania. This design uses the small transmission angle shifts therefore the spring and control wires are longer than ideal.
The MATLAB code for the inverse kinematics model is shared here. Some of the code originates from Dr. Sommer’s website as cited for reference [2].

% File name - MOSAIC_kinematic_forces_main_rev_4.m
% rev 4 - added spring in top left corner

% Greg Bruno
% MOSAIC Kinematics
% Spring 2017

% Use vector and matrix notation from Dr. Sommers Notes 04 01.

% ideas
% 1 - have interrupt to confirm the parameters are all set
% Units are in mm, sec, N, radians

clear
clear all
clc
d2r=pi/180; % convert from degrees to radians

% Load the blue print information for link 1 and link 3
MOSAIC_Link_1_Info_rev_1
MOSAIC_Link_3_Info_rev_1

% Load the path from Alex Grede
SC_equisol_20deg=load('SC_equisol_20deg.txt');
run_1_x=SC_equisol_20deg(1:55,1); %integer
run_1_y=SC_equisol_20deg(1:55,2); %mm

% run 1 data columns
% 1 - seq number
% 2 - X position mm
% 3 - Y position mm

% Spring info
% initial_length=60; % mm
k6=0.04; % N/mm
k7=0.04; % N/mm
k8=0.04; % N/mm
initial_tension_6=9; % Newtons
initial_tension_7=9; % Newtons
initial_tension_8=9; % Newtons

% configure the tilt info
mass=0;
phi_incline=45;
g=9.81;

% Path Info
% make a path with the middle sheet centered in y direction
r3_path_1=zeros(201,2);
% make a path based on actual data
r3_path_4(:,1:2)=run_1_x*2.25,run_1_y*2.25;
% make a test path
r3_path_5=[10 -10;10 -10];

adb=1;
while adb<=1
  clear r3_path info; % clear these variables
  if 1==adb
    r3_path=r3_path_1;
  elseif 2==adb
    r3_path=r3_path_2;
  elseif 3==adb
    r3_path=r3_path_3;
  elseif 4==adb
    r3_path=r3_path_4;
  elseif 5==adb
    r3_path=r3_path_5;
  else
    'error assigning adb and picking r3_path'
  end

adb=1;
adn_size=size(r3_path);
adn_max=min(adn_size(1,1));

while adn<=adn_max
  % Describe where the link 3 is relative to link 1
  r3=r3_path(adn,:); % Location of r3 relative to r1
  phi3=0*d2r; % Enter degrees, but store radians
  A3=[cos(phi3) sin(phi3); sin(phi3) cos(phi3)]; % rotation matrix
  q3=[r3; phi3]; % generalized coordinates for link 3

  r1=[0,0]'; % location of global origin (should be 0,0)
  phi1=0*d2r; % Angle of ground (should be 0)
  A1=[cos(phi1) sin(phi1); sin(phi1) cos(phi1)]; % rotation matrix
  q1=[r1; phi1]; % generalized coordinates for link 1 (should be 0,0)

  % global locations - Eq 2.4.8, page 33 (Dr. Sommer)

  r1A = r1 + A1*s1pA;
  r1B = r1 + A1*s1pB;
  r1C = r1 + A1*s1pC;
  r1D = r1 + A1*s1pD;
  r1E = r1 + A1*s1pE;
  r1F = r1 + A1*s1pF;
  r1G = r1 + A1*s1pG;
  r1H = r1 + A1*s1pH;
  r1I = r1 + A1*s1pI;
  r1J = r1 + A1*s1pJ;

  r3A = r3 + A3*s3pA;
  r3B = r3 + A3*s3pB;
  r3C = r3 + A3*s3pC;
  r3D = r3 + A3*s3pD;
  r3E = r3 + A3*s3pE;
  r3F = r3 + A3*s3pF;
  r3G = r3 + A3*s3pG;
  r3H = r3 + A3*s3pH;
  r3I = r3 + A3*s3pI;
  r3J = r3 + A3*s3pJ;

  % find the link vectors
  % length of wires and direction relative to the frame
  D2=r3A-r1A;
  D4=r3C-r1C;
  D5=r3B-r1B;
  D6=r3D-r1D;
  D7=r3I-r1I;
  D8=r3J-r1J;

  % length of the wires from frame to link 3
  L2=sqrt(D2(1,1)^2+D2(2,1)^2);
  L4=sqrt(D4(1,1)^2+D4(2,1)^2);
  L5=sqrt(D5(1,1)^2+D5(2,1)^2);
  L6=sqrt(D6(1,1)^2+D6(2,1)^2);
  L7=sqrt(D7(1,1)^2+D7(2,1)^2);
  L8=sqrt(D8(1,1)^2+D8(2,1)^2);

  % angle of the wires relative to the frame
  phi2=abs(atan(D2(2,1)/D2(1,1)));
  phi4=abs(atan(D4(2,1)/D4(1,1)));
  phi5=abs(atan(D5(2,1)/D5(1,1)));
  phi6=abs(atan(D6(2,1)/D6(1,1)));
  phi7=abs(atan(D7(2,1)/D7(1,1)));
  phi8=abs(atan(D8(2,1)/D8(1,1)));

  if [D2(1,1)>=0 & D2(2,1)>=0]
phi2 = phi2;
elseif (D2(1,1)<0 && D2(2,1)>=0)
% quadrant 2
'quadrant2'
phi2 = pi - phi2;
elseif (D2(1,1)<0 && D2(2,1)<0)
% quadrant 3
'quadrant3'
phi2 = pi + phi2;
elseif (D2(1,1)>=0 && D2(2,1)<0)
% quadrant 4
'quadrant4'
phi2 = 2*pi - phi2;
else
% error
'error1'
end

if (D4(1,1)>=0 && D4(2,1)>=0)
% quadrant 1
'quadrant1'
phi4 = phi4;
elseif (D4(1,1)<0 && D4(2,1)>=0)
% quadrant 2
'quadrant2'
phi4 = pi - phi4;
elseif (D4(1,1)<0 && D4(2,1)<0)
% quadrant 3
'quadrant3'
phi4 = pi + phi4;
elseif (D4(1,1)>=0 && D4(2,1)<0)
% quadrant 4
'quadrant4'
phi4 = 2*pi - phi4;
else
% error
'error2'
end

if (D5(1,1)>=0 && D5(2,1)>=0)
% quadrant 1
'quadrant1'
phi5 = phi5;
elseif (D5(1,1)<0 && D5(2,1)>=0)
% quadrant 2
'quadrant2'
phi5 = pi - phi5;
elseif (D5(1,1)<0 && D5(2,1)<0)
% quadrant 3
'quadrant3'
phi5 = pi + phi5;
elseif (D5(1,1)>=0 && D5(2,1)<0)
% quadrant 4
'quadrant4'
phi5 = 2*pi - phi5;
else
% error
'error3'
end

if (D6(1,1)>=0 && D6(2,1)>=0)
% quadrant 1
'quadrant1'
phi6 = phi6;
elseif (D6(1,1)<0 && D6(2,1)>=0)
% quadrant 2
'quadrant2'
phi6 = pi - phi6;
elseif (D6(1,1)<0 && D6(2,1)<0)
% quadrant 3
'quadrant3'
phi6 = pi + phi6;
elseif (D6(1,1)>=0 && D6(2,1)<0)
% quadrant 4
'quadrant4'
phi6 = 2*pi - phi6;
else
% error
'error4'
end

if (D7(1,1)>=0 && D7(2,1)>=0)
% quadrant 1
'quadrant1'
phi7 = phi7;
elseif (D7(1,1)<0 && D7(2,1)>=0)
% quadrant 2
'quadrant2'
phi7 = pi - phi7;
elseif (D7(1,1)<0 && D7(2,1)<0)
    % quadrant 3
    phi7=pi+phi7;
elseif (D7(1,1)>=0 && D7(2,1)<0)
    % quadrant 4
    phi7=2*pi-phi7;
else
    % error
    'error5'
end
if (D8(1,1)>=0 && D8(2,1)>=0)
    % quadrant 1
    phi8=phi8;
elseif (D8(1,1)<0 && D8(2,1)>=0)
    % quadrant 2
    phi8=pi-phi8;
elseif (D8(1,1)<0 && D8(2,1)<0)
    % quadrant 3
    phi8=pi+phi8;
elseif (D8(1,1)>=0 && D8(2,1)<0)
    % quadrant 4
    phi8=2*pi-phi8;
else
    % error
    'error6'
end
% Find the transmission angles
phi2_nominal=0;  % radians
phi4_nominal=0;  % radians
phi5_nominal=pi/2;  % radians
phi6_nominal=pi;  % radians
phi7_nominal=0;  % radians
phi8_nominal=1.5*pi;  % radians
phi2_deviation = phi2/d2r - phi2_nominal/d2r;  % degrees
phi4_deviation = phi4/d2r - phi4_nominal/d2r;  % degrees
phi5_deviation = phi5/d2r - phi5_nominal/d2r;  % degrees
phi6_deviation = phi6/d2r - phi6_nominal/d2r;  % degrees
phi7_deviation = phi7/d2r - phi7_nominal/d2r;  % degrees
phi8_deviation = phi8/d2r - phi8_nominal/d2r;  % degrees
theta2=phi2+pi;
theta4=phi4+pi;
theta5=phi5+pi;
theta6=phi6-pi;
theta7=phi7-pi;
theta8=phi8-pi;
% Find the forces in the wires
% force due to the weight of the middle sheet
Fw=mass*g*sin(phi_incline);
% solve for F6, F6x, F6y
F6=(L6-L6i)*k6+initial_tension_7;
F6x=F6*cos(theta6);
F6y=F6*sin(theta6);
% solve for F7, F7x, F7y
F7=(L7-L7i)*k7+initial_tension_7;
F7x=F7*cos(theta7);
F7y=F7*sin(theta7);
% solve for remaining forces
syms F2 F4 F5
eqnl = F2*cos(theta2)+F4*cos(theta4)+F5*cos(theta5)...
    +F6*cos(theta6)+F7*cos(theta7)+F8*cos(theta8)==0;
eqnm = F2*sin(theta2)+F4*sin(theta4)+F5*sin(theta5)+F6*sin(theta6)...
    +F7*sin(theta7) +F8*sin(theta8)+ mass*g*sin(phi_incline)==0;
eqnx = F2*cos(theta6)*s3pD(1,1) + F2*sin(theta6)*s3pD(2,1)...
    + F4*cos(theta4)*s3pB(1,1) + F4*sin(theta4)*s3pB(2,1)...
    + F5*cos(theta5)*s3pC(1,1) + F5*sin(theta5)*s3pC(2,1)...
    + F6*cos(theta6)*s3pI(1,1) + F6*sin(theta6)*s3pI(2,1)...
    + F7*cos(theta7)*s3pJ(1,1) + F7*sin(theta7)*s3pJ(2,1)...
    + F8*cos(theta8)*s3pK(1,1) + F8*sin(theta8)*s3pK(2,1)==0;
[A,B]=equationsToMatrix([eqn1,eqn2,eqn3],...
    [F2,F4,F5]);
forces=linzolve(A,B);
F2=double(forces(1,1));
F4=double(forces(2,1));
F5=double(forces(3,1));

F2x=F2*cos(theta2);
F2y=F2*sin(theta2);
F4x=F4*cos(theta4);
F4y=F4*sin(theta4);
F5x=F5*cos(theta5);
F5y=F5*sin(theta5);

% if the wires come out of tension (have negative force) then
% the status is set to 1

% if(F2x<=0 & F2y>=0)
%  F2_status=0;
% else
%     F2_status=1;
%     F2=-F2;
% end
%
% if(F4x<=0 & F4y<=0)
%  F4_status=0;
% else
%     F4_status=1;
%     F4=-F4;
% end
%
% if(F5x<=0 & F5y<=0)
%  F5_status=0;
% else
%     F5_status=1;
%     F5=-F5;
% end
%
% Store information
info(adn,1:37)=[adn,
            ...r3(1,1),
            ...r3(2,1),
            ...phi3,
            ...L2,
            ...phi2,
            ...L4,
            ...phi4,
            ...L5,
            ...phi5,
            ...L6,
            ...phi6,
            ...phi2_deviation,
            ...phi4_deviation,
            ...phi5_deviation,
            ...phi6_deviation,
            ...F2x,
            ...F2y,
            ...F4x,
            ...F4y,
            ...F5x,
            ...F5y,
            ...F6x,
            ...F6y,
            ...F7x,
            ...F7y,
            ...F8x,
            ...F8y,
            ...F9x,
            ...F9y,
            ...F2,
            ...F4,
            ...F5,
            ...F6,
            ...F7,
            ...F8,
            ...F2_status,
            ...F4_status,
            ...F5_status];

% info columns
% 1 - adn
% 2 - r3x - global - mm
% 3 - r3y - global - mm
% 4 - phi3 - radians
% 5 - L2 - mm
% 6 - phi 2 - radians
% 7 - L4 - mm
% 8 - phi 4 - radians
% 9 - L5 - mm
% 10 - phi 5 - radians
% 11 - L6 - mm
% 12 - phi 6 - radians
% 13 - Phi2 Deviation
% 14 - Phi4 Deviation
% 15 - Phi5 Deviation
% 16 - Phi6 Deviation
% 17 - F2x
adn=adn+1;
end % end adn while loop

if 1==adb
    info1=info;
elseif 2==adb
    info2=info;
elseif 3==adb
    info3=info;
elseif 4==adb
    info4=info;
elseif 5==adb
    info5=info;
else
    'error assigning adb and picking r3_path'
end
adb=adb+1;
end % end adb while loop

%% put phi's in degrees
phi2_degrees=phi2/d2r;
phi3_degrees=phi3/d2r;
phi4_degrees=phi4/d2r;
phi5_degrees=phi5/d2r;
phi6_degrees=phi6/d2r;
phi7_degrees=phi7/d2r;
phi8_degrees=phi8/d2r;

%% test sum of forces and moments
Fx_sum = F2*cos(theta2)+F4*cos(theta4)+F5*cos(theta5)
+ F6*cos(theta6)+F7*cos(theta7)+F8*cos(theta8);
Fy_sum = F2*sin(theta2)+F4*sin(theta4)+F5*sin(theta5)+F6*sin(theta6)
+ F7*sin(theta7)+F8*sin(theta8)+ mass*g*sin(phi_incline);
Moment_sum = F6*sin(theta6)*s3pD(1,1)
- F6*cos(theta6)*s3pD(2,1)
+ F2*sin(theta2)*s3pA(1,1)
- F2*cos(theta2)*s3pA(2,1)
+ F4*sin(theta4)*s3pB(1,1)
- F4*cos(theta4)*s3pB(2,1)
+ F5*sin(theta5)*s3pC(1,1)
- F5*cos(theta5)*s3pC(2,1)
+ F7*sin(theta7)*s3pI(1,1)
- F7*cos(theta7)*s3pI(2,1)
+ F8*sin(theta8)*s3pJ(1,1)
- F8*cos(theta8)*s3pJ(2,1);

%% plot figure
figure('Name','Run 1 State College Equisol 20 Degrees - Position')

% plot y location relative to x location
subplot(2,2,1);
plot(info(:,2),info(:,3));
title('Y vs X Location')
ylabel('Y Position (mm)');
xlabel('X Position (mm)');
axis([-15,15,-15,15]);

% plot the lengths of the wires relative to x location
subplot(2,2,2);
plot(info(:,2),info(:,5),'*'); hold on
plot(info(:,2),info(:,7))
plot(info(:,2),info(:,9))
title('Control Wire Lengths vs X Location')
ylabel('Control Wire Lengths (mm)')
xlabel('X Position (mm)');
\begin{verbatim}
\% Legend for wire angles and deviation angles.
legend('L2', 'L4', 'L5')
axis([-15, 15, 0, 50]);

\% Plot the angles of the wires
subplot(2,2,3);
plot(info1(:,2),info1(:,6)/d2r,'*'); hold on
plot(info1(:,2),info1(:,8)/d2r)
plot(info1(:,2),info1(:,10)/d2r)
title('Control Wire Angles vs X Location')
ylabel('Control Wire Angles (degrees)')
xlabel('X Position (mm)')
legend('\Phi2', '\Phi4', '\Phi5')
axis([-15, 15, 0, 360]);

\% Plot the deviation angles of the wires
subplot(2,2,4);
plot(info1(:,2),info1(:,13),'*'); hold on
plot(info1(:,2),info1(:,14))
plot(info1(:,2),info1(:,15))
title('Control Wire Transmission Angles vs X Location')
ylabel('Control Wire Transmission Angles (degrees)')
xlabel('X Position (mm)')
legend('L2 T. Angle', 'L4 T. Angle', 'L5 T. Angle')
axis([-15, 15, -90, 90]);

\% Plot the wire forces
figure('Name','Run 1 State College Equisol 20 Degrees - Forces')

\% Plot wire forces location relative to x location
plot(info1(:,2),info1(:,29)); hold on
plot(info1(:,2),info1(:,30))
plot(info1(:,2),info1(:,31))
plot(info1(:,2),info1(:,32),'*')
plot(info1(:,2),info1(:,33))
plot(info1(:,2),info1(:,34))
title('Link Forces vs X Location')
ylabel('Link Forces (N)')
xlabel('X Position (mm)')
legend('F2', 'F4', 'F5', 'F6', 'F8')
axis([-15, 15, -15, 15]);

\% Plot wire forces location relative to x location in X Direction
plot(info1(:,2),info1(:,17)); hold on
plot(info1(:,2),info1(:,19))
plot(info1(:,2),info1(:,21))
plot(info1(:,2),info1(:,23),'*')
title('Wire Force X Direction vs X Location')
ylabel('Wire Forces in X Direction (N)')
legend('F2x', 'F4x', 'F5x', 'F6x')
axis([-15, 15, -15, 15]);

\% Plot wire forces location relative to x location in Y Direction
plot(info1(:,2),info1(:,18)); hold on
plot(info1(:,2),info1(:,20))
plot(info1(:,2),info1(:,22))
plot(info1(:,2),info1(:,24))
title('Wire Force Y Direction vs X Location')
ylabel('Wire Forces in Y Direction (N)')
legend('F2y', 'F4y', 'F5y', 'F6y')
axis([-15, 15, -15, 15]);

\% Plot the location of the middle sheet
figure('Name','Visual Geometry')

\% Plot the middle sheet and frame location
link1_outline=[r1E,r1F,r1G,r1H,r1E];
link2_outline=[r1H,r3A];
link3_outline=[r3B,r3F,r3G,r3H,r3E];
link4_outline=[r3G,r3C];
link5_outline=[r3B,r3E];
link6_outline=[r1D,r3D];
link7_outline=[r1I,r3I];
link8_outline=[r1J,r3J];

plot(link1_outline(1,:),link1_outline(2,:),'-k'); hold on
plot(link2_outline(1,:),link2_outline(2,:),'g')
plot(link3_outline(1,:),link3_outline(2,:),'-r')
plot(link4_outline(1,:),link4_outline(2,:),'-b')
plot(link5_outline(1,:),link5_outline(2,:),'-k')
plot(link6_outline(1,:),link6_outline(2,:),'-w')
plot(link7_outline(1,:),link7_outline(2,:),'-k')
plot(link8_outline(1,:),link8_outline(2,:),'-r')
title('Visual Geometry')
ylabel('Location (mm)')
xlabel('Location (mm)')
legend('Link 1', 'Link 2', 'Link 3', 'Link 4', 'Link 5', 'Link 6',....
\end{verbatim}
% File name - MOSAIC_Link_1_Info_rev_1.m

% Greg Bruolo
% MOSAIC Kinematics - Info for link 1
% Spring 2017

% Use vector and matrix notation from Dr. Sommers Notes 04 01.

% Blue print information for if the coordinate system was at the bottom
% left hand corner
s1pA=[0,575]';
s1pB=[425,0]';
s1pC=[0,425]';
s1pD=[180,165]';
s1pE=[1000,1000]';
s1pF=[1000,0]';
s1pG=[0,0]';
s1pH=[0,1000]';
s1pI=[1000,0]';
s1pJ=[425,1000]';

% Put the coordinate system in the middle of the link
original_to_new=[90,90]';

s1pA=s1pA-original_to_new;
s1pB=s1pB-original_to_new;
s1pC=s1pC-original_to_new;
s1pD=s1pD-original_to_new;
s1pE=s1pE-original_to_new;
s1pF=s1pF-original_to_new;
s1pG=s1pG-original_to_new;
s1pH=s1pH-original_to_new;
s1pI=s1pI-original_to_new;
s1pJ=s1pJ-original_to_new;

% End of - MOSAIC_Link_1_Info_rev_1.m
% File name - MOSAIC_Link_3_Info_rev_1.m

% Greg Brulo
% MOSAIC Kinematics - Info for link 3
% Spring 2017

% Use vector and matrix notation from Dr. Sommers Notes 04 01.

% Blue print information for if the coordinate system was at the bottom
% left hand corner
s3pA=[0,150]';
s3pB=[0,0]';
s3pC=[150,150]';
s3pD=[150,150]';
s3pE=[150,0]';
s3pF=[0,0]';
s3pG=[0,150]';
s3pH=[150,0]';
s3pI=[0,150]';
s3pJ=[0,150]';

% Put the coordinate system in the middle of the link
original_to_new=[75,75]';

s3pA=s3pA-original_to_new;
s3pB=s3pB-original_to_new;
s3pC=s3pC-original_to_new;
s3pD=s3pD-original_to_new;
s3pE=s3pE-original_to_new;
s3pF=s3pF-original_to_new;
s3pG=s3pG-original_to_new;
s3pH=s3pH-original_to_new;
s3pI=s3pI-original_to_new;
s3pJ=s3pJ-original_to_new;

% End of - MOSAIC_Link_3_Info_rev_1.m
Arjun Singh Chauhan

Abstract

This report focuses on analyzing the kinematics and dynamics of a 3 RRR Planar Parallel Manipulator using Haug’s Method.

2. Methodology

2.1. Topology and Mobility

The mechanism has eight links (ground referred to as link ‘1’) and nine revolute joints. It is a two-loop mechanism with a mobility of 3. The blue links 2, 3 and 4 are the active links, that is, they are the ones that are actuated. The red links 5, 6 and 7 are passive and the green link 8 is the end-effector.

2.2. Coordinates

The mechanism has seven links other than the ground and hence twenty one coordinates. The coordinates of the i<sup>th</sup> link are given as follows.

\[ \{q_i\} = \begin{bmatrix} \{r_i\} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \phi_i \end{bmatrix} \]

The origins of local coordinate frames attached to the links w.r.t the global coordinate frame are given by \{r_i\}. The coordinates of a point ‘P’ attached to link ‘i’ w.r.t the global coordinates are given by \{s_i\}<sup>i</sup>. The coordinates of ‘P’ attached to link ‘i’ w.r.t the local coordinates are given by \{s_i\}<sup>i</sup>. The following relation holds:

\[ \{r_i\}<sup>i</sup> = \{r_i\} + [A_i]\{s_i\}<sup>i</sup> \]

\[ [A_i] = \begin{bmatrix} C\phi_i & -S\phi_i \\ S\phi_i & C\phi_i \end{bmatrix} \]

Another important matrices that would be used are:

\[ [B_i] = \begin{bmatrix} -S\phi_i & -C\phi_i \\ C\phi_i & -S\phi_i \end{bmatrix} = [R][A_i][R] \]

\[ [R] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \]

\[ [R]^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \]

2.3. Constraint Vector

The mechanism has nine revolute joints and hence eighteen kinematic constraints. Also, three driver constrains need to be specified.
The driver constraints specify the orientation of the active links.

2.4. Jacobian

The Jacobian, \( \Phi_q \), may be derived easily using the following general formula for the partial derivative of a revolute joint constraint,

\[
\{ r_i \}^p_w = \begin{bmatrix} \mathbb{I}_2 \end{bmatrix} \{ B_j \} \{ S_j \}^p
\]

\[
\{ r_i \}^p_s = \{ 0_{2x3} \}
\]

The Jacobian turns out to be 21x21.

2.5. Position Solution

The position solution may be obtained by giving an initial guess and then employing Newton-Raphson using the constraint vector and the Jacobian.

\[
q_{n+1} = q_n - [\Phi_q]^{-1} \Phi
\]

2.6. Velocity and Acceleration

The velocity and acceleration solutions are given by:

\[
\{ \Phi_q \} \{ \dot{q} \} = \{ \nu \}
\]

\[
\{ \Phi_r \} \{ \ddot{q} \} = \{ \gamma \}
\]

\[
\{ \nu \} = -\{ \Phi_r \}
\]

\[
\{ \gamma \} = -\{ \Phi_q \} \{ \dot{q} \} - 2 \{ \Phi_r \} \{ \ddot{q} \} - \{ \Phi_{rr} \}
\]

The vectors \( \{ \nu \} \) and \( \{ \gamma \} \) may be obtained using the following relations for revolute joints:

\[
\{ \Phi_{rev} \} = \{ r_i \}^p - \{ r_i \}^p = \{ 0_{2x1} \}
\]

2.7. Dynamics

For the dynamics, we need to make sure that the local coordinate frames are centered at the center-of-mass of the links. We can then use the dynamic equation to get the constraint forces.

\[
[M] \{ \ddot{q} \} = \{ Q \}_F = \{ Q \}_F\text{APPLIED} + \{ Q \}_F\text{ CONSTRAINTS}
\]

\[
\{ Q \}_F\text{ CONSTRAINTS} = -[\Phi_q]^T \{ \lambda \}
\]

\[
[M] \{ \ddot{q} \} + \{ \Phi_q \}^T \{ \lambda \} = \{ Q \}_F\text{APPLIED}
\]

\[
\{ \{ \ddot{q} \} \} = \left[ \begin{bmatrix} M \{ \Phi_q \}^T \end{bmatrix}^{-1} \{ Q \}_F\text{APPLIED} \right]
\]

Once we know \( \lambda \), the constraint forces may be calculated.

2.8. Singularities

Singularities of a mechanism lie where the determinant of the Jacobian goes to zero and are associated with undesirable behaviors. The types of singularities encountered for 3 RRR planar parallel manipulator are discussed below. The passive links may become parallel to each other. Any force on the end-effector perpendicular to the passive links would cause undesirable deflections in links.
The passive links may become concurrent. Any torque applied on the end-effector about the point of concurrency would lead to deflections in the links.

An active link and attached passive link may become parallel or anti-parallel. The point on end-effector attached to the passive link may only move perpendicular to the orientation of the passive link.

Two singularities may occur together. As shown in figure 5, links 2 and 5 are anti-parallel and the passive links are concurrent in the same configuration.

4. Conclusion

Using Haug’s method, the kinematics and dynamics of the 3 RRR planar parallel manipulator are just as easy as those of the 4 bar. However, singularities are much more commonly encountered and have to be planned for.

References

ME 581: SIMULATION OF MECHANICAL SYSTEMS – JOINT FORCE REACTIONS ELLIPTICAL VERSUS TREADMILL

Danny C. Duenas

ABSTRACT

Viewing joint force reactions on the human body for different exercise equipment can help to mitigate long term side effects of poor exercise behavior. An elliptical trainer and a treadmill were attempted to be 2-D modeled to compare the joint forces to see which machine is best to use for low joint stress. The Elliptical trainer showed the highest forces in the hip, and increased forces at higher speeds. The treadmill model could not function properly due to issues with center of mass balancing and body collisions within Working Model. Future testing is needed to properly compare the machines.

2. BACKGROUND

2.1. Cardiac Exercise Methods

As a way to facilitate exercising at any time, and as a way to control exercise intensity, various equipment has been created to allow for stationary aerobic exercise. Two of these options, treadmills and Elliptical trainers, are highly popular conceptions used today. Additionally, treadmill speed and incline can be easily manipulated to control exercise tests in medical and research scenarios. However, repetitive use of exercise equipment can lead to long term damage to the joints of the legs. With the Elliptical trainer’s cyclical motion, it has the potential to ease joint damage from the repetitive action performed.

2.2. Method 1 – Treadmill

As mentioned earlier, the treadmill is a standard method of exercise that can be easily manipulated to different cardiac intensities. The basic design involves a belt that moves underneath the user to simulate forward movement. Therefore, the motion is similar to that of walking, jogging, or running on normal ground. Walking on a treadmill is a simple, easy motion that is not expected to put much pressure on the joints.

2.3. Method 2 – Elliptical

A potential way to prevent the joint damage that could be caused by improper use of a treadmill is to resort to another equipment machine that reduces the shock forces as higher speeds. One of these such devices is an Elliptical trainer. Elliptical trainers utilize a consistent circular motion, regardless of speed. The user’s feet are always expected to remain on the machine as it rotates, so no hard landing or shock force is produced. Unlike treadmills, the force used to keep the machine in motion is a more consistent downward motion, where as normal running has additional force backwards in order to help propel forwards. One of the potential setbacks of an Elliptical is that its main methods to control exercise output is resistance and stride length, such that the user can move at a variable speed, making observation between groups of individuals potentially challenging.
3. METHODS

3.1. Model Development

In order to attempt to simulate the human joint interaction on a treadmill and an Elliptical trainer, Working Model was chosen to create 2-Dimensional simulations. Working Model has the capability to produce a realistic physical environment, and can focus on specific locations of the model to map various statistics over time, such as position analysis and force analysis. The focus of this experiment is to perform force analysis on the ankle, knee, and hip joints during the motion of exercise. This analysis will be done at various speeds to observe the impact that increased speed has on the joint forces.

To create the Elliptical model, the back wheel concept was used. This was chosen in order to make modeling easier in Working Model, as the front-wheel Elliptical design includes a slider attached to the foot piece in order to prevent it from hitting the floor. The rear-wheel model utilizes a rotating pin where the user’s arms would be, which is an overall easier design to conceptualize. The placement of the wheel base will change the running angle on the Elliptical, but the impact of that detail was not considered within this project.

After models are created in working model, the data was exported to .dta files. Using these files, graphs were generated in MatLAB to better compare the data outputs.

3.2. Elliptical Trainer Model

As mentioned above, the Elliptical trainer modeled utilized a rear-wheel design. The Elliptical design consists of 2 connected four bars, where link 2 is replaced by a wheel, and each four-bar’s link 3 is located at opposite ends of the wheel. The upper body and upper half of the Elliptical trainer were ignored, as the focus of the project was on the leg joints. The human model consisted of models of the foot, calf, thigh, and waist areas. Weights were added to each section individually, with the waist being a full torso weight. The weights used were as follows: foot - .959 kg; calf – 2.842 kg; thigh – 6.749 kg; waist (‘torso’) – 33.312 kg [2]. The model was difficult to build using motors at the joints, so instead the motor was placed on the Elliptical wheel itself. The change of the driver location is expected to alter the results of the force output, but if this was not done the model would not work properly. Additionally, the feet where anchored onto the Elliptical trainer, although this is not expected to have much of an impact on the results, as the feet are generally always on the trainer during exercise anyway. Finally, the body had to be anchored into a stationary location above the Elliptical model. Without this anchor, the body model would fall and no data could be obtained. Due to natural bobbing of the body while walking, this anchor may cause a minor change in the results.

![Figure 2: Image of Elliptical Trainer Model from Working Model.](image)

3.3. Treadmill Trainer Model

Unfortunately, I was unable to design a functional model of treadmill motion on Working Model. The major design concerns I ran into were joint motor control, center of gravity of the body, and collision reactions between the body and the treadmill. These issues were the same as the ones seen on the Elliptical trainer, but the issues could not be mitigate through a workaround as they were on the trainer. Moving all of the joints in synchronization with a moving treadmill belt caused issues that stemmed from the difficulty keeping the center of gravity on the body. As a result, the body would consistently fall when running the model. Additionally, although the model was set to collide, the constraints would be lost at runtime and the body would fall in on itself. These were the reasons the
model ultimately failed to work properly, and thus the subsequent results were only those of the Elliptical Trainer model.

4. RESULTS

4.1. Elliptical Force Reactions at 30 rotations per minute.

For each graph above, as for all subsequent Elliptical trainer force graphs, the maximum magnitude of the force distribution occurs during the vertical pushing motion. Of the three graphs, the hip force has the highest force applied to it, with a maximum viewed force of around 175N. This unexpected result is believed to be due to the motor being positioned on the wheel instead of the joints, so the applied forces are upwards into the body instead of downward into the trainer. Additionally, all 3 graphs display negative forces applied in the y-direction (representing a downward force), along with a positive force for the knee and hip in the x-direction, and a negative force for the ankle in the x-direction. As expected, the downward force is dominant in the y-direction due to the weight of the body pushing upon the joints. The x-direction forces is less apparent, but is believed to be due to the tendency of the body structure to have a forward momentum while moving, and therefore have forward forces applied to the joints.

Another interesting observation is that the forces in the x-direction are comparable to the forces in the y-direction. Although forces are expected in that direction, it was expected for the y-direction force magnitude to be significantly greater. Viewing the model, this observation may be from the angles created
by the joints during the cycle. In figure 2, the back leg has the hip and knee joints at a 90 degree angle from the standing position, and the front leg has high angles for the knee and ankle joints.

Of the 3 graphs displayed at 30 rotations per minute, the ankle has the most notable variation in force over 1 graph cycle. There are multiple peaks and reversals during 1 cycle, with 2 of the peaks in the vertical direction being almost the same magnitude. Because the ankle is the closest joint to the rotational movement on the Elliptical trainer, this behavior is expected. Similarly, the hip joint has the smoothest motion over the cycle, which can be attributed to it being the furthest away from the Elliptical trainer’s motion, and has the least rotation it has to cover.

![Figure 6: Force Distribution of the Ankle over 2 Rotations at 60 Rotations per Minute.]

An increase in speed has the same pattern of forces as the slower speed, but there are some notable differences. Doubling the speed had significant force yield increases, with a 10 fold increase from ~25N to ~225N in the ankle, around a 130N increase in the Knee, and almost a 300N increase in the Hip. Note that both the negative and positive forces increased in both the x and y directions. There were also notable peak changes. In the ankle, the first of the 2 similar magnitude peaks present in the y-direction is now significantly lower than the second peak, which is the peak that comes from the downward vertical push. Contrastingly, a new peak appeared in the x-direction on the knee force graph, which has a similar magnitude to the initial peak of the slower speed.

![Figure 7: Force Distribution of the Knee over 2 Rotations at 60 Rotations per Minute.]

![Figure 8: Force Distribution of the Hip over 2 Rotations at 60 Rotations per Minute.]

The results at 120 rotations per minute follow a similar pattern to the changes seen at 60 rotations per minute, in that the force magnitudes increase at a large rate (3 to 5 times the force at 60 rotations per minute), but there are no notable peak changes between the 60 and 120 rotations per minute graphs.

5. CONCLUSION

Between the different speeds, the highest speed produces the most forces, as expected. However, the hip is displayed as the highest force output, reaching 1500N at 120 rotations per minute, which was not anticipated. This could be due to the force being applied from the Elliptical trainer instead of the joints, and from the body torso being stationary throughout the cycle. Unfortunately, there is no data for comparison from a treadmill model. The expectation is that the joints would have undergone additional force than the Elliptical trainer due to the shock force that occurs upon the foot hitting the treadmill belt. This force could further be exasperated by landing on the heel instead of the ball of the foot, which supplies a spring motion to alleviate some of the additional force.

In future testing, a treadmill model is necessary to view comparative forces. Additional modifications would be to allow a proper center balance on the Elliptical trainer body such that the torso does not have to be anchored, and producing the forces at the joints so the force output is more realistic. Additional testing could be viewing different stride lengths and inclines, along with viewing the difference in landing on a treadmill on the heel versus the ball of the foot.
References


MATLAB CODE

clear all;
close all;

%Gathering Data from Working Model
ForceMatrix = importdata('JointForces30rotmin.dta');
ForceMatrix2 = importdata('JointForces60rotmin.dta');
ForceMatrix3 = importdata('JointForces120rotmin.dta');
t30=ForceMatrix(:,1);
Ankx30 = ForceMatrix(:,2);
Anky30 = ForceMatrix(:,3);
Kneex30= ForceMatrix(:,6);
Kneey30 = ForceMatrix(:,7);
Hipx30 = ForceMatrix(:,10);
Hipy30 = ForceMatrix(:,11);

t60=ForceMatrix2(:,1);
Ankx60 = ForceMatrix2(:,2);
Anky60 = ForceMatrix2(:,3);
Kneex60= ForceMatrix2(:,6);
Kneey60 = ForceMatrix2(:,7);
Hipx60 = ForceMatrix2(:,10);
Hipy60 = ForceMatrix2(:,11);

t120=ForceMatrix3(:,1);
Ankx120 = ForceMatrix3(:,2);
Anky120 = ForceMatrix3(:,3);
Kneex120= ForceMatrix3(:,6);
Kneey120 = ForceMatrix3(:,7);
Hipx120 = ForceMatrix3(:,10);
Hipy120 = ForceMatrix3(:,11);

%Plotting Data
figure(1);
plot(t30,Ankx30,t30,Anky30)
title('X and Y Ankle Forces Over Time, 30 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(2);
plot(t30,Kneex30,'r',t30,Kneey30,'b')
title('X and Y Knee Forces Over Time, 30 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(3);
plot(t30,Hipx30,'r',t30,Hipy30,'b')
title('X and Y Hip Forces Over Time, 30 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(4);
plot(t60,Ankx60,t60,Anky60)
title('X and Y Ankle Forces Over Time, 60 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(5);
plot(t60,Kneex60,'r',t60,Kneey60,'b')
title('X and Y Knee Forces Over Time, 60 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(6);
plot(t60,Hipx60,'r',t60,Hipy60,'b')
title('X and Y Hip Forces Over Time, 60 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(7);
plot(t120,Ankx120,t120,Anky120)
title('X and Y Ankle Forces Over Time, 120 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(8);
plot(t120,Kneex120,'r',t120,Kneey120,'b')
title('X and Y Knee Forces Over Time, 120 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
figure(9);
plot(t120,Hipx120,'r',t120,Hipy120,'b')
title('X and Y Hip Forces Over Time, 120 rotations/minute');
xlabel('Time [s]');
ylabel('Force [N]');
legend('Fx','Fy');
Automotive suspension systems are critical components of automobiles and play a significant role in the performance of the car and comfort of the occupants. Modern suspension systems consist of often complex geometry and 3D kinematic chains. To design suspensions that achieve their performance goals engineers must rely on extensive modeling to understand the dynamic relationships between the road, the vehicle tire, and the vehicle body.

Two common types of front suspension in cars are double wishbone suspensions and MacPherson Strut suspensions. They both hold particular advantages and disadvantages based on their application. The MacPherson Strut’s overriding advantage is its simplicity, requiring relatively few components, and its small packaging space requirements. These factors make it an inexpensive suspension that can easily be implemented in a broad range of vehicles. The MacPherson Strut does however have some drawbacks in performance inherent in the design. Due to the simplicity that benefits it in cost and packaging space, its kinematics are limited.

I used the Matlab scripts developed in C05 for kinematics of a MacPherson Strut and modified it to fit a general MacPherson Strut (figure 1) with geometry standardized for applying to other suspensions as well. The constraint vector and jacobian matrix are shown below (\( \Phi_{MacPherson} \) and \( JAC_{MacPherson} \), respectively).

*Figure 1: MacPherson Strut Diagram*
\[
\Phi_{\text{MacPherson}} = \begin{bmatrix}
\{r_1\}^A - \{r_2\}^A \\
\{\bar{f}_2\}^T \{\bar{h}_1\} \\
\{\bar{g}_2\}^T \{\bar{h}_1\} \\
\{r_2\}^B - \{r_3\}^B \\
\{d_{EF}\}^T (d_{EF}) - L^2, \\
\{d_{EF}\} = \{r_1\}^F - \{r_3\}^F \\
\{\bar{f}_3\}^T \{\bar{h}_4\} \\
\{\bar{h}_3\}^T \{\bar{g}_4\} \\
\{r_1\}^P - \{r_4\}^P \\
\{p_2\}^T \{p_2\} - 1 \\
\{p_3\}^T \{p_3\} - 1 \\
\{p_4\}^T \{p_4\} - 1 \\
y_3 - y_{3\text{START}} - \Delta y_3 t
\end{bmatrix}
\]
Next, I modified the script for the MacPherson Strut to fit a simplified, general double A-arm suspension (figure 2). The double A-arm suspension geometry was similarly generalized to be proportional to the MacPherson Strut. The constraint vector and jacobian matrix are shown below ($\Phi_{\text{Double A-Arm}}$ and $\{JAC_{\text{Double A-Arm}}\}$, respectively).

Another suspension model was made by modifying the double A-arm by shortening the upper control arm making a short-long arm, or SLA suspension (figure 3). One last suspension model was formed by skewing the upper control arm on the SLA model, angling the upper arm up (figure 4). Changing geometry like this can result in significantly different kinematics so these types of changes are often used for achieving certain performance goals in automotive suspensions.

*Figure 2: Double A-Arm (DAA) Diagram, Parallel Equal Length Arms*
\[
\{ \Phi_{\text{Double } A-Arm} \} = \begin{pmatrix}
\{ r_1 \}_A - \{ r_2 \}_A \\
\{ f_2 \}^T \{ \hat{h}_1 \} \\
\{ \bar{g}_2 \}^T \{ \hat{h}_1 \} \\
\{ r_2 \}_B - \{ r_3 \}_B \\
\{ d_{EF} \}^T \{ d_{EF} \} - I^2, \quad \{ d_{EF} \} = \{ r_1 \}_F - \{ r_3 \}_E \\
\{ r_3 \}_C - \{ r_4 \}_C \\
\{ r_1 \}_D - \{ r_4 \}_D \\
\{ f_4 \}^T \{ \hat{h}_1 \} \\
\{ \bar{g}_4 \}^T \{ \hat{h}_1 \} \\
\{ p_2 \}_T \{ p_2 \} - 1 \\
\{ p_3 \}_T \{ p_3 \} - 1 \\
\{ p_4 \}_T \{ p_4 \} - 1 \\
y_3 - y_{3\text{START}} - Ay_3 t
\end{pmatrix}
\]

\[
\{ JAC_{\text{Double } A-Arm} \} = \\
\begin{bmatrix}
-\{ I_3 \} & 2\{ A_2 \} [\bar{x}_2]^A [G_3] \\
0_{1 \times 3} & -2\{ h_1 \}_T \{ A_1 \}_T \{ A_2 \} [\bar{f}_3] \{ G_3 \} \\
0_{1 \times 3} & -2\{ h_1 \}_T \{ A_1 \}_T \{ A_2 \} [\bar{g}_2] \{ G_3 \} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{3 \times 3} & 0_{3 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 2\{ p_2 \}_T \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
0_{1 \times 3} & 0_{1 \times 4} \\
[0 \quad 1 \quad 0] & 0_{1 \times 4} \\
0_{1 \times 4} & 0_{1 \times 4} \\
0_{1 \times 4} & 0_{1 \times 4} \\
0_{1 \times 4} & 0_{1 \times 4} \\
0_{1 \times 4} & 0_{1 \times 4} \\
0_{1 \times 4} & 0_{1 \times 4} \\
0_{1 \times 4} & 0_{1 \times 4} \\
0_{1 \times 4} & 0_{1 \times 4}
\end{bmatrix}
\]
Figure 3: Short-Long Arm Suspension Diagram (DAA with parallel, unequal length arms)

Figure 4: SLA-Skewed Suspension Diagram (unequal length, non-parallel arms)
Appendix A: Matlab Script for MacPherson Strut

% McP_main.m - MacPherson Strut 3D
% main

% initialize
clear all
clc
McP_ini;
McP_kin;
% one crank revolution
tpr = 3*pi/2;
% time for one cycle

% timer loop
keep = [];
for t = pi/2 : pi/12 : tpr,
% kinematics
McP_kin;
% save kinematics
x = r3(1)-10;
y = r3(2)-3;
toe = asin(h3(1));
cambera = asin(f3(2));
camberx = f3(1);
cambery = f3(2);

keep = [keep; t x y chi2 toe cambera camberx cambery];
% bottom of timer loop
end

time = keep(:,1);
x = keep(:,2);
y = keep(:,3);
chi2 = keep(:,4)/d2r;
toe = keep(:,5)/d2r;
cambera = keep(:,6)/d2r;
camberx = keep(:,7);
cambery = keep(:,8);

figure(1)
hold on
grid on
plot(y, toe, 'r')
xlabel('Relative Height Above Resting Position [in]')
ylabel('Toe Angle [deg] - Positive OUT, Negative IN')
title('Toe Angle vs Suspension Height')

% axis equal

figure(2)
subplot(1,4,1)
hold on
grid on
scale = 0;
quiver(x, y, camberx, cambery, scale, 'r')
plot(x, y, 'ro')
%xlabel('x-Position of Knuckle')
ylabel('y-Position of Knuckle')
title({'MacPherson Strut'})
axis([-1 1 -2.5 2.5])
% axis equal

figure(3)
hold on
grid on
plot(y, cambera, 'r')
xlabel('Relative Height Above Resting Position [in]')
ylabel('Camber Angle [deg] - Positive UP, Negative DOWN')
title('Camber Angle Angle vs Suspension Height')
%axis equal

% McP_ini.m - McPherson Strut
% initialize constants and assembly guesses

% general constants
d2r = pi / 180;
R = [ 0 -1; 1 0 ];

f = [1; 0; 0];
g = [0; 1; 0];
h = [0; 0; 1];

% mechanism constants inches
s1pA = [0; 0; 0];
s1pD = [8.00; 12.00; 0];
s1pF = [2.00; 2.00; 2.00];

s2pA = [ 0 0 0]';
s2pB = [ 10.00 0 0]';

s3pB = [ 0 -3.00 0]';
s3pC = [ -1.00 3.00 0]';
s3pE = [ 0 -1.00 2.00]';

s4pC = [ 0 0 0]';
s4pD = [ 0 3.00 0 ]';

s2pAskew = [0 -s2pA(3) s2pA(2); s2pA(3) 0 -s2pA(1); -s2pA(2) s2pA(1) 0];
s2pBskew = [0 -s2pB(3) s2pB(2); s2pB(3) 0 -s2pB(1); -s2pB(2) s2pB(1) 0];

s3pBskew = [0 -s3pB(3) s3pB(2); s3pB(3) 0 -s3pB(1); -s3pB(2) s3pB(1) 0];
s3pCskew = [0 -s3pC(3) s3pC(2); s3pC(3) 0 -s3pC(1); -s3pC(2) s3pC(1) 0];
s3pEskew = [0 -s3pE(3) s3pE(2); s3pE(3) 0 -s3pE(1); -s3pE(2) s3pE(1) 0];

s4pCskew = [0 -s4pC(3) s4pC(2); s4pC(3) 0 -s4pC(1); -s4pC(2) s4pC(1) 0];
s4pDskew = [0 -s4pD(3) s4pD(2); s4pD(3) 0 -s4pD(1); -s4pD(2) s4pD(1) 0];

EF = 8.00;
% initial guesses - angles measured by protractor
chi2 = 0*d2r;
chi3 = 10*d2r;
chi4 = chi3;

% r2
q(1,1) = 0;
q(2,1) = 0;
q(3,1) = 0;

% p2
q(4,1) = cos(chi2/2);
q(5,1) = 0;
q(6,1) = 0;
q(7,1) = sin(chi2/2);

% r3
q(8,1) = 10.00; % AB*cos(chi2)-BG*sin(chi3);
q(9,1) = 3.00;  % AB*sin(chi2)+BG*cos(chi3);
q(10,1) = 0;

% p3
q(11,1) = cos(chi3/2);
q(12,1) = 0;
q(13,1) = 0;
q(14,1) = sin(chi3/2);

% r4
q(15,1) = 8.50; % q(8,1)-GC4*sin(chi4);
q(16,1) = 9.00; % q(9,1)+GC4*cos(chi4);
q(17,1) = 0;

% p4
q(18,1) = cos(chi4/2);
q(19,1) = 0;
q(20,1) = 0;
q(21,1) = sin(chi4/2);

% driver for crank - r4C-r3C = Strut_Start + jounce_dot*t
y3_start = 3.17; % Strut Jounce Start
strut_start = y3_start;
y3_dot = 4;
t = 0;

% McPherson_phi.m - McPherson Strut
% evaluate constraints and Jacobian for linear distance driving constraint
% global location of local frames and rotation matrices
% Eq 2.4.4, page 33 - Eq 2.6.5, page 42
r1 = [0 0 0]';
p1 = [1 0 0 0]';
r2 = q(1:3);
p2 = q(4:7);
r3 = q(8:10);
p3 = q(11:14);

r4 = q(15:17);
p4 = q(18:21);

% phi2 = chi2;
% phi3 = chi3;
% phi4 = chi4;

A1 = [ 1 0 0; 0 1 0; 0 0 1];
A2 = 2*[ p2(1)^2+p2(2)^2-1/2 p2(2)*p2(3)-p2(1)*p2(4) p2(2)*p2(4)+p2(1)*p2(3); p2(2)*p2(3)+p2(1)*p2(4) p2(1)^2+p2(3)^2-1/2 p2(3)*p2(4)-p2(1)*p2(2); p2(2)*p2(4)-p2(1)*p2(3) p2(3)*p2(4)+p2(1)*p2(2) p2(1)^2+p2(4)^2-1/2];
A3 = 2*[ p3(1)^2+p3(2)^2-1/2 p3(2)*p3(3)-p3(1)*p3(4) p3(2)*p3(4)+p3(1)*p3(3); p3(2)*p3(3)+p3(1)*p3(4) p3(1)^2+p3(3)^2-1/2 p3(3)*p3(4)-p3(1)*p3(2); p3(2)*p3(4)-p3(1)*p3(3) p3(3)*p3(4)+p3(1)*p3(2) p3(1)^2+p3(4)^2-1/2];

f1 = f;
g1 = g;
h1 = h;
f2 = A2*f;
g2 = A2*g;
h2 = A2*h;
f3 = A3*f;
g3 = A3*g;
h3 = A3*h;
f4 = A4*f;
g4 = A4*g;
h4 = A4*h;

fpskew = [0 -f(3) f(2); f(3) 0 -f(1); -f(2) f(1) 0];
gpskew = [0 -g(3) g(2); g(3) 0 -g(1); -g(2) g(1) 0];
hpskew = [0 -h(3) h(2); h(3) 0 -h(1); -h(2) h(1) 0];

G2 = [-p2(2) p2(1) p2(4) -p2(3); -p2(3) -p2(4) p2(1) p2(2); -p2(4) p2(3) -p2(2) p2(1)];
G3 = [-p3(2) p3(1) p3(4) -p3(3); -p3(3) -p3(4) p3(1) p3(2); -p3(4) p3(3) -p3(2) p3(1)];
G4 = [-p4(2) p4(1) p4(4) -p4(3); -p4(3) -p4(4) p4(1) p4(2); -p4(4) p4(3) -p4(2) p4(1)];

y3 = r3(2);

% global locations - Eq 2.4.8, page 33
r1A = r1 + A1*s1pA;
r1D = r1 + A1*s1pD;
r1F = r1 + A1*s1pF;

r2A = r2 + A2*s2pA;
r2B = r2 + A2*s2pB;

r3B = r3 + A3*s3pB;
r3C = r3 + A3*s3pC;
r3E = r3 + A3*s3pE;
\[ r_{4C} = r_4 + A_4s_{4pC}; \]
\[ r_{4D} = r_4 + A_4s_{4pD}; \]
\[ d_{CC} = r_{4C} - r_{3C}; \]
\[ d_{EF} = r_{1F} - r_{3E}; \]

\[ \Phi = \text{zeros}(21,1); \]

\% Joint A - Revolute
\[ \Phi(1:3) = r_{1A} - r_{2A}; \]
\[ \Phi(4) = f_{2}^{'}h_{1}; \]
\[ \Phi(5) = g_{2}^{'}h_{1}; \]

\% Joint B - Spherical
\[ \Phi(6:8) = r_{2B} - r_{3B}; \]

\% Joint E - Double Spherical (Steering Tie Rod)
\[ \Phi(9) = d_{EF}^{'}d_{EF} - EF*EF; \]

\% Joint C - Prismatic
\[ \Phi(10) = f_{3}^{'}g_{4}; \]
\[ \Phi(11) = h_{3}^{'}g_{4}; \]
\[ \Phi(12) = f_{3}^{'}d_{CC}; \]
\[ \Phi(13) = h_{3}^{'}d_{CC}; \]
\[ \Phi(14) = f_{3}^{'}h_{4}; \]

\% Joint D - Spherical
\[ \Phi(15:17) = r_{1D} - r_{4D}; \]

\% Euler Parameters
\[ \Phi(18) = p_{2}^{'}p_{2}-1; \]
\[ \Phi(19) = p_{3}^{'}p_{3}-1; \]
\[ \Phi(20) = p_{4}^{'}p_{4}-1; \]

\% vertical translation driving constraint
\[ \Phi(21) = y_{3} - y_{3_{\text{start}}} - 2\sin(t); \]

\% Jacobian by rows - Eq 3.3.12, page 66 for revolutes
\[ JAC = \text{zeros}(21,21); \]

\% Joint A - Revolute
\[ JAC(1:3,1:3) = -\text{eye}(3); \]
\[ JAC(1:3,4:7) = 2A_2s_{2pAskew}G_2; \]
\[ JAC(4,4:7) = -2h^{'}A_{1}^{'}A_{2}^{'}fpskewG_{2}; \]
\[ JAC(5,4:7) = -2h^{'}A_{1}^{'}A_{2}^{'}gpskewG_{2}; \]

\% Joint B - Spherical
\[ JAC(6:8,1:3) = \text{eye}(3); \]
\[ JAC(6:8,4:7) = -2A_2s_{2pBskew}G_{2}; \]
\[ JAC(6:8,8:10) = -\text{eye}(3); \]
\[ JAC(6:8,11:14) = 2A_3s_{3pBskew}G_{3}; \]

\% Joint E - Double Revolute
\[ JAC(9,8:10) = -2d_{EF}^{'}; \]
\[ JAC(9,11:14) = 4d_{EF}^{'}A_{3}^{'}s_{3pEskew}G_{3}; \]

\% Joint C - Prismatic
\[ JAC(10,11:14) = -2g^{'}A_{4}^{'}A_{3}^{'}fpskewG_{3}; \]
\[ JAC(10,18:21) = -2f^{'}A_{3}^{'}A_{4}^{'}gpskewG_{4}; \]
\[ JAC(11,11:14) = -2g^{'}A_{4}^{'}A_{3}^{'}hpskewG_{3}; \]
\[ JAC(11,18:21) = -2h^{'}A_{3}^{'}A_{4}^{'}gpskewG_{4}; \]
\[
\begin{align*}
\text{JAC}(12,8:10) &= -f'*A3'; \\
\text{dCC}'*A3*fpskew)*G3; \\
\text{JAC}(12,15:17) &= f'*A3'; \\
\text{JAC}(13,8:10) &= -h'*A3'; \\
\text{dCC}'*A3*hpskew)*G3; \\
\text{JAC}(13,15:17) &= h'*A3'; \\
\text{JAC}(14,11:14) &= -2*h'*A4'*A3*fpskew*G3; \\
\text{JAC}(13,18:21) &= -2*h'*A3'*A4*s4pCskew*G4; \\
\text{JAC}(14,18:21) &= -2*f'*A3'*A4*hpskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(15:17,15:17) &= \text{eye}(3); \\
\text{JAC}(15:17,18:21) &= 2*A4*s4pDskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(18,4:7) &= 2*p2'; \\
\text{JAC}(19,11:14) &= 2*p3'; \\
\text{JAC}(20,18:21) &= 2*p4';
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(21,8:10) &= [0 1 0];
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(12,11:14) &= 2*(f'*s3pCskew- \\
\text{JAC}(12,18:21) &= -2*f'*A3'*A4*s4pCskew*G4; \\
\text{JAC}(13,11:14) &= 2*(h'*s3pCskew- \\
\text{JAC}(13,18:21) &= -2*h'*A3'*A4*s4pCskew*G4; \\
\text{JAC}(14,18:21) &= -2*f'*A3'*A4*hpskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(14,11:14) &= -2*h'*A4'*A3*fpskew*G3;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(13,11:14) &= 2*(h'*s3pCskew- \\
\text{JAC}(13,18:21) &= -2*h'*A3'*A4*s4pCskew*G4; \\
\text{JAC}(14,18:21) &= -2*f'*A3'*A4*hpskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(12,18:21) &= -2*f'*A3'*A4*s4pCskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(13,18:21) &= -2*h'*A3'*A4*s4pCskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(14,18:21) &= -2*f'*A3'*A4*hpskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(15,17,18:21) &= 2*A4*s4pDskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(15:17,15:17) &= \text{eye}(3);
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(15:17,18:21) &= 2*A4*s4pDskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(18,4:7) &= 2*p2'; \\
\text{JAC}(19,11:14) &= 2*p3'; \\
\text{JAC}(20,18:21) &= 2*p4';
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(21,8:10) &= [0 1 0];
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(12,11:14) &= 2*(f'*s3pCskew- \\
\text{JAC}(12,18:21) &= -2*f'*A3'*A4*s4pCskew*G4; \\
\text{JAC}(13,11:14) &= 2*(h'*s3pCskew- \\
\text{JAC}(13,18:21) &= -2*h'*A3'*A4*s4pCskew*G4; \\
\text{JAC}(14,18:21) &= -2*f'*A3'*A4*hpskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(14,11:14) &= -2*h'*A4'*A3*fpskew*G3;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(13,11:14) &= 2*(h'*s3pCskew- \\
\text{JAC}(13,18:21) &= -2*h'*A3'*A4*s4pCskew*G4; \\
\text{JAC}(14,18:21) &= -2*f'*A3'*A4*hpskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(12,18:21) &= -2*f'*A3'*A4*s4pCskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(13,18:21) &= -2*h'*A3'*A4*s4pCskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(14,18:21) &= -2*f'*A3'*A4*hpskew*G4;
\end{align*}
\]

\[
\begin{align*}
\text{JAC}(14,11:14) &= -2*h'*A4'*A3*fpskew*G3;
\end{align*}
\]
% DAA_main.m - Double A-Arm Suspension Main

% initialize
clear all
clc
DAA_ini;
DAA_kin;

% time for one cycle

% timer loop
keep = [];
for t = pi/2 : pi/12 : tpr,

% kinematics
DAA_kin;

% save kinematics
x = r3(1)-10;
y = r3(2)-3;
toe = asin(h3(1));
cambera = asin(f3(2));
camberx = f3(1);
cambery = f3(2);

keep = [keep; t x y chi2 toe cambera camberx cambery];

% bottom of timer loop
end

time = keep(:,1);
x = keep(:,2);
y = keep(:,3);
chi2 = keep(:,4)/d2r;
toe = keep(:,5)/d2r;
cambera = keep(:,6)/d2r;
camberx = keep(:,7);
cambery = keep(:,8);

figure(1)
hold on
plot(y, toe, 'b')
xlabel('Relative Height Above Resting Position [in]')
ylabel('Toe Angle [deg] - Positive OUT, Negative IN')
title( 'Toe Angle vs Suspension Height')
axis equal

figure(2)
subplot(1,4,2)
hold on
grid on
scale = 0;
quiver(x, y, camberx, cambery, scale, 'b')
hold on
plot(x, y, 'bo')
xlabel('x-Position of Knuckle')
ylabel('y-Position of Knuckle')
title({}'Knuckle Position & Spindle Axis Direction'; 'Double A-Arm')
axis([-1 1 -2.5 2.5])
hold off
axis equal

figure(3)
hold on
grid on
plot(y, cambera, 'b')
xlabel('Relative Height Above Resting Position [in]')
ylabel('Camber Angle [deg] - Positive UP, Negative DOWN')
title('Camber Angle Angle vs Suspension Height')
axis equal

% DAA_ini.m - Double A-Arm Suspension
% initialize constants and assembly guesses

% general constants
d2r = pi / 180;

f = [1; 0; 0];
g = [0; 1; 0];
h = [0; 0; 1];

% mechanism constants

% Parallel, Equal Length Upper & Lower Arms
s1pA = [0; 0; 0];
s1pD = [0; 6.00; 0];
s1pF = [2.00; 2.00; 2.00];
s2pA = [0 0 0]';
s2pB = [10.00 0 0]';
s3pB = [0 -3.00 0]';
s3pC = [0 3.00 0]';
s3pE = [0 -1.00 2.00]';
s4pC = [0 0 0]';
s4pD = [10.00 0 0]';

---------------------------------------------

s2pAskew = [0 -s2pA(3) s2pA(2); s2pA(3) 0 -s2pA(1); -s2pA(2) s2pA(1) 0];
s2pBskew = [0 -s2pB(3) s2pB(2); s2pB(3) 0 -s2pB(1); -s2pB(2) s2pB(1) 0];
s3pBskew = [0 -s3pB(3) s3pB(2); s3pB(3) 0 -s3pB(1); -s3pB(2) s3pB(1) 0];
s3pCskew = [0 -s3pC(3) s3pC(2); s3pC(3) 0 -s3pC(1); -s3pC(2) s3pC(1) 0];
s3pEskew = [0 -s3pE(3) s3pE(2); s3pE(3) 0 -s3pE(1); -s3pE(2) s3pE(1) 0];
s4pCskew = [0
- s4pC(3) s4pC(2);
s4pC(3) 0
- s4pC(1); - s4pC(2) s4pC(1) 0];
s4pDskew = [0
- s4pD(3) s4pD(2);
s4pD(3) 0
- s4pD(1); - s4pD(2) s4pD(1) 0];

EF = 8.00;

% initial guesses - angles measured by protractor
chi2 = 0*d2r;
chi3 = 0*d2r;
chi4 = 0*d2r;

% r2
q(1,1) = 0;
q(2,1) = 0;
q(3,1) = 0;

% p2
q(4,1) = cos(chi2/2);
q(5,1) = 0;
q(6,1) = 0;
q(7,1) = sin(chi2/2);

% r3
q(8,1) = 10.00;  % AB*cos(chi2)-BG*sin(chi3);
q(9,1) = 3.00;   % AB*sin(chi2)+BG*cos(chi3);
q(10,1) = 0;

% p3
q(11,1) = cos(chi3/2);
q(12,1) = 0;
q(13,1) = 0;
q(14,1) = sin(chi3/2);

% r4
q(15,1) = 10.00;  % q(8,1)-GC4*sin(chi4);
q(16,1) = 6.00;    % q(9,1)+GC4*cos(chi4);
q(17,1) = 0;

% p4
q(18,1) = cos(chi4/2);
q(19,1) = 0;
q(20,1) = 0;
q(21,1) = sin(chi4/2);

% driver for crank - r4C-r3C = Strut_Start + jounce_dot*t
y3_start = 3;    % Strut Jounce Start

% DAA_phi.m - Double A-Arm Suspension
% evaluate constraints and Jacobian for linear distance driving constraint

% global location of local frames and rotation matrices
% Eq 2.4.4, page 33 - Eq 2.6.5, page 42
r1 = [0 0 0]';
\[ p1 = [1 0 0 0]'; \]
\[ r2 = q(1:3); \]
\[ p2 = q(4:7); \]
\[ r3 = q(8:10); \]
\[ p3 = q(11:14); \]
\[ r4 = q(15:17); \]
\[ p4 = q(18:21); \]

\[ A1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \]
\[ A2 = 2\times^{[p2(1)^2+p2(2)^2-1/2 \ p2(2)\ p2(3)^2 p2(2)^2 p2(3)^2 p2(2)^2 p2(3)^2]}^{p2(2)\ p2(3) p2(1)\ p2(3) p2(1)\ p2(3) p2(1)\ p2(3) p2(1)\ p2(3) p2(1)\ p2(3) p2(1)\ p2(3)} \]
\[ A3 = 2\times^{[p3(1)^2+p3(2)^2-1/2 \ p3(2)\ p3(3)^2 p3(2)^2 p3(3)^2 p3(2)^2 p3(3)^2]}^{p3(2)\ p3(3) p3(1)\ p3(3) p3(1)\ p3(3) p3(1)\ p3(3) p3(1)\ p3(3) p3(1)\ p3(3) p3(1)\ p3(3)} \]
\[ A4 = 2\times^{[p4(1)^2+p4(2)^2-1/2 \ p4(2)\ p4(3)^2 p4(2)^2 p4(3)^2 p4(2)^2 p4(3)^2]}^{p4(2)\ p4(3) p4(1)\ p4(3) p4(1)\ p4(3) p4(1)\ p4(3) p4(1)\ p4(3) p4(1)\ p4(3) p4(1)\ p4(3)} \]

\[ f1 = f; \]
\[ g1 = g; \]
\[ h1 = h; \]
\[ f2 = A2*f; \]
\[ g2 = A2*g; \]
\[ h2 = A2*h; \]
\[ f3 = A3*f; \]
\[ g3 = A3*g; \]
\[ h3 = A3*h; \]
\[ f4 = A4*f; \]
\[ g4 = A4*g; \]
\[ h4 = A4*h; \]

\[ \text{fpskew} = [0\ -f(3)\ f(2);\ f(3)\ 0\ -f(1);\ -f(2)\ f(1)\ 0] ; \]
\[ \text{gpskew} = [0\ -g(3)\ g(2);\ g(3)\ 0\ -g(1);\ -g(2)\ g(1)\ 0] ; \]
\[ \text{hpskew} = [0\ -h(3)\ h(2);\ h(3)\ 0\ -h(1);\ -h(2)\ h(1)\ 0] ; \]

\[ G2 = \begin{bmatrix} -p2(2) & p2(1) & p2(4) & -p2(3) \\ -p2(3) & -p2(4) & p2(1) & p2(2) \\ -p2(4) & p2(3) & -p2(2) & p2(1) \end{bmatrix} ; \]
\[ G3 = \begin{bmatrix} -p3(2) & p3(1) & p3(4) & -p3(3) \\ -p3(3) & -p3(4) & p3(1) & p3(2) \\ -p3(4) & p3(3) & -p3(2) & p3(1) \end{bmatrix} ; \]
\[ G4 = \begin{bmatrix} -p4(2) & p4(1) & p4(4) & -p4(3) \\ -p4(3) & -p4(4) & p4(1) & p4(2) \\ -p4(4) & p4(3) & -p4(2) & p4(1) \end{bmatrix} ; \]

\[ y3 = r3(2); \]

\% global locations - Eq 2.4.8, page 33
\[ r1A = r1 + A1*s1pA; \]
\[ r1D = r1 + A1*s1pD; \]
\[ r1F = r1 + A1*s1pF; \]

\[ r2A = r2 + A2*s2pA; \]
\[ r2B = r2 + A2*s2pB; \]
r3B = r3 + A3*s3pB;
r3C = r3 + A3*s3pC;
r3E = r3 + A3*s3pE;

r4C = r4 + A4*s4pC;
r4D = r4 + A4*s4pD;

dCC = r4C-r3C;
dEF = r1F-r3E;

PHI = zeros(21,1);

% Joint A - Revolute
PHI(1:3) = r1A - r2A;
PHI(4) = f2'*h1;
PHI(5) = g2'*h1;

% Joint B - Spherical
PHI(6:8) = r2B - r3B;

% Joint E - Double Spherical (Steering Tie Rod)
PHI(9) = dEF'*dEF-EF*EF;

% Joint C - Spherical
PHI(10:12) = r3C - r4C;

% Joint D - Revolute
PHI(13:15) = r1D - r4D;
PHI(16) = f4'*h1;
PHI(17) = g4'*h1;

% Euler Parameters
PHI(18) = p2'*p2-1;
PHI(19) = p3'*p3-1;
PHI(20) = p4'*p4-1;

% vertical translation driving constraint
PHI(21) = y3 - y3_start - 2*sin(t);

% Jacobian by rows - Eq 3.3.12, page 66 for revolutes
JAC = zeros(21,21);

% Joint A - Revolute
JAC(1:3,1:3) = -eye(3);
JAC(1:3,4:7)=2*A2*s2pAskew*G2;
JAC(4,4:7) = 2*h'*A1'*A2*fpskew*G2;
JAC(5,4:7) = 2*h'*A1'*A2*gpskew*G2;

% Joint B - Spherical
JAC(6:8,1:3) = eye(3);
JAC(6:8,4:7) = -2*A2*s2pBskew*G2;
JAC(6:8,8:10) = -eye(3);
JAC(6:8,11:14) = 2*A3*s3pBskew*G3;

% Joint E - Double Revolute
JAC(9,8:10) = -2*dEF; JAC(9,11:14) = 4*dEF'*A3*s3pEskew*G3;
% Joint C - Spherical
JAC(10:12,8:10) = eye(3); JAC(10:12,11:14) = -2*A3*s3pCskew*G3;
JAC(10:12,15:17) = -eye(3); JAC(10:12,18:21) = 2*A4*s4pCskew*G4;

% Joint D - Revolute
JAC(13:15,15:17) = -eye(3); JAC(13:15,18:21) = 2*A4*s4pDskew*G4;
JAC(16,18:21) = -2*h'*A1'*A4*fpskew*G4;
JAC(17,18:21) = -2*h'*A1'*A4*gpskew*G4;

% Euler Parameters
JAC(18,4:7) = 2*p2';
JAC(19,11:14) = 2*p3';
JAC(20,18:21) = 2*p4';

% driving constraint in Jacobian - Eq 3.1.9, page 52
JAC(21,8:10) = [0 1 0];

% SLA_kin.m - Short-Long Arm Suspension
% position, velocity, and acceleration at desired position

% Newton-Raphson position solution - Eq 3.6.7 and 3.6.8, page 100
assy_tol = 0.00001;
DAA_phi;
while max(abs(PHI)) > assy_tol,
    q = q - inv(JAC) * PHI;
    DAA_phi;
end
Abstract

Musculoskeletal geometry of the human body affects how people move and how efficiently they move. For this project, a musculoskeletal model of the lower limb was developed in Simscape Multibody and run using MATLAB to investigate the effect of foot structure and musculotendon properties on performance and energy efficiency in a loaded ankle extension task. A test case was simulated and a sensitivity analysis was performed that suggested shorter heels (plantarflexor moment arms) may increase energy efficiency during certain movements. The simulation was also used to design a corresponding experiment that will be used for further exploration and validation of the model.

2. Introduction

Energy efficiency is an important factor in human locomotion and understanding how humans take advantage of the geometry and properties of their musculoskeletal system to increase this efficiency has many applications ranging from prosthetic and exoskeleton design to surgery to performance training for elite athletes. A main area currently under study is moment arms about different joints in the body, particularly the ankle (due to its importance in locomotion). The heel length determines the moment arm between the plantarflexor and the ankle while the toe length determines the approximate moment arm between the ground reaction force and the ankle. Studies have suggested that these moment arms affect the performance and efficiency of different movements. For example, sprinters have been shown to have longer toes and shorter heels than the general population [1,2]. The longer toes provide more contact time with the ground for the plantarflexors to develop force. The shorter heels allow the plantarflexor muscle to operate in a more optimal location on the force-length and force-velocity curves that govern the force production capabilities of muscles shown in Figure 1(a) and 1(b). A shorter heel requires a smaller length change and lower velocity of muscle for the same ankle motion, increasing the amount of force that can be produced.

Figure 1: Characteristics of the (a) muscle force-length relationship, (b) muscle force-velocity relationship, and (c) tendon force-length relationship and (d) diagram of a Hill-type muscle model with a contractile element CE in series with an elastic tendon element [3].

It has also been found in a study by Scholz, et al. that running economy increases with shorter heels [4]. The suggested explanation was based on the spring-like behavior of the tendon as represented in the Hill-type muscle model (Figure 1(c)).
Tendons operate on a force-length curve as shown in Figure 1(d). Therefore, the greater plantarflexor force required to produce the same ankle moment with shorter heels increases the strain on the tendon. The further the tendon stretches, the more energy it stores and then later returns, reducing the amount of energy that the muscles need to generate. However, a following study by van Werkhoven did not find this shorter heel-energy efficiency relationship. While energy storage may increase, there will also be an increase in metabolic cost due to the greater amount of required force and muscle activation [5].

So far, studies have had inconsistent results on which effect dominates. In order to further explore the effect of foot structure on performance and energy efficiency in humans, a musculoskeletal simulation of the lower limb was developed for this project.

3. Methodology

3.1. Simulation Design

Main considerations in the simulation design were that it (1) be a single-joint motion that focuses on the plantarflexor musculotendon complex and minimizes the contribution of other muscles, (2) has a clear objective that can be measured, and (3) allows for a good comparison with experimental work. The task chosen for the simulation was a loaded ankle extension (Figure 2). The objective of the task was to push a load up an incline. The load, distance the load was pushed, and angle of the incline determine the work done on the load.

In an experiment, the mass of the load or load application point could be varied in order to change the force transferred to the plantarflexors. The use of a simulation adds the ability to control internal variables, such as heel moment arm and muscle and tendon properties.

3.2. Musculoskeletal Model

A simple musculoskeletal model of the human body was used in this simulation to represent a single lower limb consisting of two segments: a foot and a lower leg. The upper leg and the rest of the body were assumed to remain stationary and the model was constrained to the sagittal plane (only flexion/extension of the ankle was considered). The geometric, inertial, and tissue properties for the model were based on average values from the commonly used Arnold model (originally from cadaver studies) [3]. An adaption of the Arnold model used for previous studies in the lab (Figure 3) [6] was referenced in the development of the current model.

Two lumped muscle-tendon units, representing the plantarflexors (medial and lateral gastrocnemius, soleus) and dorsiflexor (tibialis anterior) were used to drive the model. A Hill-type muscle model was used which included the force-length and force-velocity relationships of the muscle, as well as the force-length relationship of the tendon [7]. Metabolic energy rates for each muscle group were calculated and integrated over the duration of the simulation to determine total energy expenditure [8].

Figure 2: Experimental concept for the loaded ankle extension task showing heel (r) and toe (R) moment arms.
A mobility (M) analysis was performed on the design using equation (1). With four links \((nL = 4)\) and four joints that allow a single degree of freedom \((nJ1 = 4\) and \(nJ2 = 0)\), the mechanism was determined to have one degree of freedom, as desired. A mobility analysis of an earlier version of the design resulted in the addition of the revolute joint at the knee (previously a weld joint) in order to allow the mechanism to move.

\[
M = 3(nL - 1) - 2n/1 - n/2
\]  

(1)

### 3.4. Model Implementation

The model was implemented in Simscape Multibody (Simulink, Mathworks, Natick, MA) as shown in Figure A-1 (Appendix A). Blocks were used to represent the different bodies and joints in the model. Coordinate frames were set up at the joints and centers of mass for each body as shown in Figure 5(a). The muscle models were included as subsystems that received input about the position of the musculotendon attachment sites on the foot and leg segments in the world frame (Figure 6). This was used to calculate the current length and velocity states of the muscles and tendons which, along with the muscle excitation curves, determined the force produced.

The forces were returned to the main model and applied to the segments as tension along the muscle paths. The muscle often does not travel directly between attachment points on two segments. There are physical constraints, such as bone and soft tissue, that require a more complex path and are modeled as via points or wrapping surfaces. In the case of the dorsiflexor, two via points were included on the leg segment as shown in Figure 5(b). An example of the implementation of the coordinate system transformations in the leg subsystem of the model is shown in Figure A-2 (Appendix A).
Figure 5: Graphic representation of Simscape model showing (a) the coordinate frames for the joints, centers of mass, and muscle-tendon attachment via points and (b) the lines of action for the plantarflexor and dorsiflexor.

Another component of the model that is required in biological systems is limiting the range of motion of the joints that is constrained due to passive joint mechanics. To accomplish this, a torsional spring was added to the ankle that started applying torque to the joint when it went beyond the normal range of motion (50° plantarflexion to 20° dorsiflexion) as shown in Figure 6.

Figure 6: Flow of information between muscle models and body segments and ankle spring limiting the range of motion.

3.5. Forward Dynamic Simulation

The forward dynamic simulation of the model was executed in MATLAB (Mathworks, Natick, MA). A script was used to initialize the required parameters and positions. Variables were used for a majority of the parameters in the Simscape model (including the foot geometry, musculotendon properties, mass of the load, and angle of incline of the load path) and could be easily changed in the script. The inputs to control the model were muscle excitation curves for the plantarflexor and dorsiflexor (each consisting of 11 nodes evenly spaced over the duration of the simulation). The excitation values (representing electrical signal to the muscles) ranged from 0 (no excitation) to 1 (maximum excitation). After the setup, the script called the simulation and then read and compiled the results.

4. Simulation Results

For a test case, parameters were chosen that created a physically reasonable movement of the model. A mass of 20 kg, incline of 30°, and duration of 0.5 seconds were used. Only the plantarflexor was activated and the excitation curve was tuned (Figure 7) so that the model performed a semi-periodic motion (Figure 8).
Figure 7: Plantarflexor excitation curve used for the simulation.

Figure 8: Motion of model during simulation from initial position to full extension and back to the approximate initial position.

Figure 9: Position from start and velocity of load along the incline.

The resulting kinematics and kinetics of the load and ankle are shown in Figures 9 and 10. The load traveled a distance of 8.52 cm along the ramp before returning to near its original position.

The ankle covered a range of motion from 4.6° dorsiflexion (+) to 28.5° plantarflexion (-). The high extension torque of the ankle can be seen at the beginning of the movement that provides a spike in the velocity of the ankle and the load. The high velocity of the ankle and therefore the plantarflexor causes a decrease in possible force production and a decrease in the torque. As the load reaches its maximum distance and starts to move back down the incline, the plantarflexor muscle stays activated and works against the gravitational force to slow the travel of the load in order to approximate the periodic motion.

Internal variables that are difficult or impossible to currently measured in humans can also be determined from the simulation. These variables include the force, rate of energy consumption, and tendon strain of the plantarflexor which are shown in Figure 11. The force shows the same pattern as the ankle torque (as it causes the ankle torque). The rate of energy consumption and strain in the tendon also follow a similar pattern as both are dependent upon the force. The energy rate is estimated using the activation, force, and velocity of the muscle, with a total energy of 18.29 J consumed by the plantarflexor. The tendon acts as a spring and the amount of tendon strain (and energy stored in the tendon) increases with the force.
A sensitivity analysis was performed to examine the effect of varying moment arms on performance and energy expenditure during the ankle extension task. The simulation was run multiple times with small changes in the heel length. The original heel length was 4.38 cm so 0.1 cm increments were chosen to provide about a 2% change. An adjustment of 1 cm in each direction was also simulated to check if the observed relationships held for larger changes.

For each heel length, the resulting distance traveled by the load and estimated energy expenditure of the plantarflexor are shown in Figure 12. Though the differences are fairly small, a trend towards greater distance traveled coupled with a decrease in energy use suggests greater energy efficiency with shorter heels. However, this relationship needs to be explored further to determine its dependency on other model parameters and the setup of the testing device.
5. Conclusion

For this project, a lower limb musculoskeletal model and simulation were developed to investigate the effect of musculoskeletal geometry on the performance and energy efficiency of a simple ankle extension task. The simulation was used for an initial exploration into the effect of heel moment arm on performance and energy consumption. The results support the theory that shorter heels allow for increased energy efficiency during human movement and suggest the need for further study of the mechanisms behind this potential relationship. The simulation can also be used to study the effect of other factors, such as optimal muscle fiber length, tendon stiffness, and movement frequency and the interactions between them.

The planned next step is to put the simulation into an optimization that will determine the control strategy (muscle excitation curves) to perform a specific ankle extension task while minimizing energy. The results will be compared to an experimental study that was developed during the process of this project to correspond as closely as possible to the simulation. The experimental study will also be used to help validate this model.

References

Figure A-1: Full Simscape Multibody model showing structure of bodies, joints, and actuators.
Figure A-2: Expanded leg subsystem from the Simscape Multibody model showing the implementation of the transformations from the knee coordinate frame to the frames for the center of mass (COM), ankle joint, and muscle attachment and via points for the plantarflexor (PF) and dorsiflexor (DF).
ME 581: Simulation of Mechanical Systems
Multi-Link Manipulator with Compliant End Effector

Justin A. Jones

Introduction

According to Credence Research [1] the medical robotic industry is valued at 7.24 billion in 2015 and will grow to over 20 billion by 2023. It is important that the robotic mechanism is well designed since a poor design could be potentially life threatening.

Probably the most well-known surgical robot would be the da Vinci system developed by Intuitive Surgical, as seen in Figure 1 (A). These surgical robotic systems utilize specialized end effectors to grasp and cut in order for the physician to perform the procedure. An example of such a design can be seen in Figure 1(B). These rigid mechanisms consist of multiple moving small parts. A compliant mechanism design may provide benefits over these rigid end effector mechanisms due to the ability to make highly accurate mechanisms with a lower number of parts as well as the use of micro manufacturing process.

This report develops the forward kinematic equations for a four degree of freedom (DOF) robotic manipulator using Denavit-Hartenberg (DH) convention as well as analysis of a potential compliant end effector tool for grasping. The kinematic analysis of the compliant gripper will utilize Ed Haug notation. A multibody dynamic simulation was also performed on the compliant mechanism to determine joint stresses with respect to the driver position.

Figure 1. (A) The da Vinci robotic assisted surgery system developed by Intuitive Surgical [2] and (B) a rigid body end effector used for cutting[3].

2. Methodology

2.1. Robotic Manipulator

A robotic manipulator with four revolute joints and five links was chosen for analyses. A common method to assign coordinate systems is that of the Denavit-Hartenberg method [4]. This method has four constraints that must be adhered to in order to generate the coordinate system. They are as follows:

1: $X_n \perp Z_{n-1}$
2: $X_n \cap Z_{n-1} \neq 0$
3: Origin of joint $n \cap X_n$ and $Z_n$
4: Coordinate system must be right handed.

Additionally, there are four parameters that are used to describe the system described as follows:

$a_n =$ the distance from $Z_n$ to $Z_{n+1}$ along $X_n$
$\alpha_n =$ the angle from $Z_n$ to $Z_{n+1}$ measured about $X_n$
$d_n =$ the distance from $X_{n-1}$ to $X_n$ along $Z_n$
$\Theta_n =$ the angle from $X_{n-1}$ to $X_n$ about $Z_n$

The 4-DOF robotic arm consisting of four revolute joints with the coordinate frames attached using the DH convention can be seen in Figure 2. The corresponding DH parameters can be seen in table 1.
The overall transformation matrix consists of the rotational matrix $R_{ij}$ and the positional vector $P$. The vector $P$ in Equation 7 represents the global
coordinates of the arm end point P shown in Figure 2.

To check the forward kinematic equations Figures 3 and 4 show the rotation about the wrist and shoulder joint, varying from 0 to $2\pi$ while holding the other joints fixed.

![Figure 3 Rotational position of wrist joint.](image)

![Figure 4 Rotational position of manipulator by iterating $\vartheta_1$.](image)

### 2.2. Compliant Gripper

Compliant mechanisms provide many benefits when compared to their rigid body counter parts. Some of these benefits include ease of manufacturing, reduction of wear and friction, and reduced need for lubrication [5, 6]. Additionally, compliant mechanisms are widely used in the manufacturing of MEMs and NEMS as well as in photolithography. The ability to make very small mechanisms that have reduced wear and maintenance could prove large benefits to the medical field, such as in the area of non-invasive surgeries like laparoscopic surgery.

To design a compliant mechanism, at least when only the kinematics are of concern, it is possible to replace the flexible members with revolute joints and treat the mechanism as a rigid body mechanism. However, it should be noted that for compliant mechanisms the flexible member degree of rotation is constrained. Further, they are prone to failure due to overstress and fatigue. For the dynamic response of the mechanism it is possible to use the Pseudo-Rigid-Body Model that was developed by Howell and Midha [7]. They developed this model by applying constant rate torsional springs to the revolute joints among other simplifications.

#### 2.2.1 Slide Crank

To drive the compliant gripper, the first option would be to use a linear actuator due to its compact nature and ability to apply large loads. However, due to the higher cost of micro linear actuators a slider crank mechanism was used for the driver. To determine the stroke length of the mechanism a geometric analysis was performed using equation 8:

$$s = R \cos \vartheta + L \cos \phi$$  \hspace{1cm} (8)

Where $s$, $R$, $L$, $\vartheta$, and $\phi$ are defined in Figure 5.

![Figure 5 Generic inline slider crank mechanism used to determine stroke length for compliant gripper.](image)
By using an $R$ of 5 mm and an $L$ of 15 mm, a stroke length of 10 mm was determined. The stroke profile versus the crank angle can be seen in Figure 6.

Using Ed Haugs method the kinematic constraints, as well as the Jacobian were developed and are provided in the appendix. Using the Newton-Raphson method the position of position P and Q with respect to the driver angle were calculated and can be seen in Figure 8.

2.2.2 Kinematic Analysis
The final compliant gripper design can be seen in Figure 7(A) with its rigid body kinematic diagram in Figure 7(B). It can be seen that the mechanism consists of eight links, nine revolute joints, and one prismatic joint. This will give an overall mobility of one.

![Figure 7](image)

Figure 7 (A) Compliant gripper design and (B) skeletal diagram with joint and links labeled.

It can be seen that for a stroke length of 5 mm the grippers close at 115 degrees and remain closed until the driver is at 240 degrees. This over closure could be useful if it is determined that it allows for a large closing force without causing the flexure hinges to fail. To determine this it is necessary to perfume a dynamic analysis.

2.3. Dynamic Simulation
To perfume the dynamic analysis the finite element method (FEM) is used. The use of FEM has been utilized in the design and analyses of compliant mechanisms in previous works [6, 8, 9]. For the compliant gripper design a multibody dynamic simulation was ran using ABAQUS software. For the material properties AISI 304 stainless steel, with a Young’s modulus of 200 Gpa, Poisson ratio of 0.29 and density of 1800 kg/m$^3$ was used for the crank, crank shaft and pins. For the gripper PLA was chosen with a Young’s modulus of 3.5 Gpa, poisons ration of 0.35 and density of 1300 kg/m$^3$. This material was chosen due to the fact that the part would eventually be manufactured using a 3d printer and PLA is a
common material choice for this manufacturing process. The mesh quality as well as loading and boundary conditions can be seen in Figure 9.

A constant rotational velocity, $\omega$, of 6.2 rads/s was applied to the driver. The end of the gripper was fixed in all directions. The crank shaft, pins and crank were fixed from translation in the y as well as rotation about x and z. A dynamic study was chosen for 0.5 seconds. The joint Von Mises stress as well as the position of P and Q were recorded with respect to the driver angle.

The results for the dynamic simulation can be seen in Figure 10. It can be seen the peak stress occurs on joints I and G, which makes intuitive sense since this is the main point of rotation. Further, it can be seen that the peak stress occurs when the crank is at 180 degrees. Joints G and I are the locations of potential failure and future designs will have to take these factors into consideration.

When the crank angle is at 180 degrees, the gripper is at its peak closure. Figure 11 shows the gripping contact pressure for this specific design as the crank rotates. This information can be combined with the fact the in general cases the force necessary to pick up an object is defined by:

$$F = \frac{m(a + g)}{\mu n}$$  \hspace{1cm} (#)

Where $F$ is the force required to grip the object, $m$ is the mass of the object, $a$ is the acceleration of the object, $g$ is acceleration due to gravity, $\mu$ is the coefficient of friction, and $n$ is the number of fingers on the gripper. With this it is possible to set the driver angle to get the closing force needed to lift an object.

The position of point P and Q were also tracked with the simulation and can be seen in Figure 12. These results agree with the previous kinematic analysis. The peak displacement occurs when the driver is at approximately 120 degrees. The peak...
displacement is constrained to approximately 10 mm, the closing length.

**Figure 12 Relative displacement of point P and Q.**

### 4. Conclusion and Future Works

The Devenat Heartinber convention was used to develop the forward kinematic equations for the four DOF robotic manipulator. This method is an efficient way to describe the kinematics for any robotic manipulator.

The compliant gripper showed to be a successful mechanism that may provide benefits over the traditional rigid body mechanism that are currently used as end effectors.

For future work a 4 DOF robotic arm is being manufacture and the inverse kinematic equations will be developed. The goal will be to use an external camera to generate the desired end location and feed the data into the inverse kinematic equations to calculate the required joint angles. These angles will be provided to an ARDUINO that controls the servo motors. The initial design can be seen in Figure 13.

**Figure 13 Four degree of freedom robotic manipulator for a future build.**

### References


[2] [https://www.intuitivesurgical.com/](https://www.intuitivesurgical.com/).


General coordinates

\[
\{q\} = \begin{bmatrix}
\{q_2\} \\
\{q_3\} \\
\{q_4\}
\end{bmatrix} = \begin{bmatrix}
\phi_2 \\
\phi_3 \\
\phi_4 \\
\phi_5 \\
\phi_6 \\
\phi_7 \\
\phi_8
\end{bmatrix}
= \begin{bmatrix}
x_2 \\
y_2 \\
\phi_2 \\
x_3 \\
y_3 \\
\phi_3 \\
x_4 \\
y_4 \\
\phi_4 \\
x_5 \\
y_5 \\
\phi_5 \\
x_6 \\
y_6 \\
\phi_6 \\
x_7 \\
y_7 \\
\phi_7 \\
x_{78} \\
y_8 \\
\phi_8
\end{bmatrix}
\]

Kinematic Constraints

\[
\{\Phi\} = \begin{bmatrix}
\{\Phi\}_{\text{KINE}} \\
\{\Phi\}_{\text{REV\_A}} \\
\{\Phi\}_{\text{REV\_B}} \\
\{\Phi\}_{\text{REV\_C}} \\
\{\Phi\}_{\text{REV\_D}} \\
\{\Phi\}_{\text{REV\_E}} \\
\{\Phi\}_{\text{REV\_F}} \\
\{\Phi\}_{\text{REV\_G}} \\
\{\Phi\}_{\text{REV\_H}} \\
\{\Phi\}_{\text{REV\_I}} \\
\{\Phi\}_{\text{DRIVER}}
\end{bmatrix} = \begin{bmatrix}
\{r_2\}_j - \{r_1\}_j \\
\{r_3\}_j - \{r_1\}_j \\
\{r_4\}_j - \{r_1\}_j \\
\{r_5\}_j - \{r_1\}_j \\
\{r_6\}_j - \{r_1\}_j \\
\{r_7\}_j - \{r_1\}_j \\
\{r_8\}_j - \{r_1\}_j \\
\{r_9\}_j - \{r_1\}_j \\
\{r_{10}\}_j - \{r_1\}_j \\
\{r_{11}\}_j - \{r_1\}_j \\
\{r_{12}\}_j - \{r_1\}_j \\
\phi_j - \phi_{\text{STAB\_REV}} - \omega_j
\end{bmatrix} = \{0\}
\]
Kinematic Constraints

\[
\begin{align*}
\{r_2\} + [A_2] \{s_{2r}\}^A & - \{r_1\}^A \\
\{r_3\} + [A_3] \{s_{3r}\}^B & - \{r_2\} - [A_2] \{s_{2r}\}^B \\
\{r_4\} + [A_4] \{s_{4r}\}^C & - \{r_3\} - [A_3] \{s_{3r}\}^C \\
y_4 \phi_4^T & \\
\{r_5\} + [A_5] \{s_{5r}\}^D & - \{r_4\} - [A_4] \{s_{4r}\}^D \\
\{r_7\} + [A_7] \{s_{7r}\}^E & - \{r_6\} - [A_6] \{s_{6r}\}^E \\
\{r_8\} + [A_8] \{s_{8r}\}^H & - \{r_7\} - [A_7] \{s_{7r}\}^H \\
\{r_8\} + [A_8] \{s_{8r}\}^I & - \{r_7\}^I \\
\phi_2 - \phi_{2,\text{START}} & - \omega_2 t
\end{align*}
\]
DESIGN AND ANALYSIS OF A LOW COST PATIENT-SPECIFIC WRIST-HAND ORTHOSIS

Maryam Tilton
Department of Mechanical Engineering
Pennsylvania State University
University Park, PA 16803
Email: mxj63@psu.edu

Soumyabrata Maiti∗
Engineering Science and Mechanics
Pennsylvania State University
University Park, PA 16803
Email: sxm710@psu.edu

ABSTRACT

In this work, we present a simple yet effective design modification of hand orthosis to help with gradual recovery of stroke patients through different stages of rehabilitation. Here, we have used a main lever with adjustable length attached to the husk. We have experimentally found that as we increase the length of the lever, the mechanism does more work. We also have supported the result by simulating the system numerically in ADAMS.

1 Introduction

Wrist-Hand Orthosis (WHO) are mainly used for maladjustment correction with both mobile and immobile functionality. This is done by providing a required support for weekend muscle structures and increasing the function of a totally or partially immobile body members [2]. Because of variety of patients and types of deformity, wrist-hand orthoses is often customized to meet the unique need and anthropometric measurements of each individuals. Among all the causes of movement disability, stroke is considered as the primary cause, especially in developed countrie [1]. For many stroke patients hypertonia, spasticity, abnormal synergies, and extension weakness result hyperflexion of the wrist and fingers which leads to limitation of their ability to open and use their hand. This has a negative impact on their daily activity. Currently, majority of the post stroke rehabilitation takes place in physical therapy centers [1]. In this project, our aim was to design a wrist-hand orthoses, specifically for stroke patients, that can eliminate some of the problems exist in currently available designs. We set the following desired features as our design objectives:

1. Alignment of fingers, wrist, and arm
2. Providing a support for arm
3. Adjustable for different stages of rehabilitation
4. Can be used independent by patient
5. Patient specific

2 Design Procedures

Our design is inspired by Eliza Wrobel’s WHO. We modified her design by adding adjustability to main lever position. This modification allows us to evaluate the main hypothesis for this study: which states that "positioning the main lever further from the wrist increases the amount of work done for a same task". Dimensions used for the CAD model are obtained from anthropometric measurements taken from one of the group members. After preparing the CAD model in Geomagic Design X, we proceeded with 3D printing all the parts, which were in total 77 parts, using PLA filament.

3 Experimental analysis

3.1 Experimental procedure

3.1.1 Objectives of the experiment

Here, the objective of our experiment is to find the work done by the whole mechanism under different lever lengths. In order to that, we attached one spring in between the tips of the index finger and the thumb and let it compressed by the user. We performed the experiment in near quasi-static condition so that the inertial effects of all the links become insignificant. In that case, the energy
stored inside the spring will be proportionate to the work done by the abductor pollicis brevis muscle. Measuring this energy will help us to understand the effect of the adjustable lever length on the above mentioned muscle.

Furthermore, we will study the time domain response of the angle between the distal phalange and the middle phalange (θ₁) and that between palm and the proximal phalange (θ₃) (see Fig. 2) to obtain the insight to the muscle behaviour. Later, the observations made from the experimental results will be used to numerically model governing actuating muscle.

3.1.2 Measurement of system variables  We have done the experiment in two stages. In the first stage, we choose a random spring for the experiment. We have measured its stiffness by measuring its frequency of oscillation when a lump mass is attached to its end. Since the the amount of the lump mass is known, we have used the formulae  \( K = \omega^2 m \) to measure the stiffness. Here, \( K \) is the stiffness, \( \omega \) is the frequency of oscillation, and \( m \) is the mass. The natural frequency \( \omega \) is calculated by finding the amount of time taken to perform a finite number of oscillations. Then we plot the number of oscillations against the time and find the slope of the least square fitted line through the data points.

At the second stage of experiment, we allow the user to gradually compress the spring for two cases, in the first case we set the lever length to 57.4 mm and in the second one we changed that to 77.4 mm. We record the movement using a 30 fps camera for both cases and then digitized the frame to extract the angles and the spring displacements.

Table 1: Displacement initial conditions for case I

<table>
<thead>
<tr>
<th>Variable</th>
<th>Angle (in degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ₁</td>
<td>139.896</td>
</tr>
<tr>
<td>θ₂</td>
<td>138.043</td>
</tr>
<tr>
<td>θ₃</td>
<td>123.490</td>
</tr>
</tbody>
</table>

Table 2: Displacement initial conditions for case II

<table>
<thead>
<tr>
<th>Variable</th>
<th>Angle (in degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ₁</td>
<td>132.562</td>
</tr>
<tr>
<td>θ₂</td>
<td>125.1420</td>
</tr>
<tr>
<td>θ₃</td>
<td>117.2640</td>
</tr>
</tbody>
</table>

3.2 Experimental results  
In this section, we will discuss the experimental observations and try to construct the background for our numerical study. During the experiment, we move all the four fingers simultaneously and focus on the movement of the index fingers. From our observation, we have found that the angular displacement of the thumb is negligible in compare to the other fingers. In both of experimental study, initial angular displacements are listed Tables 3 and 2.

In both the cases, we see the response of θ₃ has a dominating sinusoidal trend. Whereas the underlying trend of θ₁ has a frequency which is half of the frequency if θ₂. For example, the
Next we calculate the compressive work done on the spring for a half cycle. We have experimentally found that the stiffness of the spring is 125.4966 N/m. We also have calculated the displacements of the spring at each time step for different cases. Since the work done is defined by $\int Kx dx$, we employ trapezoidal rule to find the work done for half cycle. Note that, the process is nearly quasi-static. So we can assume that the work done by the muscle is nearly equal to the sum of the energy stored in the spring and the energy dissipated due to various joint frictions. As the mechanism is same for both case, the energy dissipated due to friction can be considered nearly constant in each case. Hence, the work done on the spring is proportionate to the work done by the muscle.

We have calculated the energy stored in the spring during the half cycle of the process are 0.404 and 0.426 Joule for case I and II respectively. This result indicates that the muscle has to do more work as we increase the length of the main lever. However, the data obtained from the experiment is fairly noisy. Hence we have employed a numerical technique to a simplified version of the hand mechanism with the wrist hand orthosis, which we have discussed in details in the next section.

4 Numerical Analysis

4.1 Assumptions and modelling

In previous section, we have seen that the response of $\theta_3$ has a dominating sinusoidal feature. So if we consider the abductor pollicis brevis to be a slider mechanism attached to the palm and the proximal phalanges to be the crank, the angular displacement response of the crank is found to be similar to that obtained from the experiment. Furthermore, the constraints imposed by orthosis restricts the motion of all the fingers (except thumb) on a single plane. Hence, we can consider the real mechanism to be a two dimensional one with a slider crank mechanism representing the actuating muscle. With this motivation, we now proceed to the qualitative details of our ADAMS model.

In our numerical model, we will study the two dimensional movements of the mechanism comprised with the index finger, the thumb, and the parts of orthosis joining them. Since, the thumb movement is insignificant in our plane of observation, thumb is considered to be the ground. We further assume that all the links are rigid and of cuboid shape. The width and the height of the links, depicting the hand, are taken directly from the subject. The values of both the parameters are set equal to the diameter of each link. The lengths are directly measured from the index finger and thumb of the subject as well as the orthosis mechanism. All the links are connected with revolute joints. The joints A, B, and C, on the index figures (see Fig. 4), also contain torsional spring (with stiffness $O(10^2)$ N-mm/deg). The density of the links, belonging to the finger, was set to 1900 kg/m$^3$ and of those belonging to orthosis was set to 1250 kg/m$^3$. The initial configuration was set in such a way that the whole mechanism satisfied the geometrical constraints and the initial conditions of the angles were in close range of Case I.

A displacement profile of $2\cos(1.57*t)$ mm was imposed on the slider to represent a constant power input to the system. A spring of stiffness 126 N/m was placed in between the index and thumb (see Fig. 4) to study its displacement variation over time.
As noted in the experiment, the variable $\theta_3$ is not significantly affect by the nonlinear stiffness of the muscle if we keep our observation well within the fast time scale. So we will observe the numerical response till 5 sec to avoid the effect of nonlinearity. Next we discuss the observations from numerical results.

4.2 Results

The time domain responses of $\theta_3$ and spring displacement are given in Figs. 5 - 6. The profiles of $\theta_3$ are very much similar to that obtained from the experiment in both the cases. When the length of link 3 is 78 mm, we see that the amplitude of the displacement curve is about 2.5 mm. As we increase the length of link 3 to 112 mm, the displacement amplitude is found to be decreased to about 0.5 mm. This has occurred because of the considerable fraction of input energy is diverted to the system to satisfy the kinematic constraints as we increase the length of link 3. Hence, the net available energy for the compression of the spring is reduced in later case.

This observation validates our experimental observation that increasing the length of link 3 will force the muscle to do more work to achieve a targeted mechanical goal.

5 Conclusion

From the experimental results we can conclude that placing the main lever further from the wrist, takes more power from the patient to perform a simple task of grabbing (using thumb and index finger) comparing to when the main lever is positioned closer to the wrist. This result justifies the adjustability feature of the design. Adjusting the position of the main lever at different stages of rehabilitation helps with gradual improvement of muscles. Furthermore, from the obtained results, we observed that muscle can act as nonlinear torsional spring with quadratic nonlinearity. These observations lead to need of future work in this area. For the future studies, we can incorporate adjustability of the constraints into the design. Also, more in depth analysis of system response can help us to improve the existing mathematical model of the muscles. And lastly, implementation of feedback control system is suggested to mimic the hand’s natural mechanism with more precision.

REFERENCES


Table 3: My caption

<table>
<thead>
<tr>
<th></th>
<th>Index Distal Phalange (cm)</th>
<th>Index Middle Phalange (cm)</th>
<th>Index Proximal Phalange (cm)</th>
<th>Thumb Proximal Phalange (cm)</th>
<th>Thumb Distal Phalange (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L = 2.529</td>
<td>L = 4.520</td>
<td>L = 2.612</td>
<td>L = 2.800</td>
<td>L = 3.340</td>
</tr>
<tr>
<td></td>
<td>D = 1.84</td>
<td>D = 2.19</td>
<td>D = 2.52</td>
<td>D = 2.03</td>
<td>D = 2.96</td>
</tr>
</tbody>
</table>

Figure 6: Numerically obtained time domain response of $\theta_3$ and displacement when the effective length of the main lever is 112 mm.
ME 581: Analysis and Simulation of Rotary Engine
Two—Dimensional Kinematics of a Three Flank Rotary Engine

Hugo McMenamin

Abstract

Engine design and analysis has varied throughout the history of internal combustion engines. The most popular design in modern day passenger vehicles is a reciprocating engine with pistons sliding within cylinders. One of the major detractors of this design is that the piston velocity will be zero at the top and bottom of each stroke [1].

A less popular passenger vehicle engine design is the internal combustion rotary engine which was first designed and produced by Felix Wankel with NSU [2]. This design uses the same four-stroke cycle as in reciprocating engines, however rather than a piston sliding in a cylinder, uses a rotor spinning within a housing. Because of this, rotary engines do not have any reciprocating parts including intake or exhaust valve mechanisms, and are able to have much higher power to weight ratios [3]. This report explores the two-dimensional movement of the crankshaft and rotor within the engine housing of a production rotary engine.

2. Methodology

2.1. Background

The automotive company Mazda implemented rotary engines in the RX-7 vehicles between 1978 and 2002 [4]. The engine analyzed in this report is the 13B-REW as seen in Figure 1-1, and is from the final generation of the RX-7.

The peritrochoid is defined by [1]

\[ x = e \cos(\alpha) + R \cos \left( \frac{\alpha}{m} \right) \]  
\[ y = e \sin(\alpha) + R \sin \left( \frac{\alpha}{m} \right) \]

Where \( x \) and \( y \) are the coordinates of each point on the curve, \( e \) is the eccentricity or offset of the rotor from the crankshaft, \( R \) is the generating radius or tip radius of the rotor, \( \alpha \) is the angle of the crankshaft, and \( m \) is the number of flanks of the rotor.

As shown in Figure 1-2, eccentricity is also referred to as \( l \), \( O_1 \) is the center of the rotor, \( O_2 \) is the center of the crank shaft, \( r_2 \) is the radius of the crank shaft, \( r_1 \) is the radius of the pitch circle within the rotor, and \( r_{root} \) is the distance from the center of the rotor to the center of the flank.
Given that the rotor is a three-sided closed shape, each apex or corner is 120° from the next and so the location of each can be defined by modifying (1) and (2) to be

\[ x = e \cos(\alpha) + R \cos\left(\frac{\alpha + n2\pi}{m}\right) \quad (3) \]
\[ y = e \sin(\alpha) + R \sin\left(\frac{\alpha + n2\pi}{m}\right) \quad (4) \]

Where \( n = 0,1,2 \) depending on the apex. The equations were used to validate the kinematic model of the motion of the rotor and crankshaft.

### 2.2. Kinematic Model Development

The first step in creating the kinematic model of the engine was to determine the two-dimensional links and joints. As shown in Figure 2-3 the housing, rotor, and crankshaft are links 1, 2, and 3.

![Figure 2-3. Links 1-3 and coordinates.](image)

The apexes were then modeled as pin in slot joints accounting for 3 more links and joints as shown in Figure 2-4.

![Figure 2-4. Links 4-6 and coordinates.](image)

Finally, the connection between the rotor and crankshaft was modeled as an internal gear at \( P \).

Using Gruebler’s equation, this yielded a mobility of 2 meaning there is a hidden constraint.

### 2.3. Coordinate and constraint vectors

Based on the links and joints determined previously, the coordinate vector \( \{q\} \) and constraint vector \( \{\Phi\} \) can be derived. The coordinate vector is based on the global positions and angles of each coordinate for links 2 through 6.

\[
\{q\}_{15 \times 1} = \begin{bmatrix}
R_2 \\
\varphi_2 \\
R_3 \\
\varphi_3 \\
R_4 \\
\varphi_4 \\
R_5 \\
\varphi_5 \\
R_6 \\
\varphi_6 
\end{bmatrix}
\]

The constraint vector was generated based on the constraints for revolute joints, prismatic joints, and the internal gear join. The constraint for each revolute is defined by Haug as

\[
\{r_i\}_P - \{r_j\}_P
\]

However here it was defined as

\[
\{r_j\}_P - \{r_i\}_P = \{0_{2 \times 1}\}
\]

The constraint for each prismatic joint was defined as

\[
\{y_i\} \\
\varphi_i
\]

The constraint for the internal gear was defined by Haug as

\[
\Phi_{gear(i,j)} = \{(x^P_i - x^P_j) \sin(\theta) - (y^P_i - y^P_j) \cos(\theta)\}
\]

[8]

And was based on the values of \( x^P_i, x^P_j, y^P_i, y^P_j \), and \( \theta \) in Figure 2-5.

![Figure 2-5. Internal gear set schematic](image)
The driver of the system is then the angle of the crank, \( \Phi_2 \). For this system, the crankshaft was rotating at 6000 rpm, corresponding to the maximum power output, 268kW, of the engine. The final constraint vector was

\[
\Phi_{14x1} = \begin{bmatrix}
\Phi_{\text{revolute.A}} \\
\Phi_{\text{revolute.B}} \\
\Phi_{\text{revolute.C}} \\
\Phi_{\text{prismatic.A}} \\
\Phi_{\text{prismatic.B}} \\
\Phi_{\text{prismatic.C}} \\
\Phi_{\text{Gear}} \\
\Phi_{\text{Driver}}
\end{bmatrix}
= \begin{bmatrix}
r_A^A - r_I^A \dot{x}_1 \\
r_B^B - r_I^B \dot{x}_1 \\
r_C^C - r_I^C \dot{x}_1 \\
\gamma_4 \\
\gamma_5 \\
\gamma_6 \\
\phi_4^2 \dot{x}_1 \\
\phi_5^2 \dot{x}_1 \\
\phi_6^2 \dot{x}_1 \\
\phi_1^2 \dot{x}_1 \\
\phi_2 - \phi_2(0) - \omega t
\end{bmatrix}
\]

2.4. Jacobian Development

The next step in the kinematic analysis was to write the Jacobian matrix however, as described above, the constraint vector had only 14 rows and the coordinate vector had 15 rows. This would make the Jacobian not-invertible and make the Newton-Raphson iteration impossible. Still, the Jacobian could be established using the entries below.

\[
\begin{align*}
[\Phi_{q,i}]_{\text{revolute}} &= -[I_2 \quad [B_i] [s_i]^p]_{2x3} \\
[\Phi_{q,j}]_{\text{revolute}} &= [I_2 \quad [B_j] [s_j]^p]_{2x3} \\
[\Phi_{q,i}]_{\text{slider}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{align*}
\]

The Jacobian entry for the internal gear is then defined by Haug as [8]

\[
[\Phi_{q,i}]_{\text{gear}} = \left[ -u^r, -s_i^r v^r B_i^T u + (r_i^r - r_i^p)^T u^i \left( \frac{R_i}{R_1 + R_i} \right) \right]_{2x3}
\]

\[
u = \left[ \cos(\theta), \sin(\theta) \right]^T_{2x1}
\]

Although the Jacobian was not successfully derived, this is likely due to an issue in the initial model development. A mobility of 2 likely indicates there is an underlying issue with the initial set of joints and links. One potential solution would be to substitute a revolute joint between links 2 and 3 rather than an internal gear. However, given that there is an internal gear in the actual mechanism, this may cause other unforeseen issues that cause the kinematic system to incorrectly model the physical system.

3. Simulation and Results

Although the kinematic analysis of the system was not accurate as described in the previous section, the basic dimensions in the two-dimensional model could still be simulated. Using MATLAB, the equations 1, 2, 3, and 4, and the relations shown in section 2.1. Background, a model was created that stepped through three revolutions of the crank angle. Figure 3-6 shows the initial location of the rotor, crank, and P and Q when the crank angle is 0°.

![Figure 3-6. Initial location of rotor at \( \Phi_2 \) of 0°.](image)

Figure 3-7 below shows the mechanism at three different crank angles that successfully show how the rotor translates through the engine housing and rotates at one third the speed of the crank.

![Figure 3-7. Mechanism at three different crank angles.](image)
Figure 3-7. Rotor and crank position simulation at varying crank angles.

4. Conclusion

Although the Newton-Raphson simulation of the kinematics of the rotary engine were not successful, the two-dimensional simulation of the system based on the governing geometrical equations, proved the underlying logic. The results of the simulation showed the actual motion of the rotary engine. The rotor rotated at one third the speed of the crankshaft, which follows based on the operation of a rotary engine [3]. In future work, the initial coordinates and constraints need to be revisited to determine the underlying issue. Once adjusted, the Newton-Raphson solution can be completed and compared with the two-dimensional simulation results.

References


KINEMATIC ANALYSIS AND SIMULATION OF AN AMUSEMENT RIDE

Nicholas C. Papavizas
Department of Mechanical & Nuclear Engineering
The Pennsylvania State University
University Park, Pennsylvania 16802
Email: nkp5117@psu.edu

ABSTRACT
This paper investigates the kinematics of the “Crazy Couch”, a family-oriented amusement ride designed by Skyline Attractions, LLC. The ride is modeled as a 2-degree-of-freedom (2-DOF) planar mechanism driven by two constant speed rotational drivers and analyzed using a method developed by E.J. Haug. The mathematical model is verified against two other computer simulations of the ride, which were carried out in the programs Working Model 2D and SolidWorks. Results provide insight into the dependence of ride experience on seating location as well as the effects of driving speeds on accelerations and ride cycle duration.

1 INTRODUCTION
Amusement rides typically employ mechanical structures and linkages to move people in pre-defined paths through horizontal or vertical planes as well as three-dimensional (3D) space. These motion paths often utilize quick and even surprising changes in speed and/or direction to cause enjoyable physical sensations.

One such ride is the Crazy Couch, shown in Figure 1(a). Designed by Skyline Attractions, LLC, the Crazy Couch is a family-oriented ride with a compact 18’ x 7’ footprint [1]. It bounces riders up and down and tilts them side-to-side in manner that is difficult to decipher upon first glance. The ride utilizes a 5-bar planar linkage with four moving bodies: two cranks and two intermediate coupler links. Traditionally, a 5-bar mechanism requires two independent drivers to have fully defined motion, but the Crazy Couch operates on a single input as shown in Figure 1(b). The output shaft of a single AC Motor is connected to a miter gear box, which redirects the driveline to two drive shafts so that torque may be supplied to the cranks. Each crank is connected to its respective driveshaft via a reducer gear box with its own unique reduction ratio, meaning that the cranks rotate at different speeds. This is what ultimately gives the Crazy Couch its varied tilting and bouncing motions.

FIGURE 1: THE CRAZY COUCH AMUSEMENT RIDE: (a) FRONT VIEW (IMAGE PROPERTY OF SKYLINE ATTRACTIONS) AND (b) REAR VIEW.
This paper explores the motions and sensations one may experience while riding the Crazy Couch via theoretical kinematic analysis and computer simulation. Kinematic equations of motion for the fundamental mechanism of the Crazy Couch are derived using the analysis methods developed by E.J. Haug for constrained multibody systems [2]. These equations are programmed into MATLAB [3] and the results are compared with those obtained from computer simulations of the Crazy Couch carried out using two different simulation programs: Working Model 2D [4] and SolidWorks + Motion Analysis [5]. Finally, a sensitivity study is performed in which the gear ratios of the reduction gear boxes are varied in order to investigate effects on ride aspects such as maximum acceleration magnitudes at different locations on the mechanism and ride cycle duration.

2 METHODOLOGY

2.1 Simple Model and Mobility

The fundamental mechanism of the ride is a 5-bar planar mechanical linkage as depicted in the skeletal diagram of Figure 2, which views the ride from the front as in Figure 1(a). The mechanism exists in a vertical plane with X taken to be the horizontal direction and Y taken to be the vertical. The mechanism comprises four moving bodies: a first crank (link 2), the seat span (link 3), a slider block or trolley (link 4), and a second crank (link 5). Ground is traditionally labeled link 1. Points A, B, C, and D represent revolute joints connecting link 2 to ground, link 2 to link 3, link 4 to link 5, and link 5 to ground, respectively. On the Crazy Couch the slider trolley sits in a linear bearing on the underside of the seat span, so the interface between links 4 and 5 in Figure 2 is a translational/prismatic joint that allows the seat span to slide over the second crank.

\[ \{q_i\} = \left\{ \begin{array}{c} \phi_1^T \\ \phi_2^T \end{array} \right\} = \left\{ \begin{array}{c} x_1 \\ y_1 \\ \phi_1 \end{array} \right\} \quad i \in \{2,3,4,5\} \]  

\[ \{q\} = \left\{ \begin{array}{c} q_2^T \\ q_3^T \\ q_4^T \\ q_5^T \end{array} \right\} \]  

where \( n_L \) is the number of links (including ground), \( n_J_1 \) is the number of full joints, and \( n_J_2 \) is the number of half joints. An unconstrained body in planar space has three degrees-of-freedom (DOF). Full joints, such as revolute and prismatic joints, allow movement in one direction, thereby restricting two DOF. Half joints allow movement in two directions and restrict only one. In this case \( n_L = 5 \), \( n_J_1 = 5 \), and \( n_J_2 = 0 \), so the mobility of the Crazy Couch is \( M = 2 - DOF \). As expected, a 5-bar mechanism requires two drivers in order to have fully defined motion. In this case, the two required inputs are the constant angular velocities \( \phi_2 \) and \( \phi_3 \) as shown in Figure 2.

2.2 Kinematic Equations

In accordance with Haug’s method [2], the motion of the mechanism can be described by a set of generalized coordinates that represent the global positions and attitudes of the moving links. The generalized coordinates for this mechanism are defined as follows:

\[ \{q_i\} = \left\{ \begin{array}{c} r_1^T \\ \phi_i \end{array} \right\} = \left\{ \begin{array}{c} x_i \\ y_i \\ \phi_i \end{array} \right\} \quad i \in \{2,3,4,5\} \]  

\[ \{q\} = \left\{ \begin{array}{c} q_2^T \\ q_3^T \\ q_4^T \\ q_5^T \end{array} \right\} \quad \]  

In total, the mechanism has twelve generalized coordinates.

Figure 3 shows local body-fixed coordinate frames that are attached to each of the links. These local coordinate frames allow the positions of various points on the links to be described in terms of invariant distances; they stay the same for all time, no matter the position or orientation of the body. The orientations angles \( \phi_i \) of the local frames with respect to the global directions are measured the counter-clockwise (CCW) from the global X axis to the local \( x_i \) axes. For convenience and simplicity, the origins of the local coordinate frames are placed at the centers of revolute joints.

In Figure 3(b), the points \( R_4 \) and \( S_4 \) on link 4 are collinear with the translational path of link 4 along the length of link 3. In Figure 3(d), the points \( P_3 \) and \( Q_4 \) on link 3 are also collinear with the translational path of link 4 along the length of link 3. These points are necessary for writing vectors and constraints that enforce the prismatic joint, as will be described below. The local positions of all the points on the links in Figure 3 are necessary pieces of information about the geometry of the mechanism to be able to perform kinematic analysis. Specific values of these geometries will not be given in this paper per a non-disclosure agreement between the author and Skyline Attractions, LLC.
The mechanism is 2-DOF and has twelve generalized coordinates, so, per Haug [2], a total of twelve constraints are needed to enforce the constrained motion of the mechanism. Specifically, ten kinematic constraints are needed to limit the mobility of the mechanism to 2-DOF, and then two driving constraints are needed to fully define its motion. Algebraic position and driving constraints presented in Haug [2] are written in Eq. (4):

\[
\{\Phi\} = \begin{bmatrix}
\{\Phi\}_{\text{REV,A}} \\
\{\Phi\}_{\text{REV,B}} \\
\{\Phi\}_{\text{REV,C}} \\
\{\Phi\}_{\text{REV,D}} \\
\{\Phi\}_{\text{PARALLEL}} \\
\{\Phi\}_{\text{PIN SLOT}} \\
\Phi_{\text{DRIVER,A}} \\
\Phi_{\text{DRIVER,D}}
\end{bmatrix} = \begin{bmatrix}
\{r_1\}_A - \{r_1\}_A \\
\{r_1\}_B - \{r_1\}_B \\
\{r_3\}_C - \{r_3\}_C \\
\{r_3\}_D - \{r_3\}_D \\
\{a_3\}_1^T \{R\}^T \{a_4\}_1 \\
\{a_3\}_1^T \{R\}^T \{d_{34}\} \\
\phi_2 - \phi_{2,\text{START}} - \omega_2 t \\
\phi_5 - \phi_{5,\text{START}} - \omega_5 t
\end{bmatrix} = 0_{12 \times 1} \tag{4}
\]

Rows one through eight of the 12 x 1 constraint vector \(\{\Phi\}\) are revolute constraints that enforce the pin joints at Points A, B, C and D. The expressions of the form \(\{r_j\}_B - \{r_j\}_B = \{0_{2x1}\}\) state that global position of point “P” on two separate bodies “j” and “i” must be coincident at all times. The next two rows enforce the prismatic joint between link 3 (seat span) and link 4 (slider trolley) by stating that the vectors \(\{a_3\}_1\) and \(\{d_{34}\}_i\) must always remain parallel to the vector \(\{a_3\}_1\) (see Figure 3). The term \(\{R\}\) is a 2 x 2, 90-degree CCW rotator matrix. The last two rows are relative angle driver constraints that apply constant speed rotation to the cranks, where link 2 (first crank) is driven from Point A at constant angular velocity \(\omega_2\) and link 5 (second crank) is driven from Point D at constant angular velocity \(\omega_5\). CCW is taken to be the positive direction.

These generalized coordinates \(\{q\}\) and constraints \(\{\Phi\}\) are then used to determine the position of each link in the global reference frame. Due to the nonlinearity of position equations, a Newton-Raphson algorithm Eq. (5) is employed to solve for updated values of the generalized coordinates at a new time step \(k+1\) based on initial guesses of \(\{q\}\) taken at the current time step \(k\):

\[
\{q\}_{k+1} = \{q\}_k + \{\Delta q\}_k = \{q\}_k - \{\Phi\}_{k}^{-1} \{\Phi\}_{k} \tag{5}
\]

The term \(\{\Phi\}_k\) is a matrix of partial derivatives of the constraints taken with respect to the generalized coordinates known as the Jacobian matrix, as shown in Eq. (6):

\[
\{\Phi\}_q = \left[ \frac{\partial \{\Phi\}_i}{\partial q_j} \right], \quad i \in \{1,2,\ldots, nc\}, \quad j \in \{1,2,\ldots, nq\} \tag{6}
\]

where nc is the number of constraints and nq is the number of generalized coordinates. For the Crazy Couch, the Jacobian is a 12 x 12 matrix. The Jacobians terms for planar revolute and prismatic joints and the relative angle drivers can be found in Chapter 3 of [2].

To start the Newton-Raphson algorithm Eq. (5), initial global positions and orientations of the links are estimated from the initial pose of the Crazy Couch shown in Figure 4, where the seat span is horizontal with respect to the ground and at its
lowest position relative to Points A and D. It is in this position that the ride starts and ends and in which riders load onto and disembark from the ride.

FIGURE 4: INITIAL POSE OF THE MECHANISM.

With the position solution obtained from Eq. (5), velocity and acceleration solutions are straightforward to obtain thereafter and can be calculated during each time step. The velocity solution presented by Haug [2] is given by Eq. (7):

\[ \{ \Phi_q \} = \{ v \} \text{ for } \{ v \} = -\{ \Phi_t \} \]

(7)

where \( \{ q \} \) are the generalized velocities, \( \{ v \} \) is the velocity right-hand side (RHS) and \( \{ \Phi_t \} \) is the first partial of the constraint vector with respect to time. For scleronomic constraints that are independent of time, such as mechanical joints, the velocity RHS is zero. The acceleration solution presented by Haug [2] is given by Eq. (8):

\[ \{ \Phi_q \} = \{ \gamma \} \text{ for } \{ \gamma \} = -\{ \Phi_t \} \frac{\partial^2 \{ q \}}{\partial t^2} - 2\{ \Phi_q \} \frac{\partial \{ q \}}{\partial t} - \{ \Phi_{tt} \} \]

(8)

where \( \{ q \} \) are the generalized accelerations and \( \{ \gamma \} \) is the acceleration RHS, which includes second partials derivatives of the constraint vector with respect to the generalized coordinates and time. For constrained multibody systems like the Crazy Couch with only scleronomic constraints, all second partials of the kinematic constraints are zero and Eq. (8) reduces to

\[ \{ \Phi_q \} = \{ \gamma \} \text{ for } \{ \gamma \} = -\{ \Phi_t \} \frac{\partial^2 \{ q \}}{\partial t^2} \]

(9)

The velocity RHS’s and acceleration RHS’s presented by Haug [2] for the joints and drivers employed in the Crazy Couch are shown in Eq. (10) and Eq. (11):

\[
\begin{align*}
\{ v \}_{\text{REV}} & = \{ 0_{2 \times 1} \} \\
\{ v \}_{\text{PARALLEL}} & = 0 \\
\{ v \}_{\text{PIN SLOT}} & = 0 \\
\{ v \}_{\text{DRIVER}} & = \omega \\
\{ \gamma \}_{\text{REV}} & = \dot{\phi}_1^T \{ A_1 \} \{ \dot{q}_1 \} + \dot{\phi}_2^T \{ A_2 \} \{ \dot{q}_1 \} \\
\{ \gamma \}_{\text{PARALLEL}} & = 0 \\
\{ \gamma \}_{\text{PIN SLOT}} & = \phi_1^T \{ 2 \dot{\phi}_1 \ { \dot{q}_1 } \} + R^T \left[ \phi_1^T \ { \dot{q}_1 } + \{ R \} \{ \gamma \}_{\text{REV}} \right] \\
\{ \gamma \}_{\text{DRIVER}} & = 0
\end{align*}
\]

(10a) \hspace{1cm} (10b) \hspace{1cm} (10c) \hspace{1cm} (10d) \hspace{1cm} (11a) \hspace{1cm} (11b) \hspace{1cm} (11c) \hspace{1cm} (11d)

The terms \( [A_i] \) and \( [A_j] \) are orthonormal rotation matrices that describe the attitudes of generic bodies “j” and “i”, which are used to convert information described in local coordinates to global coordinates and vice versa.

2.3 MATLAB Algorithm

Position, velocity, and acceleration analyses are carried out in MATLAB over one full cycle of the mechanism in mid-operation (neglecting start up and slow down). Since the cranks link 2 and link 5 rotate at different speeds, a full cycle can be defined as the time or the number of rotations of each crank it takes for the mechanism to return to its starting pose as shown in Figure 4. Nominally, the Crazy Couch operates so that the cranks rotate at a relative speed ratio of 5:4, meaning that for every five revolutions of link 2 there are four revolutions of link 5 in a single cycle. Then the cycle duration, denoted herein as \( t_{\text{cycle}} \), can be calculated using Eq. (12):

\[
t_{\text{cycle}} = \frac{2\pi}{\omega_{\text{crank}}} \times (\text{number of crank revs per cycle})
\]

(12)

where either of the crank speeds (in radians per second) and corresponding number of revolutions per cycle can be used.

An explicit time loop is utilized to perform the position, velocity, and acceleration analyses described by Eq. (2) through Eq. (11) at each time step starting from \( t = 0 \). A number of time steps per second, \( n \), is defined and the time step size, \( \Delta t \), is calculated as \( 1/n \). Therefore, the total number of time steps is \( t_{\text{cycle}}/\Delta t \) (rounded up to the next whole number if necessary).

At each time step, the global orientations of the cranks, \( \phi_2 \) and \( \phi_5 \), are updated based on the current time, \( \omega_2 \) and \( \omega_5 \). Global positions and orientations from the last time step are used as initial estimates for the positions at the current time step. The numerical convergence criteria for the Newton-Raphson algorithm Eq. (5) is when the constraints become sufficiently close to zero. In this case, Eq. (5) is iterated until \( \max(\text{abs}(\{ \Phi \})) < \varepsilon \), where \( \varepsilon \) is an assembly tolerance. An assembly tolerance of \( \varepsilon = 10^{-5} \) cm is chosen given the scale of the ride. The data from the kinematic analyses are stored and the time step is updated.

2.4 Comparison Models

In order to verify the accuracy of the mathematical model developed in Sections 2.2 and 2.3, two other models of the Crazy Couch were created using the commercial simulation packages Working Model 2D [4] and SolidWorks [5] with Motion Analysis add-on. Figure 5 shows the model created in Working Model and Figure 6 shows the model created in SolidWorks.

Working Model enforces constraints between rigid bodies in manner much like the method presented by Haug [2]. However, instead of performing exact kinematics when carrying out simulations, Working Model utilizes forward time integration in which dynamic forces are used to predict resulting motion. Forward time integration can result in
solutions “blowing up” if a mechanism is not well behaved or if the time step size is not chosen sufficiently small. Therefore, a time step of $\Delta t = 0.002$ seconds, or $n = 500$ steps per second, was chosen for all three of the simulations (MATLAB, Working Model, SolidWorks) for the verification step.

The basic implementation of SolidWorks motion analysis also performs exact kinematics when a kinematic driver such as a constant speed rotational driver is chosen.

Simulations were carried out in all three programs using identical geometries, initial positions, driving speeds, and time steps as mentioned previously.

3 RESULTS

3.1 Model Verification
For the simulations, $\omega_2$ and $\omega_5$ were set to the nominal operating speeds of 21.75 RPM and 17.4 RPM, respectively, which are based on the standard RPM of the Crazy Couch’s AC motor and reduction ratios, thus yielding the appropriate 5:4 relative speed ratio. Both $\omega_2$ and $\omega_5$ were set clockwise (CW). The rotational kinematics of the seat span (link 3) were chosen as the metric for comparison because this link goes through the most complicated motion. These rotational kinematic metrics includes the attitude angle of the seat span $\phi_3$ and its first and second derivatives, $\dot{\phi}_3$ and $\ddot{\phi}_3$.

Data from the Working Model and SolidWorks simulations were collected and plotted against the corresponding solution obtained from the MATLAB implementation of the Haug model. Data from Working Model is compared with the MATLAB solution in Figure 7. Data from SolidWorks is compared with the MATLAB solution in Figure 8. There is clear agreement between the solutions, with the dashed lines (Working Model) and dotted lines (SolidWorks) following the solid lines (MATLAB) without any appreciable difference. Also, the results indicate motion that is consistent with the

![FIGURE 5: WORKING MODEL 2D CRAZY COUCH MODEL](image5)

![FIGURE 6: SOLIDWORKS CRAZY COUCH MODEL](image6)

![FIGURE 7: ROTATIONAL KINEMATICS OF THE SEAT SPAN OVER A FULL CYCLE – HAUG’S METHOD (SOLID) COMPARED WITH WORKING MODEL 2D (DASHED).](image7)

![FIGURE 8: ROTATIONAL KINEMATICS OF THE SEAT SPAN OVER A FULL CYCLE – HAUG’S METHOD (SOLID) COMPARED WITH SOLIDWORKS (DOTTED).](image8)
real mechanism. There are five peaks corresponding to the five revolutions of link 2 in one cycle, with the largest rotations, angular velocities, and angular accelerations occurring when the two cranks are out of phase and the smallest occurring when cranks become in phase.

The errors between these solutions are plotted in Figures 9 and 10. The overall error between the Working Model simulation and the MATLAB solution is greater than the error between the SolidWorks simulation and the MATLAB solution. This can be primarily attributed to reasons mentioned in Section 2.4, where Working Model performs forward time integration as opposed to exact kinematics.

As Figure 9 shows, one effect of numerical integration is that solution accuracy decreases every time a derivative is taken and when the function being integrated changes rapidly over time. In Figure 7, the most rapid change in acceleration occurs around the zero crossing after the third peak at around 7.5 seconds into the cycle. Observation of Figure 9 shows that this is also when the largest error between the acceleration solutions occurs. Furthermore, one other possibility for the relatively large error—order of $10^{-4}$—is because the translational path of link 4 along link 3 was made collinear with the line connecting the revolute joints at Point C in the Working Model simulation (compare Figures 2 and 5). As mentioned previously in Section 2.1, the offset of the prismatic joint (or lack thereof) makes a difference in dynamic analysis. It is expected that recreating the Working Model simulation with the exact offset prismatic joint geometry as the MATLAB and SolidWorks models would significantly decrease the relative solution error. However, errors on the order of $10^{-4}$ is still very small given the scale of the mechanism and the solutions can be considered near-exact matches for practical purposes.

The error between the SolidWorks and MATLAB position, velocity, and acceleration solutions, shown in Figure 10, jumps sporadically around values on the order of $10^{-10}$, which is most likely only that large because of rounding differences incurred when entering numerical geometry data into the two programs. Because the error between the SolidWorks and MATLAB solutions is so small, it can be inferred that plotting the error between the Working Model and SolidWorks simulations would yield a similar characteristic to the one shown in Figure 9.

These results prove that the mathematical model developed following Haug [2] accurately predicts the theoretical motions of the moving bodies that comprise the Crazy Couch.

### 3.2 Rider Experience: Motion Profiles and Accelerations

The mathematical model can be used to learn pertinent information about the kinematics of the amusement ride. First and foremost, it should be noted that the kinematic analysis methods of Haug [2] provide a straightforward link to both inverse and forward dynamics. In this case, the Crazy Couch is treated as a kinematically driven problem; finding the Cartesian and rotational accelerations of the centers of the gravity of all the moving links, combined with knowing each of their mass and inertia properties, would allow an engineer to perform inverse dynamics in order to find joint reaction forces and the torques required to drive the two cranks. Dynamic analysis was not a focus of this paper, but it is certainly a prospect for future work.

However, another critical element in the design of an amusement ride is the experience of the riders: to what types of motions and accelerations will the riders be subjected, and are those motions within safe limits? Figure 11 shows a front view of the Crazy Couch. Points of interest are marked by red circles.
and local coordinate systems indicating their global positions in the XY plane defined in Figure 2. These points represent the approximate locations of a rider (chest to head level, depending on height) when they sit in the left end seat (Point L), either of the two middle seats (Point M), or the right end seat (Point R) on the seat span.

![FIGURE 11: THREE RIDER LOCATIONS ON THE SEAT SPAN CHOSEN FOR ANALYSIS: LEFT END SEAT (L), MIDDLE SEATS (M), AND RIGHT END SEATS (R).](image)

The motion profiles experienced by riders sitting in these seats can be obtained by plotting the global positions of Points L, M, and R over a full cycle of the mechanism. This is shown in Figure 12, where the black circles indicate the coincident starting and ending positions of Points L, M, and R. Point L follows the blue path (left), Point M follows the red path (middle), and Point R follows the yellow path (right). The CW circular paths followed by the cranks around their ground connection points, A and D, are plotted for reference.

![FIGURE 12: MOTION PROFILES OF POINTS L, M, AND R.](image)

As one moves from left to right on the seat span (facing the ride from the front as in Figure 11), the motion experienced varies. In the left end seat (Point L), the rider will essentially move in a clockwise circular path that follows first crank (link 2) and gets stretched and warped due to the back-and-forth tilting of the seat span. Looking at the middle seats (Point M), the overall motion envelope decreases, but the rider will start to experience direction changes that result from the sliding of the seat span over the second crank (link 5) as the two cranks rotate at different speeds. Moving all the way to the right end seat (Point R), the range of motion increases, somewhat combining and intensifying the vertical stretch seen in motion profile of Point L and the direction changes in the motion profile of Point M. The question then becomes, what does this mean for sensations felt by the riders? In other words, what accelerations are they subjected to?

ASTM F 2291 on the “Standard Practice for Design of Amusement Rides and Devices” [6] establishes strict limitations on allowable accelerations that patrons can be subjected to in a given plane of motion relative to anatomical directions of the human body. Taking definitions from [6], the Crazy Couch moves riders in the human body's coronal (YZ) plane, with the left and right lateral directions in the Y the cranial and caudal directions in the Z. Figure 13 shows a standard graphical tool from [6] that is used to visualize allowable accelerations in this plane (accelerations in g's).

![FIGURE 13: ALLOWABLE COMBINED MAGNITUDES OF Y (HORIZONTAL) AND Z (VERTICAL) ACCELERATIONS [6].](image)

Any component of total acceleration experience by riders of the Crazy Couch need to meet individual axis requirements, and
total combined acceleration magnitude must fall inside the so-called “acceleration egg”.

Because the Crazy Couch is a “family-oriented” ride—not a high-speed thrill ride like a roller coaster, a 300+ ft. drop tower, or a giant pendulum ride—it is expected that the horizontal and vertical accelerations will be relatively small, but noticeable enough to cause pleasant sensations that ultimately make the ride fun for amusement park patrons. These accelerations are sometimes referred to colloquially as “g-forces” when the quantities are normalized by the gravitational acceleration constant 9.81 m/s^2 (32.2 ft/s^2).

Horizontal accelerations felt by riders, as shown in Figure 14, are relatively mild on the Crazy Couch, with a peak value of about 0.4 g’s to the right occurring at only one instant in the right end seat (Point R). Accelerations seldom exceed 0.3 g’s in either direction for any of the points of interest. There is also little variation from left to right on the seat span; the acceleration profiles of all the points follow roughly sinusoidal paths with similar periodicities and amplitudes. What this translates to in terms of physical sensations is a gentle left-to-right rocking/swaying felt in all the seats, which is unlikely to cause any discomfort to riders, perhaps aside from mild nausea in those who are prone to motion sickness.

Vertical accelerations felt by riders is a different story, as can be seen in Figure 15. Where one sits on the seat span could make a difference in the intensity of “heaviness” and “losing one’s stomach” sensations felt. For example, a rider sitting in the left end seat will experience a near-harmonic variation in the perception of their weight similar to what one may feel while swinging on a swing set, with peaks regularly reaching 0.2 to 0.3 g’s above and below 1 g. This is consistent with the oblong oval motion path of Point L in Figure 12.

Riders sitting in the middle seat of the Crazy Couch (Point M) will feel weaker vertical accelerations overall (magnitudes less than 0.2 g above and below 1 g), but the variations from positive to negative may feel less “planned” and more “random”, as evidenced from the clear deviation from a simple sinusoidal profile. This is consistent with the smaller motion envelope and direction changes of Point M in Figure 12.

As the motion profile of Point R seems to combine and intensify those of Points L and M, so goes the acceleration. A rider sitting in the right end seat will encounter the most drastic bouncing, with two moments of noticeable heaviness where accelerations peak at just below 1.4 g’s and three moments of noticeable lightness with acceleration dipping below 0.6 g’s at one instant around 7.5 seconds into the cycle.

It is interesting and intuitive to note that motion and acceleration profiles of seats closer to the first crank (link 2) have five distinctive peaks within one cycle, whereas those of the seats closer to the second crank (link 5) seem to have four distinctive peaks within one cycle. This is a direct effect of the 5:4 speed ratio of link 2 to link 5. It also clear from Figures 14 and 15 that accelerations experienced by riders fall well within the margins imposed by ASTM 2291 F [6].

Although excitement on amusement rides is entirely subjective, and while the Crazy Couch is a milder ride in general, someone looking for the most intense forces and “quirky” bouncing motions may wish to sit in the right end seat, which sits to the outside of the sliding joint. Perhaps the greater takeaway is that each rider will have a slightly different experience from the riders he or she sits next to. Furthermore,
this means that one can ride the Crazy Couch multiple times and get a different experience each time.

### 3.3 Model Visualization

A short program was written in MATLAB to animate a visual representation of the Crazy Couch 5-bar mechanism as the ride moves through one full cycle. Real-time visualization of a mechanism can be a helpful design tool. The built-in function ‘VideoWriter’ is also used to create a video file of the animation that can be played back. At each time step, the positions of reference points and outlines of the links are plotted on the same figure, written as a single frame into the video file, and then updated at each subsequent time step. This program follows the basic structure shown below. A sample frame of the output is shown in Figure 16, which also includes real-time tracing of the rotational kinematics of the seat span.

```matlab
1 % Set up video file
2 v = VideoWriter('cc.avi');
3 v.FrameRate = fr; % set frame rate
4 open(v);
5 % sequence through positions
6 for i = 1 : length(t_cycle)
7 % plot all points on links
8 M = getframe(gcf);
9 writeVideo(v,M);
10 end
11 close(v);
```

![Crazy Couch Animation](image)

**FIGURE 16: SCREENSHOT OF A MATLAB ANIMATION OF THE CRAZY COUCH AND SEAT SPAN ROTATIONAL KINEMATICS.**

### 3.4 Sensitivity Study

The Crazy Couch is currently manufactured under one set of specifications in terms of link geometries, input motor speed, and relative crank speeds. While modifying the design from its current form may or may not be economical, an obvious question stemming from a standpoint of pure curiosity is, what types of different motions can be obtained by changing certain parameters of the mechanism? In this paper, the parameters chosen for variation are the speeds of the two cranks \( \omega_2 \) and \( \omega_5 \) and, therefore, the ratio between them, \( \omega_2 / \omega_5 \). This is achieved by assuming a constant operating speed of the input AC motor and simply adjusting the reduction ratios of the reducer gear boxes that connect to the cranks. The author realizes that adjusting gear ratios in the mechanism can also change torque and power requirements, but it certainly the simplest modification from a design standpoint and is likely to be less costly than changing entire geometries of the moving links, which would require full redesigns of numerous parts of the mechanism and the supporting structure.

Reduction ratios for each of the two reduction gear boxes are determined based on the capabilities of the manufacturer of the current reducer model employed on the Crazy Couch. Furthermore, reduction ratios are also chosen to ensure that the ratio of the crank speeds \( \omega_2 / \omega_5 \) is always a rational number, i.e., a fraction of two integers. This prevents the possibility of impossibly long ride cycle durations.

To show the effects of this parameter variation in a succinct way, the maximum acceleration magnitudes of different points on the links that are achieved during one cycle of the mechanism are plotted as functions of \( \omega_2 \) and \( \omega_5 \). From this it is possible to see whether accelerations at rider locations such as Points L, M, and R remain within the allowable ranges set forth by ASTM F 2291 [6]. Analyzing the acceleration magnitudes at the joints (Points A, B, C, D, and the prismatic joint) would additionally help determine whether the crank speeds are appropriately safe for a given construction of the mechanism—dynamic effects on the mechanism itself should also be considered.

The resulting ride cycle durations \( t_{cycle} \) are also plotted as functions of \( \omega_2 \) and \( \omega_5 \) in order to determine whether the selected gear ratios would yield a practical ride duration. If the cycle is too short, then the ride becomes repetitive. If the cycle is too long, then the ride could be hampered down by low ridership capacity.

Sample results taken at Point R (right end seat) are presented in Figure 17 for an input speed of 870 RPM. Maximum acceleration magnitudes are plotted in Figure 17(a) and the corresponding ride cycle durations are plotted in Figure 17(b). The reduction ratio applied to link 2, which produces the output speed \( \omega_2 \), is the abscissa and the resulting acceleration and \( t_{cycle} \) are the ordinates. Different curves are produced by varying the value of the reduction ratio applied to link 5, which produces the output speed \( \omega_5 \). Circled in red is the nominal operating point of the Crazy Couch, where reduction ratios of 40:1 to link 2 and 50:1 to link 5 result in \( \omega_2 = 21.75 \) RPM and \( \omega_5 = 17.4 \) RPM, respectively. With these parameters, the
maximum acceleration magnitude felt by a rider in the right end seat is just under 1.5 g’s and the ride cycle duration is just under 14 seconds.

It can be seen that under the right combinations of gear ratios, the maximum accelerations and ride cycle durations can be increased or decreased dramatically. As expected, choosing lower gear ratios for both reducers results in much faster crank speeds, which then increase the accelerations felt by the riders. Ride cycle duration shoots up when the gear ratios are slightly mismatched and drops significantly when the reduction ratios are integer multiples of one another.

CONCLUSIONS

Amusement rides often incorporate various translating and rotating parts connected through mechanical joints and linkages in order to produce exciting motions for park patrons. As such, they are constrained multibody systems. The kinematic analysis method of Haug [2] is powerful tool that can be used analyze and simulate such amusement rides and devices. In this paper, Haug’s method was used to develop a model and simulate the motion of an amusement ride known as the Crazy Couch. The accuracy of the mathematical model of the Crazy Couch was verified against simulations carried out in two different commercial simulation packages, and results were analyzed to describe rider experience on the Crazy Couch. A real-time visualization of the model was created using MATLAB animation techniques and a sensitivity study was performed, both of which can serve as design tools.

The versatility of Haug’s method allows for easy transition from kinematics into inverse and forward dynamics. The lessons learned from this report could be easily applied a dynamic analysis of joint reaction forces and required crank torques for the Crazy Couch.

ACKNOWLEDGMENTS

The author thanks Dr. H.J. Sommer III, Professor of Mechanical Engineering at Penn State, for his education and guidance; Skyline Attractions, LLC for their support of this project; and Fun Spot America in Kissimmee, FL, for allowing filming and photography of the Crazy Couch in operation from a restricted area.

REFERENCES

**ME 581: Simulation of Mechanical Systems**  
**Grand Piano Action Model**

**David Pepley**

1. **Introduction**

The basic operation of the grand piano (Figure 1) has not changed much since its invention by Bartolomelo Cristofori in the year 1700 [1]. The instrument operates by pressing a key, which transmits motion through the piano action, causing a hammer to strike a wire or string. This vibrating string then sounds a musical note based on the gauge, length, and number of wires. The goal of this project was to design and build a working physical model of a grand piano action mechanism and to understand how the mechanism operates.

![Figure 1: Baby grand piano][2]

2. **Methodology**

2.1. **Model Development**

The first step in building the model was to design the grand piano action in CAD using Solidworks. Not having access to a grand piano made determining the dimensions of the parts difficult, however a set of one to one scale 2D sketches was found online to use as a starting point for the model [3]. The 3D CAD model in figure 2 consists of six main parts: the wippen, the jack, the repetition lever, the hammer, the hammer knuckle, and the damper lever. Various small parts were also modeled to serve as anchoring points for the action.

![Figure 2: CAD model of the grand piano action mechanism][2]

The largest part of the action is the wippen seen in figure 3. The wippen is attached to a fix backing via a revolute and is rotated when it is pushed by the piano key. The jack and repetition lever are also attached to the wippen via revolutes and springs.

![Figure 3: CAD model of the wippen and rotational axes for the backboard anchor, repetition lever, and jack][2]

The next part modeled was the jack. The jack serves three purposes: it presses against the knuckle contacting the hammer, it stops the rotation of the wippen, and it uses a spring to quickly reset the position of the wippen after a key is released. The jack has a simple “L” shape and is attached to the end of the wippen. When the key is pressed, the wippen and jack raise until the end of the jack contacts a stopper which stops the wippen and slightly rotates the jack. Further rotation of the jack...
is prevented by its slotting into a gap in the repetition lever. A spring is connected between the wippen and the jack to pull the jack back into position when the key is released and spring the wippen back into its starting position.

The third part is the repetition lever. This lever is used to quickly reset the hammer knuckle on top of the jack. This in turn allows the key to be pressed and again without fully resetting to its resting position. When the hammer rebounds from the string, the repetition lever catches the hammer knuckle. A spring between the repetition lever and the wippen damps the impact of the hammer knuckle and forces the knuckle back on top of the jack. Without the repetition lever, the key would need to be fully released after every key press to play a note.

The fourth and fifth parts are the hammer and hammer knuckle. The hammer is thrown upward by the motion of the wippen and the jack cause it to strike a string and play a note. The hammer is attached to the backboard anchor using a revolute. Between the jack and the hammer is the hammer knuckle. This is a roller that serves to transmit the upward motion of the jack to the hammer.

The final part is the damper lever. The damper lever is separate from the wippen assembly and is used to raise a damper from the string to be played. When a key is pressed, the damper lever, attached to the backboard anchor via a revolute, is pressed upward by the end of the key. The lever is attached to a rod in a collar. This rod then slides upward through the collar, raising a damper off the piano string. When the key is released, the damper lowers muting the vibrating string. The damper can also be raised through the use of a foot pedal, but this is separate from the piano action.

2.2. Physical Model

The finished CAD models were then exported to STL files and 3D printed using ABS plastic and a Makerbot Replicator 2X. Through-holes on the parts were brought to their finished diameter using a drill. The key, damper, backboard, damper lever collar, and other fixtures were cut from clear acrylic sheets. The damper lever rod is made out of a steel rod and the string is simulated by a similar steel rod. Finally, the hammer head was fashioned from a soft eraser and the hammer knuckle was created using the head of a plastic bolt. The finished model can be seen in figure 4.

Finding appropriate springs for the repetition lever and the jack was difficult, however it was found that several small rubber bands acted perfectly for this function. By changing the number of rubber bands, the strength of the spring force could be altered until ideal motion was achieved. Nuts and washers were also used to properly space the model so that it would not rub against the backboard and to ensure proper alignment of the parts.

3. Results

Overall, the physical model functions very well. When the key is pressed, the hammer is rapidly launched towards the string. The damper also raises appropriately whenever the key is pressed. Using the rubber bands as springs worked extremely well. Good piano keys have a responsive “bounce” or rebound when pressed.
This is caused by properly calibrated springs on the jack, and this effect is well emulated by my model. The key is able to be played quickly because of a properly functioning repetition lever.

4. Conclusion

A working grand piano action mechanism model was constructed. The 3D printed parts were effective and motion is effectively transmitted from the key to the hammer and damper. Several challenges were encountered during the project such as proper part alignment, but this was alleviated through the use of nuts and washers. Overall, the project was a success.

References

ME 581: Simulation of Mechanical Systems – Spring 2017
Robots for Use at the Fukushima Daiichi Nuclear Power Plant Accident Site

Jacob Rosnack

Abstract

The Fukushima Daiichi nuclear power plant accident requires a seemingly insurmountable cleanup task to be completed. While impossible by human means alone, robots are being used to survey and cleanup the remains at Fukushima Daiichi. This paper provides background on the accident, information on several different robots used to meet various needs in the Fukushima Daiichi cleanup effort, and challenges faced in the design and implementation of these robots. This study concludes that safety is the highest priority in nuclear engineering and that robot designers must use strong technical rigor to create products capable of completing tasks that humans cannot.

1. Fukushima Daiichi Accident Background

1.1. Natural Disasters

The nuclear accident at the Fukushima Daiichi nuclear power plant, one of the worst nuclear accidents in history, second only to the Chernobyl nuclear accident, was caused by the conglomeration of two natural disasters of epic proportions. At 2:46 p.m. on Friday, March 11, 2011 (local Japanese time), Japan experienced the largest earthquake in its recorded history, a Magnitude 9.0 off the Sanriku coast [Ref. 1]. About one hour later, the earthquake resulted in a tsunami, roughly 15 meters in height at its largest, which inundated the Fukushima Daiichi site; something that the staff considered to be impossible with a 10-meter safety wall in place [Ref. 2]. Before the nuclear accident occurred, this combination of natural disasters left nearly 20,000 dead and 2,500 missing in its wake [Ref. 3].

1.2. Aside on Fossil vs. Nuclear Plants

It is important to note the fundamental differences between typical fossil-fueled power plants and nuclear power plants to understand the significance of such a catastrophic event. The major differences between nuclear power plants and conventional fossil-fueled power plants are the amount of energy in the system and the length of time required to shut down to levels capable of being cooled by natural circulation.

For instance, the thermal power output of Fukushima Daiichi Units 1, 2, and 3 are 1380 MWt (460 MWe), 2350 MWt (784 MWe), and 2350 MWt (784 MWe), respectively [Ref. 4] and need only be refueled once every two years. For reference, a typical natural gas-fired power plant has a thermal power output of 1275 MWt (425 MWe, similar to Fukushima Daiichi Unit 1), but must be supplied a constant stream of natural gas (i.e. refueled continuously).

Additionally, although the nuclear fission chain reaction can be shut down immediately, the heat generated by radioactive decay takes on the time scale of weeks (depending on the capacity of the plant) to dissipate to levels cooled by natural circulation. In contrast, all energy is removed from the system in fossil-fueled plants immediately after the fuel supply is cut off.

Due to these major differences, nuclear power plants need higher capacity (around 12,000 gpm), higher redundancy (around eight pumps with associated heat exchangers, valves, pipes, etc.) cooling systems than their fossil-fueled counterparts. In addition, these cooling systems are required to be available for long periods of time after plant shutdown. To satisfy these requirements of high capacity, high redundancy, and long-term availability, high voltage electricity (4-13.8 kV) must be provided to operate these cooling systems.

1.3. Plant Response

The earthquake caused peak ground acceleration levels to exceed the prescribed seismic safety limit for the plant equipment, resulting in an immediate scram, or rapid shutdown of the nuclear fission
chain reaction, of the three operating reactors at Fukushima Daiichi. The other three units at the site were already shut down for routine maintenance and refueling [Ref. 2].

After the units scrammed, but prior to being struck by the tsunami wave, the associated Emergency Core Cooling System (ECCS) initiated to inject water into the reactor. However, as previously mentioned, the ECCS requires large amounts of power in order to operate. Because of the intensity of the earthquake, the entire grid was disrupted, with rolling blackouts across much of Japan. Nuclear power plants are designed to handle such an event (designated a Design Basis Event). Because the ECCS could not be powered by its usual offsite sources, emergency onsite sources had to be provided. This emergency onsite power came by virtue of the emergency diesel generators (EDGs). These EDGs are large diesel engines, typically rated to provide 4.16 kV electricity, capable of powering the plant in emergency situations for around one week, depending on the fuel capacity of the EDGs.

One hour after the safe shutdown of the three operating reactors at Fukushima Daiichi, the majority of the site was inundated by a 15-meter tsunami wave. This water flooded out nearly all of the EDGs, leaving the site in one of the direst situations known to the nuclear power industry, the Station Black Out (SBO). With all offsite power unavailable due to the earthquake and all onsite power unavailable due to the tsunami, the only systems left available were the steam-driven systems. However, power is required in order to operate the valves used to control the steam driven systems. Without the EDGs, this power would be provided only by batteries with a four to eight hour capacity, which could not be recharged without restoring the EDGs. So, the site only had about four to eight hours remaining before the batteries depleted and, subsequently, all control was lost.

Without any power, water could not be supplied in sufficient quantity in order to cool the decay heat of the three operating reactors. Conditions degraded rapidly, as the water in the reactors continued to vaporize, further pressurizing the system. As the water continued to boil, the nuclear fuel continued generating heat. In these superheated conditions, the steam reacted with the zirconium in the cladding of the nuclear fuel, which generated large amounts of hydrogen gas. This hydrogen gas continued to build up in the buildings which hold the nuclear reactors, called containment. The containment buildings continued to pressurize, nearing, and eventually exceeding, their design limits.

The onsite employees spent several grueling days with little rest attempting to combat these severe conditions. They went to any measures possible to try to save the reactors, including running into the reactor buildings to check conditions, manually operating the valves that normally required power, and pumping water into the reactors using fire trucks and helicopters [Ref. 2].

At this point, no radioactive material had been released to the environment. However, conditions were clearly degrading to the point in which a radioactive release was all but guaranteed. Although it was not yet confirmed, most of the nuclear fuel in Units 1-3 had experienced severe melting by this time. One last ditch attempt to save the reactors came after the decision was made to vent the containment buildings. This involved opening an airway outside of the containment buildings and knowingly releasing some radiation to the environment in order to depressurize the containment in hopes of preventing a large uncontrolled release via explosion. Through the valiant efforts of the Fukushima Daiichi crews, the accident was prevented from reaching its full catastrophic potential, but Units 1, 3, and 4 all experienced hydrogen explosions. Although Unit 4 was not operating at the time of the accident, the used nuclear fuel in its spent fuel pool experienced boil-off and hydrogen generation and subsequent explosion after losing all cooling functions.

1.4. Environmental Conditions

After the hydrogen explosions at Fukushima Daiichi, the radiation levels at the site boundaries increased. These radiation levels can be described by the dose that would be received in the given area. The biological effect of ionizing radiation, which is correlated to radiation levels, is called dose. Dose is measured in terms of the amount of
energy imparted into a recipient mass. The SI unit of dose is the Sievert (Sv), which corresponds to one joule per kilogram [Ref. 5].

In order to put the post-accident radiation levels into appropriate scale and context, a few examples of dose received in common activities will be illustrative. Each year, a typical person receives 2.4 mSv of dose from natural radiation. A round-trip flight from Tokyo to New York will earn each passenger a dose of 0.19 mSv. A chest X-ray imparts a dose of 6.9 mSv into the person receiving it [Ref. 1].

Immediately following the Fukushima Daiichi nuclear accident, the dose rates measured at the plant boundaries exceeded 10 mSv/h [Ref. 6]. In other words, for every hour spent near the site after the accident, an individual would expect to receive roughly four times the dose that a typical person experiences in one year. It was due to these severe conditions that the Japanese government evacuated a 20 km (14.4 mi) radius around Fukushima Daiichi and restricted access to the area to only those fighting the disaster [Ref. 6]. This restriction displaced roughly 185,000 Japanese civilians from their homes [Ref. 3]. These dose rates decreased exponentially, such that levels had been reduced by a factor of one-thousand in about two months. Today, the dose rates have decreased to approximately 1-2 μSv/h [Ref. 7]. This has resulted in the restricted areas being reduced, allowing some people to return to their homes, but some restrictions still remain.

Environmental conditions at the Fukushima Daiichi nuclear station have improved enough that workers can occupy the area without any protective clothing (i.e. face masks, respirators, etc.), but this is not the case inside the reactor buildings where the nuclear fuel melted. As of April 2015, dose rates were measured at nearly 10 Sv/h [Ref. 8] in the Unit 1 primary containment lower levels, corresponding to a lethal (99% mortality rate) dose of radiation [Ref. 1] transferred every hour.

Knowledge of the sheer toxic conditions that exist in the Unit 1 primary containment at Fukushima Daiichi, conditions that can be reasonably assumed as similar to Units 2, 3, and 4, begs the question of how this knowledge was obtained. Further still, it illuminates the difficulty of the task of cleaning up the decimated reactor buildings. Clearly, the data gathering and cleanup tasks are beyond human intervention, as humans could not survive the conditions. The answer lies in the use of remotely-operated robots to perform these tasks.

2. Robots for Surveillance & Cleanup

Several different robot designs have been developed for use at the Fukushima Daiichi nuclear plant accident site. Each robot was designed with a different purpose in mind, with a different mission to complete. These missions include tasks such as monitoring environmental conditions, collecting data, surveying destroyed area layouts, transporting debris, and many others. Details will be provided in this report on a few key robots used at Fukushima Daiichi, including the PMORPH, Shape- Changing Robot (SCR), MHI-MEISTeR, Swimming Robot, Crawling Robot, Rosemary, and Sakura.

2.1. PMORPH

The first robot discussed in this report is the one that was most recently developed for investigation into the most brutal conditions at the Fukushima Daiichi nuclear plant accident site. The PMORPH, jointly developed by the International Research Institute for Nuclear Decommissioning (IRID) and Toshiba, is a scorpion-like, self-propelled robot used to survey the interior of the damaged Unit 2 Primary Containment Vessel (PCV) [Ref. 9]. A photograph of this robot is seen in Figure 1.

![Figure 1: Photograph of PMORPH (Ref. 9)](image-url)
The mission for this robot was two-fold: (1) to collect data for the next generation of inspection robots and (2) to survey the Unit 2 PCV to determine if fuel melted through the bottom of the reactor to the pedestal on which it stands. A cutaway view of the area of the building that was surveyed by the PMORPH is seen in Figure 2 [Ref. 9].

The first step to ensuring that the PMORPH could meet its mission was to appropriately, exhaustively define the problem it was intended to solve. The inputs to this problem include the following [Ref. 9].

1. The robot must be capable of fitting on a 60-cm-wide rail surface.
2. The robot must be transformable so that it can fit underneath of smaller obstacles (approximately 30 mm height).
3. The robot must be robust enough to traverse obstacles, such as debris, floor grates, small changes in elevation, etc.
4. The robot must be capable of surviving a 10 Sv/h dose rate for a sufficient length of time to record meaningful data.
5. The robot must be equipped to monitor environmental conditions for data collection and estimation of remaining life in the robot.

As with most designed products, this design went through many iterations and tests before reaching the “final” product displayed in Figure 1. The term “final” is used, because it is expected that this design will be further refined, redesigned, rebuilt, etc. as more operational experience is gained and more data becomes available from the accident site.

This design is a single-loop, open-loop mechanism, whose members can be effectively considered as a moving base (or ground) with an attached two-link manipulator. The body consists of three links, one revolute joint, and a universal joint. Its motion is controlled through the use of tracks, similar to a tank, rather than wheels.

A 3-dimensional model rendering of the initial phase of the PMORPH can be seen in Figure 3. For a brief analysis on mobility, topology, and kinematics for the PMORPH, see Appendix A. In this configuration, the manipulator arm is fully retracted such that the height of the PMORPH is minimized, in order to fit underneath obstacles. As seen in Figure 3, the dimensions of the PMORPH are well within the bounds of the problem statement items 1 and 2.
The tracks of this robot are quite robust. The grooved teeth are designed to grip items (such as floor grating) to enable it to traverse rugged terrain. In addition, the robot is made of various metal components, with little use of plastic, such that it can survive small falls without sustaining significant damage. If the PMORPH does topple over on a particular obstacle, the manipulator arm rotates at the universal joint about an axis parallel to the floor to push the robot back onto its base tracks so that it can continue on its way. These satisfy the problem statement item 3.

The fourth problem statement input item is perhaps one of the most difficult issues to solve. The PMORPH is made from robust materials that will not rapidly degrade in a radiation field. However, the electronics used to control the robot and the camera used to record video of the conditions inside the PCV are more susceptible to the radiation. This problem is solved by building enough material around the susceptible components in order to shield them from the radiation, but without using an excess of material that would immobilize the PMORPH.

The primary mission for the PMORPH is met by satisfying the problem statement item 5. Attached to the front of the base of the PMORPH is the main camera for surveying the area in the direction of travel. Also attached to the PMORPH’s base is a thermocouple used to measure the temperature of the Unit 2 PCV and assure that cold shutdown conditions remain. A second camera is attached to the end of the manipulator arm to survey the area in any direction desired. Lastly, attached to the cable that connects to the base of the PMORPH is a dosimeter used to measure the radiation conditions. With temperature, radiation, and visual inspection information that would be otherwise impossible to collect by human means, the overall issue of cleaning up and decommissioning the Fukushima Daiichi nuclear plant accident site can be better understood. As a result, more tools, like these robots, can be developed to solve this problem.

2.2. Shape-Changing Robot

The second robot discussed in this report is an earlier iteration of the PMORPH, referred to simply as the Shape-Changing Robot. For brevity, it will be referred to as the SCR for this report. Hitachi-GE Nuclear Energy developed this robot for use in inspection of the Fukushima Daiichi Unit 1 PCV [Ref. 10].

The mission for the SCR was similar to that of the PMORPH: (1) to survey the Unit 1 primary containment to determine if fuel melted through the bottom of the reactor to the floor of the containment and (2) to measure the radiation levels in the primary containment with an equipped dosimeter [Ref. 8].

The problem inputs for the SCR were also similar to that of the PMORPH.

1. The robot must be capable of fitting within a pipe of diameter 100 mm (≈4 in).
2. The robot must be robust enough to traverse obstacles, such as debris, floor grates, small changes in elevation, etc.
3. The robot must be equipped to monitor radiation levels for data collection and estimation of robot lifespan.
4. The robot must be capable of surviving a 10 Sv/h dose rate for a sufficient length of time to record meaningful data.

This design is simple, yet effective. It is a single-loop, open-loop mechanism, consisting of three links and two revolute joints. Similar the PMORPH, its motion is controlled through the use of tracks.
In order to meet the first problem input, the body begins in a linear state, essentially in the shape of a right, circular cylinder, such that it can fit through the 100 mm penetration into the primary containment. A 3-dimensional model rendering of this phase of the SCR and an associated cross sectional view of the penetration pipe can be seen in Figure 4.

As seen in Figure 4, the initial phase of the SCR is approximately 95 mm in diameter and 700 mm (27.6 in) in length. It travels through the pipe on its treads until reaching the Unit 1 PCV [Ref. 8]. This satisfies the first problem constraint. The mobility, topology, and kinematics may also be analyzed using similar analyses as those presented in Appendix A.

Once inside the containment, the second problem constraint becomes important. In its initial phase, the robot is unlikely to be capable of traversing any uneven or grated surface. At this point, the body folds at both revolute joints (to more closely resemble a tank) such that it has side-by-side tracks for more stability over uneven terrain. This second phase of the SCR is seen in Figure 5.

A camera with attached cable and a dosimeter are stored in the center section of the SCR. When the SCR sits upon grating, the camera/dosimeter detracts from the SCR, through the grating, to observe and record the conditions of the building below it. This camera is also rated to operate underwater, if necessary. This satisfies the third problem input and can be seen in Figure 6.
The fourth problem input is fulfilled in a similar fashion to the PMORPH, in which sensitive electronic components are shielded (to the extent possible) from the radiation, without using material in excess that would hamper the mobility of the SCR.

The SCR, while not being as refined as the PMORPH, demonstrated the viability of the design. It proved that robots of similar construction could operate in the intense environments within the PCVs at Fukushima Daiichi for a sufficient amount of time to collect data to better define the problem of cleanup.

2.3. MHI-MEISTeR

The third robot to be discussed in this report is the MHI-MEISTeR, or the Mitsubishi Heavy Industries Maintenance Equipment Integrated System of Telecontrol Robot. Rather than being used strictly for observation and massive data collection, this multi-functional robot is used for actual cleanup activities at the Fukushima Daiichi nuclear plant. A photograph of this robot is provided in Figure 7 [Ref. 11].

![Figure 7: MHI-MEISTeR Side View](image)

The MEISTeR has two arms, each with seven joints. It is approximately 1.25 m long by 0.7 m wide by 1.3 m tall (4.1’ x 2.3’ x 4.3”) and weighs 440 kg (970 lb). It moves at a rate of 2 km/h (1.24 mph) and can climb 40° slopes and steps of 220 mm (8.7”) [Ref. 11]. Similar to the PMORPH design, it moves using toothed treads rather than wheels. However, this robot has four freely moving tracks, rather than the two of the PMORPH. These separate sets of treads are what enable the MEISTeR to climb over obstacles with ease.

The MEISTeR’s design is an improvement over an earlier design that MHI developed for a different accident site in 1999 [Ref. 11]. It is intended for use in many tasks to include vacuuming debris, power blasting contaminated surfaces, drilling concrete for core samples, cutting handrails and piping, removing obstacles, performing repairs, and many others.

This large degree of versatility is the primary goal of the MEISTeR’s design. It can attach different tools to each of its arms in order to complete these various tasks, some of which may be performed simultaneously. For instance, the MEISTeR could grasp a pipe with one hand, cut it with the other, and then transport that pipe to a different area so that it was not in the way of further investigations [Ref. 11].

In fact, this robot was used for just such tasks in the Fukushima Daiichi Unit 2 reactor building. In March 2014, the MEISTeR was transported to the Unit 2 reactor building floor and, then, remotely operated to drill three separate concrete core samples, using a drill on one arm and a chisel on the other [Ref. 12]. Along the way, this larger robot cleared a pathway that can be used for future investigations into the Unit 2 PCV.

2.4. Other Robots

This section provides a brief synopsis on a few other robots used at the Fukushima Daiichi nuclear power plant. These robots include the swimming robot, crawling robot, Rosemary, and Sakura.

The first two of these robots, the swimming robot and crawling robot, were developed by Hitachi-
GE Nuclear Energy. They were used in tandem in the Torus (a.k.a. Suppression Chamber, see the donut-shaped object in Figure 2) inside Fukushima Daiichi Unit 2. The swimming robot (Figure 8), equipped with a camera and a clay-tracer, was used to inspect penetrations between the torus and the PCV and check for flow. The crawling robot (Figure 9) remained out of the water and measured the flow using an ultrasonic sonar system which targeted the clay particles [Ref. 13].

The last two robots, Rosemary (Figure 10) and Sakura (Figure 11) were used to inspect the Unit 1 reactor building. These robots were developed by Chiba Institute of Technology and modified by Hitachi-GE Nuclear Energy [Ref. 14].
Rosemary and Sakura were used to measure radioactivity prior to the SCR and likely informed the design that eventually developed into the PMORPH. These two robots were also used as a pair, much like the swimming and crawling robots. Rosemary uses a gamma camera to measure the dose rates which is operated wirelessly. Because wireless communication is limited in the reactor building, Sakura is used as transmission station in addition to a secondary inspection robot [Ref. 14].

3. Challenges

As should be expected when dealing with one of the worst nuclear accident sites in history, many challenges are faced in the design and implementation of these robots.

Firstly, a large number of iterations of designing, testing, and implementing of these robots was required. As discussed in this paper, the Rosemary and Sakura designs informed the SCR design which eventually led to the PMORPH. What was not discussed in this paper was the number of iterations that would have been required to achieve a working version of each of the mentioned designs. These types of iterations would not have been reported in any type of media outlet. In addition, the implementation of these robots was not always successful. For instance, the first SCR that was deployed was stuck and could not complete its full mission [Ref. 10]. This was not the first designed robot that failed its initial attempt.

Secondly, environmental conditions challenged the design and application processes immensely. These robots were designed with a four-day life expectancy within the high radiation fields of the reactor buildings, because that is the longest feasible timeframe in which the components can reasonably be expected to survive [Ref. 10]. In the actual application of these robots, they have not always been capable of meeting their designed life span. Another example is the MEISTeR, which can be powered both via a cable connection to a source and via a self-contained battery. The battery would allow the MEISTeR to travel further into the buildings to extend the range of cleanup activities, but its battery life is only 2 hours [Ref. 12], so this distance is limited. Yet another example is the Rosemary and Sakura pair. These were likely designed to work independently, but had to be operated in tandem because environmental conditions (the thick steel and concrete of any nuclear plant’s PCV) challenged wireless communications.

Lastly, the sheer scale of the cleanup project is a significant challenge. The decommissioning effort is likely to last for 30 to 40 years. The pathway to success at the end of this long period of time is quite unclear. In addition, most of the progress that has been made through March 2017 is in the data collection task. Advancements in robot designs have allowed the collection of more, higher quality data that would have been impossible otherwise. However, only a small percentage of the progress made has been towards the ultimate goal of cleanup and disposal of the melted nuclear fuel.

4. Conclusions

This paper provides lessons learned on two different levels: for the nuclear engineer and for the robot designer.

For the nuclear engineer, it is of the utmost importance to realize that a degree of safety beyond that used for other applications is fundamental. The nuclear engineer must consider events that most others may think impossible. Fukushima Daiichi’s nuclear accident is proof that significant natural disasters, while rare events, can occur. Therefore, the analyses which provide the framework for each nuclear power plant must be held to the highest standard. If these analyses require an expensive (financially or politically) decision to be made, then those performing the analyses must advocate their position strongly. In the specific instance of Fukushima Daiichi, because post-accident conditions exist, it is important to realize that robots are the best available option for cleanup and surveillance, even if their lifespan is shorter than desired.
For the robot designer, it is important to balance simplicity and complexity and recognize that intense technical rigor is required to achieve a working design. Simplicity and complexity are best balanced by analyzing the needs of a particular job. For instance, the SCR had a specialized mission of data collection that was successfully completed with few components or inputs. In contrast, the MHI-MEISTeR had a general mission of cleanup that was successfully completed with many components, attachments, etc. Intense technical rigor is required, as seen by the slow, incremental improvements in the various robot designs presented in this paper. However, it is important to realize that simple analyses (i.e. similar to those shown in Appendix A) can provide keen insights to robot design.

References

4. TEPCO Plant Overview.  
5. Definition of Sievert.  
   http://www.dictionary.com/browse/sievert
8. Unit 1 Primary Containment Vessel Internal Investigation. March 27, 2017.  
    http://irid.or.jp/en/research/20150420/
12. Robots working inside the buildings at Fukushima Daiichi NPS (Part II) MHI-MEISTeR.  
    http://irid.or.jp/en/research/meister/
13. Robots working inside the buildings at Fukushima Daiichi NPS (Part III) Swimming robot & Crawling robot.  
    http://irid.or.jp/en/research/gengorov_tridyver/
14. Robots working inside the buildings at Fukushima Daiichi NPS Rosemary & Sakura.  
    http://irid.or.jp/en/research/rosemary_sakura/
Appendix A: Brief Analysis of PMORPH

Topology and Coordinate Frames:

Mobility: One Free Rotation of the Revolute, Two Rotations of the Universal

\[ M = 6(nL - 1) - 5n/1 - 4n/2 - 3n/3 = 6(3 - 1) - 5(1) - 4(1) = 12 - 5 - 4 = 3 \]

Kinematic Setup: Two-Link Manipulator in 2-D

\[ \{q\} = \begin{pmatrix} \{r_2\} \\ \{\phi_2\} \\ \{r_3\} \\ \{\phi_3\} \end{pmatrix} \quad \{\Phi\} = \begin{pmatrix} \{r_1\}^A - \{r_2\}^A \\ \{r_2\}^B - \{r_3\}^B \\ \phi_2 - \phi_{2\text{START}} - \omega_2 t \\ \theta - \theta_{\text{START}} - \omega_3 t \end{pmatrix} \quad \theta = \phi_3 - \phi_2 \]
SIMULATION OF AN OPERATING YAW SYSTEM IN A WIND TURBINE

John Vieni
Mechanical Engineer
Orlando, FL, USA
E-mail: john.vieni@gmail.com

ABSTRACT
The purpose of this paper is to describe a methodology for simulating an operating yaw system in a wind turbine. When video simulating the operating system, a point on the system is tracked. With the tracked point within the video, the corresponding positions, velocities, and accelerations can be calculated at each point. The model is compared to the turbine's actual tracked model. The simulation also describes the level of detail provided within each model and future studies for completion of this analysis.

INTRODUCTION
The yaw system of a wind turbine drives the nacelle into the direction of the wind. The construction of this system consists of eight yaw gear drivers that mounted to the nacelle bedplate. The yaw ring gear is mounted directly to the tower shown in Figure 1.

To track the characteristics of the moving mechanism, the blue point on a yaw block was followed for 360 degrees of motion shown in Figure 2.

DETERMINING THE POSITION OF THE MECHANISM
Using the resolution of the video, the x and y pixels were tracked for each frame. To track the angle of the mechanism at any given time, a stationary center point of the mechanism was needed.

Because of space constraints in this wind turbine, it was not possible to capture a video of the entire mechanism. The center point was defined by scaling the video to the actual dimensions of the design. By knowing the distance from the center of the mechanism to the point tracked, the distance could be converted to an x and y pixel location.

With the x and y pixel location of the center and of each tracked position of the moving mechanism, the angle at any given point could be tracked.

COMPARING VIDEO SIMULATION TO SYSTEM TRACKED DATA
This wind turbine tracks the nacelle position, in order to direct the turbine into the direction of the wind. The system tracks the nacelle position every second. The video simulation goes into a finer level of detail of 29 frames per second. The system tracks data for only 1 frame per second.

Comparing these data points is necessary in order to determine if the simulation model is in line with the turbine.
tracked data. The comparison is also used to evaluate the behavior of the system.

**APPLYING A SAVITISKY-GOLAY FILTER**

A 7 point Savitisky-Golay filter was applied to the simulation data and the turbine data. It was applied to remove the noise from the data set. To affectively apply this filter one must calculate the position, velocity, acceleration, and jerk at each time step (h) Eq. (1). At the first three time step one is unable to use the 7 point filter without interpolating the position Eq. (1a), velocity Eq. (1b), acceleration (1c), and jerk (1d), from the fourth time step (xₙ₄).

\[
x_{i}^{*} = (-2 x_{i+3} + 3 x_{i+2} + 6 x_{i+1} + 7 x_{i} + 6 x_{i-1} + 3 x_{i-2} - 2 x_{i-3}) / 21 \quad (1a)
\]

\[
\dot{x}_{i}^{*} = (-22 x_{i+4} + 67 x_{i+3} + 58 x_{i+2} - 58 x_{i+1} - 67 x_{i+2} + 22 x_{i+1}) / 252h \quad (1b)
\]

\[
\ddot{x}_{i}^{*} = (5 x_{i+3} - 3 x_{i+1} - 4 x_{i} - 3 x_{i-1} + 5 x_{i-3}) / 42 h^2 \quad (1c)
\]

\[
\dddot{x}_{i}^{*} = (x_{i+3} - x_{i+2} - x_{i+1} + x_{i+2} - x_{i-3}) / 6 h^3 \quad (1d)
\]

After determining the values at the fourth time step, interpolator values can be calculated for the first three time steps (h) Eq. (2).

\[
b_0 = x_4^{*} \quad (2a)
\]

\[
b_1 = \ddot{x}_4^{*} h \quad (2b)
\]

\[
b_2 = \dddot{x}_4^{*} h^2 / 2 \quad (2c)
\]

\[
b_3 = \dddot{x}_4^{*} h^3 / 6 \quad (2d)
\]

Once the interpolator values are calculated the position Eq. (3), velocity Eq. (4), acceleration Eq. (5), and jerk Eq. (6) can be calculated for the first three time steps.

\[
x_{i+1} = b_0 + b_1 x_{i} + b_2 / h + b_3 / h^2 \quad (3a)
\]

\[
x_{i+2} = b_0 + b_1 x_{i} + b_2 / h + b_3 / h^2 \quad (3b)
\]

\[
x_{i+3} = b_0 + b_1 x_{i} + b_2 / h + b_3 / h^2 \quad (3c)
\]

\[
\dot{x}_{i+1}^{*} = (b_1 - 6b_2 + 27b_3) / h \quad (4a)
\]

\[
\dot{x}_{i+2}^{*} = (b_1 - 4b_2 + 12b_3) / h \quad (4b)
\]

\[
\dot{x}_{i+3}^{*} = (b_1 - 2b_2 + 3b_3) / h \quad (4c)
\]

\[
\ddot{x}_{i+1}^{*} = (2b_2 - 18b_3) / h^2 \quad (5a)
\]

\[
\ddot{x}_{i+2}^{*} = (2b_2 - 12b_3) / h^2 \quad (5b)
\]

\[
\ddot{x}_{i+3}^{*} = (2b_2 - 6b_3) / h^2 \quad (5c)
\]

\[
\dddot{x}_{i+1}^{*} = \dddot{x}_4^{*} \quad (6a)
\]

\[
\dddot{x}_{i+2}^{*} = \dddot{x}_4^{*} \quad (6b)
\]

\[
\dddot{x}_{i+3}^{*} = \dddot{x}_4^{*} \quad (6c)
\]

Additionally, one is unable to use the Savitisky-Golay Filter on the last three time steps without interpolating the position Eq. (1a), velocity Eq. (1b), acceleration (1c), and jerk (1d), from the fourth to last time step (xₙ₃). The interpolators used for the last three time steps (h) are described in Eq. (7).

\[
b_0 = x_{n-3}^{*} \quad (7a)
\]

\[
b_1 = \dddot{x}_{n-3}^{*} h \quad (7b)
\]

\[
b_2 = \dddot{x}_{n-3}^{*} h^2 / 2 \quad (7c)
\]

\[
b_3 = \dddot{x}_{n-3}^{*} h^3 / 6 \quad (7d)
\]

Once the interpolator values are calculated one can calculate the position Eq. (8), velocity Eq. (9), acceleration Eq. (10), and jerk Eq. (11) for the last three time steps.

\[
x_{n+2}^{*} = b_0 + b_1 x_{n-1} + b_2 / h + b_3 / h^2 \quad (8a)
\]

\[
x_{n+1}^{*} = b_0 + b_1 x_{n-1} + b_2 / h + b_3 / h^2 \quad (8b)
\]

\[
x_{n}^{*} = b_0 + b_1 x_{n-1} + b_2 / h + b_3 / h^2 \quad (8c)
\]

\[
\dot{x}_{n+2}^{*} = (b_1 + 2b_2 + 3b_3) / h \quad (9a)
\]

\[
\dot{x}_{n+1}^{*} = (b_1 + 4b_2 + 12b_3) / h \quad (9b)
\]

\[
\dot{x}_{n}^{*} = (b_1 + 6b_2 + 27b_3) / h \quad (9c)
\]

\[
\ddot{x}_{n+2}^{*} = (2b_2 + 6b_3) / h^2 \quad (10a)
\]

\[
\ddot{x}_{n+1}^{*} = (2b_2 + 12b_3) / h^2 \quad (10b)
\]

\[
\ddot{x}_{n}^{*} = (2b_2 + 18b_3) / h^2 \quad (10c)
\]

\[
\dddot{x}_{n+2}^{*} = \dddot{x}_{n+2}^{*} \quad (11a)
\]

\[
\dddot{x}_{n+1}^{*} = \dddot{x}_{n+3}^{*} \quad (11b)
\]

\[
\dddot{x}_{n}^{*} = \dddot{x}_{n-3}^{*} \quad (11c)
\]

**ANAYLSIS OF THE SIMULATION**

The initial analysis was done from 0 to 27 degrees. The simulated data showed that the position did not change linearly. The position of the system seemed to be very choppy during the first 10 seconds and then after the initial 10 seconds the system starts to smooth, shown in Figure 2.

![Figure 2](image-url)

**Figure 2**-Video simulated position

When analyzing the turbine’s data the position of the nacelle seemed to be moving in a linear pattern shown in Figure 3.
After applying the Savitisky-Golay filter the position of the simulated data seemed to show the position in finer detail, shown in Figure 4. The filter exemplifies the variability in the position of the nacelle in between the 7 to 9 second range.

When analyzing the turbine simulated position, the Savitisky-Golay filter does not transform the data very much. The position of the nacelle system changes linearly, shown in Figure 5.

When studying the velocity and acceleration, the analysis started to show significant differences between models. In the simulated video, the speed of the system was very volatile ranging from -2 to 2 degrees per second, shown in Figure 6. Based on the design of the system it does not seem very likely that the system could be operating at these speeds. Some external forces would have to be present to allow for the system to change position that rapidly.

The turbine simulated speed is roughly between 0.4-0.5 degrees per second, shown in Figure 7. This is more in line with the designed yawing speed of this turbine.
The acceleration analysis of the simulated video showed to be very volatile as well. The acceleration ranged from 15 to -15 degrees per second squared, shown in Figure 8.

From the position analysis the models looks fairly the same, but when looking at the velocity and acceleration the video simulation seems to be inconsistent with the turbine's simulation.

CONCLUSIONS

Based on the results found from the simulated model compared to the turbines model, the results from this simulation are inconclusive. There were several things that occurred during the simulation model that does not allow for concrete conclusions.

The first problem that I ran into was filming this mechanism. In order to film this mechanism, I had to lie directly on the yaw deck. Even with this visual, I was only able to capture part of the mechanism in the video. I was able to capture the moving mechanism with a partial frame but it still leads me into my second problem with this model.

When determining the position of the moving point, I had to scale the center point of the mechanism in order to get roughly an idea at which angle the mechanism was at. Because I was lying directly on the yaw deck my video picture was not always show from the same location as my hands could have been moving from time to time. This allowed for varied positions and an extra third dimension.

Because of the radical results of the simulation, more investigation will be done. When first looking into this it was thought that the simulation model should match with the turbine model but show a finer level of detail of what is happening.

Instead, the results between simulations were totally different. If this is the actual case, the behavior of the system can be better defined from filmed simulations of the turbine.

The next steps for completing this analysis are to figure out how to simulate the model with a fixed camera location that encompasses the entire system. This will remove
the scaling error as well as the shaky camera errors. The results can then be compared to the initial analysis and the turbine’s data.

REFERENCES
   Notes_05_02 H J Sommer
   Siemens 2.3-108 Turbine
ME 581: Simulation of Mechanical Systems
Experimental Analysis of Various Pool Shots

Grant Weekes

Abstract

There are various shooting styles available to a pool player. However, during each shot, the exact angle, speed, and position of the pool cue must be considered to ensure accuracy. Certain shots have a higher focus on power, such as a breaking shot. Others still focus on angle and position, like a draw shot. This report examines the kinematics of a pool cue-tip when making various strikes. The intent of this report is to provide a basic analysis of the shots, providing benchmark values for speed, acceleration, and jerk of each shot. The experiment also was intended to develop improved capabilities with motion capture methods.

2. Methodology

2.1. Experiment

For this experiment I recorded three types of shots recording three strikes for each type, a total of nine recordings. For each shot, 4 felt markers were placed on the shooter, allowing tracking of joint positions. The landmarks are as follows:

LM=1 at Shoulder
LM=2 at Elbow
LM=3 at Wrist
LM=4 at First Knuckle
LM=5 at Cue Tip

The intent was to vary the shot type and obtain multiple recordings of each shot type in order to account for some inconsistency in the shots. Due to the physical arrangement of the pool table and the room it was in, shooting with my right, non-dominant hand, was necessary. Additionally, the pool table was not the traditional green felt, reducing the contrast between some of the felt markers and the background. A friend began and ended each recording to minimize the amount of unnecessary footage taken for each recording.

2.2. Motion Capture

Once each recording was made, still frames were ripped from each video using vlc software. Each clip was between four and seven seconds long. The camera recorded each video at thirty frames per second, resulting in approximately one-hundred and seventy frames per video-clip. The tpsdig software was then used to compile and mark the frames with landmarks at each felt dot/ at the cue tip. Marking each video’s frames and extracting the data to a usable form was a very laborious process, resulting in the coordinates for each landmark in each frame of each video.

2.3. Data

The coordinates were plotted for each shot, providing an image of the practice strokes and strike taken. These images are shown in Figures 2.1 through 2.9.

Figure 2.1: Point positions for 1st breaking shot
3. Analysis

3.1 Position Data

With the position data, I was able to calculate the average practice stroke length and the length of the shot, including follow through, in the horizontal direction, shown in Table 3.1.

<table>
<thead>
<tr>
<th>Shot Type</th>
<th>Shot Attempt</th>
<th>Stroke Type</th>
<th>Length (inches)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breaks</td>
<td>1</td>
<td>Avg Practice</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Strike</td>
<td>16.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Avg Practice</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Strike</td>
<td>11.5</td>
</tr>
<tr>
<td>Normal Shots</td>
<td>1</td>
<td>Avg Practice</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Strike</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Avg Practice</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Strike</td>
<td>17</td>
</tr>
<tr>
<td>Draw Shots</td>
<td>1</td>
<td>Avg Practice</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Strike</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Avg Practice</td>
<td>5.5</td>
</tr>
</tbody>
</table>

Table 3.1: Stroke lengths for each strike & practice set

My initial review of the position data, and my experience performing the motion capture landmark placement, led me to believe that determination of a precise strike location was not possible due to the relatively low frame rate compared to the speed of the pool cue. However, upon further review of the data, I found that this was not necessarily the case. Table 2 shows the cue tip location for each practice shot (at full extension) and the cue tip location for the frames immediately before and after each strike (for improved granularity, these values are in pixels: 1 pixel=0.0785 inches).

<table>
<thead>
<tr>
<th>Stroke (practice number shown as decimal)</th>
<th>x-position(s)</th>
<th>y-position(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break 1.1</td>
<td>689</td>
<td>419</td>
</tr>
<tr>
<td>Break 1.2</td>
<td>703</td>
<td>417</td>
</tr>
<tr>
<td>Break 1.3</td>
<td>654.724</td>
<td>422.421</td>
</tr>
<tr>
<td>Break 1.4</td>
<td>669</td>
<td>420</td>
</tr>
<tr>
<td>Break 2.1</td>
<td>652</td>
<td>420</td>
</tr>
<tr>
<td>Break 2.2</td>
<td>654</td>
<td>420</td>
</tr>
<tr>
<td>Break 2.3</td>
<td>683</td>
<td>419</td>
</tr>
<tr>
<td>Break 2.4</td>
<td>683</td>
<td>416</td>
</tr>
<tr>
<td>Break 2.5</td>
<td>667.729</td>
<td>415.414</td>
</tr>
<tr>
<td>Break 3.1</td>
<td>711</td>
<td>421</td>
</tr>
<tr>
<td>Break 3.2</td>
<td>726</td>
<td>418</td>
</tr>
<tr>
<td>Break 3</td>
<td>679.741</td>
<td>424.423</td>
</tr>
<tr>
<td>Normal 1.1</td>
<td>706</td>
<td>419</td>
</tr>
<tr>
<td>Normal 1.2</td>
<td>705</td>
<td>420</td>
</tr>
<tr>
<td>Normal 1</td>
<td>702.728</td>
<td>418.416</td>
</tr>
<tr>
<td>Normal 2.1</td>
<td>691</td>
<td>418</td>
</tr>
<tr>
<td>Normal 2.2</td>
<td>689</td>
<td>418</td>
</tr>
<tr>
<td>Normal 2.3</td>
<td>696</td>
<td>417</td>
</tr>
<tr>
<td>Normal 2.4</td>
<td>657.710</td>
<td>418.411</td>
</tr>
<tr>
<td>Normal 3.1</td>
<td>704</td>
<td>420</td>
</tr>
<tr>
<td>Normal 3.2</td>
<td>710</td>
<td>418</td>
</tr>
<tr>
<td>Normal 3.3</td>
<td>709.735</td>
<td>420.419</td>
</tr>
<tr>
<td>Draw 1.1</td>
<td>697</td>
<td>415</td>
</tr>
<tr>
<td>Draw 1.2</td>
<td>710</td>
<td>412</td>
</tr>
<tr>
<td>Draw 1.3</td>
<td>704.735</td>
<td>413.410</td>
</tr>
<tr>
<td>Draw 2.1</td>
<td>676</td>
<td>414</td>
</tr>
<tr>
<td>Draw 2.2</td>
<td>683</td>
<td>413</td>
</tr>
<tr>
<td>Draw 2.3</td>
<td>684</td>
<td>413</td>
</tr>
<tr>
<td>Draw 2.4</td>
<td>662.691</td>
<td>417.411</td>
</tr>
<tr>
<td>Draw 3.1</td>
<td>691</td>
<td>412</td>
</tr>
<tr>
<td>Draw 3.2</td>
<td>692</td>
<td>412</td>
</tr>
<tr>
<td>Draw 3.3</td>
<td>686.725</td>
<td>413.411</td>
</tr>
</tbody>
</table>

Table 2: Practice targets and strike points
Put in another form, we get Figure 3.1.

![Figure 3.1: Practice targets vs. Strike Points](image)

In Figure 3.1, shades of blue indicate breaks, yellow are normal shots, and fuchsia for draws. In order to show the vertical variation between single pixels, the chart is not to scale. By inspection, it appears the blue arrows have the most distance from their associated target points, the yellow shows improvement, and fuchsia more improvement still. I averaged the practice target locations, and compared the height of the strike (linearly interpolated to the practiced x-coordinate) to the averaged practice height. Table 3.3 shows the results of this comparison.

<table>
<thead>
<tr>
<th>Shot</th>
<th>Practiced Height</th>
<th>Strike Height</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break 1</td>
<td>417.3</td>
<td>421.3</td>
<td>4 pix = 0.314 in</td>
</tr>
<tr>
<td>Break 2</td>
<td>419</td>
<td>415.0</td>
<td>-4 pix = -0.314 in</td>
</tr>
<tr>
<td>Break 3</td>
<td>419.5</td>
<td>423.4</td>
<td>3.9 pix = 0.306 in</td>
</tr>
<tr>
<td>Normal 1</td>
<td>419.5</td>
<td>417.7</td>
<td>-1.8 pix = -0.165 in</td>
</tr>
<tr>
<td>Normal 2</td>
<td>417.7</td>
<td>413.4</td>
<td>-4.3 pix = -0.338 in</td>
</tr>
<tr>
<td>Normal 3</td>
<td>419</td>
<td>420</td>
<td>1 pix = 0.079 in</td>
</tr>
<tr>
<td>Draw 1</td>
<td>413.5</td>
<td>413.0</td>
<td>-0.5 pix = -0.039 in</td>
</tr>
<tr>
<td>Draw 2</td>
<td>413.3</td>
<td>413.1</td>
<td>-0.2 pix = -0.024 in</td>
</tr>
<tr>
<td>Draw 3</td>
<td>412</td>
<td>412.7</td>
<td>0.7 pix = 0.055 in</td>
</tr>
</tbody>
</table>

Table 3.3: Average practice heights vs. interpolated strike height

The average magnitude of the “miss” for each type of shot is then 0.311” for breaks, 0.194”, and 0.039” for draw shots. However, I do not believe these results are significant when compared to the error from imperfect landmark placement and, potentially much more significantly, my body and the cue possibly going out of plane.

While being recorded, I did not adjust my hand position on the cue stick. As a result, if my knuckle and the cue remained in a plane parallel to the video camera lens, then the perceived distance between the landmark on my knuckle and the cue tip should remain a fixed length. I averaged the distance between my knuckle and cue tip for the first 88 frames of each video (intentionally excluding frame where I was taking a shot where my arm was severely blurred). The measured distances were distributed around this average value for each video. For eight of the nine videos, this variation was in the range of 0.66 inches to 0.95 inches (8.5 pixels to 12 pixels). For the second break, this variation spanned a full five inches. The behavior of the error (shown in figure 3.2) leads me to believe there either was an experimental error or a computational error. However, a review of the video does not indicate I was standing or stroking grossly askew (I would have needed to have swung my wrist 15 degrees out of parallel in both directions to have made such error) and a review of my calculations shows no difference from my other error calculations.

![Figure 3.2: Variation in distance between knuckle and cue tip](image)
3.2 Speed Data

Next I adapted a standard 3-point central derivative and 5-point central derivative to calculate the speed of each point at each frame (vice velocity in the horizontal and vertical directions) as shown in Equations 3.1 and 3.2.

\[
\text{Speed}_{3-\text{pt}}^n = \sqrt{\left(\frac{x_{n+1} - x_{n-1}}{2 \times dt}\right)^2 + \left(\frac{y_{n+1} - y_{n-1}}{2 \times dt}\right)^2} \times \frac{\text{inches}}{\text{pixel}}
\]

Equation 3.1: Three point speed

\[
\text{Speed}_{5-\text{pt}}^n = \sqrt{\left(\frac{-x_{n+2} + 8x_{n+1} - 8x_{n-1} + x_{n-2}}{12 \times dt}\right)^2 + \left(\frac{-y_{n+2} + 8y_{n+1} - 8y_{n-1} + y_{n-2}}{12 \times dt}\right)^2} \times \frac{\text{inches}}{\text{pixel}}
\]

Equation 3.2: Five point speed

This resulted in 18 graphs, however the plots of each strike’s speed are not particularly necessary for further analyses. Representative graphs from the first break are shown in Figures 3.3 and 3.4, below.

Figure 3.3: First break speeds (3-pt derivative)

Figure 3.4: First break speeds (5-pt derivative)

Note that each pair of peaks represents the drawing back and then the moving forward of the cue as the plot depicts the speed. Figure 3.5 shows a side-by-side comparison of the two methods of differentiation for the knuckle landmark’s speed during this first break.

Figure 3.5: Comparison of differentiation methods

It is readily observable that the strike event last approximately five to seven frames (0.3925 seconds to 0.5495 seconds). Because of this, continued use of a 5-point differentiation method would quickly lead to parameters of the strike event being influenced by data from other events of the clip (by the time I calculated jerk using 5-point differentiations for scalar acceleration and jerk, every single point of the strike event would have some weighting from points before or after the strike event). For this reason, I abandoned further use of 5-point differentiations.

The speed reached for each strike is shown in Table 3.4.
Table 3.4: Speed of cue at each strike

<table>
<thead>
<tr>
<th></th>
<th>Speed (in/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break 1</td>
<td>11.62</td>
</tr>
<tr>
<td>Break 2</td>
<td>10.31</td>
</tr>
<tr>
<td>Break 3</td>
<td>10.39</td>
</tr>
<tr>
<td>Normal 1</td>
<td>4.58</td>
</tr>
<tr>
<td>Normal 2</td>
<td>7.49</td>
</tr>
<tr>
<td>Normal 3</td>
<td>5.17</td>
</tr>
<tr>
<td>Draw 1</td>
<td>5.98</td>
</tr>
<tr>
<td>Draw 2</td>
<td>6.61</td>
</tr>
<tr>
<td>Draw 3</td>
<td>7.12</td>
</tr>
</tbody>
</table>

With these speed data, and the data from the position analysis, I compared the speed of the cue near contact with the overall stroke length past where the practice strokes stopped (an approximation of where the cue ball was), Figure 3.6 relates.

Table 3.5: Acceleration of and Force required for each shot

<table>
<thead>
<tr>
<th>Shot</th>
<th>Acceleration</th>
<th>Equivalent Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break 1</td>
<td>210.5 ft/sec²</td>
<td>1.96 lbf</td>
</tr>
<tr>
<td>Break 2</td>
<td>133.75 ft/sec²</td>
<td>2.25 lbf</td>
</tr>
<tr>
<td>Break 3</td>
<td>137.1 ft/sec²</td>
<td>2.32 lbf</td>
</tr>
<tr>
<td>Normal 1</td>
<td>34.4 ft/sec²</td>
<td>1.96 lbf</td>
</tr>
<tr>
<td>Normal 2</td>
<td>92.7 ft/sec²</td>
<td>3.42 lbf</td>
</tr>
</tbody>
</table>

I identified no relation between the two.

3.3 Acceleration Data

Using a three-point derivative I calculated the scalar-accelerations for each strike, and then using the weight of the cue (22oz for breaks and first normal strike, 19oz for remaining strikes), shown in Table 3.5. NOTE: the analysis presented 5/1/2017 via powerpoint did not account for the higher weight of the breaking cue, also used for the first normal shot.

I identified that the variation in force applied appears to vary less, by percentage, for the break shots and even less for the draw shots when compared to the normal shots. The maximum shot-force expressed as a percentage of the minimum shot force for each type of shot is: 157% for breaks, 110% for draws, and 246% for the normal shots.

3.4 Jerk Data

I took the derivative of scalar-acceleration, again using the three-point derivative. Figure 3.7 provides a representative graph of the jerk of the cue tip.
Table 3.6: Peak Jerks

<table>
<thead>
<tr>
<th>Shot</th>
<th>Peak Jerks (in/sec³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break 1</td>
<td>2526.4 -2824.3 2200.4</td>
</tr>
<tr>
<td>Break 2</td>
<td>1609.1 -2989.4 2548.8</td>
</tr>
<tr>
<td>Break 3</td>
<td>1983.1 -3012.6 1302.6</td>
</tr>
<tr>
<td>Normal 1</td>
<td>446.9 -529.1 369.4</td>
</tr>
<tr>
<td>Normal 2</td>
<td>1078.5 -1963.7 1449.7</td>
</tr>
<tr>
<td>Normal 3</td>
<td>802.7 -591.4 510.4</td>
</tr>
<tr>
<td>Draw 1</td>
<td>796.5 -978.2 793.9</td>
</tr>
<tr>
<td>Draw 2</td>
<td>1008.4 -1861.6 1507.9</td>
</tr>
<tr>
<td>Draw 3</td>
<td>805.2 -1421.1 1469.6</td>
</tr>
</tbody>
</table>

This jerk data provided insight into the physical meaning of jerk, and the behavior of jerk for physical events.

4. Conclusion

There is some evidence suggesting that shot type influences the accuracy of the strike. However, additional data points would be required for a strong conclusion and the error associated with digitization of landmarks and imperfect recording techniques would need to be reduced for reasonable significance to be attributed to the results. I further found no correlation between shot type and stroke length. Furthermore, the speed of any particular strike does not appear to have a strong correlation to the distance the cue tip travels past the cue, implying that acceleration and jerk of my shots is not uniform or consistent. However, I did find indication that my draw shots are more consistent with respect to the applied force. Analytically, I concluded that for the timescale of a pool shot requires reasonably high speed equipment for high fidelity. Without this granularity, increased order of derivation or taking multiple derivatives leads to a majority of the points pertaining to the event sourcing points from other events (such as before the strike is initiated).

Additional areas for research include analysis of the scatter in measured distance between two points of fixed distance apart. Such analysis could control for the angle of a player with respect to the camera lens. By analyzing the resulting error compared to the overall motion of the pool cue, one could attempt to separate the digitization error from the perspective error. An analytical model containing both types of error could predict the angle of a player with respect to a camera, and remove the perspective error.

References
None