Ahmad Amin
Sarah Polzer
Joshua Cho

Project 1: Kinematics of the Stewart Platform

MATLAB Code

Problem 1

```
function out = project_1(theta)
   L1 = 2;
   L2 = sqrt(2);
   L3 = sqrt(2);
   gamma = pi/2;
   p1 = sqrt(5);
   p2 = sqrt(5);
   p3 = sqrt(5) ;
   x1 = 4;
   x2 = 0;
   y2 = 4;
   A2 = L3*cos(theta)-x1;
   B2 = L3*sin(theta);
   A3 = L2*cos(theta+gamma)-x2;
   B3 = L2*sin(theta+gamma) - y2;
   N1 = B3*(p2^2-p1^2 - A2^2 - B2^2)-B2*(p3^2-p1^2-A3^2-B3^2);
   N2 = -A3*(p2^2 - p1^2 - A2^2 - B2^2) + A2*(p3^2 - p1^2 - A3^2 - B3^2);
   D = 2*(A2*B3 - B2*A3);
   out = N1^2 + N2^2 - p1^2*D^2;


%Number 1
%Code to check if function was set up appropriately
theta_one = pi/4;
theta_two = -pi/4;
project_1(theta_one)
project_1(theta_two)
```

Problem 2

```
%Number 2
%Code to generate respective f(x)-values and plot x, and f(x)
figure(1)
X = -pi:.01:pi;
n=length(X);
for i=1:n
    Y(i)=project_1(X(i));
end
plot(X,Y);
grid;
```

Problem 3

```
%Number 3
%Code to generate the planar Stewart platform with identical arm lengths as
seen in figure 1.15
u1 = 1;
u2 = 2;
u3 = 2;
```

Ahmad Amin
Sarah Polzer
Joshua Cho

```
v1 = 2;
v2 = 1;
v3 = 3;
figure(2);
plot([u1 u2 u3 u1],[v1 v2 v3 v1], 'b'); hold on;
plot([0 4 0], [0 0 4], 'mo');
plot([0 1],[0 2], 'm'); hold on;
plot([2 4], [1 0], 'm'); hold on;
plot([0 2], [4 3], 'm'); hold on;
plot([1 2 2],[2 1 3], 'mo');
```

Problem 4

```
%Code that has both updated constants within the function and also plots the
planar Stewart Platforms once theta’s were found
function out = part_four_part_2(theta)
   L1 = 3;
   L2 = 3*sqrt(2);
   L3 = 3;
   gamma = pi/4;
   p1 = 5;
   p2 = 5;
   p3 = 3;
   x1 = 5;
   x2 = 0;
   y2 = 6;
   A2 = L3*cos(theta)-x1;
   B2 = L3*sin(theta);
   A3 = L2*cos(theta+gamma)-x2;
   B3 = L2*sin(theta+gamma) - y2;
   N1 = B3*(p2^2-p1^2 - A2^2 - B2^2)-B2*(p3^2-p1^2-A3^2-B3^2);
   N2 = -A3*(p2^2 - p1^2 - A2^2 - B2^2) + A2*(p3^2 - p1^2 - A3^2 - B3^2);
   D = 2*(A2*B3 - B2*A3);
   x = N1/D
   y = N2/D
   u1= x + L2*cos(theta + gamma)
   u2= x + L3*cos(theta)
   v1= y+L2*sin(theta+gamma)
   v2= y + L3*sin(theta)
%Now that we have x and y, we were able to plot the triangles depending on
what the theta value given was for the function
   x
   y
   figure(9)
   plot([x u1 u2 x],[y v1 v2 y], 'r'); hold on;
   plot([x u1 u2 x],[y v1 v2 y], 'ro');
   plot([0 x],[0 y],'b'); hold on;
   plot([x2 u1],[y2 v1],'b'); hold on;
   plot([x1 u2],[0 v2],'b'); hold on;
   plot([0 x1 x2],[0 0 y2],'bo'); hold on;

%Code to plot the initial function to determine our roots
X_n_four = -pi:.01:pi;
n=length(X);
for i=1:n
    Y_n_four(i)=project_1_part_four(X(i));
end
```

Ahmad Amin
Sarah Polzer
Joshua Cho

```
figure(3);
plot(X_n_four,Y_n_four);
grid;

% 4 zeroes/poses = -.9376, -0.0592, 1.1569, 2.0495

% Bisect Method function to solve for the roots
function xc = bisect(f,a,b,tol)
fa=f(a);
fb=f(b);
k = 0;
while (b-a)/2>tol
 c=(a+b)/2;
 fc=f(c);
 if fc == 0 %c is a solution, done
 break
 end
 if sign(fc)*sign(fa)<0 %a and c make the new interval
 b=c;fb=fc;
 else %c and b make the new interval
 a=c;fa=fc;
 end
end
xc=(a+b)/2;
```

Problem 5

```
%Code that has both updated constants within the function for this given
problem to find 6 poses instead of 4. Function also includes later part that
was initially omitted but added later to generate the plots of the Stewart
Platforms

function out = part_five_part_2(theta)
   L1 = 3;
   L2 = 3*sqrt(2);
   L3 = 3;
   gamma = pi/4;
   p1 = 5;
   p2 = 7.01;
   p3 = 3;
   x1 = 5;
   x2 = 0;
   y2 = 6;
   A2 = L3*cos(theta)-x1;
   B2 = L3*sin(theta);
   A3 = L2*cos(theta+gamma)-x2;
   B3 = L2*sin(theta+gamma) - y2;
   N1 = B3*(p2^2-p1^2 - A2^2 - B2^2)-B2*(p3^2-p1^2-A3^2-B3^2);
   N2 = -A3*(p2^2 - p1^2 - A2^2 - B2^2) + A2*(p3^2 - p1^2 - A3^2 - B3^2);
   D = 2*(A2*B3 - B2*A3);
   x = N1/D;
   y = N2/D;
   u1= x + L2*cos(theta + gamma);
   u2= x + L3*cos(theta);
   v1= y+L2*sin(theta+gamma);
   v2= y + L3*sin(theta);
   figure()
```

Ahmad Amin
Sarah Polzer
Joshua Cho

```
   plot([x u1 u2 x],[y v1 v2 y], 'r'); hold on;
   plot([x u1 u2 x],[y v1 v2 y], 'ro');
   plot([0 x],[0 y],'b'); hold on;
   plot([x2 u1],[y2 v1],'b'); hold on;
   plot([x1 u2],[0 v2],'b'); hold on;
   plot([0 x1 x2],[0 0 y2],'bo'); hold on;

%Code to plot the initial function to determine our roots
X_n_five = -pi:.01:pi;
n=length(X_n_five);
for i=1:n
    Y_n_five(i)=project_1_part_five(X_n_five(i));
end
figure(100);
plot(X_n_five,Y_n_five);
grid;

%zeroes: -0.6415, -0.4098, 0.0576, 0.4617, 0.9774, 2.5149
```

Problem 6

```
%Code that has both updated constants within the function for this given
problem to find 2 poses instead of 4. Function also includes later part that
was initially omitted but added later to generate the plots of the Stewart
Platforms
function out = part_six_part_2(theta)
   L1 = 3;
   L2 = 3*sqrt(2);
   L3 = 3;
   gamma = pi/4;
   p1 = 5;
   p2 = 4;
   p3 = 3;
   x1 = 5;
   x2 = 0;
   y2 = 6;
   A2 = L3*cos(theta)-x1;
   B2 = L3*sin(theta);
   A3 = L2*cos(theta+gamma)-x2;
   B3 = L2*sin(theta+gamma) - y2;
   N1 = B3*(p2^2-p1^2 - A2^2 - B2^2)-B2*(p3^2-p1^2-A3^2-B3^2);
   N2 = -A3*(p2^2 - p1^2 - A2^2 - B2^2) + A2*(p3^2 - p1^2 - A3^2 - B3^2);
   D = 2*(A2*B3 - B2*A3);
   x = N1/D;
   y = N2/D;
   u1= x + L2*cos(theta + gamma);
   u2= x + L3*cos(theta);
   v1= y+L2*sin(theta+gamma);
   v2= y + L3*sin(theta);
   figure()
   plot([x u1 u2 x],[y v1 v2 y], 'r'); hold on;
   plot([x u1 u2 x],[y v1 v2 y], 'ro');
   plot([0 x],[0 y],'b'); hold on;
   plot([x2 u1],[y2 v1],'b'); hold on;
   plot([x1 u2],[0 v2],'b'); hold on;
   plot([0 x1 x2],[0 0 y2],'bo'); hold on;
```

Ahmad Amin
Sarah Polzer
Joshua Cho

```
%Script Code to plot the initial function to determine our roots
X_n_six = -pi:.01:pi;
n=length(X_n_six);
for i=1:n
    Y_n_six(i)=project_1_part_six(X_n_six(i));
end
figure();
plot(X_n_six,Y_n_six);
grid;

%zeroes = 1.3500, 1.7700
```