**Project 1: Kinematics of the Stewart Platform.**

**Group members: Shilpa Kuppili , Sathwik Kuchana,   
 Praveen Kumar Govind Reddy, Jayanth Vodnala**

**Introduction:**

"A stewart platform consist of six variable length struts, or prismatic joints, supporting a payload. Prismatic joints operate by changing the length of the strut, usually pneumatically or hydraulically. As a six-degree-of-freedom robot, the Stewart platform can be placed at any point and inclination in three-dimensional space that is within its reach." -Timothy Sauer, Numerical Analysis.

So essentially, a Stewart Platform is a very helpful, adjustable triangle. If one knows certain pieces of information about the triangle's position and orientation, they can find the proper settings for the triangle and even graph it. Sometimes, there are multiple triangles that will satisfy the parameters. One can find the proper values of the variables by doing a bit of arithmetic. This project explores doing this arithmetic to solve the planar Stewart platform.

A planar Stewart platform has many values associated to it. L1, L2, and L3 refer to the side lengths of the triangle. The end of one leg is placed at the origin. Another leg is positioned on the x-axis at x1. The end of the third leg is placed at (x1, x2). Gamma is an angle of the triangle. p1, p2, and p3 are the lengths of the legs. Theta is the orientation of the triangle, while x and y denote a single coordinate of the triangle. So given L1, L2, L3, x1, x2, y2, p1, p2, p3, and gamma, can one find x, y, and theta? To do such a thing will require solving what is called the forward kinematics problem of the planar Stewart platform.

Suggested activities:

**Activity 1.** The parameters L1, L2, L3,γ, x1, x2, y2 are fixed constants, and the strut lengths p1, p2, p3 will be known for a given pose. function out=f(theta) : : out=N1^2+N2^2-p1^2\*D^2; To test your code, set the parameters L1 = 2, L2 = L3 = √ 2, γ = π/2, p1 = p2 = p3 = √ 5 Then, substituting θ = −π/4 or θ = π/4, corresponding to Figures respectively, should make f (θ ) = 0.

**Answer :** In this activity we have written a function f1\_a which takes theta as input and calculates f(theta) and we have tested this function at -pi/4 and pi/4 and got the result closer to zero. First, python code was developed to create f(theta), a function of 1 variable and 7 constants, utilizing different values for L1, L2, L3, x1, x2, y2, p1, p2, p3, and gamma. It demonstrates how the forward kinematics problem should be solved as well as the values of the variables that were employed.

• Specify the function's inputs:

• Fixed platform and end-effector positions are at x1, x2, and y2.The platform arms' lengths are L1, L2, and L3.

• Gamma: The platform's orientation angle.

• P1, P2, and P3: the struts' lengths.

• Determine the Stewart platform equations' intermediate values. The Stewart function f(theta) can be calculated using these intermediate values.

• To calculate the outcome and the intermediate value, you can invoke the function with a given theta value.

-4.547473508864641e-13

-4.547473508864641e-13

**Activity 2.** Plot f (θ ) on [−π,π]. You may use the @ symbol as described to assign a function handle to your function file in the plotting command. You may also need to precede arithmetic operations with the “.” character to vectorize the operations. As a check of your work, there should be roots at ±π/4.

**Answer:** In this activity we have plotted f(theta) over -pi to pi and the function is intersecting x axis at two points -0.786 and -0.785 so there are roots at pi/4 and -pi/4. One can test it with various theta values using the f(theta) function with the given arguments. Additionally, it plots f(theta) in the interval [-π, π].

Keep in mind that there are two choices for theta because there are two zeros. This indicates that, given these characteristics, there are two possible positions for a Stewart platform. The basic bisection method was used to locate these zeros.

These are the roots: [-0.786 0.785]

• Create the appropriate x-axis points.

• Create the appropriate y-axis points. One might use list comprehension.

• The X-axis was created to examine the intersection points.

• Draw the x-axis and the Stewart function.

• Intersection pointed with x-axis (or roots)

A graph of a function

Description automatically generated

**Activity 3.** Plot([u1 u2 u3 u1],[v1 v2 v3 v1],’r’); hold on >> plot([0 x1 x2],[0 0 y2],’bo’) will plot a red triangle with vertices (u1,v1),(u2,v2),(u3,v3) and place small circles at the strut anchor points (0,0),(x1,0),(x2,y2). In addition, draw the struts.

**Answer:** The plots are below:

* The limits for X and Y are set.
* The red polygon is plotted
* The black lines are plotted
* The blue circles are plotted
* The plot has been displayed

The code will plot a red triangle and small blue circles to reproduce. Adjust the axis limits as needed to fit your specific scenario.

In this activity we have plotted a red triangle with vertices (u1,v1),(u2,v2),(u3,v3) and placed small circles at the strut anchor points (0,0),(x1,0),(x2,y2). and also we have drawn the structs.

**A graph with a red and black triangle

Description automatically generated**

**A diagram of a triangle

Description automatically generated**

**Activity 4.** Solve the forward kinematics problem for the planar Stewart platform specified by x1 = 5, (x2, y2) = (0,6), L1 = L3 = 3, L2 = 3 √ 2, γ = π/4, p1 = p2 = 5, p3 = 3. Begin by plotting f (θ ). Use an equation solver to find all four poses and plot them.

**Answer:** Below is the graph of the new f(theta)

This provided four new possible values of theta. These values of theta were then used to make four new plots.

1.Which equation solver did you use, and why?

Answer : We have used bisection method to find the roots because it is easy to code and convergence is guaranteed

2.How did you initialize your solver, and why? (e.g. how did you pick the initial interval for bisection, or initial condition for FPI, . . .

Answer: We plotted function f(theta) from -pi to pi. it is touching x - axis at  
four points so by looking at the plot we have taken 4 intervals they are

1. -1,-0.5
2. -0.5,0
3. 1,1.5
4. 2,2.5

3.What stopping condition did you use in your solver? How accurate is the obtained solution (root)?

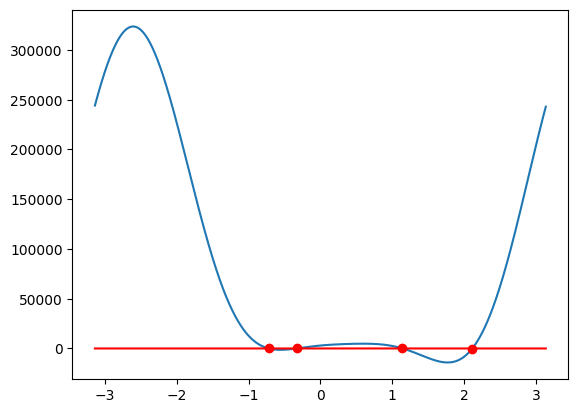
Answer : We have used np.abs(a-b)>TOL as stopping condition and initialized TOL to 10 \*\* -12 so we have obtained the solution with tolerance of 10 \*\* -12

4.How fast is your solver? (e.g. linearly convergent, quadratically convergent, . . .)

Answer : Bisection method is linearly convergent. This means that with each iteration, the method approximately doubles the number of correct digits. In other words, if you double the number of iterations, you get roughly twice as many correct digits in your approximation

--> we have plotted f(theta) from -pi to pi and from the plot we have taken four intervals and used bisection method to find four roots and for each root we have calculated p1,p2,p3 using inverse kinematics function and verified the correctness of p1,p2,p3 and for each pose we have plotted red polygon , black lines and blue circles

Roots are: [-0.721 -0.332 1.143 2.115]



theta = -0.7208492044605919

u2 = 2.8554254993170494

v2 = 5.0799210806394886

u3 = 0.8753568865678552

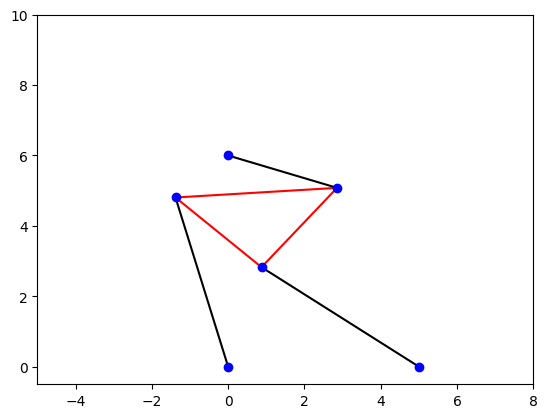
v3 = 2.8261845634739435

p1 = 5, P1 = 5

p2 = 5, P2 = 5

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = -0.33100518428409487

u2 = 2.897421145495243

v2 = 6.777785771040897

u3 = 1.9224397112210028

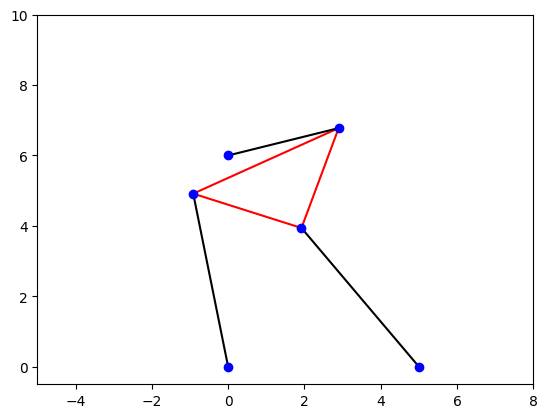
v3 = 3.9406373429849113

p1 = 5, P1 = 5

p2 = 5, P2 = 5

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = 1.1436855178212681

u2 = 2.9939796160277568

v2 = 6.189963309126738

u3 = 5.724478736893191

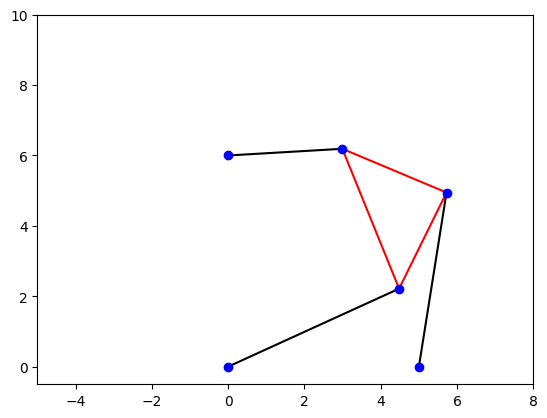
v3 = 4.9472346376328

p1 = 5, P1 = 5

p2 = 5, P2 = 5

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = 2.11590901408681

u2 = 0.45108022909668666

v2 = 3.0341060998541467

u3 = 3.016286827758401

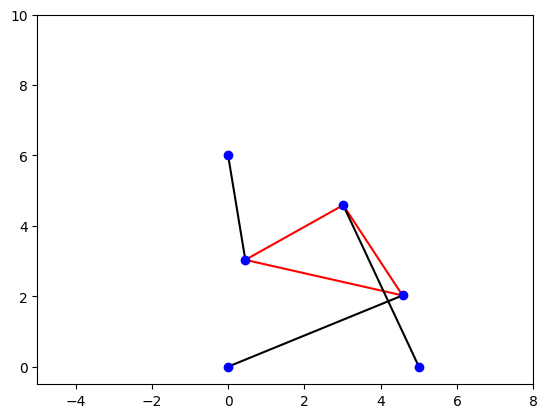
v3 = 4.589649447428363

p1 = 5, P1 = 5

p2 = 5, P2 = 5

p3 = 3, P3 = 3

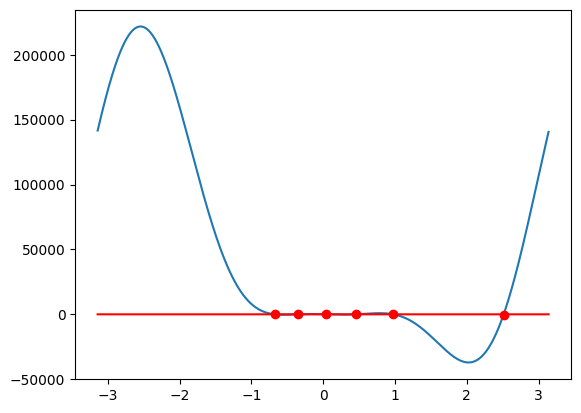
successfully verified that p1,p2,p3 values are lengths of structs in the plot



**Activity 5**. The shape of the triangle is defined by L1 = 2, L2 =L3 = √  
2, γ =π/2. 5. Change strut length to p2 = 7 and re-solve the problem. For these parameters, there are six poses.

**Answer:** Here, p2 has the strut length as p2 = 7. The graph of f(theta) now has 6 solutions, for which there are 6 poses.

Roots are: [-0.674 -0.355 0.037 0.458 0.977 2.513]



theta = -0.6731574863720198

u2 = -0.09881519735927569

v2 = 3.001627848853331

u3 = -1.9691885782391547

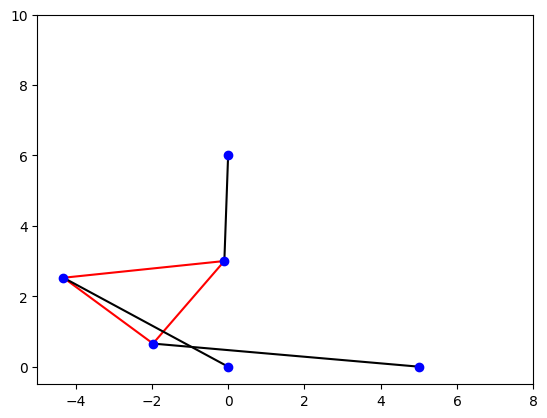
v3 = 0.6560568275248415

p1 = 5, P1 = 5

p2 = 7, P2 = 7

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = -0.3547402704157321

u2 = -0.9496458520511908

v2 = 3.154271138059392

u3 = -1.991686308996179

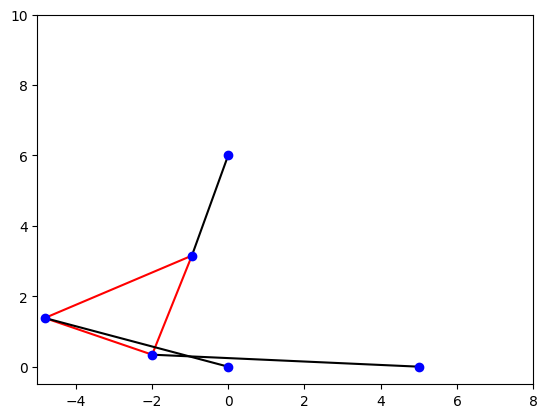
v3 = 0.34106092798089804

p1 = 5, P1 = 5

p2 = 7, P2 = 7

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = 0.03776676057568693

u2 = -2.0644372045864157

v2 = 3.823282510677434

u3 = -1.9511638548364902

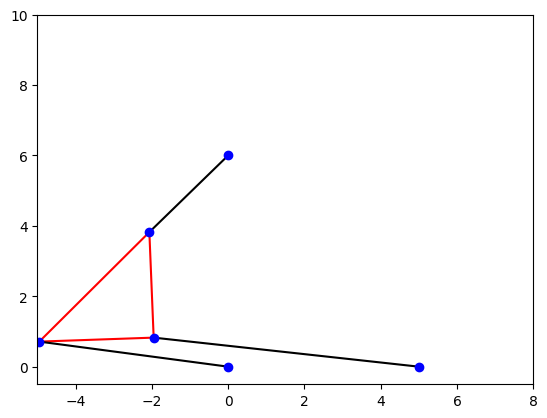
v3 = 0.8254217486945775

p1 = 5, P1 = 5

p2 = 7, P2 = 7

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = 0.4588781810493856

u2 = 0.541021862445155

v2 = 8.950812658301503

u3 = 1.869849719908682

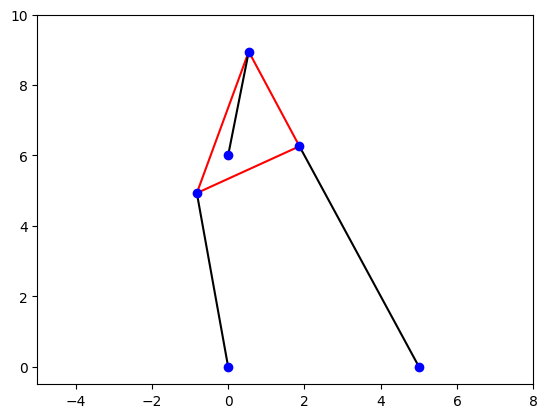
v3 = 6.261162769332886

p1 = 5, P1 = 5

p2 = 7, P2 = 7

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = 0.9776728950005236

u2 = 1.4928185543048271

v2 = 8.602209208329885

u3 = 3.980415173198234

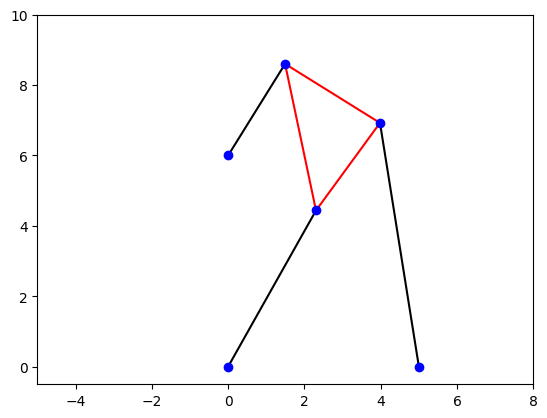
v3 = 6.925348134278105

p1 = 5, P1 = 5

p2 = 7, P2 = 7

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = 2.5138527993499338

u2 = -0.974325937443064

v2 = 3.1626263961860523

u3 = 0.7876250469533117

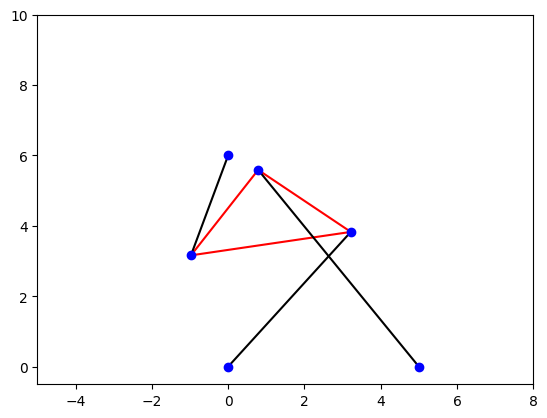
v3 = 5.590697385383988

p1 = 5, P1 = 5

p2 = 7, P2 = 7

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot

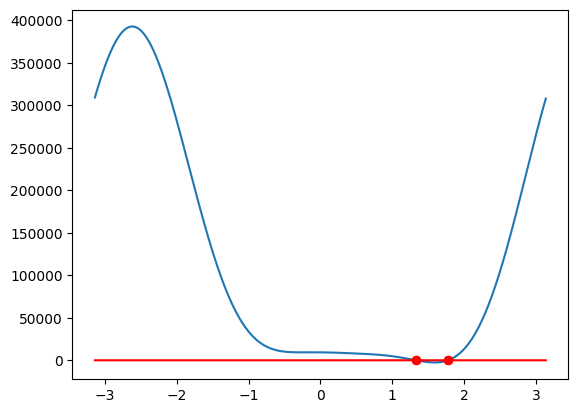


**Activity 6.** Find a strut length p2, with the rest of the parameters as in Step 4, for which there are only two poses.

**Answer:** We have searched over a range of P2 values from 1 to 10 and we have written a function count\_real\_roots which counts number of roots .If number of roots is equals to 2 we have added it to the pose counts and printed, as 4 comes first we can take 4 as answer.

{'2 poses': [4, 8, 9]}

Roots are: [1.331 1.777]



theta = 1.330999999999508

u2 = 2.6896848170899292

v2 = 4.667823571000907

u3 = 5.60384392217133

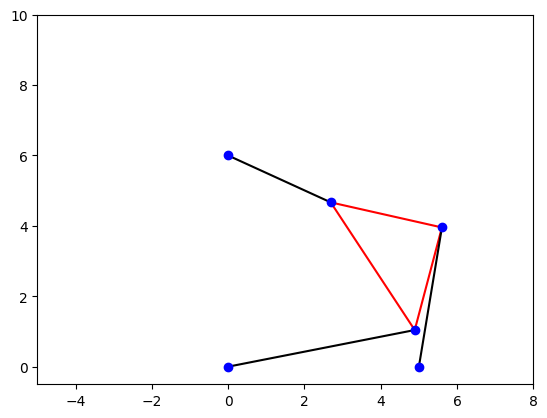
v3 = 3.9553092130605423

p1 = 5, P1 = 5

p2 = 4, P2 = 4

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



theta = 1.776999999999458

u2 = 1.348178392715623

v2 = 3.320756394512773

u3 = 4.28462413370659

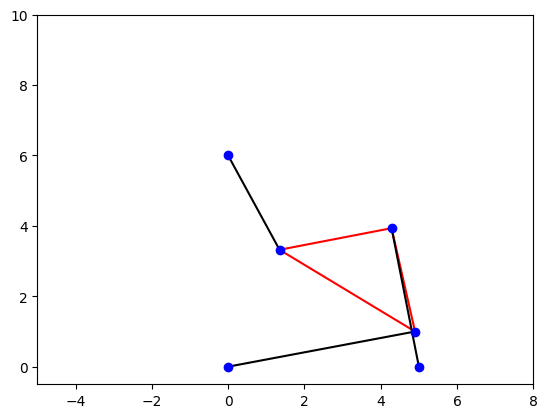
v3 = 3.934992839390489

p1 = 5, P1 = 5

p2 = 4, P2 = 4

p3 = 3, P3 = 3

successfully verified that p1,p2,p3 values are lengths of structs in the plot



**Activity 7.** Calculate the intervals in p2, with the rest of the parameters as in Step 4, for which there are 0,2,4, and 6 poses, respectively.

**Answer:** For 0 poses, P2 values are: 1, 2, 3

For 2 poses, P2 values are: 4, 8, 9

For 4 poses, P2 values are: 5, 6

For 6 poses, P2 values are: 7

For P2 = 4, root intervals in radians are:

For P2 = 4, root intervals in radians are:

Theta: 1.3264502315156905 to 1.3439035240356338 radians

Theta: 1.7627825445142729 to 1.7802358370342162 radians

For P2 = 5, root intervals in radians are:

Theta: -0.7330382858376184 to -0.7155849933176751 radians

Theta: -0.33161255787892263 to -0.3141592653589793 radians

Theta: 1.1344640137963142 to 1.1519173063162575 radians

Theta: 2.111848394913139 to 2.129301687433082 radians

For P2 = 6, root intervals in radians are:

Theta: -0.9773843811168246 to -0.9599310885968813 radians

Theta: 0.12217304763960307 to 0.13962634015954636 radians

Theta: 1.0297442586766545 to 1.0471975511965976 radians

Theta: 2.3387411976724017 to 2.356194490192345 radians

For P2 = 7, root intervals in radians are:

Theta: -0.6806784082777885 to -0.6632251157578453 radians

Theta: -0.3665191429188092 to -0.3490658503988659 radians

Theta: 0.03490658503988659 to 0.05235987755982989 radians

Theta: 0.4537856055185257 to 0.47123889803846897 radians

Theta: 0.9773843811168246 to 0.9948376736367679 radians

Theta: 2.5132741228718345 to 2.530727415391778 radians

For P2 = 8, root intervals in radians are:

Theta: 1.064650843716541 to 1.0821041362364843 radians

Theta: 2.530727415391778 to 2.548180707911721 radians

For P2 = 9, root intervals in radians are:

Theta: 1.53588974175501 to 1.5533430342749532 radians

Theta: 2.2863813201125716 to 2.303834612632515 radians

{'0 poses': [1, 2, 3], '2 poses': [4, 8, 9], '4 poses': [5, 6], '6 poses': [7]}

For 0 poses, P2 values are: 1, 2, 3

For 2 poses, P2 values are: 4, 8, 9

For 4 poses, P2 values are: 5, 6

For 6 poses, P2 values are: 7