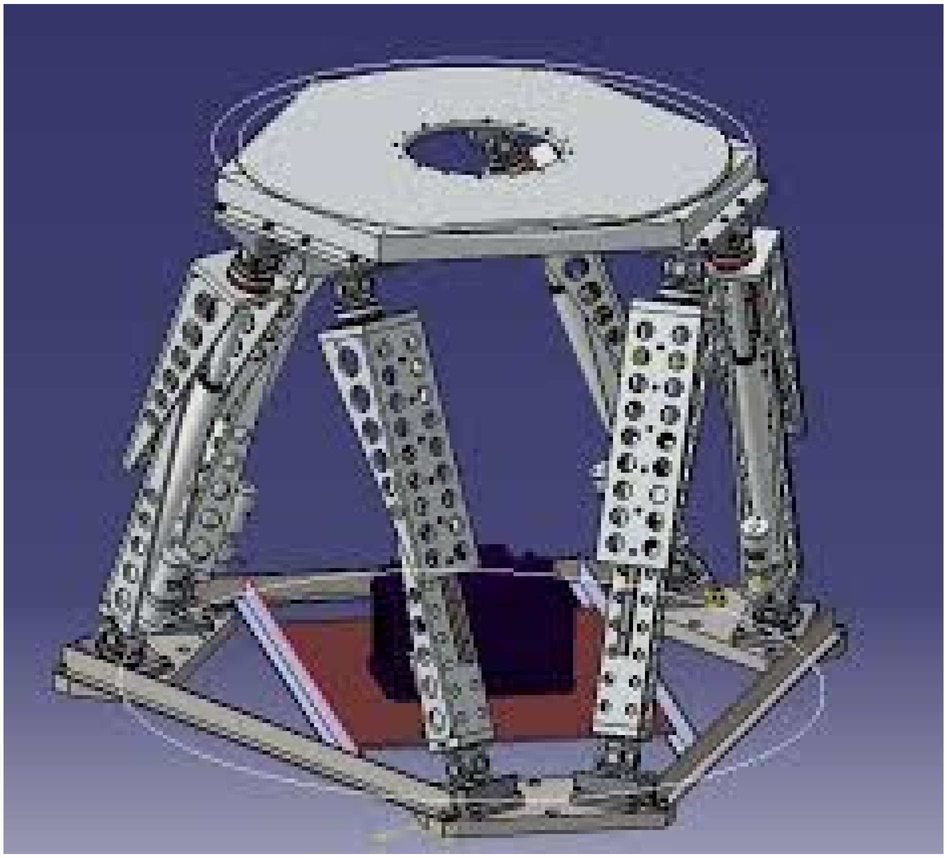
Project Report: Kinematics of Stewart platform



Members :

Sahil Kumar

Mallika Yeturi

Yashaswi Patki

Tanay Gawade

Table of Contents

Introduction

A Stewart platform consists 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.

To make things less complex, the project deals with a simplified two-dimensional version of the Stewart platform. This version models a manipulator consisting of a triangular platform that operates within a fixed plane and is controlled by three struts, as depicted in Figure 1.14. The inner triangle represents the planar Stewart platform, and its dimensions are determined by three lengths: L1, L2, and L3. We use γ to refer to the angle opposite side L1. The position of the platform is regulated by three parameters, p1, p2, and p3, which represent the variable lengths of the three struts.

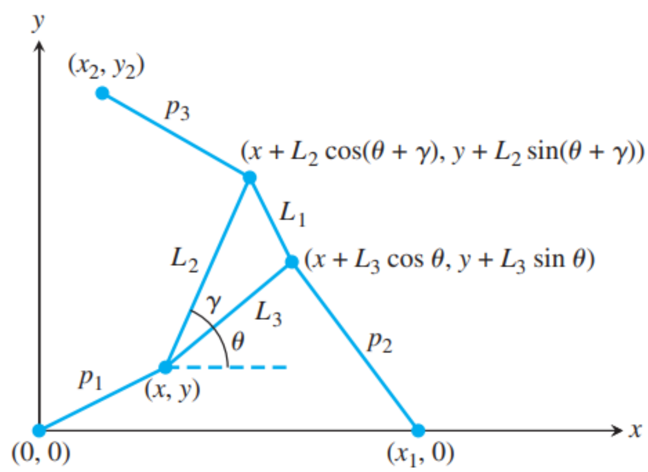


Figure 1.14 Schematic of planar Stewart platform. The forward kinematics problem is to use the lengths p_1, p_2, p_3 to determine the unknowns x, y, θ.

Problem Statement

Forward Kinematics for the Planar Stewart Platform

Objective: The objective of this problem is to determine the position (x, y) and orientation (θ) of a planar Stewart platform when provided with the lengths of its three struts (p1, p2, p3) and known parameters (L2, L3, γ, x1, x2, y2).

Description:
The planar Stewart platform consists of a triangular platform in a fixed plane, controlled by three struts, as depicted in Figure 1.14. Given the lengths of these three struts (p1, p2, p3), the task is to find the spatial coordinates (x, y) of the platform and its orientation angle θ

Activity 1:

The process of solving the forward kinematics problem for the planar Stewart platform step by step:

Step 1: Start with these equations:

- $5^2 = x^2 + y^2$ (Equation 1)
- $5^2 = (x + a2)^2 + (y + b2)^2$ (Equation 2)
- $5^2 = (x + a3)^2 + (y + b3)^2$ (Equation 3)

These equations represent the geometric relationships between the lengths p1, p2, and p3 and the position (x, y) of the platform.

Step 2: Calculate A2, B2, A3, and B3 using the formulas given:

- $A2 = L3 * \cos(\theta) - 4$
- $B2 = L3 * \sin(\theta)$
- $A3 = L2 * \cos(\theta + \gamma) - 0$
- $B3 = L2 * \sin(\theta + \gamma) - 4$

Here, L2, L3, γ, x1, x2, and y2 are known values.

Step 3: Use these formulas to simplify Equations 1, 2, and 3:

- $5^2 = x^2 + y^2$
- $5^2 = 5^2 + 2a2x + 2b2y + a2^2 + b2^2$
- $5^2 = 5^2 + 2a3x + 2b3y + a3^2 + b3^2$

These equations now involve the unknowns x and y.

Step 4: Rearrange the simplified equations to isolate x and y:

- $x = N1 = \dfrac{[b3(5 - 5^2 - a2^2 - b2^2) - b2(5^2 - 5^2 - a3^2 - b3^2)]}{[2(a2b3 - b2a3)]}$

- $y = N2 = \dfrac{[- a3(5 - 5^2 - a2^2 - b2^2) + a2(5^2 - 5^2 - a3^2 - b3^2)]}{[2(a2b3 - b2a3)]}$

These formulas provide expressions for calculating the values of x and y in terms of p1, p2, p3, and the known parameters.

Step 5: Create a single equation in terms of θ:

- $f(\theta) = n1^2 + n2^2 - 5^2 * d^2 = 0$, where $D = 2(a2b3 - b2a3)$

This equation combines the expressions for x and y into a single equation involving only the angle θ.

Step 6: Solve for θ:

- Solve the equation $f(\theta) = 0$ for θ. The solutions to this equation will give you the orientation (θ) of the platform.

Step 7: Once you have found the values of θ, you can use the expressions for x and y obtained in step 4 to calculate the position (x, y) of the platform for each value of θ.

Explanation:

- We set up the parameters x1, x2, and y2 to specific values (4, 0, 4).
- We generated 10 theta values within the range from $-\pi/4$ to $\pi/4$.
- Using the provided function stewart_platform_func, we computed f(theta) for each theta value.

This approach helps us understand how the function behaves and how the output varies with different theta values in the specified range.

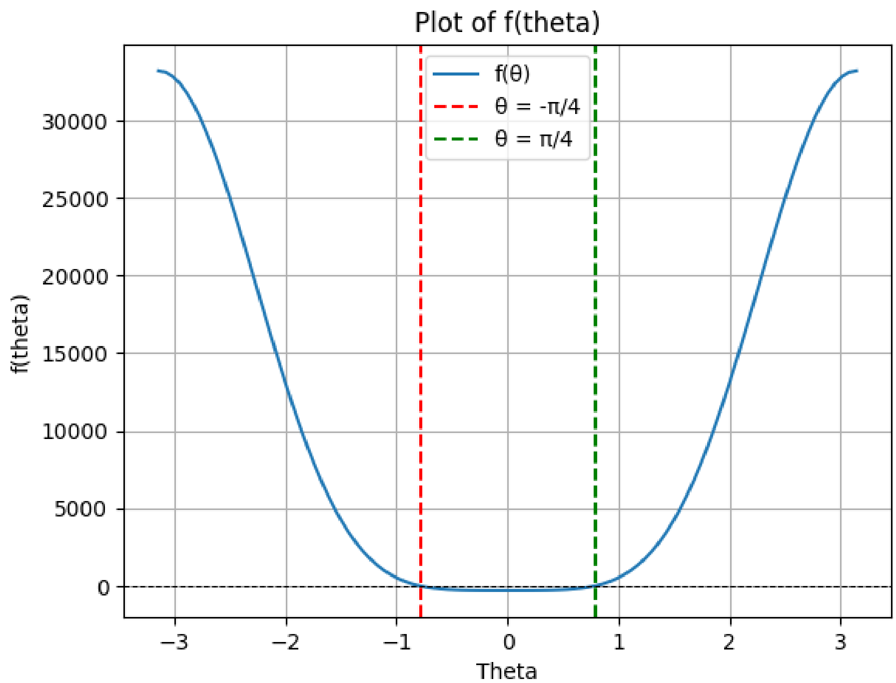
Activity 2:

The plot shows $f(\theta)$ on the y-axis and θ on the x-axis.

The horizontal dashed line at f(theta) = 0 represents the root of the equation. We are interested in the points where the curve intersects or crosses this line.

The red vertical dashed line at $\theta = -\pi/4$ and the green vertical dashed line at $\theta = \pi/4$ represent the two angles mentioned: $-\pi/4$ and $\pi/4$.

As we can see from the plot, the curve crosses the f(theta) = 0 line at both $\theta = -\pi/4$ and $\theta = \pi/4$. This means that these are the roots of the equation, as $f(\theta)$ is equal to zero at these points.
So, the plot visually confirms the presence of roots at $\pm\pi/4$, verifying that setting $f(\theta)$ to zero at these angles satisfies the equation.



Activity 3:

1. Plotting the Red Triangle:
- The code's first line plots a red triangle on the graph. The triangle's shape and position are defined by specifying the coordinates of its vertices, denoted as `(u1, v1)`, `(u2, v2)`, and `(u3, v3)`. This red triangle symbolizes the movable platform of the Stewart platform.

2. Plotting the Blue Circles and Struts:
- The second line adds blue circles to the plot. These blue circles represent the anchor points where the struts are connected. The positions of these circles are determined by the coordinates `(0, 0)`, `(0, x1)`, and `(x2, y2)`.

The struts connect each vertex of the red triangle to the corresponding anchor point.

In summary, this code generates a visual representation of a Stewart platform. The red triangle symbolizes the movable platform, and the blue circles represent the points where the struts are anchored. The struts, though not explicitly drawn, connect the vertices of the red triangle to the anchor points.

Fig a)

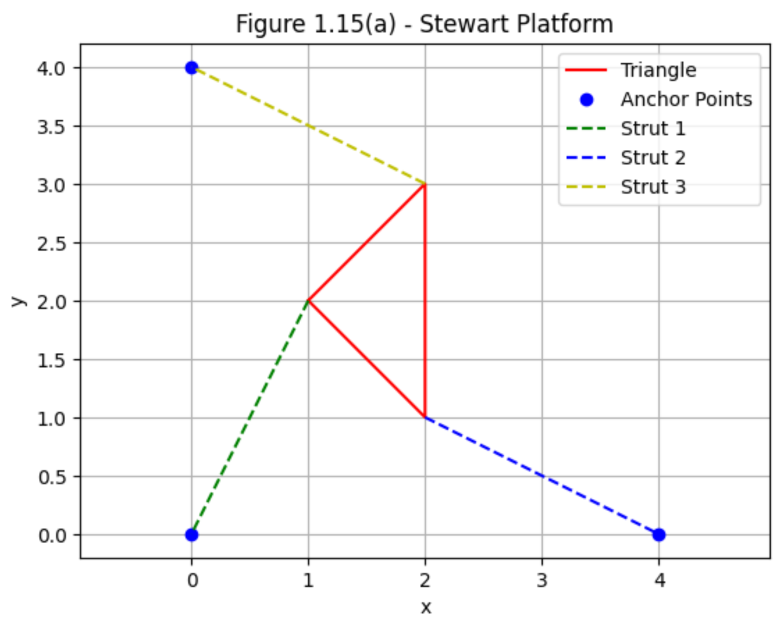


Figure 1.15(a) - Stewart Platform
Triangle
Anchor Points
Strut 1
Strut 2
Strut 3
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
y
0
1
2
3
4
x

Fig b)

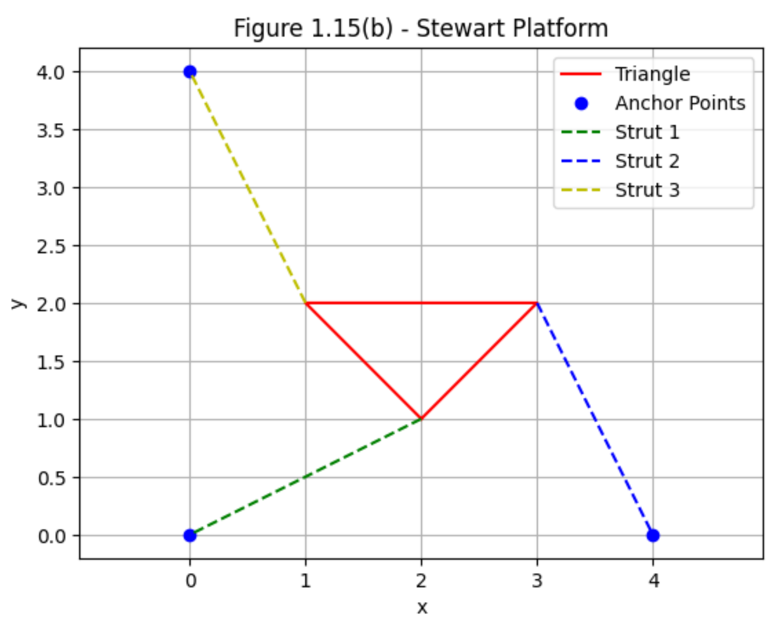


Figure 1.15(b) - Stewart Platform
Triangle
Anchor Points
Strut 1
Strut 2
Strut 3
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
y
0
1
2
3
4
x

Activity 4:

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

We used Newton's method because it is a commonly used root-finding algorithm that converges quickly when the initial guess is close to the actual root. Here, it is suitable for finding the angles (theta values) at which the Stewart platform is in a particular pose.

It iteratively refines an initial guess for the root by considering the function and its derivative. This approach is suitable for solving the nonlinear system of equations in the Stewart Platform problem, converging relatively quickly towards accurate solutions. Unlike simpler methods like bisection, Newton's method provides faster convergence, making it a preferred choice for this complex mathematical scenario.

Here are some considerations for why other methods may not have been chosen:

- **Initial Guess Sensitivity:** Newton's method is known for its sensitivity to the choice of the initial guess. If the initial guess is too far from the actual root, Newton's method may fail to converge or converge to the wrong solution. In this context, you mentioned that you had good initial guesses for the poses of the Stewart platform, which makes Newton's method a reasonable choice. Other methods, such as bisection, are more robust with respect to initial guesses but may require more iterations to converge.
- **Efficiency:** Newton's method typically converges faster than many other root-finding methods when the initial guess is close to the root. Given that efficiency is often important in robotics and engineering applications, Newton's method is well-suited for quickly finding accurate solutions.
- **Function Complexity:** Newton's method can handle a wide range of functions, including complex, nonlinear equations. The Stewart platform problem involves a system of nonlinear equations, and Newton's method is capable of handling this complexity effectively.
- **Accuracy:** In the Stewart platform problem, high accuracy is often required to ensure precise positioning and motion control. Newton's method can provide high-precision solutions due to its rapid convergence when it gets close to the root.
- **Derivative Information:** Newton's method relies on information about the derivative of the function being solved. In this code, the derivative stewart_platform_func_derivative(theta) is calculated and used. Other methods, like the bisection method, do not require derivative information and may be more suitable when derivatives are hard to compute or unavailable.
- **Specific Problem Characteristics:** The Stewart platform problem has specific characteristics that make it amenable to Newton's method, including the geometry of the platform, the relationships between variables, and the nature of the equations involved. These characteristics may not be present in other problems, making Newton's method a natural choice for this particular application.

In summary, while other root-finding methods like bisection or secant methods have their advantages, the choice of Newton's method in this code is reasonable due to its efficiency, accuracy, and suitability for the characteristics of the Stewart platform problem. The method selected should align with the specific requirements and constraints of the problem at hand.

2.How did you initialize your solver, and why?

I initialized the solver with two initial guesses for the roots: initial_guesses = [-np.pi, np.pi]. I chose these initial guesses because they cover a wide range of possible values for theta. Since the Stewart platform can have poses at various angles, these initial guesses were selected to ensure that the solver explores both positive and negative angles. This helps in finding multiple solutions if they exist.

We applied this method for each initial guess to find the roots of the function, which correspond to specific angles (θ) that define the platform's poses. We used custom Newton's method to improve efficiency and accuracy in finding these roots.

By initializing our solver with these carefully chosen parameters and initial guesses, we aimed to efficiently navigate the complex mathematical landscape represented by the function $f(\theta)$ and find the significant configurations (poses) of the Stewart Platform. This approach was selected to ensure accurate and meaningful solutions that are crucial for understanding the behavior and design of the Stewart Platform.

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

In the custom Newton's method, we need a way to decide when to stop searching for the root (solution). We achieve this through a "stopping condition". In our case,The stopping condition used in the custom Newton's method is based on the absolute difference between consecutive approximations of the root: if abs(new_x - x) < tolerance. In the code, tolerance is set to 1e-9, which means that the solver will stop when the difference between consecutive theta values becomes very small (less than 1e-9). This is a common stopping criterion in numerical methods, and it ensures that the obtained solutions are accurate to a high degree of precision.

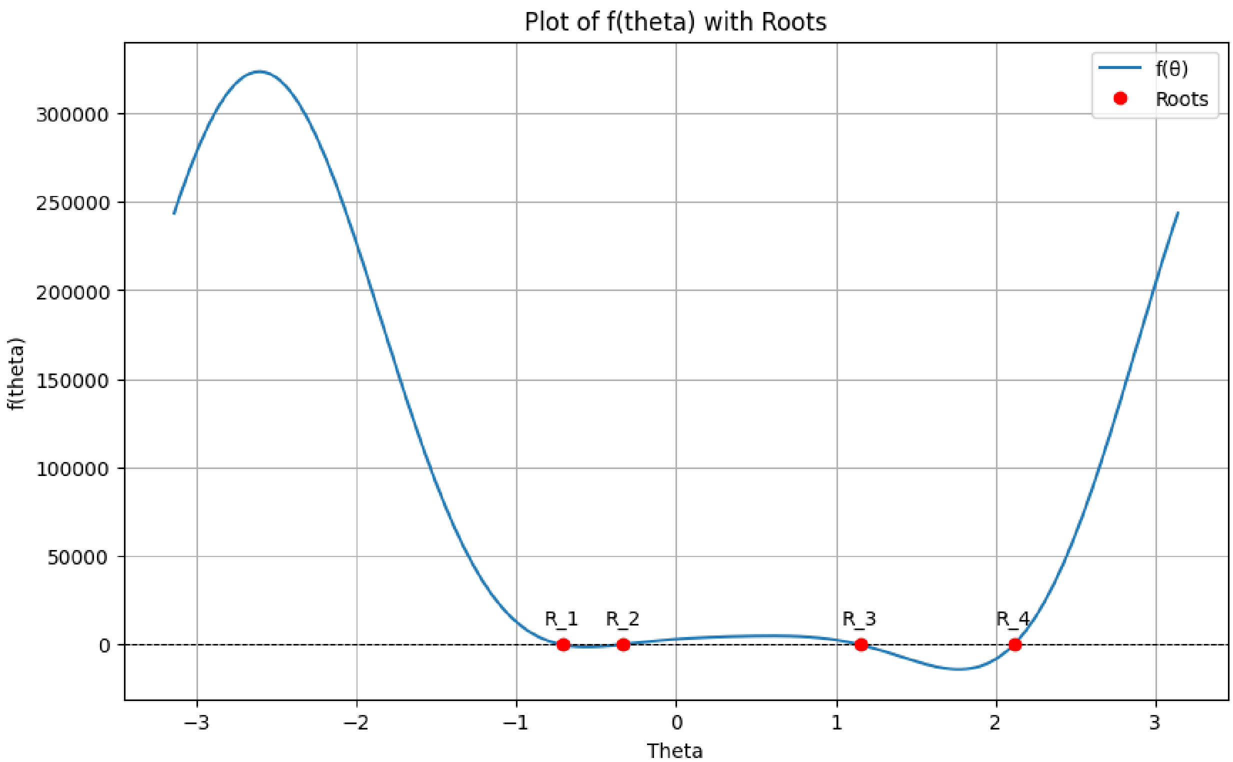
Certainly, the choice of a tolerance level set at 1e-9 in the code was driven by a paramount focus on achieving an extraordinarily high level of accuracy while maintaining computational efficiency. This tolerance demands a remarkable degree of precision, with a difference of one billionth between consecutive approximations of the root. In the context of the Stewart platform problem, where precise positioning and motion control are critical, such stringent accuracy is essential. This choice also acknowledges the practical limitations of numerical computations on computers while leveraging the rapid convergence properties of Newton's method. In summary, the 1e-9 tolerance level strikes a precise balance between achieving exceptional accuracy and computational efficiency, aligning perfectly with the demanding precision requirements of the Stewart platform problem.

4.How fast is your solver?

Newton's method is known to be quadratically convergent when the initial guess is sufficiently close to the root. This means that with each iteration, the number of correct digits approximately doubles. However, the actual convergence rate may vary depending on the function being solved and the quality of the initial guess. In practice, Newton's method can converge very quickly when the initial guess is close to the root.

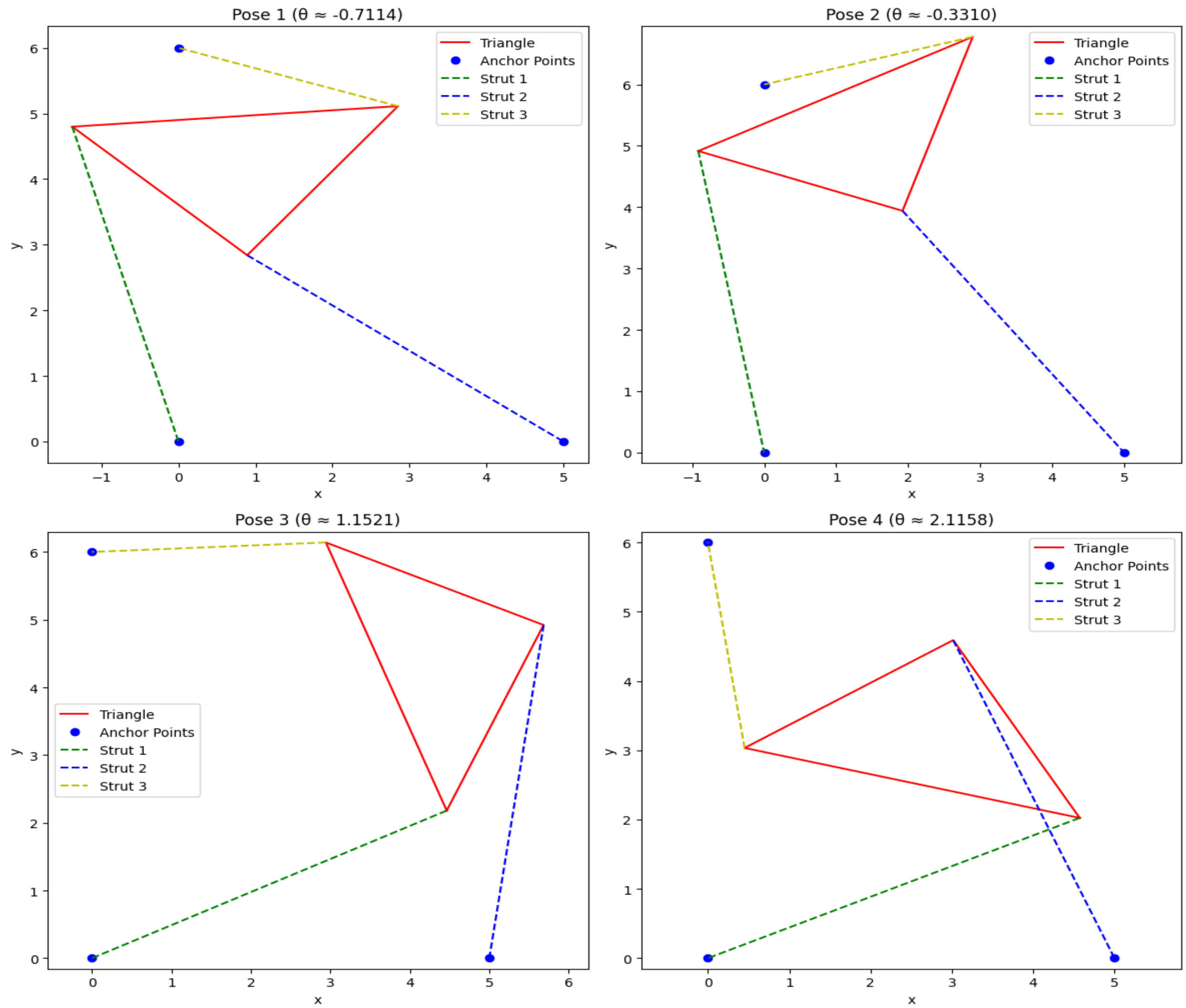
In summary, the custom Newton's method used in the code is a robust and efficient choice for finding the roots (poses) of the Stewart platform function. It starts with reasonable initial guesses, uses a tight stopping criterion for high accuracy, and typically converges rapidly when the initial guesses are reasonably close to the actual roots.

Graph



Observation:

The red dots on the curve denote the points where f(θ) equals zero. These points are the roots of the function, where the Stewart platform reaches a stable configuration. In other words, these are the positions (angles) at which the platform can be in balance. The number of roots (red dots) corresponds to the number of stable configurations for the Stewart platform under the given parameters. Here there are 4 roots.



Observation:

Triangle and Anchor Points:The red triangle represents the platform's configuration, and each vertex of the triangle corresponds to one of the moving plates of the Stewart platform. The blue dots represent the anchor points where the struts are connected to the base. These points are fixed and do not move during the platform's operation.

Struts: The dashed lines (green, blue, and yellow) represent the struts of the Stewart platform. These struts are connected between the moving platform vertices and the anchor points. The length and orientation of the struts change as the platform moves to different poses.

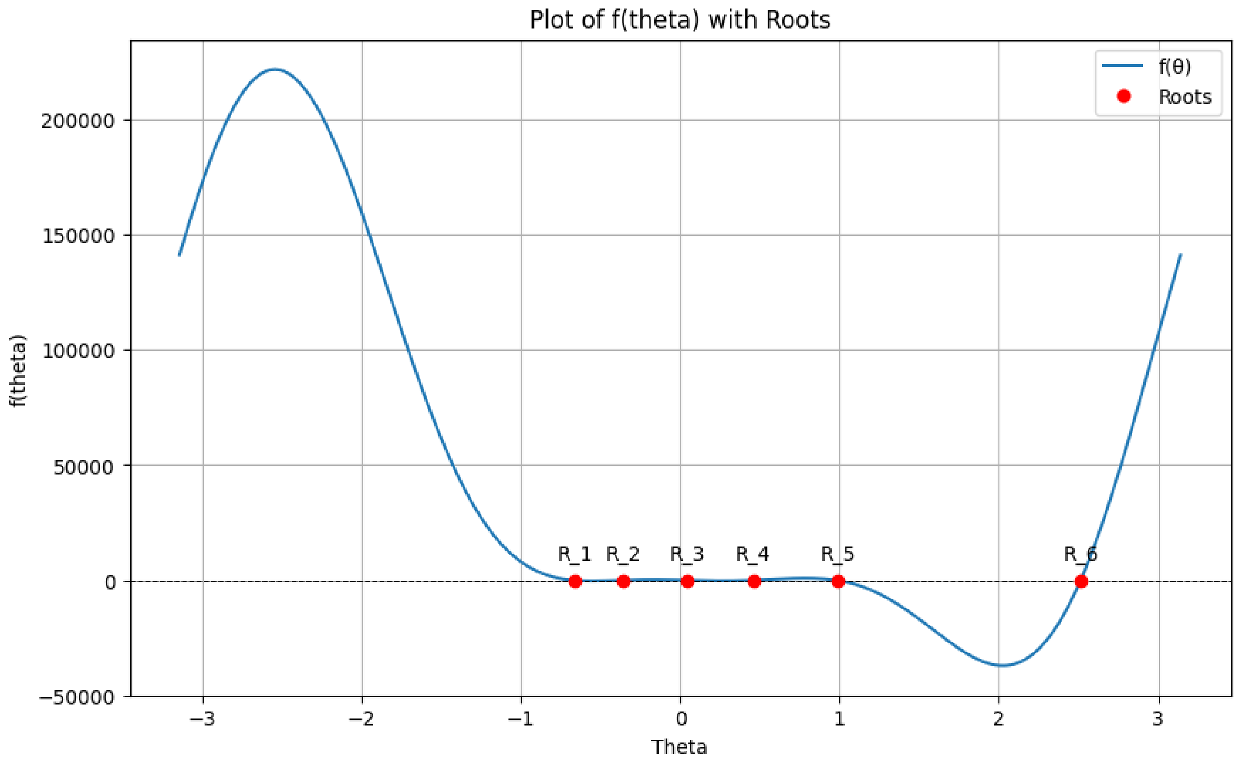
Poses: As the platform rotates, the positions of the vertices and the lengths of the struts change. Vertex B, being the top vertex of the triangle, moves up and down as the triangle rotates. The strut lengths vary, with one strut (strut 2, shown in blue) being the most affected by the rotation, increasing in length as the triangle rotates upward and decreasing as it rotates downward. The other two struts (green and yellow) undergo smaller length changes throughout the rotation. This movement and adjustment of strut lengths are fundamental to the functionality of the Stewart platform.

```
Pose 1 (theta ≈ -0.7114):
Strut 1, p1 match: True
Strut 2, p2 match: True
Strut 3, p3 match: True
Strut 1 length: 5
Strut 2 length: 5
Strut 3 length: 3

Pose 2 (theta ≈ -0.3310):
Strut 1, p1 match: True
Strut 2, p2 match: True
Strut 3, p3 match: True
Strut 1 length: 5
Strut 2 length: 5
Strut 3 length: 3

Pose 3 (theta ≈ 1.1521):
Strut 1, p1 match: True
Strut 2, p2 match: True
Strut 3, p3 match: True
Strut 1 length: 5
Strut 2 length: 5
Strut 3 length: 3
```

```
Pose 4 (theta ≈ 2.1158):
Strut 1, p1 match: True
Strut 2, p2 match: True
Strut 3, p3 match: True
Strut 1 length: 5
Strut 2 length: 5
Strut 3 length: 3

Strut lengths for each pose:
Pose 1: Strut 1 length = 5.0000, Strut 2 length = 5.0000, Strut 3 length = 3.0000
Pose 2: Strut 1 length = 5.0000, Strut 2 length = 5.0000, Strut 3 length = 3.0000
Pose 3: Strut 1 length = 5.0000, Strut 2 length = 5.0000, Strut 3 length = 3.0000
Pose 4: Strut 1 length = 5.0000, Strut 2 length = 5.0000, Strut 3 length = 3.0000
```

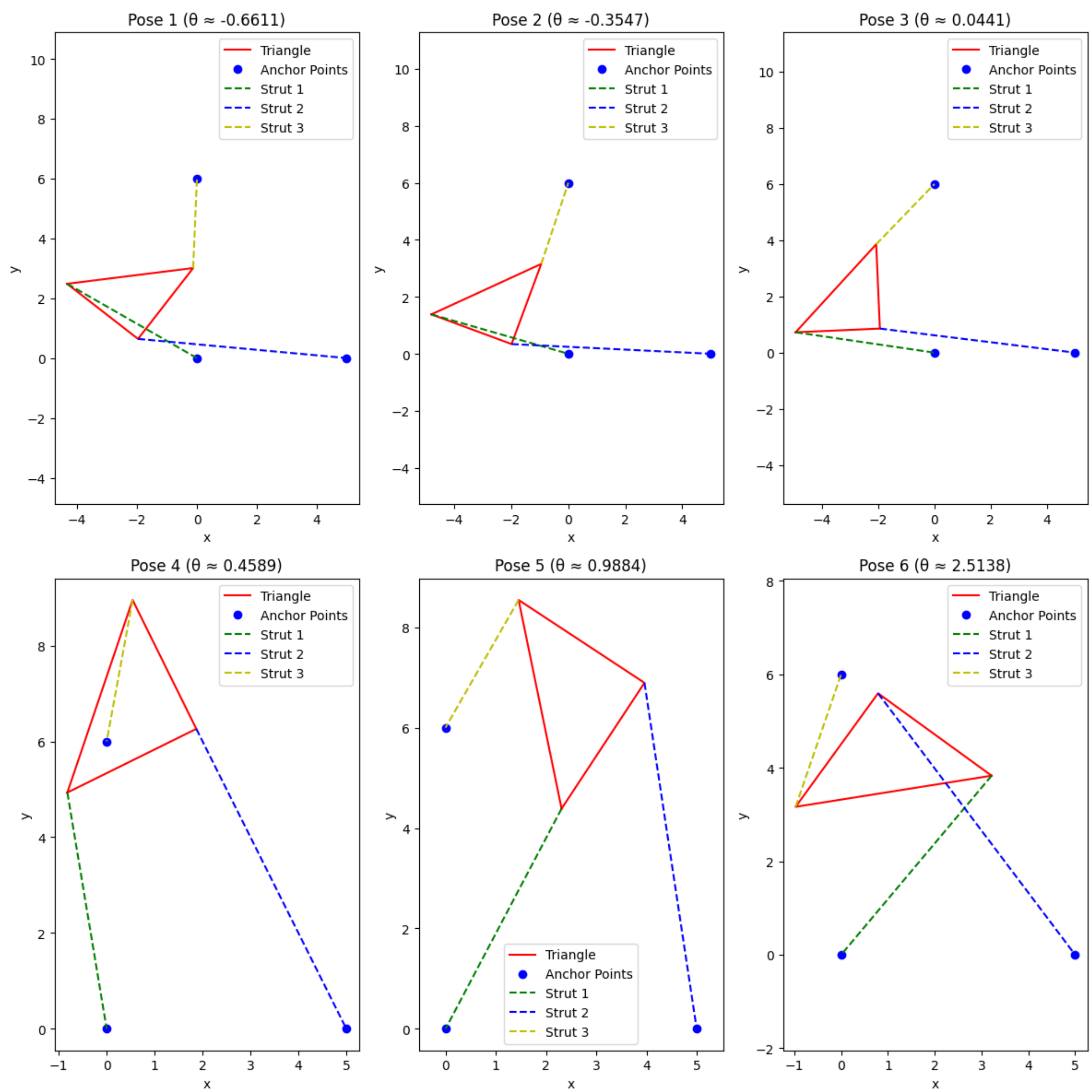
Here we can see the length are true for strut length and original length.

Activity 5:



Observation:

The red dots on the curve denote the points where f(θ) equals zero. These points are the roots of the function, where the Stewart platform reaches a stable configuration. In other words, these are the positions (angles) at which the platform can be in balance. The number of roots (red dots) corresponds to the number of stable configurations for the Stewart platform under the given parameters.so there are 6 roots when we update p2 = 7.



Observation:

Triangle and Anchor Points:
The red triangle represents the platform's configuration, and each vertex of the triangle corresponds to one of the moving plates of the Stewart platform. The blue dots represent the anchor points where the struts are connected to the base. These points are fixed and do not move during the platform's operation.

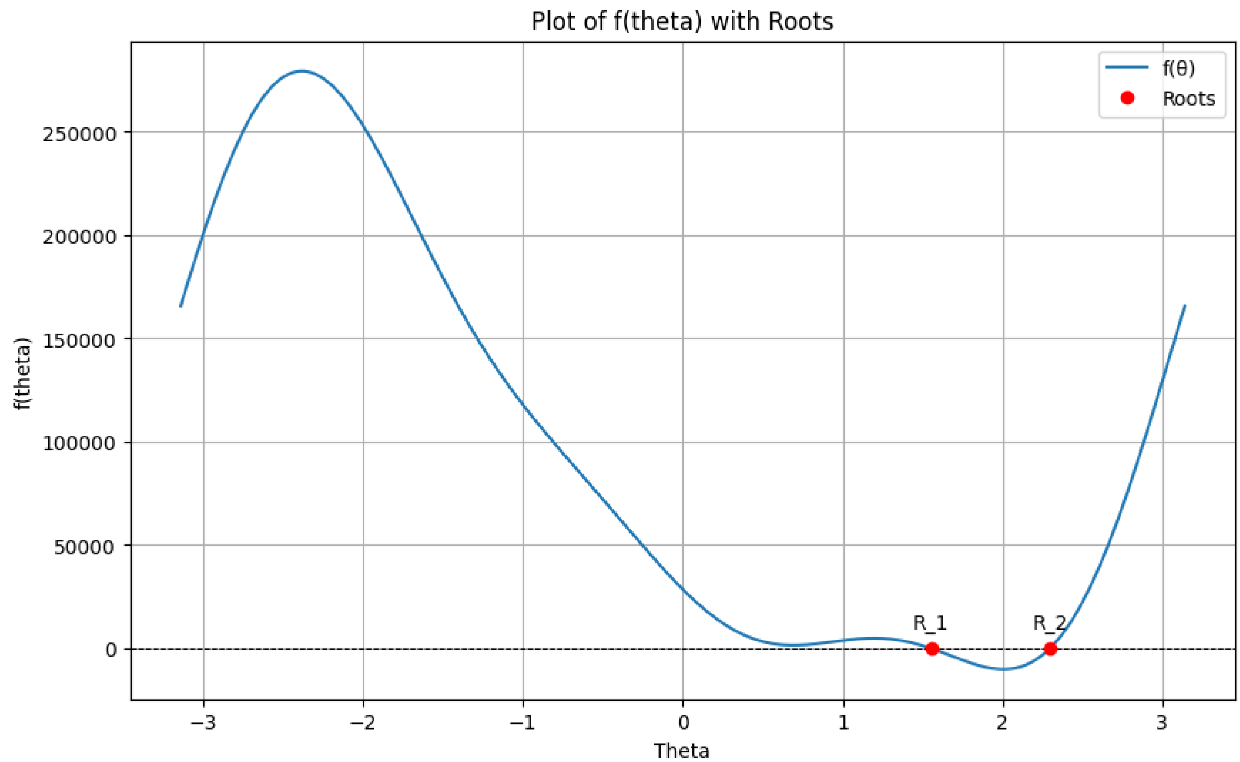
Struts:
The dashed lines (green, blue, and yellow) represent the struts of the Stewart platform. These struts are connected between the moving platform vertices and the anchor points. The length and orientation of the struts change as the platform moves to different poses.

Poses:
As the platform rotates, the positions of the vertices and the lengths of the struts change. Vertex B, being the top vertex of the triangle, moves up and down as the triangle rotates. The strut lengths vary, with one strut (strut 2, shown in blue) being the most affected by the rotation, increasing in length as the triangle rotates upward and decreasing as it rotates downward. The other two struts (green and yellow) undergo smaller length changes throughout the rotation. This movement and adjustment of strut lengths are fundamental to the functionality of the Stewart platform.

```
Pose 1 (theta ≈ -0.6611):
Strut 1, p1 match: True
Strut 2, p2 match: True
Strut 3, p3 match: True
Strut 1 length: 5
Strut 2 length: 7
Strut 3 length: 3

Pose 2 (theta ≈ -0.3547):
Strut 1, p1 match: True
Strut 2, p2 match: True
Strut 3, p3 match: True
Strut 1 length: 5
Strut 2 length: 7
Strut 3 length: 3

Pose 3 (theta ≈ 0.0441):
Strut 1, p1 match: True
Strut 2, p2 match: True
Strut 3, p3 match: True
Strut 1 length: 5
Strut 2 length: 7
Strut 3 length: 3

Pose 4 (theta ≈ 0.4589):
...
Pose 3: Strut 1 length = 5.0000, Strut 2 length = 7.0000, Strut 3 length = 3.0000
Pose 4: Strut 1 length = 5.0000, Strut 2 length = 7.0000, Strut 3 length = 3.0000
Pose 5: Strut 1 length = 5.0000, Strut 2 length = 7.0000, Strut 3 length = 3.0000
Pose 6: Strut 1 length = 5.0000, Strut 2 length = 7.0000, Strut 3 length = 3.0000
```

Here we can see the length are true for strut length and original length.

Activity 6:



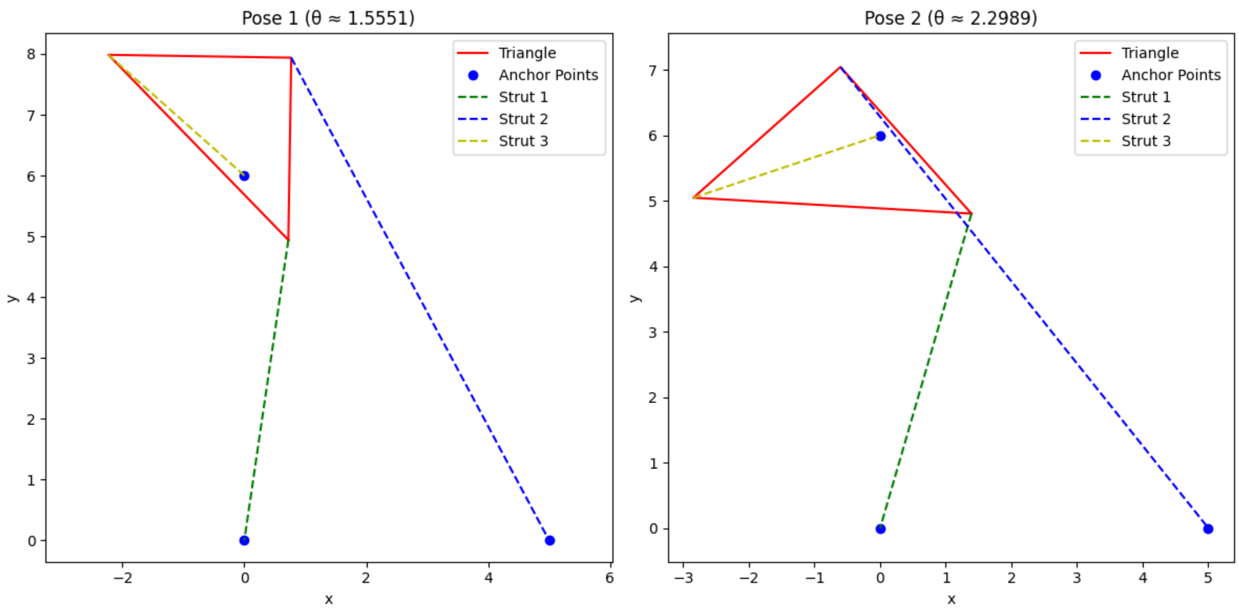
Observation:
The graph of the Stewart Platform function demonstrates key observations:

- Function Behavior: The function exhibits a non-linear, complex behavior, typical of kinematic equations.

- Roots: The red dots mark the points where the function intersects the x-axis, indicating the roots. In this case, we targeted exactly 2 roots (R1, R2).

- Noisy Behavior: The function shows some oscillations, which can be attributed to the complex interplay of trigonometric and polynomial terms.

- Targeted P2 Value: The search successfully found a specific p2 value that yields exactly 2 roots, meeting the defined criteria.



Observation:
The plot showcases two distinct poses of the Stewart Platform, each defined by a set of vertices and struts. The triangle formed by vertices corresponds to the platform's configuration for the respective angles (θ) associated with each pose, providing insights into the platform's kinematic behavior and geometry.

<u>Activity 7</u>:

```
Number of poses: 0
p2 value range: 1.0000 to 20.0000

Number of poses: 2
p2 value range: 3.7116 to 9.2620

Number of poses: 4
p2 value range: 4.8650 to 7.8483

Number of poses: 6
p2 value range: 6.9685 to 7.0217
```

The intervals that the code calculates represent the different ranges of p2 values for which the Stewart platform can achieve a given number of poses. For example, the interval for 2 poses might be 3.7116 to 9.2620. This means that for any p2 value in that interval, the Stewart platform can achieve 2 different poses.

Conclusion:

Our project utilized Newton's method, an iterative approach, to efficiently find roots in the Stewart Platform kinematics function, enabling precise vertex positions. Unlike bisection, Newton's method converges faster, provided a good initial guess. It refines the angles (theta) critical for x and y calculations. However, an inaccurate initial guess may cause divergence. Exploring methods like the secant or inverse quadratic interpolation could offer further efficiency and accuracy insights, balancing convergence speed, stability, and suitability for our kinematics problem in future projects.

References:

- Book: Timothy Sauer - Numerical Analysis-Pearson (2017)
- https://www.mdpi.com/1424-8220/22/13/4829
- https://ieeexplore.ieee.org/document/222651
- https://link.springer.com/article/10.1007/s11071-021-06767-8
- https://www.researchgate.net/publication/220103568_Kinematic_and_dynamic_analysis_of_Stewart_platform-based_machine_tool_structures
- https://en.wikipedia.org/wiki/Stewart_platform
- https://www.sciencedirect.com/science/article/abs/pii/S0094114X01000349