Stewart Platform Project Report

Initial CAD Model

Progress Made:

- The project's first step was to create a basic CAD model in Solidworks to learn about the structure of the Stewart Platform.
- The main components of the CAD model were the base and top plate, motors, ball joints, rods, and screws.
- Below is the image of the initial CAD model.

Shortcomings:

- The movement of the Stewart platform was not correctly defined.
- The rod and servo arms had no constraints, so they collided multiple times.

- Due to the random dimensions of the components, the ratio of Stewart platform components was improper, which affected the Stewart platform's motion.

Gazebo Simulation:

Progress Made:

- Gazebo simulator uses URDF description of the robot model, so the CAD model was converted in URDF using SWURDF extension.
- After configuring the joints of the Stewart platform, it was loaded in Gazebo.

Shortcoming:

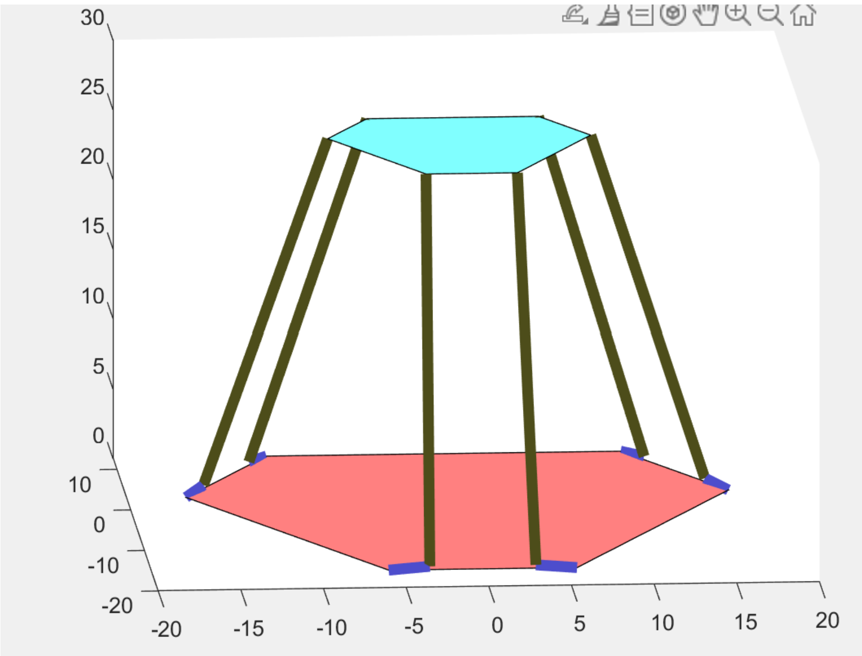
- In the URDF description of a robot model, joints are represented as a parent-child link. For example, the base link is the parent and the wheel link in the child, and their joint type will be the revolute joint. A single parent link can have multiple child links, but it is impossible to have multiple parents and single child links. In the case of the Stewart platform, the top plate can be considered a child link, and the 6 rods connected to the top plate can be regarded as six parent links.
- Therefore it was not possible to control the Stewart platform in Gazebo Simulator.

MATLAB simulation:

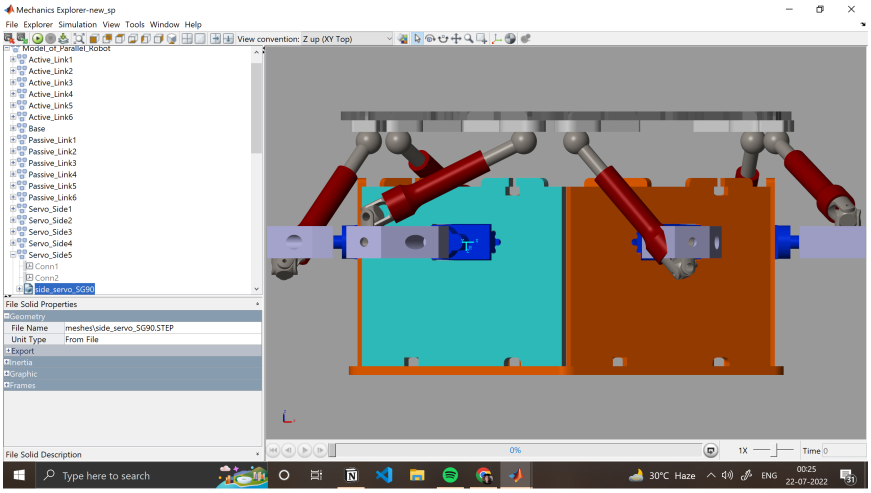
Progress Made:

- There are two types of Stewart platforms - actuator-based and motor based.
- To create the MATLAB simulation first task was to understand the kinematics of the Stewart platform.

- The motion of the Stewart platform is explained by inverse kinematics. Given the desired robot's end-effector positions, inverse kinematics (IK) can determine an appropriate joint configuration for which the end-effectors move to the target pose. Here the end-effector is the top plate of the Stewart platform.
- The kinematics of the actuator-based Stewart platform is explained here
- The kinematics and code explanation of motor based Stewart platform is explained here.
- MATLAB simulation image of Stewart platform.



- A function calculate_servo_angle takes six input parameters: pitch, roll, yaw, x, y, and z.
- From inverse kinematics, the function calculates the angle of each servo motor to reach that configuration.
- The function calculates the coordinates of the components and plots them as shown above.
- Simulink simscape multibody toolbox provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, etc.
- The Stewart platform was simulated in Simulink and controlled by the MATLAB script created earlier.



The Simscape multibody model of the Stewart platform

- The next step after manually controlling the Stewart platform was integrating it with the hand gesture recognition AI model.
- The integration of the AI model written in Python language and MATLAB was done using the MATLAB engine.
- The initial gesture recognition AI model had four gestures: front, back, right, and left. Front and back gestures were used to increase the pitch angle, whereas the left and right gestures were used to improve the roll angle by 1 degree.

Shortcomings:

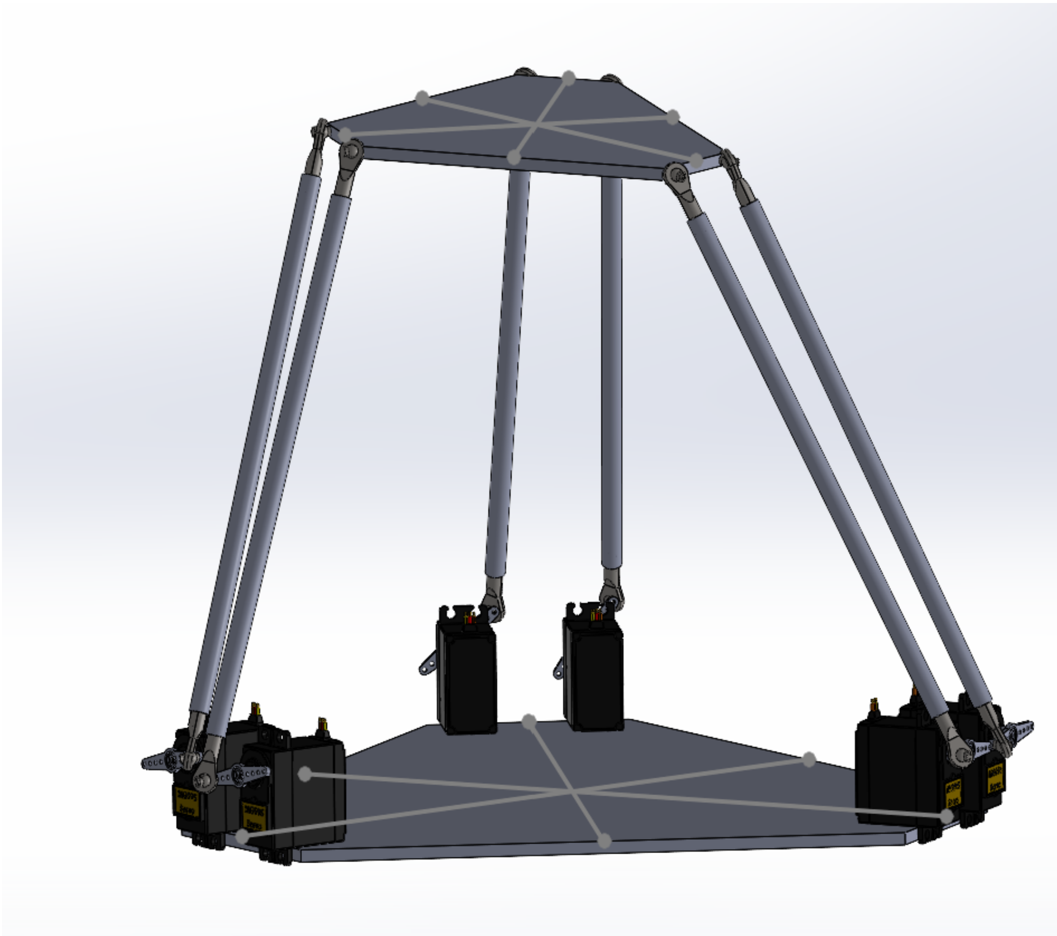
- The CAD model made in Solidworks could not be imported to the Simscape multibody because only a few joint types were allowed in Simscape multibody.
- The AI model could not be integrated with the Simulink model because there was no direct/ indirect way to control the Simulink model from the Python script.

Physical Prototype:

Progress made:

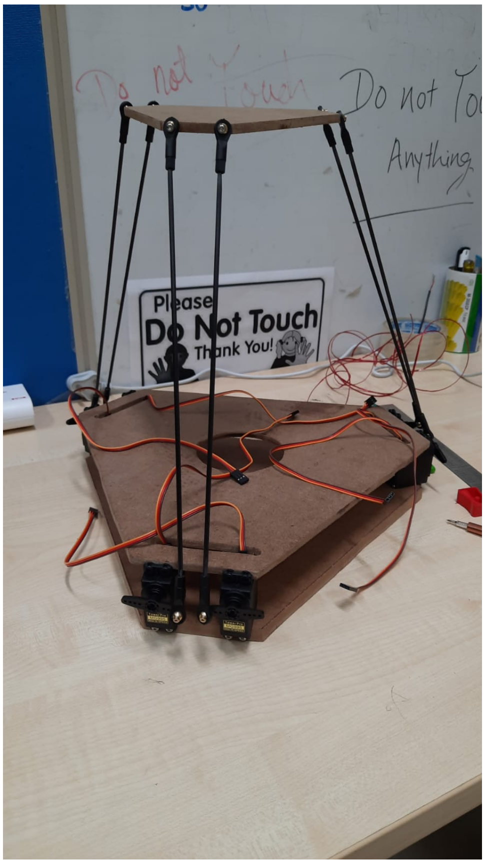
- After completion of the simulation, necessary components like motors, acrylic sheets, ball joints, rods, etc., were ordered. A detailed list of parts can be found here.
- Servo motors were selected based on their stall torque and limiting angles.

- Before the physical prototype of the Stewart platform, a final CAD model was made with the proper dimensions of the parts. The dimensions of the parts were decided from the motion analysis of the stewart platform simulation made in MATLAB.
- Final CAD Model:

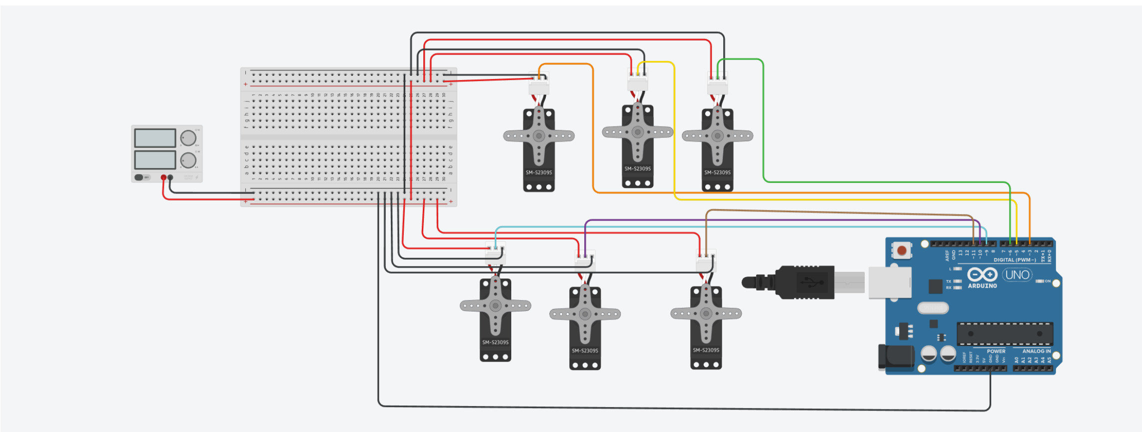


- The final prototype was made using the above CAD model as a reference.
- Part Dimensions:

Part	Dimension(mm)	Material
Top plate	120x40(PD=80.8)	plywood (6mm)
Bottom plate	240x80 (BD=160.66)	plywood(12mm)
Rods	230	Carbon fiber
M3 ball joints	3mm diameter	plastic
servo arm-full length	4mm	plastic



- After assembling all the parts together next step was to control the motor angles using an Arduino board.
- The Arduino board can supply a maximum 5 V voltage which is not enough to control six motors simultaneously. Six servo motors require 5.5V-6V of voltage and 0.7A current.
- The DC power supply was used to provide the motors with the necessary voltage and current.
- Circuit Diagram:



- By Using MATLAB script as a reference, Arduino code was written to calculate the servo angles.
- Below is Arduino code used to control the Stewart Platform.
- The ***calculations*** function takes translational and rotational values as input values and out six servo angles. Using the ***Servo.h*** library, the servo angles are written to the servos.
- The isnan() function checks whether any output values are nan. If any of the out values is nan, the Stewart platform will not move.
- After controlling the Stewart platform from Arduino code, an IMU sensor (MPU6050) was installed in the Stewart platform to analyze its movements.

Shortcomings:

- The acrylic sheet ordered online had 2mm thickness which was less than the diameter of the screws.
- While ordering the materials , the length of metal rods was not fixed. After making the CAD model the metal rods were bought from local shops. The diameter of the threaded metal rod was 8mm which was greater than the required diameter to fit in the ball joints.
- While testing the motors one of the motors failed and no extra motor was available.
- The Stewart platform has jerky movements due to high rpm of motors and loose fitting of the screws to the top plate.
- Because of think and bad quality plywood used in the top plate the screws are loosely fitted.
- The metal rods were unavailing so carbon rods of 3mm diameter were ordered. But their diameter was greater than the M3 ball joint groove diameter. Therefore they were manually grinded to reduce diameter.
- Difference in final rod lengths due to improper grinding of the Carbon rods.
- The yaw angle of the IMU sensor changed every time the code is uploaded because of which it is difficult to know the z-axis rotation of the Stewart platform.

- Unable to combine Stewart platform code and MPU6050 code in single .ino file due to FIFO error.