Balancing Stewart Platform

Omar Hoblos, Abhishek Kashyap, & Namrita Madhusoodanan

***Abstract*—The use of parallel manipulators can be advantageous for applications that require high rigidity and precise motion. A Stewart Platform is an example of a parallel manipulator that is able to achieve six degrees of freedom using six prismatic joints. This paper describes the control architecture of a modified version of the Stewart Platform to be used on a cruise ship as a pool table with the goal of keeping the balls stationary despite the constant motion while sailing. The platform is equipped with an Inertial Position Sensor that measures the 3-D time-varying transformation and its derivatives between any two coordinate frames attached to it. Each actuator is also equipped with a sensor to read position and velocity. A PID controller is used to ensure that the surface of the platform's top plate is always perpendicular to the direction of the gravity vector (which is assumed to be in the negative *z* direction).**

I. Introduction

Parallel manipulators have become increasingly popular in the recent years due to robotic applications that require high rigidity. A parallel manipulator can be defined as a closed-loop mechanism in which the end-effector is connected to the base by at least two independent kinematic chains [1].

Parallel manipulators have numerous advantages over serial manipulators, the most important one being a high nominal load per weight ratio, i.e. parallel manipulators have a much higher payload compared to their serial counterparts of the same mass. This is so because of the geometry of a parallel manipulator, which is essentially a platform supported by parallel independent links. In such a configuration, the weight on the platform is supported by all the parallel links, thus distributing the weight. Also, the high rigidity of parallel manipulators makes them highly suitable for applications that require high precision and speed, such as in flight simulators.

Parallel manipulators have other interesting features as well. The process of finding the link lengths for a given configuration of the mobile top platform, also known as inverse kinematics, is very straightforward. On the other hand, inverse kinematics for serial manipulators is quite complex at times, as there may be more than one solution [4, 7]. In the case of parallel manipulators, this is true when it comes to calculating forward kinematics, as a fixed set of link lengths result in many different platform configurations.

Force sensors can easily be added to parallel manipulators. This is so because the measurement of the traction-compression stress in the links enables to calculate the forces and torques acting on the mobile platform [1].

Parallel manipulators have certain disadvantages as well. They have limited workspace. Also, obtaining the dynamic model of a parallel manipulator in a closed form is a very difficult process, and has been a topic of research. There have been various algorithms proposed to address this problem, which include Newton-Euler, Lagrange, and Virtual-Work method [2].

An example of a parallel manipulator is the Stewart Platform which has 6 prismatic joints attached in pairs to three positions on the platform's base plate and top plate. Stewart Platforms (SPs) have a wide variety of uses in diverse fields. Most notably, the SP is frequently utilized by NASA in specific applications such as flight simulators and precise positioning of large telescopes. SPs are especially useful in applications where accurate positioning and six degree-of freedom motion is required.

II. Problem Statement

For our project, we are looking to utilize a Stewart Platform to keep a pool table on a cruise ship perfectly balanced despite the random swaying of the ship due to the waves of the ocean. A regular pool table on a cruise would result in the balls rolling all over the table and not allow a full game to be played. However, if the table had the ability to counteract the imbalance caused by the motion of the ship, the table can remain balanced despite the disturbances.

We used a Stewart Platform made up of six linear actuators to give the table six degrees-of-freedom to overcome all possible disturbances. The SP gives the table the ability to move in three translational motions (x, y, z) and three rotational motions (pitch, roll, yaw). An inertial position sensor is used to measure the 3-D time-varying transformation between frames to keep the top plate perpendicular to the gravity vector . The block diagram below shows the system architecture.

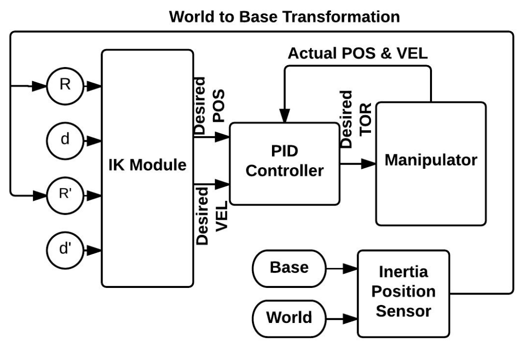

Fig. 1. Block diagram of Stewart Platform control structure

III. Dynamical Model and Control

The block diagram in Fig. 1 shows the various subsystems that comprise the Stewart Platform and its controller. There are a total of 4 subsystems: the inverse kinematics module (IK), the PID controller, the Manipulator and the Inertial Position Sensor. The translation and rotation matrices from the base frame of the platform to the desired position and orientation of the frame attached to the top plate are given as inputs to the IK module, along with their derivatives. The IK module produces an output that represents the desired velocity and position of the frame attached to the top plate, which is fed into the PID controller as its input. Using these inputs and feedback from the manipulator, the controller produces an output signal that represents the force that must be applied to the prismatic joints of the platform. The Inertial Position Sensor is employed to compute the rotation matrix needed to get the platform's top plate perpendicular to the direction of gravity (with respect to the frame attached to the base plate). This matrix, and its derivative, are fed into the IK module. Since the goal is to make the surface of the top plate always perpendicular to the direction of gravity, we introduce disturbances to the base plate of the platform to recreate the pool table on a cruise ship problem. The following paragraphs explain these subsystems in detail.

A. *Inverse Kinematics*

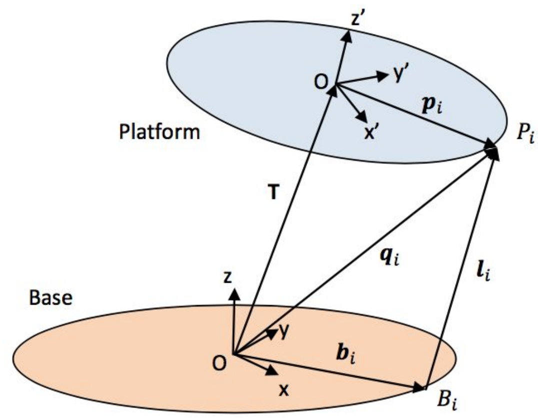

Fig. 2. Representation of transformation from base plate to platform

Fig. 2. shows the schematic of the mechanical structure of a standard Stewart Platform. An equation is derived that relates the desired pose of the top plate in terms of leg lengths. The leg length of leg i is denoted by l_i. Coordinate frames are attached to both the plates: F_{top} for the top plate, and F_{base} for the base plate. B_i and P_i are the connection points of the i^{th} leg with the base plate and the top plate respectively. Position vectors b_i and p_i represent those connections points relative to F_{base} and F_{top} respectively, whereas position vector q_i denotes P_i relative to F_{base}. T represents the translation between the origins of the two frames. Let $({}^{base}R_{top})$ represent the rotation matrix of F_{top} relative to F_{base} .

By inspection:

$$q_i = T + ({}^{base}R_{top} * p_i) \tag{1}$$

$$q_i = b_i + l_i \tag{2}$$

P_i and B_i are not time-varying, while T and ${}^{P}R_B$ are obtained in real-time from the Inertial Position Sensor mounted on the top plate. From (1) and (2):

$$l_i = T + ({}^{base}R_{top} * p_i) - b_i \tag{3}$$

While each leg extends or retracts based on (3), the PID controller also requires their velocities. Differentiating (3):

$$v_i = T' + ({}^{base}R_{top})' * p_i \tag{4}$$

Therefore, the Inverse Kinematics block contains position computation and velocity computation sub-blocks for the 6 prismatic joints. (3) provides the desired leg lengths as a 3x1 vector whose elements are length components in the x, y, z directions. (5) calculates the norm of that vector to obtain the desired leg length as a single value which can be fed into the PID controller.

$$\|l_i\| = \sqrt{(l_{i_x})^2 + (l_{i_y})^2 + (l_{i_z})^2} \tag{5}$$

Similarly, (4) outputs the leg velocity for each leg as a 3x1 vector representing velocities in the x, y, z directions. As with position computation, the norm of v_i is calculated using (6):

$$\|v_i\| = \sqrt{(v_{i_x})^2 + (v_{i_y})^2 + (v_{i_z})^2} \tag{6}$$

The results obtained from (5) and (6) are fed into the PID controller as the desired leg lengths and velocities respectively.

From (3) and (4), it can be noted that the Inverse Kinematics block depends on the variables T, ${}^{base}R_{top}$, and their derivatives T', $({}^{base}R_{top})'$. T, which represents the translation between the origins of F_{base} and F_{top} is initially set to $[0, 0, 72]^T$ where the units are in centimeters. The z component can be anywhere between $71 - 82\ cm$, outside which the prismatic joints either mash into themselves or detach completely. T', which denotes the relative linear velocity between the two plates, is set to 0 because in equilibrium there should be no relative velocity between the two plates. $({}^{base}R_{top})$ and $({}^{base}R_{top})'$ are obtained from the Inertial Position Sensor.

B. PID Control

To control the Stewart Platform, a PID controller was implemented. From the IK module, the PID Control block receives the desired position and desired velocity of each each leg. The goal of the PID block is to give the manipulator the desired torque required to achieve the desired position and velocity. The control law for a PID controller is listed below in equation (7).

$$\tau = K_p\tilde{q} + K_d\tilde{q}' + K_i\int_0^t \tilde{q}(\sigma)d\sigma \tag{7}$$

τ represents the leg torque, $\tilde{q}$ is the position error, $\tilde{q}'$ is the velocity error, and $\tilde{q}(\sigma)d\sigma$ is the integral term obtained by integrating, with respect to time, the position error term. K_p, K_d, and K_i represent the proportional, derivative and integral gains, respectively. As mentioned, the desired terms are received from the IK module and are subtracted from the actual positions and velocities to create the position and velocity errors. The actual position and velocity are obtained from the manipulator block. The calculated torques for each of the six legs are then fed into the manipulator block.

C. Manipulator Block

As shown in Fig. 1, the manipulator block receives the desired torques of each leg required to obtain the proper top plate configuration. The manipulator block consists of the rigid components that make up the Stewart Platform and how they are connected to each other.

Each of the six actuators consist of an upper and lower part. The upper part has a slightly smaller diameter to allow for linear actuation by sliding the upper part into the lower part. These two solid parts are connected as cylindrical joints to allow for this motion. Each of the six actuators are connected at specific points to the top and base plates using a universal joint at both locations. The universal joint allows for a 2DOF motion (rotation about x and y) which gives the Stewart platform the ability to reach all 6 degrees-of-freedom. Each actuator also has a sensor which records the actual position and velocity of the leg which are later fed back into PID to obtain the error terms.

The top and base platforms are both shaped as hexagons to better illustrate the paired actuator connection points. The base platform is also slightly thicker than the top platform. As previously mentioned and shown in Fig. 2, the top and base platforms each have their own reference frames based on the world frame. The base platform is modeled as a universal joint to allow for disturbances in the form of rotation about x and rotation about y. A pool table on a cruise ship would only need to account for this 2DOF disturbance since no translational motion would be necessary since that will be fixed and no rotation about z is necessary since that won't help in stabilization. The disturbance D(t) was modeled as a sine wave shown in equation (8) below.

$$D(t) = 0.15 sin(t) \tag{8}$$

The amplitude of the wave was chosen as 0.15 radians but the algorithm is able to work with any amplitude of range $-0.25 \leq A \leq 0.25$. The SP is unable to account for disturbances outside of this amplitude range and will result in a broken SP. The frequency was set at 1 rad/s to simulate realistic ship disturbance.

D. Inertial Position Sensor

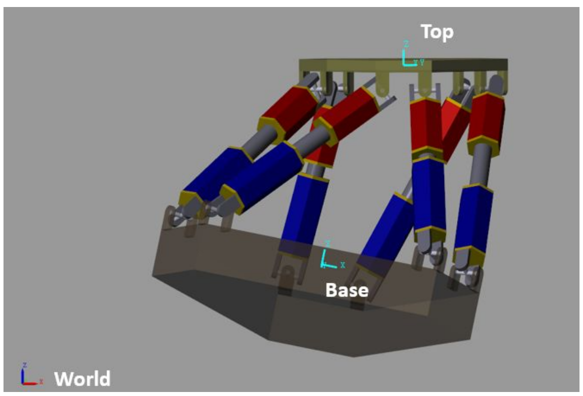

Fig. 3. Representation of the Coordinate Frames attached to the Top Plate, Base Plate and World. The frame attached to the Top Plate has the same orientation as the World Frame in the desired configuration.

The Inertial Position Sensor measures the 3D time varying transformation between any two frames given as its inputs. In this problem, the objective is to get the surface of the top plate perpendicular to the direction of the gravitational force. This goal can be achieved if the frame attached to the top plate is always in the same orientation as the frame that represents the directions of x, y, and z in the world. This idea can be clearly understood from Fig. 3. Therefore, in the desired configuration of the robot, equation (9) holds true.

$$F_{top} = F_{world} \tag{9}$$

To get the desired configuration, we need the rotation matrix from the frame attached to the base of the platform to the desired top plate frame, which is essentially the world frame from (9). As the base of the platform is subject to disturbances, the coordinate frame attached to it keeps changing its orientation and position with time. The Inertial Position Sensor measures the instantaneous rotation matrix from the dynamic frame attached to the base of the platform to the static world frame. This rotation matrix (R), is fed into the Inverse Kinematics module for computing the leg lengths corresponding to the desired configuration. The Inertial Position Sensor also calculates the angular velocities of rotations between frames. This property is used to compute the derivative of the rotation matrix R, using (10). Here, $S(\omega)$ is a skew symmetric matrix which is the linear function of the angular velocities. The derivative R' is also fed into the IK module.

$$R' = S(\omega) * R \tag{10}$$

IV. Results

The effectiveness of the PID controller in maintaining stability of the top platform relative to the world frame is visually apparent through the simulation. Alternatively, to demonstrate the controller's ability to counteract the x-y disturbances, the angular velocities about x, y, and z of the top plate with respect to the world and base frames was plotted in Fig. 4. In all three plots in Fig 4, the angular velocity of the top frame with respect to the world frame was 0 rad/s. This indicates the controller is performing as expected since the top frame should match the world frame at all times to ensure the top platform is perpendicular to the gravity vector. However, the plot showing the angular velocities of the top frame with respect to the base frame about x and y clearly shows an angular velocity is present, in the form of the disturbance equation (8) above. In the z plot, both lines wrt Base and wrt World show 0 rad/s since there was no disturbance applied to the z axis. As mentioned previously, disturbance about the z-axis is unrealistic in the context of a pool table on a ship.

In Fig. 5, the plot of the leg position errors of all six actuators is shown. These plots show the difference between the desired position and actual position for each leg. The errors are relatively minor with leg 2 and 5 reaching the highest error at only about 0.2 cm. Overall, the results show that the implementation of the PID controller was successful in ensuring the top platform is able to counteract x-y rotational disturbances to maintain stability by remaining perpendicular to the gravity vector.

V. Conclusion and Future Work

The control system designed in this project was able to keep the surface of the top plate of the Stewart Platform always perpendicular to the gravity vector (which points in the negative z direction), despite disturbances being injected into the base of the platform. The project was inspired by the problem of balancing a pool table on a cruise ship, which is subject to the swaying motion caused by waves. A PID controller was used to control the Stewart Platform.

This system is a little different from the pool table problem. In this implementation, the top plate of the Platform does translate laterally. However, a pool table must only rotate, and not translate with respect to the ship's base, as it is fixed. Future work in this area would involve modifying the current design so that the top plate does not translate laterally with respect to the base.

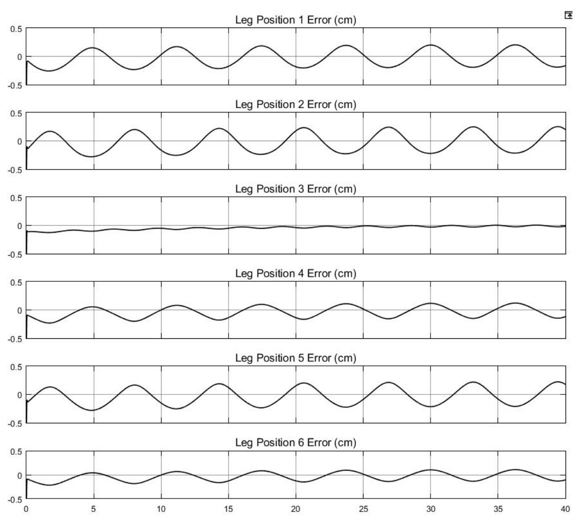

Fig. 5. Leg Position Error(cm) vs Time(s) for all the six legs of the Stewart Platform

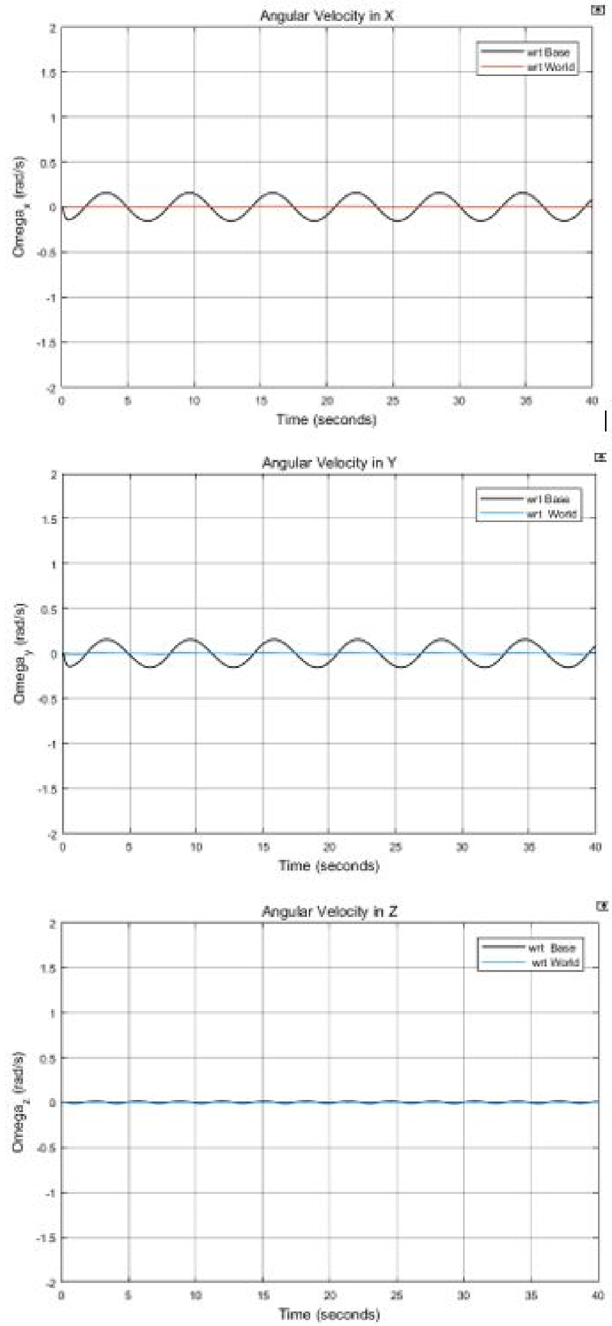

Fig. 4. Angular velocity (rad/s) vs time (s) plots for the rotations about the x (top), y (middle), and z (bottom) axes. The black lines represent the angular velocity of the top frame with respect to the base frame and the colored lines represent the angular velocity of the top frame with respect to the world frame.

VI. References

[1] Merlet, J-P. "Parallel manipulators: state of the art and perspectives." Advanced Robotics 8.6 (1993): 589-596.

[2] Abedloo, Ebrahim, Amir Molaei, and Hamid D. Taghirad. "Closed-form dynamic formulation of spherical parallel manipulators by gibbs-appell method." Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on. IEEE, 2014.

[3] Gough, V. E. "Contribution to discussion of papers on research in automobile stability, control and tyre performance." Proc. Auto Div. Inst. Mech. Eng. Vol. 171. 1956.

[4] Bingul, Zafer and Karahan, Oguzhan. "Dynamic Modeling and Simulation of Stewart Platform." InTech, March 2012: 19-42

[5] Jakobovic, Domagoj and Jelenkovic, Leonardo. "The Forward and Inverse Kinematics Problems for Stewart Parallel Mechanisms." University of Zagreb.

[6] Mathworks. "Stewart Platform (r2017b)." Accessed November 2017: https://www.mathworks.com/help/physmod/sm/examples/stewart-platform.html.

[7] Dasgupta, Bhaskar, and T. S. Mruthyunjaya. "The Stewart platform manipulator: a review." *Mechanism and machine theory* 35.1 (2000): 15-40.