Visual Servoing of a Ball on a Stewart Platform

Group 3
Robotics Case Study Project – B51RO
Department of Electrical and Electronic Computing Engineering
January - April 2023

Submitted by:

1. Sankalp Arora (H00427326)
2. Raj Aryan Yadav (H00424954)
3. Alistar Chapman (H00320987)
4. Jaya Deepak Reddy (H00415978)
5. Akash Manoharan (H00310954)

Contents

Abstract

The Stewart platform, also known as a Hexapod, is a parallel mechanism consisting of a platform and a base coupled with six links. Despite its relatively small workspace and low manoeuvrability, it is capable of 6-DOF, load distribution, and high structural rigidity. The platform's motion can be produced by varying the lengths of six parallel linear actuators, often driven by electrical or fluid power drives. The platform has been used in numerous applications, such as radio telescopes, robot manipulators, and service robots.

Continuous and accurate information about the ball's position is required to balance the ball on the Stewart platform. This is accomplished by mounting a visual sensor, which is the Pixy2 camera here, on the platform's top. The camera captures real-time images of the ball on the plate, and image processing techniques are used to detect and monitor the ball's position.

The ball's position is used as feedback by the controller to determine the control signal to be applied to the six parallel linear actuators on the Stewart platform. The controller compares the feedback signal to the desired reference signal and uses the resulting error signal to modify the control signal to minimise the error and maintain the ball's balance on the plate.

The project demonstrates the following:

- The inverse kinematics of the Stewart platform, which is used to calculate the lengths of the legs of the platform, necessary to achieve the desired position and orientation.
- The Dynamic model of a ball and plate system which refers to the mathematical description of how the system behaves over time.
- Hardware designing of the Stewart platform using 3D printers.
- Image processing for monitoring and detecting the ball’s position and sending the data as feedback to the controller.
- PID tuning for balancing the ball on the platform.

SECTION I

Introduction

Objective: To develop a control system that uses visual feedback to track the position of a ball on the platform and generate appropriate control signals to maintain the ball's position and balance it against external disturbances while ensuring that the platform remains within safe operating limits.

A Stewart platform is a parallel manipulator that consists of a fixed base, a movable platform, and six linear actuators that connect the two. The actuators are connected to the platform through six ball joints, and they can independently control the position and orientation of the platform. A vision-based self-calibration and control system can be used to control the platform's motion using a camera. A ball balancing on a plate is an example of a category of mechatronic systems. The basic mechanical process for implementing such a system can be outlined as follows:

- Mount the camera on top of the moving platform and ensure that it has a view of the surroundings that is undisturbed and clear.
- We used the Pixy2 camera to extract information about the position and orientation of the moving platform concerning the stationary base.
- Use the information obtained from the camera to compute the desired position and orientation of the moving platform based on a predetermined control strategy. It may involve feedback control techniques to ensure the platform moves smoothly and accurately to its target position.
- Use the servo motors with tie rods to move the platform to its desired position and orientation based on the outputs received from the control system.
- Continuously monitor the position and orientation of the moving ball using the camera and adjust the control strategy as necessary to ensure that the platform movements try to bring the ball back to the centre.

Since the introduction of the Stewart platform manipulator for the training of helicopter pilots in 1965, it has been widely utilised. As a motion device, Stewart platform manipulators have been utilised for many decades in driving and flight simulators.

The disadvantage of these systems is that, due to mechanical misalignments, the actual process may differ significantly from its mathematical model. Due to the difficulty of obtaining a precise model, the controller design must account for modelling uncertainties and maintain closed-loop performance at various operating points.

SECTION II

Stewart Platform

A. 3D printing

Before building a physical prototype, designing a robot in AutoCAD Fusion 360 entails creating a virtual model of the robot and simulating its movements and behaviours. The 3D-printed components are integral to the robot's physical structure, providing the necessary support and mechanical functionality to move and carry out its assigned tasks.

In this instance, the automaton consists of PLA filament-printed 3D-printed components. Polylactic acid, or PLA, is a common 3D printing material renowned for its usability, affordability, and eco-friendliness. It is also a robust and long-lasting material that is suitable for a variety of applications.

To print these parts, we utilised the Ender 3 3D printer to create these parts, a popular choice for 3D printing enthusiasts. We followed the generally recommended settings, setting the nozzle temperature to 220 degrees Celsius and the bed temperature to 60 degrees Celsius. In addition, we selected the "Gradual Infill" option in Ultimaker Cura to enhance the strength of the models. This option allows the infill density to gradually increase towards the top layers of the model, resulting in stronger and more durable prints.

The project utilised 16 3D-printed components, including six links, one lower surface-base plate, one middle plate, one upper surface-base plate, two holders, four square-shaped models, and one ball. Each component served a specific purpose and was vital to the final product's functionality. The uses and dimensions of these models are given below:-

1.) Links: We utilised six connections to connect the robot's legs and motors.

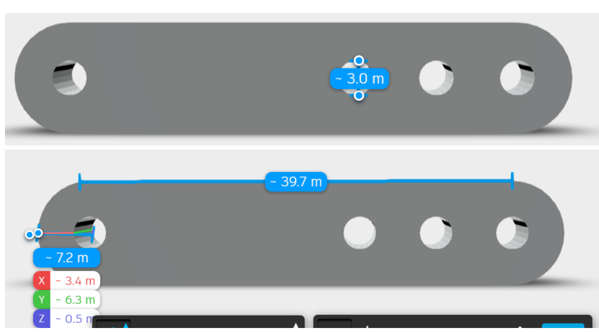


Fig1. 3D Model of the links with dimensions

2.) Lower surface base plate: The robot's lower surface base plate provided stability and support to the framework.

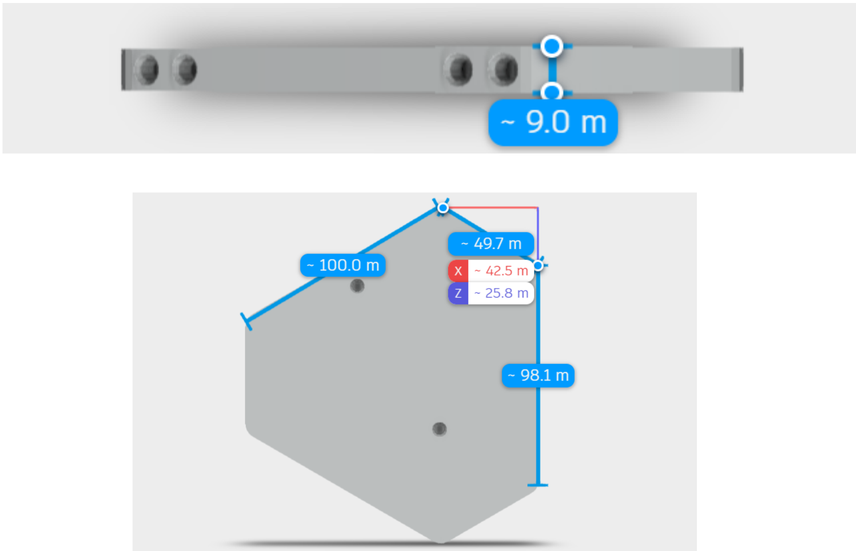


Fig2. 3D Model of the Lower surface base plate with dimensions

3.) Middle plate: We used one middle plate in the project to prevent wires from tangling and to enhance the project's appearance. Additionally, it includes a switch for turning the project on and off. Dimensions are the same as the Lower base plate.

4.) Upper surface base plate: One upper surface base plate rotates the polycarbonate sheet's topmost layer. It connects to the ends of the six legs.

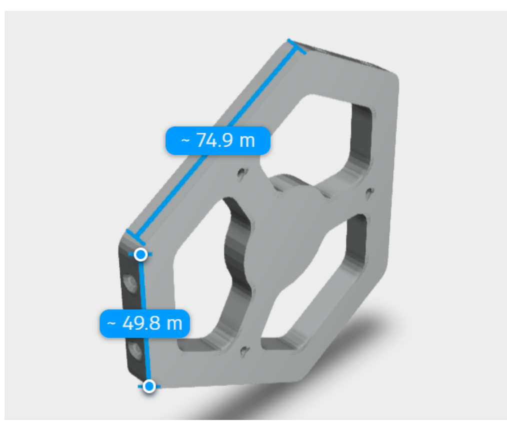
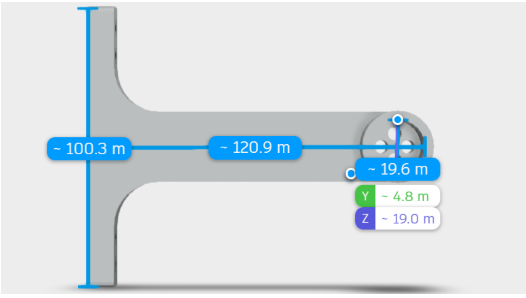


Fig3. 3D Model of the Upper surface base plate with dimensions

5.) Holders: We used two holders to hold the dowel that supports the camera. Both holders attach to either end of the dowel.



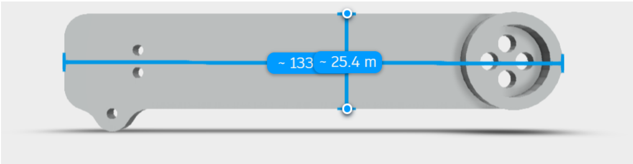


Fig4. 3D Model of the Holder with dimensions

6.) Markers: square-shaped blocks to mark the centre for object detection.

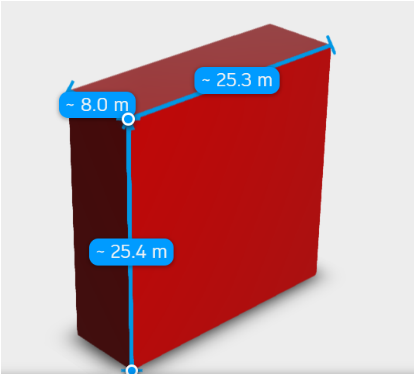


Fig5. 3D Model of Square markers with dimensions

7.) Ball: A ball is an object used in the project.

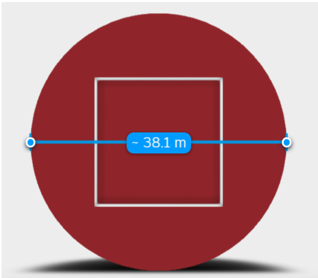


Fig6. 3D Model of the ball with dimensions

B. Inverse Kinematics

S. Küçük's, Serial and Parallel Robot Manipulators – Kinematics Dynamics Control and Optimization[1] describes the detailed inverse kinematics of parallel robots. Below in this section is a closed-form solution for inverse kinematics. It is a transformation of the corresponding position and orientation of the platform relative to the base into six servo motor angles which are necessary to control for the platform to reposition accordingly to prevent the ball from falling.

Two right-hand coordinate frames, {A} and {B}, are designated to the base and platform. The platform's centroid A is the origin of the frame {A}, and the *x, y,* and *z* axes are depicted in Fig. 1. θp represents the angle formed by the joints *A1* and *A2*. Likewise, the platform's centroid B is the origin of the frame {B} corresponding to the *xp, yp,* and *zp* axis. θb represents the angle formed between universal joints B1 and B2. θp and θb are both 20°. rp and rb represent the platform's and base's radii, respectively. Both r_p and r_b measure 105 millimetres.

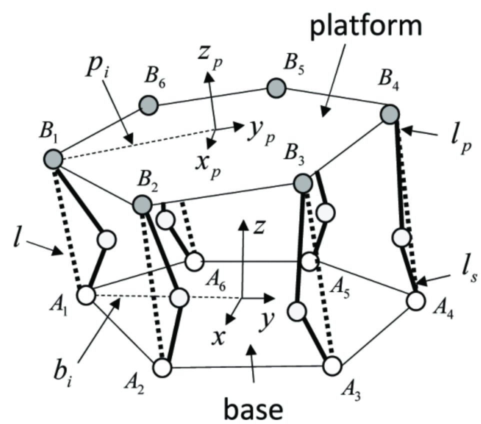


Fig7. Stewart Platform Model (Source:[2])

pi is the vector from centroid A to the joint *Ai* , *bi* is the vector from centroid B to the universal joint *Bi.*

$$p_i = \begin{bmatrix} p_{xi} \\ p_{yi} \\ p_{zi} \end{bmatrix} = \begin{bmatrix} r_p \cos a_i \\ r_p \sin a_i \\ 0 \end{bmatrix} \tag{1}$$

When,
$$a_i = \frac{i\pi}{3} - \frac{\theta_p}{2}, i = 1,3,5$$
$$a_i = a_{i+1} + \theta_p, i = 2,4,6$$
$$b_i = \begin{bmatrix} b_{xi} \\ b_{yi} \\ b_{zi} \end{bmatrix} = \begin{bmatrix} r_b \cos c_i \\ r_b \sin c_i \\ 0 \end{bmatrix}$$
When,
$$c_i = \frac{i\pi}{3} - \frac{\theta_b}{2}, i = 1,3,5$$
$$c_i = c_{i+1} + \theta_b, i = 2,4,6$$

The orientation of frame {B} with respect to frame {A} can be described by the orientation matrix R_T, which requires r_{ij} for the values *1, 2, and 3* for *i* and *j*. Roll-Pitch-Yaw angles (α, β, and γ) indicate the orientation of Frame {B} about the *x, y,* and *z* axes, respectively. The rotation matrix is:

$$R_T = R_x(\alpha)R_y(\beta)R_z(\gamma) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$
$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$
$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$
$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This rotation matrix transforms a vector expressed in frame {B} into its equivalent representation in frame {A}, regarding a linear actuator. From *i = 1, 2,..., 6,* the length of the limb can be computed from

$$l_{l,i} = \sqrt{(b_{xi} - S_{xi})^2 + (b_{yi} - S_{yi})^2 + (b_{zi} - S_{zi})^2}$$

$$\text{When,} \quad S_i = \begin{bmatrix} S_{xi} \\ S_{yi} \\ S_{zi} \end{bmatrix} = \begin{bmatrix} X + p_{xi}r_{11} + P_{yi}r_{12} \\ Y + p_{xi}r_{21} + P_{yi}r_{22} \\ Z + p_{xi}r_{31} + p_{yi}r_{32} \end{bmatrix}$$

In this instance; the legs comprise a servo arm and platform links connected with universal joints. The length of the servo arm link is l_s = 40 mm, while the length of the platform link is l_p = 80 mm. The servo arm angles from *i = 1, 2,..., 6* can be expressed as follows:

$$\theta_i = \sin^{-1}\left(\frac{k_i}{\sqrt{m_i^2 + n_i^2}}\right) - \tan^{-1}\left(\frac{n_i}{m_i}\right)$$
$$k_i = {l_{l,i}}^2 - \left({l_{p,i}}^2 - {l_{s,i}}^2\right)$$
$$m_i = 2 \times l_s \times S_{zi}$$
$$n_i = 2 \times l_s \times (X_i \times b_{xi}) \times (S_{yi} - b_{yi})$$

By substituting all these values for different legs in MATLAB, we got the visuals of our project.

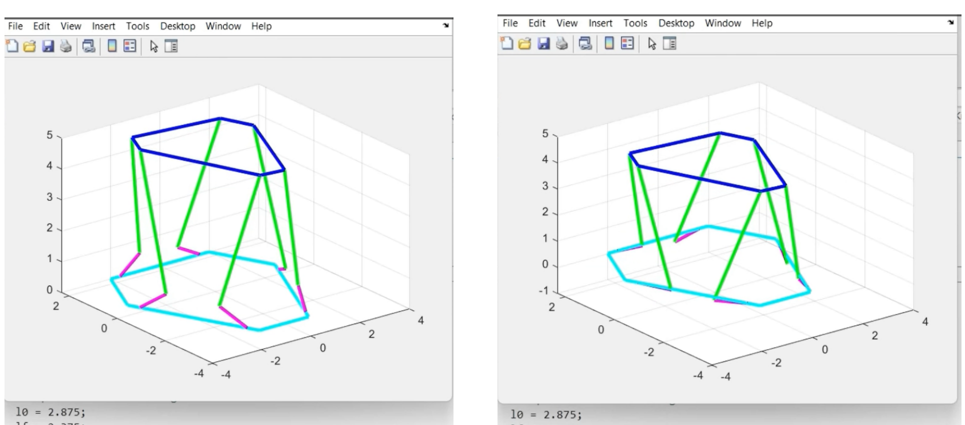


Fig8. Matlab simulations of the platform movement

C. Dynamic Model

Using the Euler-Lagrange method, the dynamic model of the ball and plate system can be described as

$$\frac{d}{dt}\frac{\partial L}{\partial q} - \frac{\partial L}{\partial q} + \frac{D}{\partial q} = Q$$

The Lagrangian (L) is a concept in physics and mechanics that is used to describe the dynamics of a mechanical system in terms of its kinetic(T) and potential(V) energies. The ball's translation *(x, y)* and rotation (around its centre) add together to form the ball's kinetic energy. The platform's rotational energy is not taken into consideration. The ball has a mass of *m*, a radius of r_b, and an inertial moment of *J*. The following is a formula for the kinetic energy:

$$T = \frac{1}{2}\left(m + \frac{J}{r_b^2}\right)(\dot{x}^2 + \dot{y}^2)$$

θx and θy are the rotations of the platform along the x-axis and y-axis. The potential energy can be written as follows:

$$V = mg(x \sin\theta_x + y \sin\theta_y)$$

The Rayleigh dissipation function (D) is a function that accounts for energy dissipation due to damping in a mechanical system. The vector Q represents the external forces acting on the system, such as gravitational, applied, or frictional forces and $q = [x, y]T$.

Together, the Lagrangian, Rayleigh dissipation function, and external force vector provides a complete description of the dynamics of a mechanical system. In the case of a ball and plate system, From the Euler-Lagrange method, the dynamic model obtained is expressed as

$$\left(m + \frac{J}{r_b^2}\right)\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} = -mg\sin\begin{bmatrix}\theta_x\\ \theta_y\end{bmatrix}$$

SECTION III

System Setup

A. Microcontroller

There were numerous options to choose from, thanks to recent advancements in microcontroller technology. First, we looked at the teensy 4.1 microcontroller's feasibility. A new version of the microcontroller that uses the ARM cortex-M7 CPU is called the Teensy 4.1. With a clock speed of 600MHz, this processor is commonly considered a fast and dependable CPU. Although the Teensy 4.1 has sufficient processing power and would be ideal for the robotic stabiliser, it exceeded the project's budget.

The viability of the Arduino uno R3 microcontroller was then examined. The components in the Arduino, which is renowned for being accessible and simple to use, are not of the same calibre as those in the teensy 4.1. A 16MHz CPU and 32KB of flash memory are features of the Arduino r3. The Arduino is a low-cost, easily-programmable board with an integrated development environment (IDE) that makes it considerably easier to write sophisticated code. The microcontroller's lack of processing power is a drawback, but its simplicity, and affordable price made it a good option for the project. Moreover, The Arduino board can run various programming languages such as C++, Python, and others, and you can use libraries such as the Arduino Motor library to simplify motor control. Therefore, Arduino was chosen as the board for the case study.

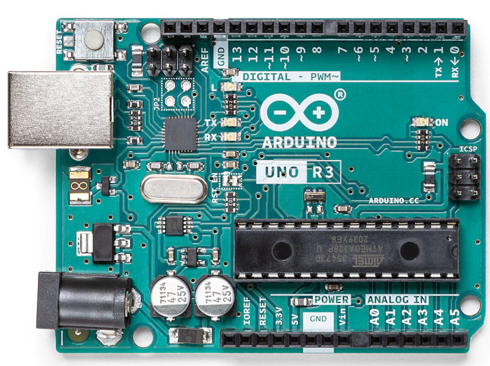


Fig9. Arduino UNO Rev3

B. Servo Controller

Since the microcontrollers under consideration lacked the power to support all six actuators, a servo controller was necessary to power the servos employed in the platform. This servo controller had to work with the chosen microcontroller and support at least six channels.

The PCA 985 12-bit, 16-channel servo controller was chosen as the servo controller. This controller board was trustworthy, affordable, and adequate to manage six servos. Because the PCA985 employs an I2C-controlled PWM driver with a built-in clock, it can operate independently and is not dependent on the Arduino for steady signals that could impair the microcontroller's performance. Two power inputs are necessary for the servo controller. First, to power the board and the other to power the servos. The Arduino's 5V output channel would power the board, while the servos were intended to be powered by a separate 5V power supply.

Fig10. PCA9685 16-Channel 12-bit servo motor driver

C. Servo Motors

The parallel manipulator-type stabilisation robot employed in this case study uses multi-axis motion control with six actuators. Although the actuators move the platform, the manipulator's base remains fixed. The platform may be stabilised to the fullest extent possible thanks to the placement of these actuators in pairs, which produce six degrees of freedom. The actuators must be able to make accurate and powerful movements because the platform's design necessitates it to stabilise the object on top. The precision, controllability, and reliability of servo motors make them an excellent choice as an actuator in this situation.

Servos use a closed-loop feedback system that tracks and adjusts the actuator's speed, torque, and location. Usually, a potentiometer is used to control the current and, consequently, the torque of the motor by providing feedback on the location of the motor. The servo's ability to produce high torque and low-speed output makes it perfect for accurate movement. These elements work together to provide an accurate and reliable task-specific motor.

The Servo motors used here are The HiTec HS-311 Standard Servo with a feedback system that adjusts the position of the motor shaft based on the input signal. The HS-311 servo motor is known for its high-impact gears, making it more durable and reliable in rugged environments. It has a torque rating of 3.0 kg-cm and a speed of 0.19 sec/60 degrees, which makes it suitable for various applications, including robotics, remote control vehicles, and model aircraft. The motor operates on a voltage range of 4.8V to 6.0V and consumes a maximum current of 700mA. It also features a standard three-pin connector for easy integration with control systems.

Fig11. PCA9685 16-Channel 12-bit servo motor driver

D. Pixy2.0 Camera

The Pixy 2 camera is a vision sensor designed for robotics and automation projects. It is a small and lightweight camera that can detect and track objects based on their colour codes or shapes. The Pixy 2 camera is compatible with various microcontrollers, including Arduino, Raspberry Pi, and others, making it easy to integrate into different projects. Depending on the platform, it communicates with the microcontroller using SPI, I2C, UART, or USB interfaces.

The camera features a 60 frames-per-second image sensor and can detect objects at a range of up to 2 meters. It also has built-in colour-based object recognition and tracking algorithms, which can be programmed using the PixyMon software. The Pixy 2 camera captures images of the surrounding environment and processes them to detect and track objects based on their colour codes or shapes. Additionally, it can detect multiple objects simultaneously, making it useful in complex robotics projects.

Overall, the Pixy 2 camera is a powerful and flexible vision sensor that can be used in various applications, including robotics, automation, and computer vision projects. Its compact size, ease of integration, and advanced features make it an ideal camera for this project.

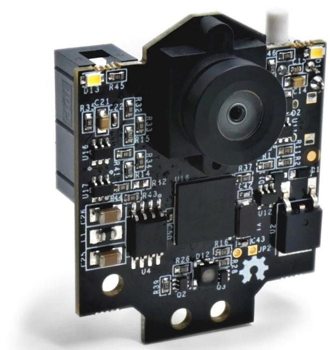


Fig12. Pixy2.0 camera

E. Circuit

When connecting servos to a servo controller, it is common practice to provide power to the controller separately from the power source for the rest of the circuit. This is because servos require a significant amount of power to operate. Drawing power from the same source as the rest of the circuit may cause voltage drops and potentially damage the other components. Therefore, each servo was likely connected to the servo controller with its power source to ensure that each servo received enough power to function correctly.

On the other hand, the Pixy 2.0 camera was connected directly to the Arduino Uno R3, which means it draws power from the board itself. This may be possible because the Pixy 2.0 camera requires less power than the servos. The Arduino Uno R3 is designed to provide power to connected devices through its built-in regulator. The ground pins on all of the components are connected. This is known as common ground. This can be confirmed with a multimeter by checking connectivity. Figure.13 shows the schematics of the circuit:

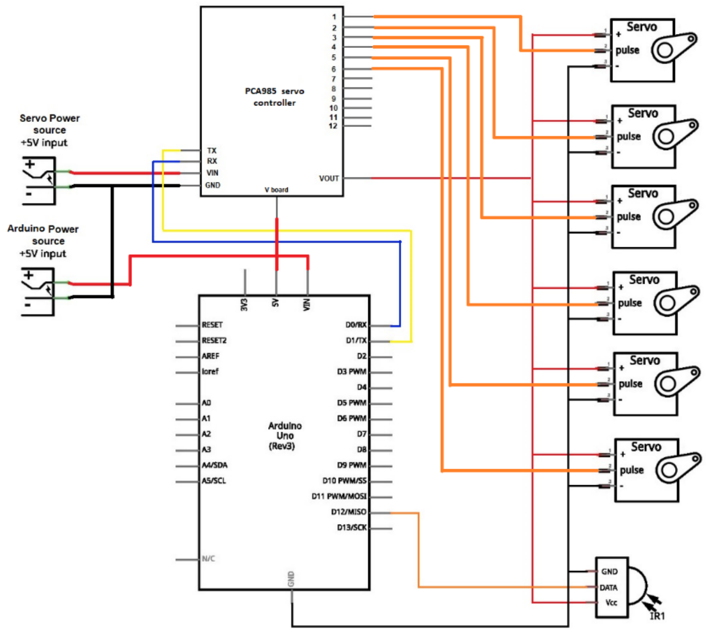


Fig13. Schematic diagram of the circuit

F. Assembly

The following steps are followed to put the robotic platform together:

1. Six servo motors are evenly spaced around the board's edge in a hexagonal arrangement on the base plate. The tie rods of the motors are faced upwards as they are put in place. Motors are firmly fixed to the base plate, so they will not move when the machine operates.

2. Using the tie rods, a 3D-printed platform is fastened to the tops of the six motors. The tie rod is fastened to the motor's face up, and the platform is suspended above the plate. Assemble the six tie rods by attaching two tie rod ends to each pushrod connector.

3. Cover plate is set to retain the tie rods in place after mounting the platform. The base plate and cover plate are fitted tightly and connected.

4. Centre markers are attached to each platform corner using m3 x 5mm screws. Once the centre markers are attached, the Pixy2 camera uses them as reference points to track objects accurately. By analysing the position of the centre markers relative to the camera, Pixy2 can determine the position and movement of the platform and track objects within its field of view.

5. Using the higher and lower holders with a wooden dowel, the camera is mounted on top of the platform to face downward, ensuring that the camera is firmly fixed to the platform so that it will not move while in use.

6. Attached is a power supply and a controller board to each of the six motors. To make sure that the actuators can be controlled independently.

7. Pixy2 is attached to a computer using a USB cable to handle the video feed and regulate the platform's position. The software is installed to process the camera feed and move the platform. The Pixy2 camera has software called PixyMon, which can configure the camera and view its output.

Paying close attention to every step's specifics is crucial to guarantee that the robotic platform stands correctly and securely. All potential risks can be avoided with proper assembly and wiring, guaranteeing that the platform will function as intended.

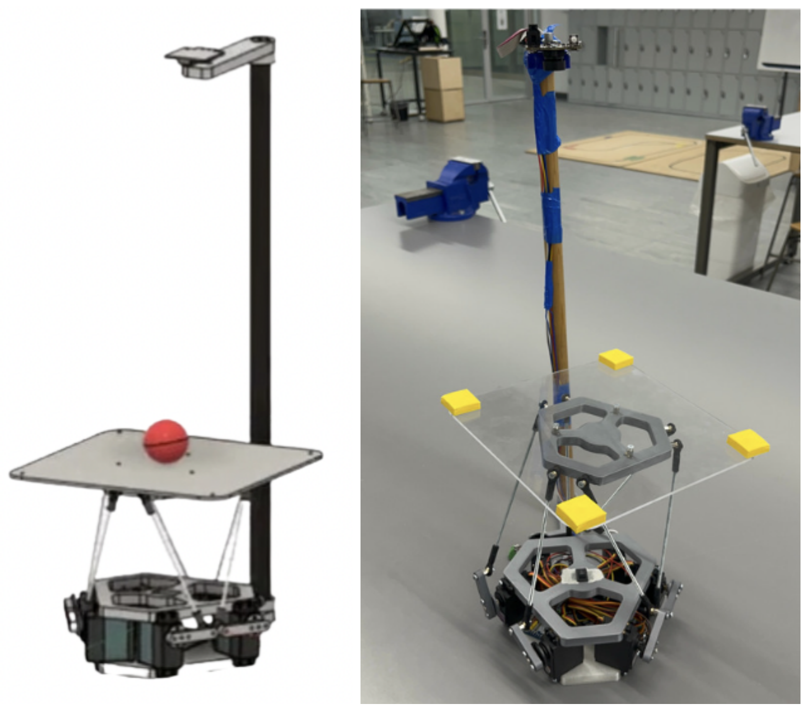


Fig14. (A) 3D model of the assembly (B) Real-life model implementation

SECTION IV

Vision and Control

A. Image Processing

The Pixy2 camera is a vision sensor designed for object recognition and tracking. It uses a powerful processor and a colour sensor to capture and analyse images in real-time. The working of the Pixy2 camera can be summarised in the following steps [3]:

1. Capture images: The Pixy2 camera captures images of its surroundings using its built-in colour sensor. It can capture images at 60 frames per second, making it ideal for tracking fast-moving objects.

2. Object detection: Once an image is captured, the Pixy2 camera analyses it in real-time to detect objects that match the user-defined colour codes or shapes. It can detect multiple objects simultaneously, and each can be assigned a unique signature.

3. Object tracking: Once an object is detected, the Pixy2 camera tracks its movement in subsequent frames. This allows it to predict the object's future position and keep it in focus even if it moves around.

4. Communication: The Pixy2 camera communicates with external devices using a variety of interfaces such as USB, SPI, UART, and I2C. This allows it to send object data to devices such as microcontrollers, computers, or robots.

Once the Pixy2 camera module has detected the ball's(yellow) position, it sends this information to the PixyMon software, which displays the X and Y coordinates of the object on the screen, which are in the camera's field of view. These X and Y coordinates are then communicated to the Arduino board, which uses the kinematics and control algorithms to maintain the ball's position on the platform.

The Arduino board receives the (x,y) coordinates from the Pixy2 camera module using the I2C protocol. These coordinates represent the position of the yellow object's centre concerning the camera image's centre.

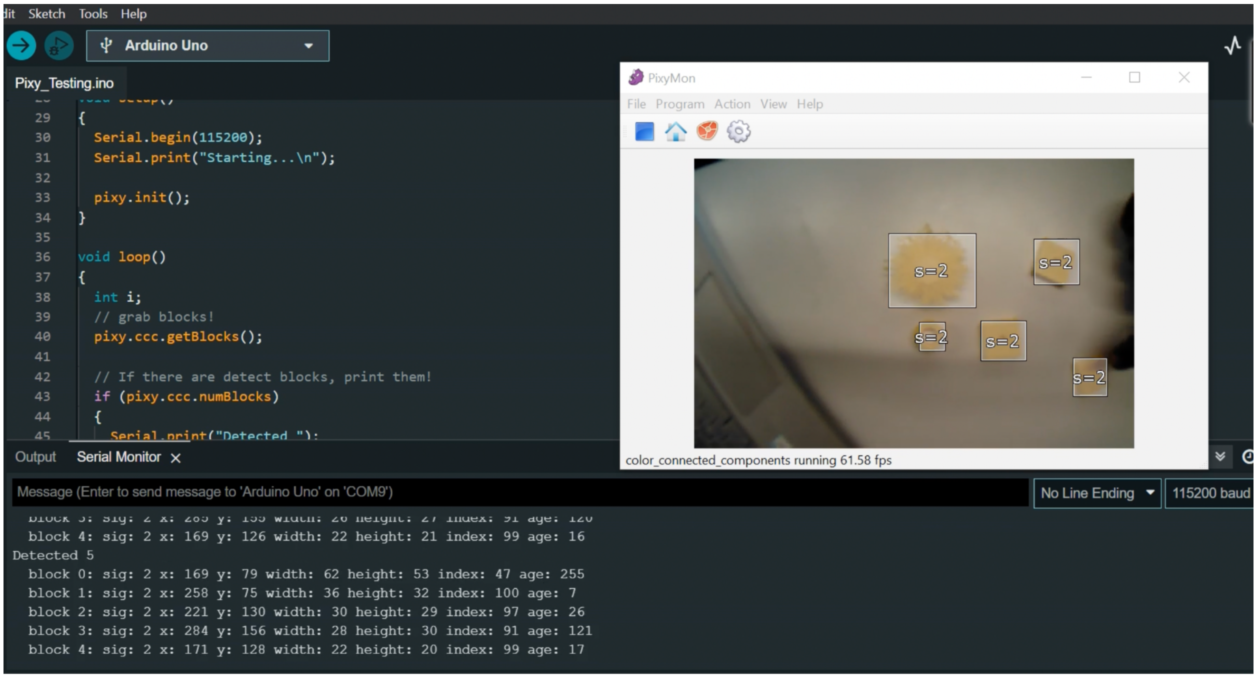


Fig15. PixyMon sends the coordinates information to Arduino

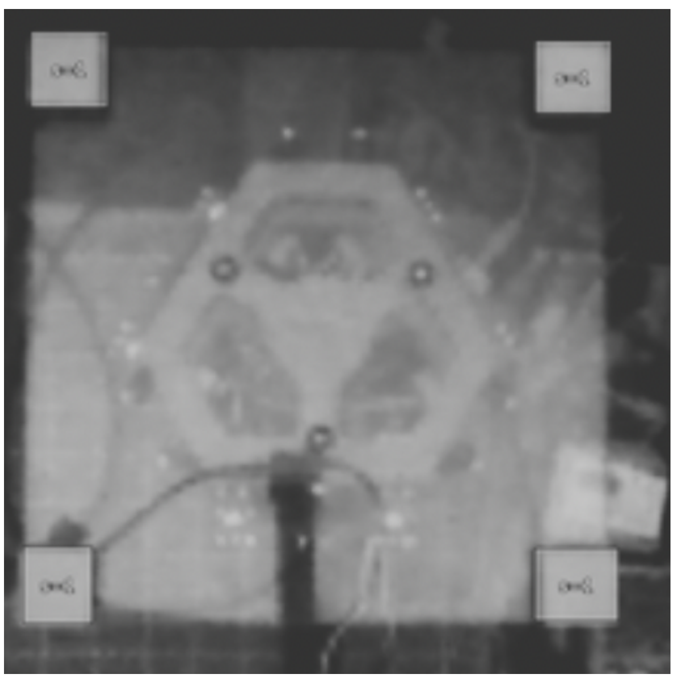


Fig16. Visual of Pixy2.0 marking the centre using the markers at corner

B. PID Tuning

PID tuning is an important process in controlling the motion of a ball balancing the Stewart platform. PID tuning aims to adjust the controller's proportional, integral, and derivative gains of the controller to achieve stable and accurate platform motion. [4]

In the case of a ball-balancing Stewart platform, the PID controller receives input from sensors that measure the ball's position. The controller then calculates the error between the desired position of the ball and its actual position and uses this error to adjust the output of the platform's actuators to bring the ball into the desired position.

To perform PID tuning, the first step is to set the proportional gain (Kp) to a reasonable value and observe the system's response. If the system exhibits overshoot or oscillation, the gain must be decreased. If the system responds too slowly, the gain needs to be increased.

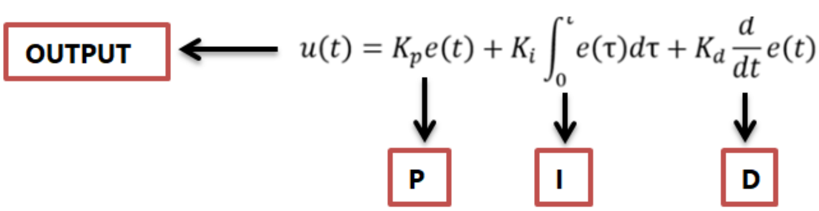
Next, the integral gain (Ki) is adjusted to reduce steady-state error. Ki determines how much the controller should accumulate errors over time. If there is a significant steady-state error, Ki needs to be increased.

Finally, the derivative gain (Kd) is adjusted to reduce overshoot and oscillation. Kd determines how much the controller should react to changes in the error over time. If the system exhibits overshoot or oscillation, Kd needs to be increased.

The process of PID tuning is iterative and involves adjusting each gain while observing the system's response until the desired performance is achieved.

We used integral control (I) in this project because the managed system shows no steady-state error or has a limited tolerance for overshoot or output oscillations [10].

The output of the PID Algorithm is given by:-



So the redefined equation for this system is:-

$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t)$$

Where Kp = Proportional Constant

Kd = Derivative Constant

By tuning P and D values and giving proper limits to Kp and Kd, we can perfectly balance the robot. The inverse kinematics model in Fig. 17's Stewart platform system model converts the reference pitch and roll to six servo motor angles (θi, ref). Within each digital servo motor is a PID controller that controls the position of each motor independently. This causes the platform to move and the ping pong ball to move. The visual system of the web camera functions as a feedback sensor.

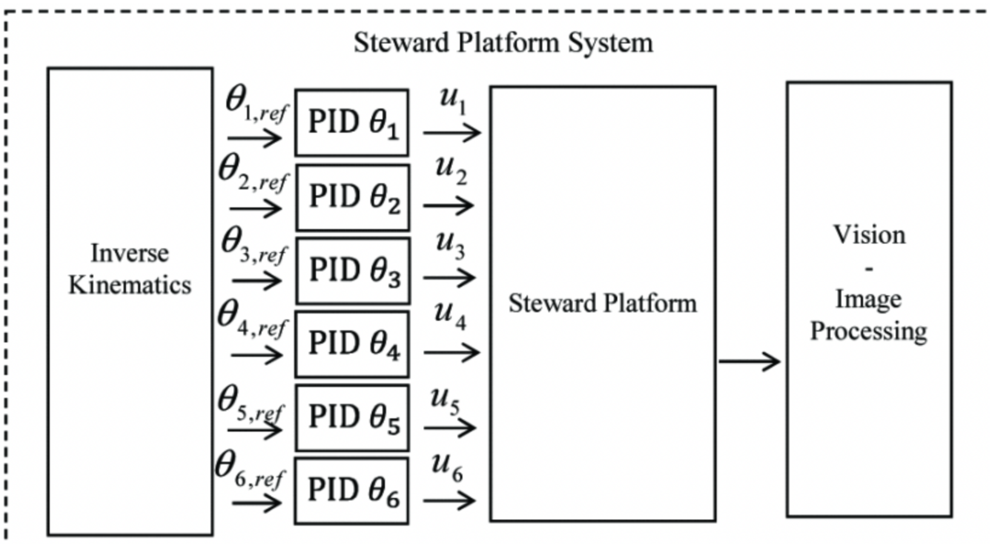


Fig17. Stewart Platform System (Source: [2])

C. Experimentation

On the Stewart platform, the tuned controller was tested by balancing balls. The ball was subjected to external force during the testing at a rate of one second.

The ball positions in the x and y directions are shown in Figure.18 A. The Stewart platform's orientation was altered with the help of the control to balance the ball. The PID control brought the ball's location under control and allowed it to stabilise after 7 seconds. The solid blue lines show the x-positions, while the red dashed lines show the y-positions.

The blue lid lines in Figure.18 B show the pitch angles, while the red dashed lines represent the roll angles.

The six servo motors changed their angular locations, as shown in Figure.18 C. The lines numbered 1 through 6 are solid blue lines, red dashed lines, dark green dash-dot lines, green dotted lines, magenta dashed lines, and solid black lines each referring to a servo motor.

The system is nonlinear due to the friction between the ball and the plate, and steady-state errors were discovered. However, the ball was balanced and controlled around the origin using a few human intuitive criteria [11].

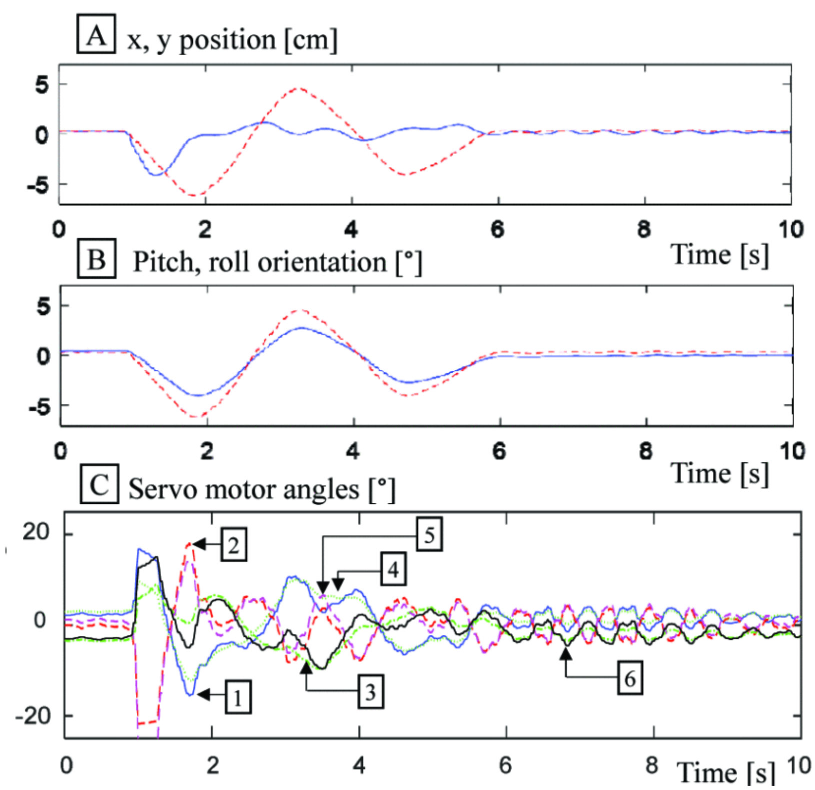


Fig18. Experimental results of PID control (Source: [2])

SECTION V

Applications and Conclusion

A. Real-life applications

Many real-world applications requiring precise positioning and motion control are available on Stewart platforms. These are a few of them:

1.) Medicine: The Stewart Platform is used for surgical training by simulating breathing and heartbeats, tremor-compensation for surgery, fixation devices for correcting deformities called Taylor Spatial Frames. [5]

2.) Flight simulators: Flight simulators use the six degrees of freedom provided by the framework to simulate direction and movement. The Stewart Platform's high dynamic capacity enables them to produce linear and nonlinear pressures, such as a bump, and it can perfectly orient to the appropriate roll, pitch, and yaw. [6]

3.) Experimenting with terrain-based vehicle dynamics has a variety of uses, most of which are connected to suspension systems. Take seat suspension as an illustration; it may be found on anything from a bicycle to a space launcher. [7]

4.) Everything above 3D cinema experiences is considered an XD cinematic experience in the gaming and entertainment industries that use augmented reality (ranging from 4D to 12D). Every cinematic experience that uses seats that move in time with the action uses a certain kind of platform, sometimes known as a particular kind of Stewart Platform. [8]

5.) Applicants for Stabilization: Stewart Platforms are also used in stabilisation applications, which need an object to hold a fixed position while moving by its ground plane. To retain the location of a moving object, commonplace items like satellite or radar antennae on board a ship or aeroplane and landing platforms may need a continuous position fix. [9]

B. Conclusion

The robot creation discussed in this study needed a solid grasp of mechanical, electrical, and object-tracking fundamentals. The machine was developed to attain optimum precision and accuracy during its motions and to give a solid and effective approach to a practical situation.

During the design phase, suitable components and materials were chosen, in addition to ensuring the robot could resist the stresses and strains of its intended purpose. Six linear actuators control the platform's position and orientation through six ball joints. The robot's creation needed meticulous planning and collaboration among team members. 3D printing enabled the precise manufacture of the robot's components. The Pixy2 sensor, which detects the ball's location, is processed by the Arduino microcontroller, which commands the six actuators to change position appropriately. To keep the ball in place on the platform, the lengths of the linear actuators are calculated by applying inverse kinematics equations.

Inverse kinematics was critical for the robot's functioning since it allowed the group to set up the robot to move precisely and consistently. A PID algorithm is employed to balance the ball, having the set point indicating the platform's centre, the error indicating the ball's position relative to the platform's centre, and the derivative term being the ball's velocity. PID tuning is crucial for assessing system stability, precision, and reaction time.

The connectivity of many systems and innovations, such as sensors and motors, necessitated a thorough grasp of software and hardware engineering. Throughout the task, the team members collaborated extensively, synchronising their efforts to make sure that every element was smoothly incorporated into the overall design. The project's achievement depended on open communication and an eagerness to work together. Each team member offered their own set of talents and knowledge to the project, allowing the group to conquer obstacles and discover creative solutions to difficulties.

Overall, the team's success in developing the robot was remarkable. The project demanded a great degree of technical competence, imagination, and teamwork, and the outcome reflected the team's devotion and commitment to excellence. This concept has the potential to motivate future robotics and engineering breakthroughs.

SECTION VI

References

[1] Küçük, S. (2012) *Serial and Parallel Robot Manipulators: Kinematics, Dynamics, Control and Optimization*. IntechOpen. Available at: https://books.google.co.uk/books?id=W8ycDwAAQBAJ.

[2] K. Yaovaja, "Ball Balancing on a Stewart Platform using Fuzzy Supervisory PID Visual Servo Control," *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, Krabi, Thailand, 2018, pp. 170-175, doi: 10.1109/ICAICTA.2018.8541349.

[3] https://pixycam.com/wp-content/uploads/2018/07/Pixy2-Review-Comp-Add.pdf

[4] C. Copot, C. Ionescu and R. De Keyser, "Visual servo control of a Steward platform using fractional-order PID controller," *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, 2014, pp. 70-75, doi: 10.1109/ICSTCC.2014.6982393.

[5] Stewart D. A Platform with Six Degrees of Freedom. Proceedings of the Institution of Mechanical Engineers. 1965;180(1):371-386. doi:10.1243/PIME_PROC_1965_180_029_02

[6] K. Ravichandran, "Driving simulator for tracked vehicles using Stewart platform manipulator," INTERACT-2010, Chennai, India, 2010, pp. 245-249, doi: 10.1109/INTERACT.2010.5706149.

[7] H. -E. Lee, Y. Kim, K. -H. Park and Z. Zenn Bien, "Development of a steward robot for human-friendly interaction," 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, Munich, Germany, 2006, pp. 551-556, doi: 10.1109/CACSD-CCA-ISIC.2006.4776705.

[8] J. M. Rossell, J. Vicente-Rodrigo, J. Rúbio-Massegú and V. Barcons, "An effective strategy of real-time vision-based control for a Stewart platform," 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, France, 2018, pp. 75-80, doi: 10.1109/ICIT.2018.8352155.

[9] Kumar P., R. and Bandyopadhyay, B. (05 2013) ‘Stabilization of stewart platform using Smooth Second Order Sliding Mode Controller’, in, pp. 1–6. doi: 10.1109/ISIE.2013.6563760.

[10] K. Åström and T. Hägglund, "Advanced PID control", *ISA-The Instrumentation Systems and Automation Society*, 2006.

[11] J. M. Rossell, F. Palacios-Quiñonero, J. Rubió-Massegú and J. Vicente-Rodrigo, "Tracking control for a Stewart platform prototype," 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 2015, pp. 58-63, doi: 10.1109/ICAMIMIA.2015.7508003.

NOTE: Code for the entire project is uploaded on GitHub: Link