



ISSN NO. 2320-5407

Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/9802
DOI URL: <http://dx.doi.org/10.21474/IJAR01/9802>



INTERNATIONAL JOURNAL OF
ADVANCED RESEARCH (IJAR)
ISSN 2320-5407
Journal Homepage: <http://www.journalijar.com>
Journal DOI: 10.21474/IJAR01

RESEARCH ARTICLE

DESIGN AND IMPLEMENTATION OF FEEDBACK CONTROL SYSTEM FOR 6-DOF STEWART PLATFORM.

Gopiya Naik ^{S1}, Manojgowda S. P² and Kishor G².

¹Associate Professor, ²Students, Department of Electrical & Electronics Engineering, PES College of Engineering, Mandya, Karnataka (India).

Manuscript Info

Manuscript History

Received: 02 August 2019

Final Accepted: 04 September 2019

Published: October 2019

Key words:-

Feedback Control System, 6-Degree of Freedom, Stewart Platform, Modelling.

Abstract

The goal of this paper is to design and implement feedback control system for a Stewart Platform to control a ball-on-plate system. In this work, a complete physical system and controller design of a 6-Degree of Freedom (6-DoF) Stewart platform explored from conception to modelling to testing and completion is done. Stewart platform is parallel manipulator having 6 actuators to control the position and orientation of a plat-form. The ranges of motion achieved by this manipulator are three translation motions (each in X; Y and Z axes) and three rotational motions around each axis, these motions can be controlled either independently or in any combination (The range and combinations of motion are subjected to physical limitation). The ball-on-plate system is an unstable open-loop System in which a ball placed on a plate (i.e., Platform) and the position of the ball is indirectly controlled by tilting the plate on its X and Y axes.

Copy Right, IJAR, 2019,. All rights reserved.

Introduction:-

Motivation

Nowadays, robots can be found in the fields of medicine, the military, the industry or even in the household sector. This leads to assume that robotics field is expanding, involving in many other techno-scientific sectors and reaching, as well, the daily life. Drones, which were only used in the military industry, are proof of this. All this reflects the control we have achieved over robotics to this day. The problem of controlling an open-loop unstable system presents many unique and interesting challenges. A specific example of an open-loop unstable system is the ball-on-plate system, a two-dimensional extension of the ball-and-beam problem. Among the interesting challenges of such a system is the indirect control of the ball using the angles of the plate. Therefore, it is a field that expands the accessibility and capacity of the human being in all possible and imaginable directions. Robotics, then, can be studied in all modalities, shaping them in different ways and creating robots of all sizes, appearances, and functionalities; making the analysis and the classification of robots a really diverse task.

This paper is structured as follows. Section-2 deals with the Modelling and Simulation of the Stewart platform. Stewart Platform is parallel manipulator with 6 actuators, this makes deriving the equations of motions of the platform complex and highly non-linear solving these equations is highly computationally taxing which is not suitable for our control design. This section explains the inverse kinematic approach to control the platform and also provides details about modelling of the physical Stewart Platform in SolidWorks CAD software, and MATLAB simulation of the Physical system. Section-3 deals with building the Physical Prototype of the platform using,

Corresponding Author:- Gopiya Naik S.

Address:-Associate Professor, Department of Electrical & Electronics Engineering, PES College of Engineering, Mandya, Karnataka (India).

servos, micro-controller, joints, PWM module, touch panel etc., This section also includes building GUI for controlling the platform in a Java based programming language/Processing". Section-4 deals with the control system design for the ball-on-plate system; this includes mathematical modelling of the system and design of control blocks using two methods PID and LQG. Section-5 explains implementing the control strategies on the actual prototype. This includes programming the control logic in Arduino and Processing. Finally, Section-6 gives results and discussion.

Modelling and Simulation of Stewart Platform

Stewart Platform is a parallel manipulator which uses 6 actuators to position the top platform. Usually these actuators are hydraulic pistons whose lengths can be varied by pumping hydraulic fluid into them, this kind of actuation is suitable in large load application on the platform. Since we are building a prototype model cost of a hydraulic actuators are a major constraint. Linear actuators also give linear motion just like hydraulic actuators but they are relatively slow for real-time control of the platform. So we are using fixed length rods connected to a servo motor arm, thus by rotating the servo arm we can emulate the varying length between the servo center of rotation and the top platform. Literature survey was done to find research papers that explain the control of the Stewart platform, [1] [2] [3] most of the papers go through control using hydraulic actuators there are very few papers deal with rotary actuators. [4], [5] show that there are two ways of controlling the platform.

Forward Kinematics:

This is the method of obtaining the orientation of the platform by solving a bunch of simultaneous non-linear equations having length of actuators (in this case servo angles) as inputs and platform orientation as output.

Inverse Kinematics:

This is the method of obtaining the length of actuators (in this case servo angles) by providing the required orientation of the platform. This method employs geometric transformation to reverse calculate the servo angles from the platform orientation.

Since the system to be controlled is ball-on-plate, it is required to control the roll and pitch (rotation along X and Y axes respectively) of the platform. By designing a controller that takes in roll and pitch as input of the system and positions the ball, we need to calculate the servo angles required to position the platform in the right pitch and yaw. So it is convenient to use Inverse Kinematics for ball-on-plate system. 3D Simulation of Stewart platform gives a visual understanding of the system and its physical limitations. This allows to actually animate all the movements of the platform before building actual prototype. The Sections below give a brief description on the steps taken to building and simulating the Stewart Platform.

3D Modelling in Solid Works

Each part of the Stewart platform shown in Fig. 2.1 is modelled as 3D Part (.ipt file) in Solid Works (2015 Student edition), Then assembled with appropriate joints to obtain a full dynamic model with all the movements. This allowed us to make necessary changes to the model without having built the model to get the required motion and test it. MATLAB add-on in the Solid Works allows us to open this entire 3D model in the MATLAB.

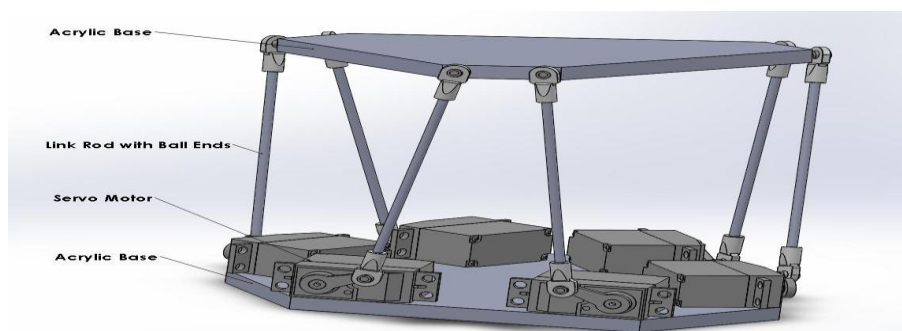


Fig. 2.1:-3D rendering of the Stewart Platform in SolidWorks

SolidWorks

SolidWorks (Fig. 2.2) is a solid modelling computer-aided design (CAD) and computer-aided engineering (CAE) computer program that runs on Microsoft Windows. Solid Works is published by Dassault Systems.

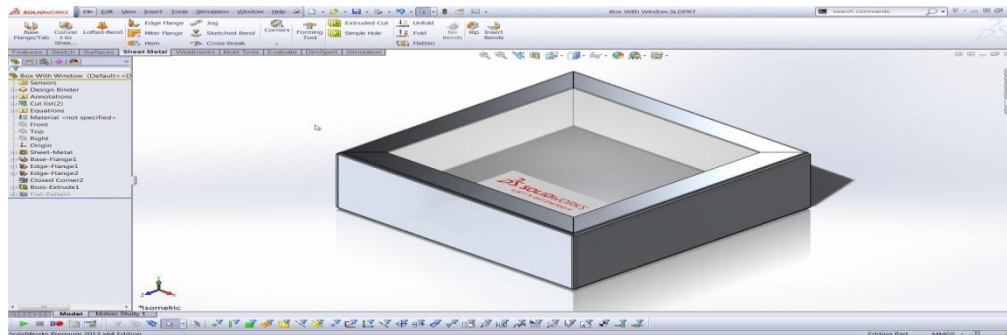


Fig. 2.2:-SolidWorks Window

Solid Works is a solid modeller, and utilizes a parametric feature-based approach which was initially developed by PTC (Creo/Pro-Engineer) to create models and assemblies. The software is written on Parasolid-kernel. Parameters refer to constraints whose values determine the shape or geometry of the model or assembly. Parameters can be either numeric parameters, such as line lengths or circle diameters, or geometric parameters, such as tangent, parallel, concentric, horizontal or vertical, etc. Numeric parameters can be associated with each other through the use of relations which allows them to capture design intent. Building a model in SolidWorks usually starts with a 2D sketch (although 3D sketches are available for power users). The sketch consists of geometry such as points, lines, arcs, conics (except the hyperbola), and splines. Dimensions are added to the sketch to define the size and location of the geometry. Relations are used to define attributes such as tangency, parallelism, perpendicularity, and concentricity. The parametric nature of SolidWorks means that the dimensions and relations drive the geometry, not the other way around. The dimensions in the sketch can be controlled independently, or by relationships to other parameters inside or outside of the sketch.

In an assembly, the analog to sketch relations are mates. Just as sketch relations define conditions such as tangency, parallelism, and concentricity with respect to sketch geometry, assembly mates define equivalent relations with respect to the individual parts or components, allowing the easy construction of assemblies. Solid Works also includes additional advanced mating features such as gear and cam follower mates, which allow modelled gear assemblies to accurately reproduce the rotational movement of an actual gear train. Finally, drawings can be created either from parts or assemblies. Views are auto-matically generated from the solid model, and notes, dimensions and tolerances can then be easily added to the drawing as needed. The drawing module includes most paper sizes and standards (ANSI, ISO, DIN, GOST, JIS, BSI and SAC).

SimMechanics Model

The Solid Works assembly is converted into a Simechanics model shown in Fig. 2.3 (Package in MAT-LAB) using STP and xml files generated by the Solid Works simmechanics add-on. The automated conversion is not efficient, Hence the converted model has to be cleaned by removing redundant joints, adjusting the co-ordinate system, rearranging the block orientation (i.e., Some blocks may have their driven and Driving side interchanged), reorganizing the blocks into subsystems to obtain a manageable Simulink model. Inputs to the Servo angles are created by changing the input method in blocks, this makes the model to respond to the input rather than simulating just random dynamics due to gravity.

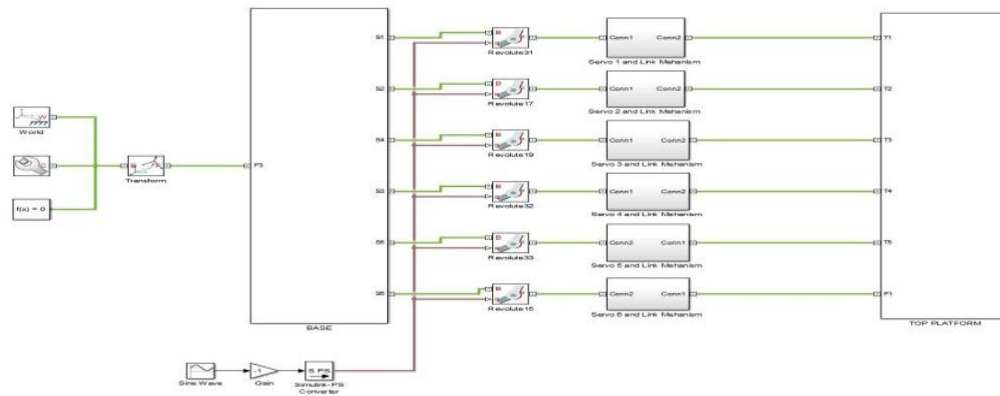


Fig 2.3:-Simulink Model of the Stewart Platform using simmechanics blocks

Inverse Kinematics Simulation in Simmechanics (MATLAB)

Stewart Platform is a Parallel Manipulator (i.e., the one input does not directly influence one output, all 6 inputs have to synchronously move to obtain desired output), the equations of motion of the platform are Highly non linear and forward dynamic simulation is computationally taxing and also real time control is not easily feasible. So a reverse approach is taken, instead of calculating output based on 6 inputs, inputs are calculated to a given set of outputs this method is known as Inverse Kinematics.

Inverse Kinematics Formulation and Simulation

We modelled our Inverse Kinematics (Fig. 2.4) based on [6], kinematic equations are coded in MATLAB as functions and are called by the Simulink model, All 6 motions are Simulated , verified and correctness of inverse Kinematic equations are conformed.

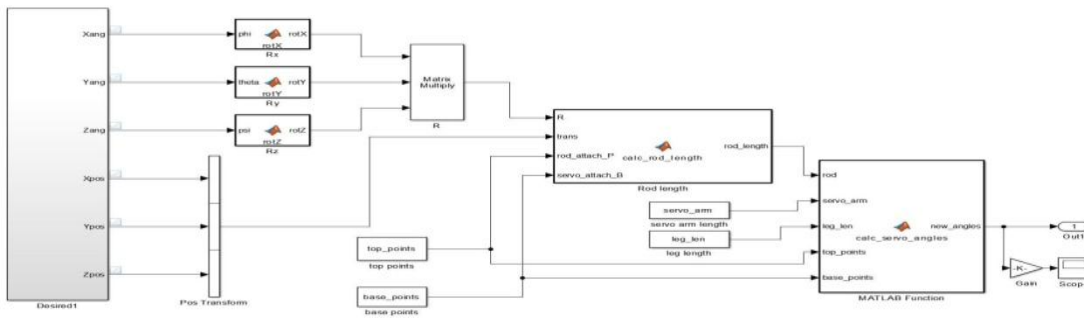


Fig 2.4:-Inverse Kinematic Bolck in Simulink

Simulink

Simulink (Fig. 2.5), developed by MathWorks, is a graphical programming environment for modelling, simulating and analyzing multi domain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in automatic control and digital signal processing for multi domain simulation and Model-Based Design. MathWorks and other third-party hardware and software products can be used with Simulink. For example, State flow extends Simulink with a design environment for developing state machines and ow charts. MathWorks claims that, coupled with another of their products, Simulink can automatically generate C source code for real-time implementation of systems. As the efficiency and flexibility of the code improves, this is becoming more widely adopted for production systems,[5], [6] in addition to being a tool for embedded system design work because of its flexibility and capacity for quick iteration[citation needed]. Embedded Coder creates code efficient enough for use in embedded systems.

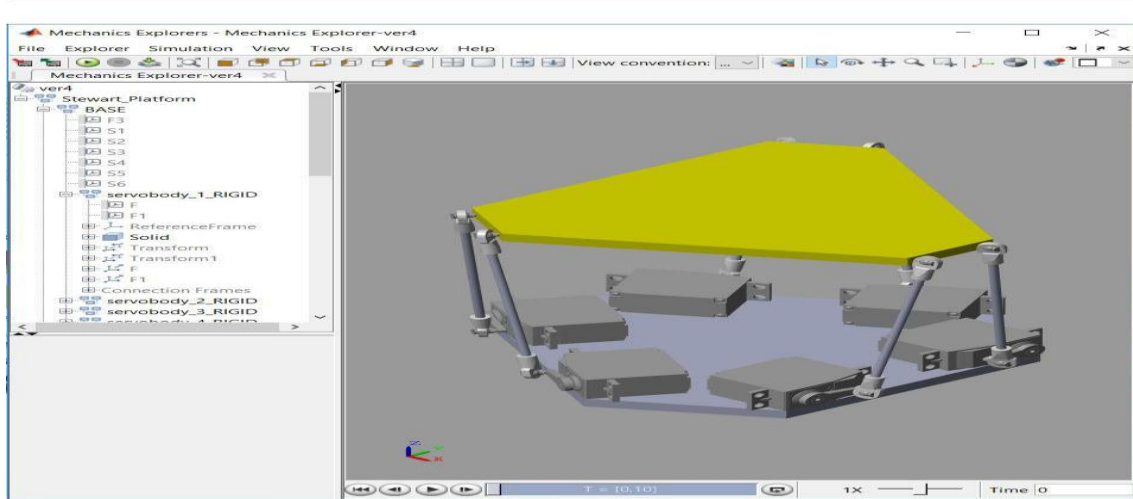


Fig. 2.5:-Simmechanics Simulation of the Stewart Platform

Simulink Real-Time (formerly known as xPC Target), together with x86-based real-time systems, is an environment for simulating and testing Simulink and State-flow models in real-time on the physical system. Another MathWorks product [10] also supports specific embedded targets. When used with other generic products, Simulink and State-flow can automatically generate synthesizable VHDL and Verilog. Simulink Verification and Validation enables systematic verification and validation of models through modeling style checking, requirements traceability and model coverage analysis. Simulink Design Verifier uses formal methods to identify design errors like integer overflow, division by zero and dead logic, and generates test case scenarios for model checking within the Simulink environment. SimEvents is used to add a library of graphical building blocks for modeling queuing systems to the Simulink environment, and to add an event-based simulation engine to the time-based simulation engine in Simulink. Therefore in Simulink any type of simulation can be done and the model can be simulated at any point in this environment. Different type of blocks can be accessed using the Simulink library browser. And therefore the benefit could be taken out from this environment efficiently.

Stewart Plat form Prototype

After simulating the platform, Actual prototype is built and a GUI to control the platform is also built using Java based programming language \Processing". The process followed in building the prototype (Fig. 3.3) is given below.

Physical model

Using the Simulated model as reference a physical model was built.

Top and Base of the platform are made from Plywood (Fig. 3.1).

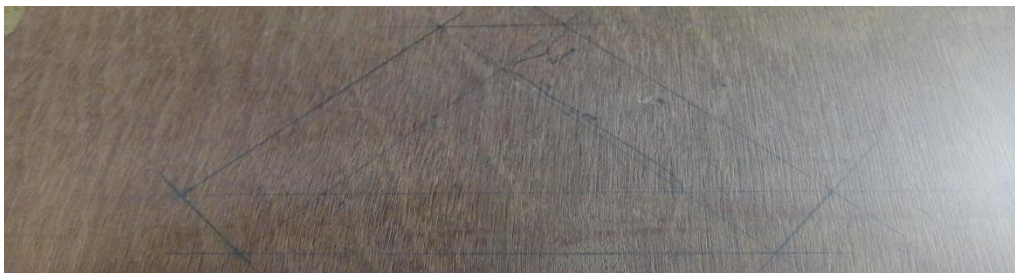


Fig 3.1:-Plywood Base

Used futuba s006 servo motor for actuating the arms (Fig.3.2), this uses pulse width modulated signal to give a precise angular rotation (upto +/- 1 degree accuracy). It was found by experimenting that pulse width of 120 ms to 470ms in 1000 ms period gives a 0 to 180 degree rotation without over turning or physical damage to the servo.



Fig 3.2:-Servo Placed on the base

1. Normal Screws are used to attach servo to the base and also to attach universal joints, top platform.
2. Universal joints with plastic rods are used as connecting rods between base and top platform the length of these are longer than the simulated version.
3. Arduino Uno is used to control the servos using PWM signals over an I2C bus, the arduino is programmed to receive angles from the computer and send corresponding PWM signal to servos, and also programmed to read touch sensor data and send it back to computer using Serial communication with a baud rate of 115200 bits/second.
4. Touch Sensor is used as a feedback device that give the position of a ball when is placed on it, this is 11inch 4 wire touch screen (diagonally) with a 10ms refresh rate.
5. I2C communication protocol is used to send PWM signal to servos. Even though the Arduino has more than 6 PWM ports we have used I2C module to send PWM signals because each servo draws about 250-400mA (depending on load) so 6 servos will easily consume about 2-2.5A but Arduino can only supply about 250mA of current, So this module allows us to power the servos using an external power supply and this also helps is easy routing of cables.

Description of components

Arduino Uno

The Arduino UNO (Fig. 3.4) is an open source microcontroller board based on the ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of dig-ital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board features 14 Digital pins and 6 Analog pins. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. The ATmega328 on the Arduino Uno comes pre-programmed with a boot loader that allows uploading new code to it without the use of an external hardware programmer.

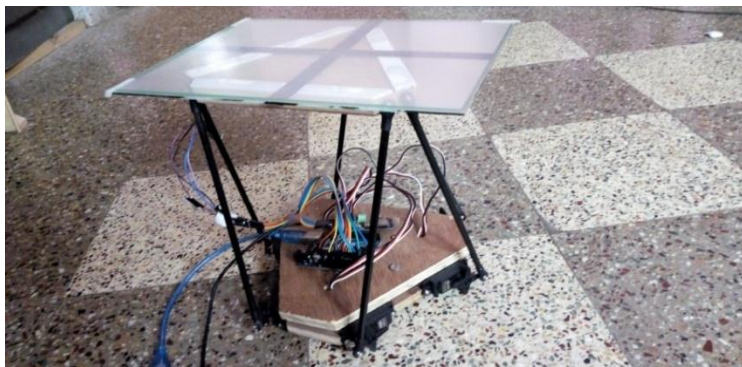


Fig 3.3:-Image of the Stewart platform

Technical Specifications:

1. Microcontroller: ATmega328P
2. Operating Voltage: 5v
3. Input Voltage: 7-20v

4. Digital I/O Pins: 14 (of which 6 provide PWM output)
5. Analog Input Pins: 6
6. DC Current per I/O Pin: 20 mA
7. DC Current for 3.3V Pin: 50 mA
8. Flash Memory: 32 KB of which 0.5 KB used by bootloader
9. SRAM: 2 KB
10. EEPROM: 1 KB
11. Clock Speed: 16 MHz
12. Length: 68.6 mm
13. Width: 53.4 mm
14. Weight: 25 g



Fig 3.4:-Pin Configuration of Arduino Uno

Servo motor

A servomotor Fig. 3.5 is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Most hobby servos use a standard type of 3-pin plug, with the same control signalling. The connector is a female, 3-pin, 0.1" pitch header. One thing that can be confusing is that the wiring colour code is not always consistent there are several colour codes at play. The good news is that the pins are usually in the same order, just that the colours are different. Table I gives the wiring colour code of Servo motor

Table I:-Wiring colour code of Servo motor

Pin Number	Signal Name	Colour Scheme
1	Ground	Black
2	Power Supply	Red
3	Control Signal	White

The third pin of the servo connector carries the control signal, used to tell the motor where to go. This control signal is a specific type of pulse train. The pulses occur at a 20msec (50 Hz) interval, and vary between 1 and 2msec in width. The Pulse Width Modulation hardware (Fig. 3.6) available on a microcontroller is a great way to generate servo control signals. Common servos rotate over a range of 90 as the pulses vary between 1 and 2msec they should be at the center of their mechanical range when the pulse is 1.5msec.



Fig. 3.5 Futubas3003 servo motor

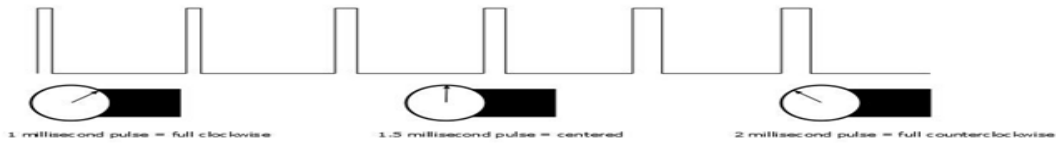


Fig. 3.6 PWM control of servo motors

Touch Panel

The sensor used to obtain the position of the ball is a 4-wire resistive touch panel (Fig. 3.7). A transparent screen with four wires coming out of it is used for interfacing. Though transparent, but consists of 2 resistive layers lying parallel to each other one above the other having an air gap (a kind of insulation) in between. What happens is that when the user touches the touch screen, the two resistive layers gets connected and a voltage divider resistive network is established. In order to conform that, we will have to calculate the co-ordinates. To calculate the coordinates on the touch screen where the user has touched, we play with four wires coming out of the touch screen (Fig. 3.8). The two wires indicating the x-axis and the other two the y-axis (x and y axis here are the relative directions). So, after the formation of a potential divider network, we apply voltages to X+ and X- and making Y+ OR Y- as a probe (means we take voltages from Y+ or Y-), in the same fashion we take voltage from the X+ or X-. These two voltages we get are function of the resistance which will vary throughout as you move in the touch screen.

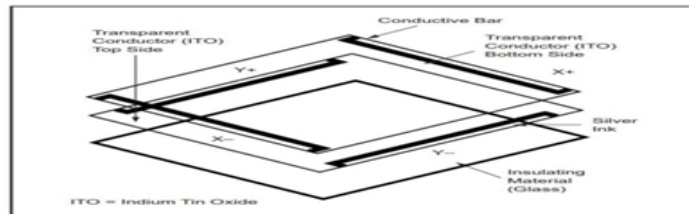


Fig. 3.7 Touch Screen Configuration

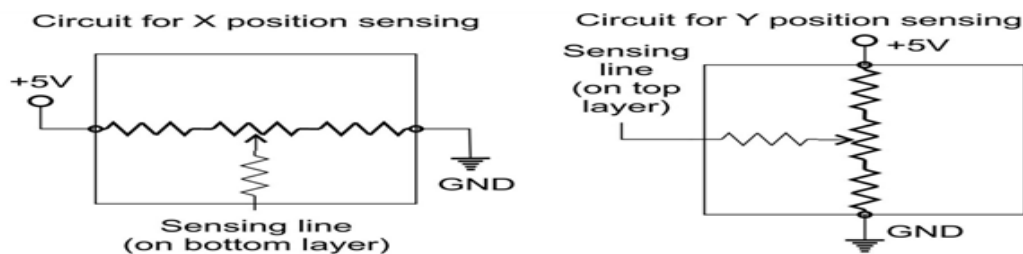


Fig. 3.8 Circuit to measure the X and Y position

PWM module

The PCA9685 is an I2C-bus controlled 16-channel PWM controller (Fig. 3.9). Each PWM output has its own 12-bit resolution (4096 steps) and frequency individual PWM controller that operates at a programmable frequency from a typical of 24 Hz to 1526 Hz with a duty cycle that is adjustable from 0 % to 100 %. All outputs are set to the same PWM frequency. Each output can be on or off (no PWM control), or set at its individual PWM controller value.

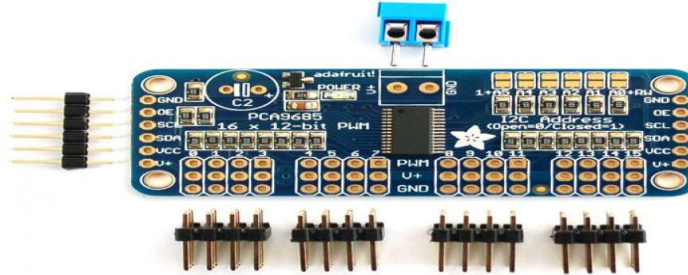


Fig 3.9:-PCA9685 16-channel I2C PWM module

The Inter-integrated Circuit (I2C) Protocol (Fig. 3.10) is a protocol intended to allow multiple slave digital integrated circuits (chips) to communicate with one or more master chips. Like the Serial Peripheral Interface (SPI), it is only intended for short distance communications within a single device. Like Asynchronous Serial Interfaces (such as RS-232 or UARTs), it only requires two signal wires to exchange information.

Communication via I2C is more complex than with a UART or SPI solution. The signalling must adhere to a certain protocol for the devices on the bus to recognize it as valid I2C communications. Fortunately, most devices take care of all the odd details for you, allowing you to concentrate on the data you wish to exchange.

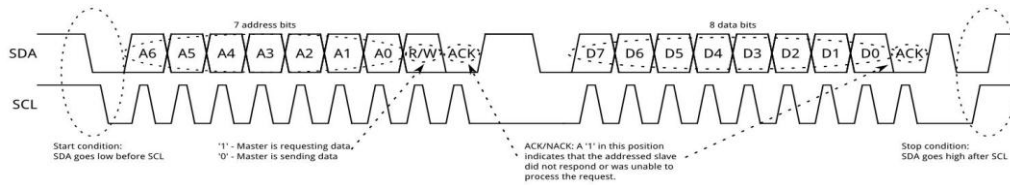


Fig 3.10:-I2C protocol Signal diagram

Messages are broken up into two types of frame: an address frame, where the master indicates the slave to which the message is being sent, and one or more data frames, which are 8-bit data messages passed from master to slave or vice versa. Data is placed on the SDA line after SCL goes low, and is sampled after the SCL line goes high. The time between clock edge and data read/write is defined by the devices on the bus and will vary from chip to chip.

Software to Control the Stewart platform

We wanted a Simple program that is cross platform and light to Control the platform (Fig. 3.11). We use Processing instead of MatLab to communicate to our Arduino in the platform. Processing is Java base programming language it is much easier to create GUI and Visualizations, It has built in libraries to communicate to Arduino which works well for our application. We translated the Inverse Kinematics Code from Matlab to Processing.

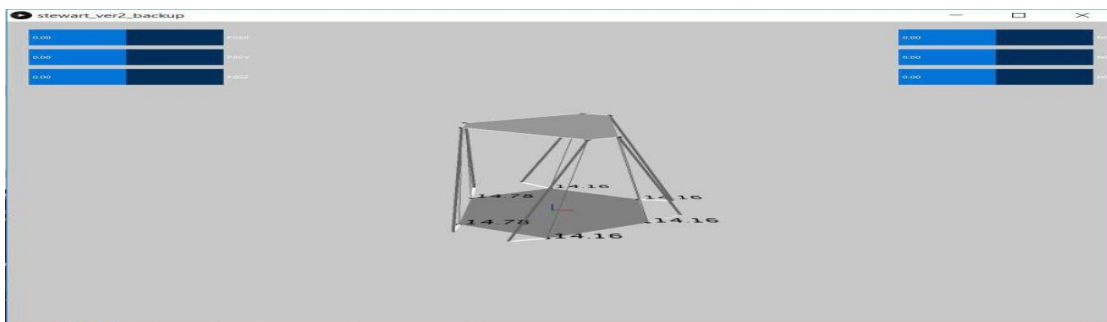


Fig 3.11:-GUI Software to control the Stewart Platform

Prototype testing

The physical model and Software were successfully integrated. A Game controller is used to test range to desired motion i.e., X, Y, Z translations and rotation about each axis of top platform.

Control system design and simulation

Modelling of the System

The Control problem is to balance a ball on the top platform of the Stewart platform using feedback control techniques.

The general name for ball balancing system is "Ball on Plate System". The papers referred to obtain the mathematical model of the system are [7, 8]. With some modification such as ball inertia and neglecting friction etc., nonlinear State Space model is obtained.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} x_2 \\ B(x_1x_4^2 + x_4x_5x_8 - g \sin x_3) \\ x_4 \\ 0 \\ B(x_5x_8^2 + x_1x_4x_8 - g \sin x_7) \\ x_6 \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

Fig 4.1:-Non linear State Space model of ball-plate system

Design and Simulation of PID Controller

The modelled system is linearized over position x, y = [0, 0] and angles about axis to [0, 0] and transfer function of the system is obtained. This transfer function is used in Simulink and PID tuner (Fig. 4.3) to design a robust PID controller (Fig. 4.2-4.4).

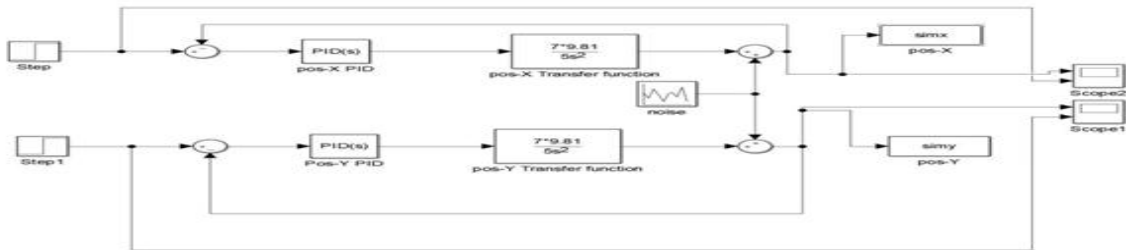


Fig. 4.2 Simulink Block Diagram to simulate PID controller

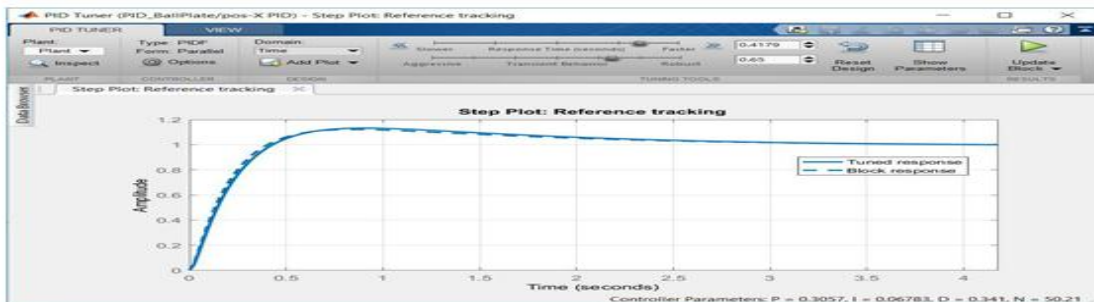


Fig. 4.3 PID tuner to obtain required response

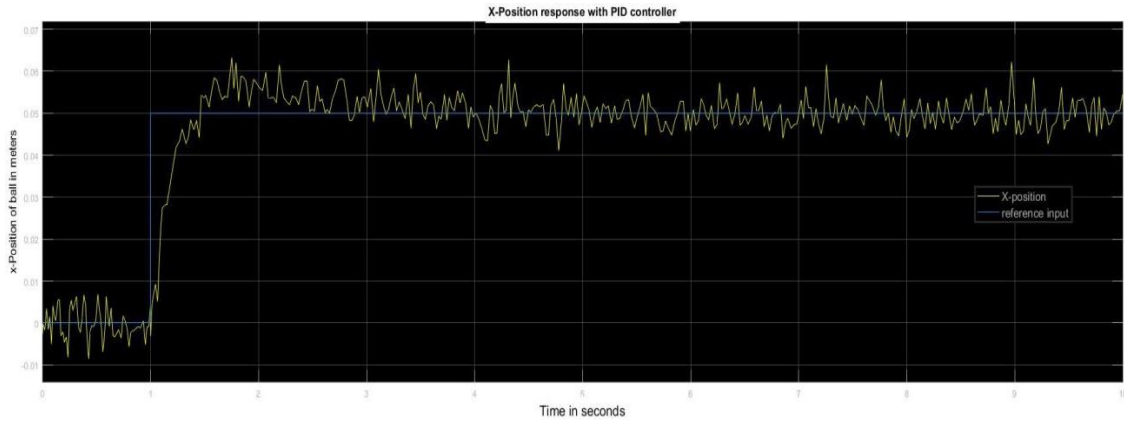


Fig 4.4:-Position-X response and input with PID controller

Design and simulation of LQG Controller

Linear Quadratic Gaussian or state feedback controller with Kalman filter as state estimator. We use Control law $u = K x$ to control input based on the current state x of the system, to obtain a optimal K we use `lqr` command in matlab to solve the optimal control problem using "Ricatti Equation". Since Touch sensor can only give position information of the ball we are using a state-estimator to estimate other states like angle of plate, velocity of ball, etc. This is done by using `lqg` command and spacing the Noise and Disturbance Co-variance matrix. The matlab code for obtaining K and Kalman filter is written.

The designed State estimator and regulator are used to simulate ball-plate system in Simulink (Fig. 4.5 and 4.6).

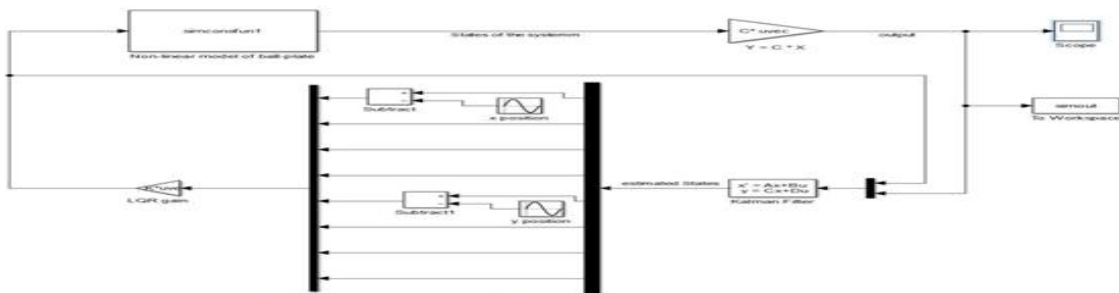


Fig. 4.5 Simulink model to simulate ball-plate system with State estimation

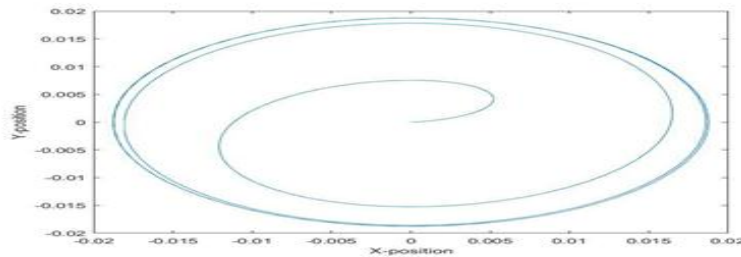


Fig 4.6:-Ball position output when the system is made to follow circular trajectory

Implementation of Designed controller in actual system

Implementing PID control

The continuous PID Designed in previous step is converted in to discreet system and are coded into the Stewart Platform Control Software designed in Section 2. However, including of Integral controller tends to destabilize the system, so I controller is removed and only PD controller are being used. The PD gains adjusted to obtain smooth motion.

Successfully implemented the PD controller, the platform can autonomously balance the ball in the given position.

The class PID is written in processing which takes inputs of Proportional Derivative and Integral gains and outputs the controlled system input this class is called twice for X-axis and Y-axis.

Implementing LQG control

Implementing of LQG controller involves discretization and online ODE solving, the coding of Kalman filter into the Java is left for future work

Results and Conclusion:-

Results

Modelling and Simulation

1. Stewart platform is successfully 3D modelled as shown in Fig. 2.1.
2. The 3D model of the platform is successfully simulated along with Inverse Kinematic solution to obtain required motion (Fig. 2.5).

Control System Design

1. Mathematical Model of ball-on-plate is obtained and coded into matlab State-Space model.
2. PID controller is designed using PID tuner and Proportional, derivative and integral gains are found as shown in Fig. 4.3.
3. LQG Controller is designed as in section 4.4. Optimal regulator gain K and State estimator is obtained as sysKF. The system is simulated along with LQG as shown in Fig. 4.5.

Prototype Implementation

1. The prototype of Stewart Platform was successfully built as shown in Fig. 3.3.
2. GUI program to control the Platform was successfully implemented in Processing.
3. Successfully implemented the PID controller on the prototype (section 5.1). The prototype balances a ball on the platform autonomously.

Conclusion:-

Stewart Platform is complex manipulator whose orientation can be easily controlled using Inverse Kinematics. Ball-on-Plate is a nonlinear system with 8-states. This system can be linearized and control system can be designed using this linearization. The designed control can be implemented to actual non-linear system, as long as the system (ball-on-plate) is within linearized limits the designed control is able to control the plate in a steady state. In Control design section we have discussed two methods of control but only PID control is implemented in the actual system. Implementation of LQG controller can be done in our future work. The PID is designed to have a steady state time period of 1.5 to 2 seconds. This can be improved by using different feedback types such as H1 and Machine Learning Control are the scope for further work for the control of ball-on-plate system.

Acknowledgements:-

The F.A. would like to express the deepest thanks to the principal, PESCE, Mandya, and the Management, PET(R), Mandya, for their timely support and constant encouragement in successful completion of this research work.

References:-

1. L. G., L. K., and L. F. L., "Dynamic analysis and control of a Stewart platform manipulator", *Journal of Robotic Systems*, Vol. 10, No. 5, pp. 629-655 (Available Online): <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.4620100506>.
2. D. Li and S. E. Salcudean, "Modeling, simulation, and control of a hydraulic Stewart platform," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, April 1997, pp. 3360-3366.
3. S.-H. Lee, J.-B. Song, W.-C. Choi, and D. Hong, "Position control of a Stewart platform using inverse dynamics control with approximate dynamics," *Mechatronics*, vol. 13, no. 6, pp. 605-619, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957415802000338>
4. C. GUYEN, S. ANTRAZI, and Z.-L. ZHOU, "Analysis and design of a six-degree-of-freedom stewart platform-based robotic wrist", semi annual technical report, 1 Aug. 1990- 1 Jan. 1991.
5. D. T. Gee Kwang Lim, Robert A. Freeman, "Modeling and simulation of a Stewart platform type parallel structure robot", NAL report; 1 Dec. 1987 - 30 Nov. 1988, Apr 01, 1989, no. 89N23875, 1988.
6. Ramrez Gomez A, "Inertial stabilization system based on a Gough-Stewart parallel platform," B.S. thesis, Universitat Politecnica de Catalunya, 2017.
7. M. Nokhbeh, D. Khashabi, and H. Talebi, "Modelling and control of ball-plate system," B.S. thesis, Teheran: Amirkabir University of Technology, 2011.
8. F. Dusek, D. Honc, and K. R. Sharma, "Modelling of ball and plate system based on first principle model and optimal control", in *Process Control (PC)*, 2017, 21st International Conference on. IEEE, 2017, pp. 216-221.