iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VIII **Month of publication:** August 2022

DOI: https://doi.org/10.22214/ijraset.2022.46190

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VIII Aug 2022- Available at www.ijraset.com

Ball Balancing PID System using Image Processing

Pradeepa S C[1], Abhijith N[2], Amoghavarsha S G[3], Kiran C N[4], Niranjana Jois H C[5]
[1,2] *Asst Professor, ECE Department. JNNCE , Shivamogga*
[3,4,5] *4th year, ECE Department. JNNCE , Shivamogga*

Abstract: ***Control theory and its applications are crucial when operating within the area of dynamic systems. Compensating for disturbances and external actions imposed on a given system being inherently unstable or semi-stable. This project consists in the realization of a Ball-Plate system with 2 degree of freedom which will control the position of the ball by tilting the plate. Two servomotors will allow the table to be oriented with a certain angle of inclination to counterbalance the movements of the ball. A Pi camera placed above the device will transmit live images of the plate to a Raspberry Pi which will be responsible for analyzing them to draw conclusions such as the position of the all, its speed and acceleration.***
A Python program will then be responsible for processing these data, feed it to designed PID controller and transmitting the result by USB to the Arduino which will control the rotation of both servos independently. All the data will be sent to the Python program running in another computer using MQTT protocol. And set point, error and PID v/s time graphs would be plotted. The experiment is done by dropping a ball on the plate and waiting for the system response to balance the ball in the center of the plate and to follow the set trajectories.
The plate is made of wood to ensure smooth slipping of the ball without friction. Balancing experiments were done, following the set trajectories, and positioning the ball at the center of the plate.
Through the use of multithreading, image processing at the rate of 90 FPS was achieved. And results were plotted. In conclusion it should be emphasized that in control systems acceptable real time performance can be achieved by decentralizing the processing unit into several PUs. It is notable to mention that with small additional piece of software and control checkpoints PU redundancy feature can be added to the plant, so in the event of physical damage and PU failure the other PU can be notified and take over the system.
Keywords: Ball balancing, PID, Image Processing, Raspberry Pi, Arduino, Python.

I. INTRODUCTION

Digital image processing consists of the manipulation of images using digital computers. Its use has been increasing exponentially in the last decades. Its applications range from medicine to entertainment, passing by geological processing and remote sensing. One of the advantages of machine vision is the ability of sensing moving objects from distance and without contact with the object. However, the data in the image can be easily affected by the disturbance and changes like sunlight and shadow that occur in the surrounding area of the object being sensed. Such issue can be overcome by using digital image processing and advanced image filtering techniques to get rid of such noises and disturbances. The goal of this project is to develop a Ball-Plate balancing system which uses camera above it to get feedback on the position of the ball. The Ball-Plate system has 2 degree of freedom (DOF) the goal is to balance the ball on a horizontal plane on a given set point. Position on of the ball on plate is represented by 2 coordinates system X-axis and Y-axis, once the position of the ball is known, error is calculated based on set point and developed PID controller is used to actuate the 2 servomotors independently, one servo is used to tilt the plate on X-axis and other to tilt the plate on Y-axis both will use separate PID controllers. The camera above the pate will capture continuous frames that will be processed using Python in order to get the coordinates of the ball. The servomotors will be controlled by Arduino which will receive the angles of the servomotors from software developed using Python which is capable of image processing, designing a controller and serial communication. Document is a template. For questions on paper guidelines, please contact us via e-mail.

II. LITERATURE SURVEY

We referred many papers among which some will be discussed in later section we have gone through many concepts by referring these papers, many old papers and articles were also referred but latest papers had many improvements over the previous ones. New methodologies and approaches were discussed. We referred many latest papers related to this project out which five of them are discussed below.

A. *Lefrouni, K., Moubachir, Y. and Taibi, S., 2021. Control of Ball-Plate System with observer-Based State-Feedback and Geometric Considerations. International Journal of Difference Equations (IJDE), 16(1), pp.35-46.*

1) Linear model of the ball and plate system was developed by applying Lagrange's method.
2) Ball-plate system was studied using a frequency domain approach.
3) The developed controller is based on a state feedback controller and Luenberger observer.
4) Choice of observer gains within the stability regions is difficult

Conclusions

- The stability of the closed loop system is consequently proven.
- Simulations and experimental results of demonstrations were not so excellent in tracking performance and control design
- Back stepping is classic method to solve the control design problem cascaded under-actuated system by transforming the control design problem of entire system into several lower order system.
- Model was done using load gear and driver gear which was introducing more friction component than usual.

B. *Ali, E., and Aphiratskan, N.(2015, December).AU ball on plate balancing robot. IEEE. International Conference on Robotics and Biomimetics(pp. 2031-2034).*

1) PC-Based Controller, Linux
2) More than one PUs.
3) 2 axis PID control
4) Matlab/Simulink-based
5) Raspberry Pi 2
6) Slower FPS due to single threading

Conclusions

- In system with high number of active actions the derivation of mathematical model becomes somewhat impossible, therefore adopting a model free approach using feedback to calculate the responses is the only rescue.
- For control mechanism raspberry pi 2 is used which equips 700Mhz ArM 11762P-S processor solely for image processing task and a STM 3219407 board the includes 168MHz ARM Processor solely for control of the servos.

C. *Gustavo and L. Juan and J. B. Jovani, "Control of a ball-and-plate system using a State-feedback controller". Ingeniare. 28. 6-15. 10.4067/S0718-33052020000100006, 2020*

1) Mechanical constraints with which the central ball joint (central piece) was designed only allows inclinations of ± 15º in the x-y plane.
2) State feedback controller using the pole assignment method was designed to maintain a desired position of the sphere over the plane
3) Phenomenological analysis to obtain a state-space representation of the system

Conclusions

- Actuation mechanism is based on RC servos motion which are connected to the beneath of plate through arms and sphere joints.
- The ball position detection is done by capturing frames with a single Raspberry Pi camera module that is placed on top of the plate.
- Two processing units Raspberry Pi and STM32F407.
- Main disadvantage is that using such a system with high quality images leads to decrease in FPS.
- This gives a period of 66ms between each ball coordinate being send to second PU.
- This was not enough to let the second PU to calculate the PID values while giving the system acceptable precision in real time.
- PID tuning was manually by Ziegler-Nicholas method
- Due to this the ball dropped on plate came to rest at about 28 secounds.

D. Mohsin, M., T. (2015). Development of a Ball and Plate System.122nd ASEE Annual Conference and Exposition.Washignton.

1) Three-degree-of-freedom system.
2) Simulations using Simulink/MATLAB™.
3) Touch panel platform.
4) The control system utilizes Euler-Lagrange equations.
5) An irregularity of the system performance may result from the sensitivity of the touch screen and control algorithm.

Conclusions

- In this paper mainly four stages of engineering analysed and designs were addressed.
- 1st state corresponds to the phenomenological analyses of the movement of ball on plate from which its representation in space state is obtained.
- In 2nd stage with the previous results a state feedback controller using the pole assignment method is developed and the establishment time and
- Percentage overshoot as the main design parameter are considered.
- In 3rd stage simulations are performed in Simulink in order to verify the fulfillment of desired parameters.
- In the final stage, ball plate system controlled and implementation is experimentally validated

E. M. Yaseen, J. A. Haider, "Modeling and control for a magnetic levitation system based on SIMLAB platform in real time", Results in Physics, Volume 8, Pages 153-159, ISSN 2211-3797, 2018.

1) Maglev system model.
2) 4 electromagnets as actuators for applying magnetic forces to achieve levitation and position control.
3) LQR based controller, PID control and Lead Compensation.
4) Real-time control Simulink feature of (SIMLAB) microcontroller

Conclusions

- As the work use that touch panel for the main point locator a sensor mount was fabricated for the touch panel to fit within the platform to eliminate any sliding, so the risk of damaging of the costly sensor was a bit less.
- This 4-wire resistive feather touch panel obtains the ball location very easily.
- It consisted of two layers, as the top layer was flexible it causes the surface to deform when a force is applied the operation force range is around 0.02 to 0.3 Newton, which in turn produces current as an indication.
- The project also used linear actuators to change the angle of the plate with respect to ground.
- The main delivers due to the linear actuators only.
- The ball location was to updated within milliseconds but the movements of the linear actuators wasn't up to the required level.

III.DESIGN

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

A. System Design

The Ball-Plate system has 2 degree of freedom (DOF) the goal is to balance the ball on a horizontal plane on a given setpoint. Position on of the ball on plate is represented by 2 coordinates system X-axis and Y-axis, once the position of the ball is known; error is calculated based on setpoint and developed PID controller is used to actuate the 2 servomotors independently, one servo is used to tilt the plate on X-axis and other to tilt the plate on Y-axis both will use separate PID controllers.

The camera above the plate will capture continuous frames that will be processed using Python on the Raspberry Pi in order to get the coordinates of the ball, these coordinates are sent to the PID controller which is also running on the Raspberry Pi and the computed servo angles are sent to the Arduino. The Arduino then receives the angles of the from the Raspberry Pi and change the servomotors angle accordingly.

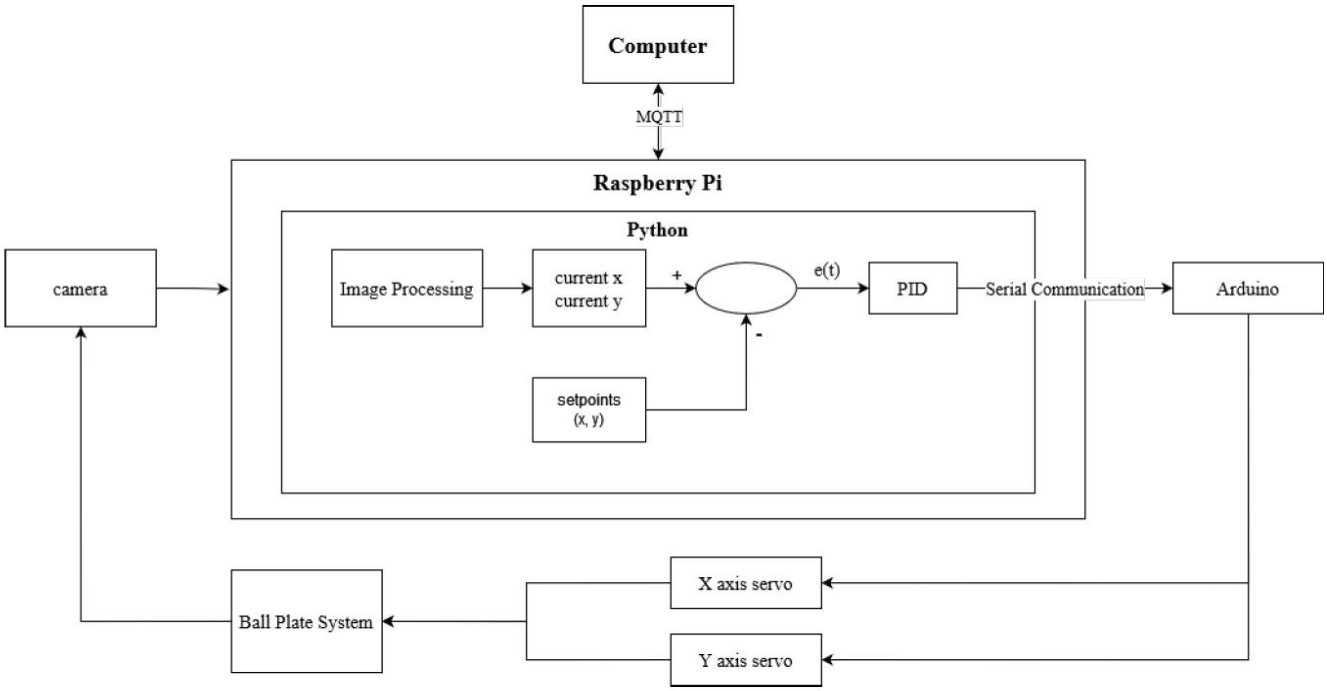


Fig. 1 Block diagram of the PID system.

B. Circuit Descriptions

As follows below in Fig.2 is the electrical circuit and the respective components. Besides wiring the components are the Arduino Uno microprocessor, two servo motors, a small breadboard and a voltage step down component. The Arduino and the servo motors are powered from an external power source stepped down to 5V. The capacitors are used to prevent temporary power deficit and assure continuity in the power supply to the servo motors.

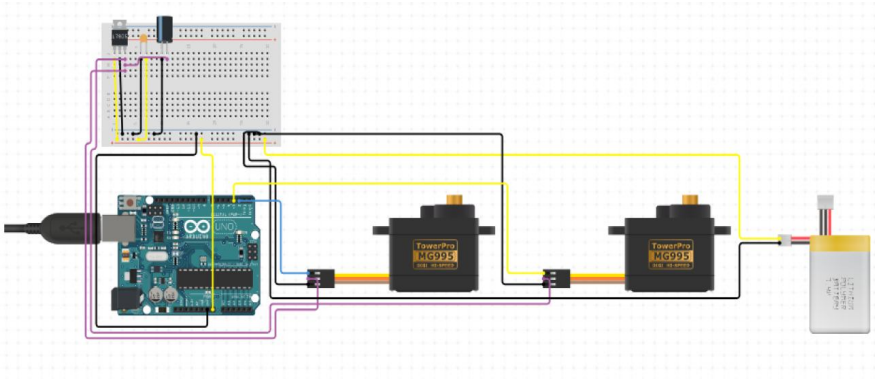


Fig.2 Servo Circuit Diagram

IV. IMPLEMENTATION

A. Developing Multithreaded Image Processing System

First stage is to get frames from Pi camera. We use picamera library to get frames from pi camera in python. Since we needed to get frames at 90 FPS resolution should be set to 640×480 and frame rate should be 90 and video port should be set to True.

B. Finding the Corners of the Table

Since we use the video port the frames we get will be encoded and we are not in numpy array format so we need to convert it to numpy array format that is of shape (x, y, 3) y is number of rows, x is number of columns and 3 represents colour channels (R, G, B) in opencv (B, G, R).

Fig.3 Table Sketch.

C. Finding the Position of the Ball on the table

First as discussed earlier we don't get the image in *numpy* array format so we have to decode it using *np.from* buffer and *cv2.imdecode* functions. After getting the frame in BGR format we call *self.find.ball* method. We can find the ball by performing color segmentation.Which can be done in RGB or BGR or HSV colour space. The problem we found in RGB colour space is that whenever there is a small change in lighting conditions in the Room all the values in RGB would change i.e., R & G & B differently, which was making it harder and inaccurate ball detection.So first we convert the BGR frame to HSV using the function *cv2.cvtcolor*. Then we threshold the image using lower limit and upper limit in HSV space which we found using experimentation. *cv2.inRange*. Then in order to reduce the noise we do morphological transformations like open and close. Then find all the contours in the frame and pick the contour with the maximum area. Then we set the ball-pose to be the position and radius obtained which will be used in PID controller.

D. Building the PID Controller

We need some kind of controller in order to make use of the ball positions data we get from the image processing threads and bring the ball to the requested setpoint. We choose PID controller since it is widely used in industrial systems and it is easy to implement.

$$u(t) = K_p e(t) + K_i \int e(t)\mathrm{d}t + K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t) \qquad (1.1)$$

Where,

Kp: proportional gain that is proportional to the error
Ki: integral gain that interprets for past errors.
Kd: derivative gain that interprets for future errors.

PID stands for Proportional-Integral-Derivative which is represented by Equation 1.1. As we can observe in the equation Kp is an proportional constant meaning Kpe(t) increases proportionally as e(t) (error) increases. This will help the ball to get closer to the setpoint however when e(t) = 0, Kpe(t) = 0, so the plate is flat but e(t-1)≠0. So the ball will have momentum so the ball will move past the setpoint and Kpe(t) will account for that again the same process will continue thus inducing oscillations.

Kd is the derivative constant and Kd d/dt is a derivative form as the ball approaches the setpoint Kpe(t) decreases but derivative term Kd de/dt does not since de/dt is the velocity of the ball hence derivative term increases which tilts the table opposite to the motion of the ball trajectory hence reducing the oscillations.

Ki is an integral constant and Ki ∫ e(t) dt is an integral term. This helps us to reduce the static error i.e., when the ball is very close to the setpoint the proportional term maybe very less and since the ball will be stationary. Derivative term will also be 0. But the integral term adds the error in each iteration thereby reducing the static error.

E. Deciding the Layout so as to Build the PID Controller

Things to be keep in mind while designing the table servo mounting

1) Increasing the servo angle should move the table up.

2) Decreasing the servo angle should move the table down.

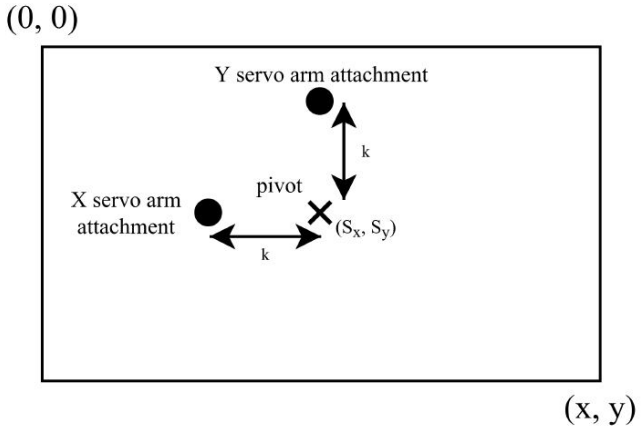


Fig.4 Table for dimensions.

F. To select Appropriate Actuators Parameters and Mechanical Structure for the Control of ball-plate System

While implementing the PID controller layout of the table was decided Fig.5 and now the distance from the centre of the table to the server armed attachments was decided to be 8 cm i.e., K=8cm. And the height at which the pi camera is mounded is 75 cm from the base of the platform.

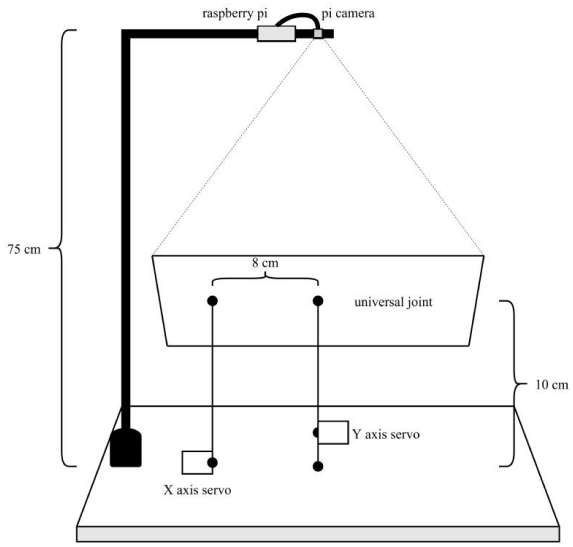


Fig.5 Designed Mechanical Structure.

G. To send position of ball to the computer using MQTT

By default MQTT broker listens on port 1883. In order to communicate with MQTT broker from Python we installed a library called paho mqtt *pip install paho_mqtt*. In order to send messages we need to first establish connection to MQTT broker. We can connect to MQTT broker by using a method called connect.

Since we are connecting to broker from raspberry pi where MQTT broker is also installed brocker_url = "localhost" and brocker_port = 1883. Once we are connected to broker we can subscribe to a topic by using the method client.subscribe We can publish by using the method client.publish Since we use these methods multiple times we made ourselves a library called iot.py.

H. Receiving the setpoint and ball position

In order to receive the ball position we need to subscribe to the topic *ball_set_pose* we can do that by importing our library 7.3 and using *subscribe_thread_start* method.

The payload we receive will be a JSON string of format:

"{"setpoint":[x,y],"ball_pose":[[x,y], r],"pid_values":[x,y]}"

We can convert it back to Python directory by using a method json.load from JSON library. In case of animating the ball we need only set point and ball pose so following call back function is used.

where ball_pose_plot, PID_plot and setpoint_plot are array objects from multiprocessing library which offers multithreading protection.

I. Animating the ball position

In order to animate the ball position first we take a static table or platform image and add the ball image of size 2*r x 2*r at the position received from MQTT where 'r' is the radius of the ball received from MQTT. Since the radius of the ball changes and resizing the image is computation intensive, this would result in decrease in FPS which would make animation look choppy. Thus we resize the ball image from 2 x 2 till 140 x140 and made a dictionary with keys as radius of ball and value as images. And we stored it using pickle library. So now we can just use pickle to load the dictionary into memory and there is no need to compute and resize the image. We can just get the required image by passing the radius of ball as key.

J. Plotting the graphs

Graph plotter will run as a separate process so it again uses iot library in order to get ball position, setpoint and PID value data from MQTT. In order to plot any live data we need to have a buffer which stores the data and when new data is received we need to delete oldest data and insert the new data at the start so that it gives us that rolling effect. We use double ended queue data structure for this purpose. Where we insert the new data at the end and remove old data from index zero.

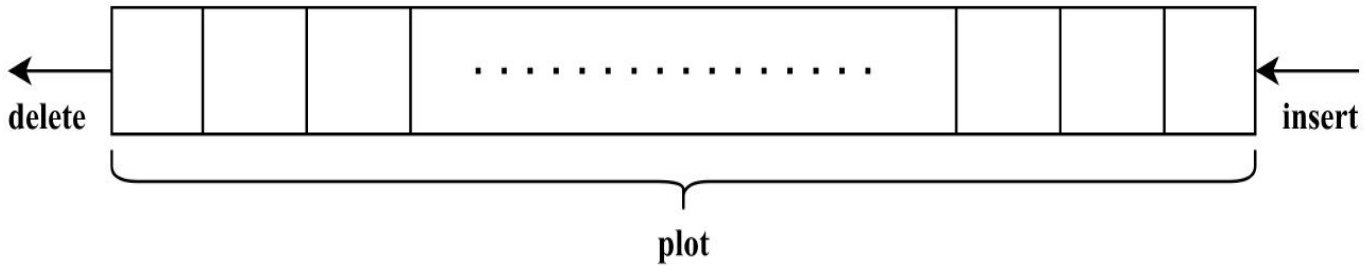


Fig.6 Dequeue Operation

First we used *matplotlib* in order to plot the graph but it was too slow so we switched to by *pyqtgraph* instead which uses Qt. Then we used timer from *Qtimer* in order to call graph plotter function which plots the data in dqueue every 1/800 second. The *dqueue* will be updated in parallel in a separate thread whenever the data comes from the MQTT.

V. RESULTS AND DISCUSSIONS

A. Developed the image processing system and PID controller using Python on Raspberry Pi

1) Installed Raspbian OS on Raspberry pi.
2) Interfacing the PiCamera.
3) Wrote efficient multithreaded image processing code in python which

The system has to work with the following steps

- Finds the corners of the table.
- Finds the position of the ball on the table.
- Process the frames at 90FPS

Decided the layout so as to build the PID controller.

Fig.7 Raspberry Pi and PiCamera

Fig.8 Platform

Fig.9 Servo Placement

Fig.10 Overall Mechanical Structure

B. Send position of ball using MQTT and animate and plot the ball position

Mosquito MQTT broker is installed on Raspberry Pi through which position of the ball, setpoint and the PID values is sent to the computer.Using these values ball position is animated using existing background image.And setpoint, error and PID vs time graphs are plotted.

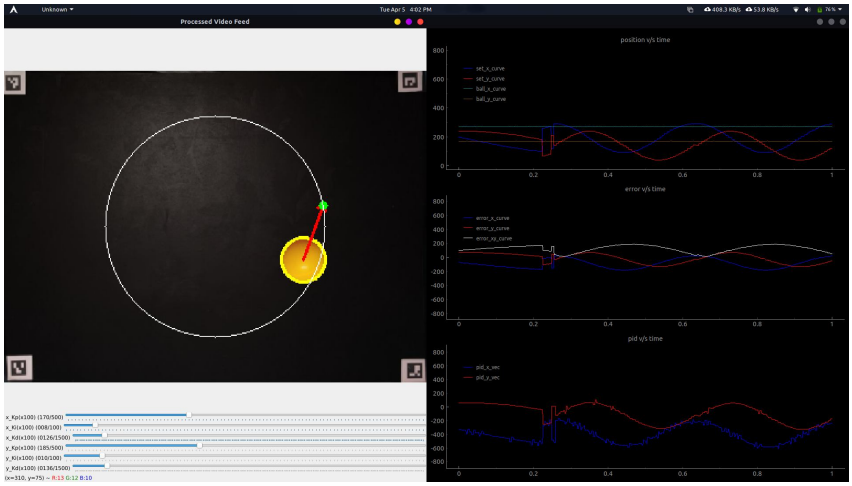


Fig.11 Interface

C. PID and Serial communication parameters used

Parameter	Value
kp_x	1.70
ki_x	0.08
kd_x	1.26
x_offset	1535
exp_filter_alpha_x	0.85

TABLE I: X axis PID tuning parameters

Parameter	Value
kp_y	1.85
ki_y	0.1
kd_y	1.36
y_offset	1410
exp_filter_alpha_y	0.9

TABLE III: Y axis PID tuning parameters

Parameter	Value
port	/dev/ttyACM0
baudrate	115200
write_timeout	30 ms
seperator	'#'
stopByte	'$'

TABLE IIIII: Serial communication parameters

VI. CONCLUSIONS & FUTURE SCOPE

A. Conclusions

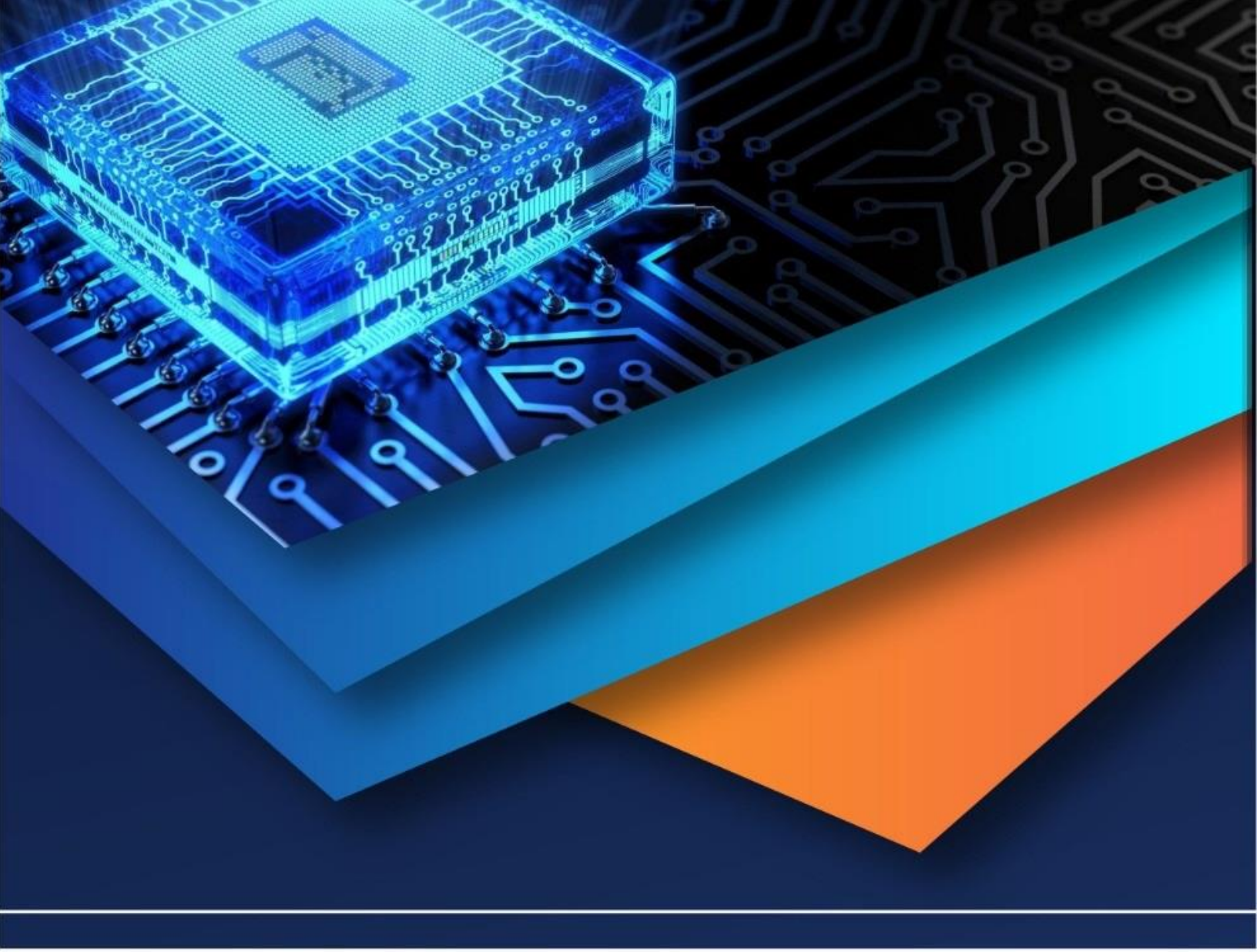
The overall system will consist of two distinct processing units, one controls the servo motors while the other runs the PID control and the image processing task to track the ball position. Both PUs will be coupled via a serial UART communication channel. The control mechanism is a free-model approach based on PID control. The PID tunning will be done manually by applying Ziegler-Nichols method. The experiment will be conducted by dropping a ball on the plate and waiting for the ball equilibrium around the center of the plate. A disturbance will be introduced after having the ball stabilized and the motion of servo arms and ball coordinates will be recorded and sent using MQTT and plotted on another computing system. The image processing rate 90 FPS is achieved. In conclusion it should be emphasized that in control systems acceptable real time performance can be achieved by decentralizing the processing unit into several PUs. It is notable to mention that with small additional piece of software and control checkpoints PU redundancy feature can be added to the plant, so in the event of physical damage and PU failure the other PU can be notified and take over the system.

B. Future Scope

There is a bit of play in the servo motors and arm joints and links in our construction, we could address this by using better precession 3D printed parts. And there is also vibrations in the whole structure due to shattering in the servo motors and we would like to address this by better shock absorbing material at the bottom of the mechanical structure than the double sided tape we are using now. And we would also like to address mounting errors of the platform allowing play and Deformation and give in solid parts. And investing in better servos will also increase precession and will reduce the vibrations in the structure.

REFERENCES

[1] Ali, E and Aphiratsakun, N., AU ball on plate balancing robot, International Conference on Robotics and Biomimetics, pp.2031-2034, IEEE December 2015.
[2] Liu, A., Peng, C., and Chang., The Wavelet analysis of satellite images for coastal watch IEEE, 1997, vol. 22.
[3] A. Gustavo & L. Juan & J. B. Jovani, "Control of a ball-and-plate system using a State feedback controller" Ingeniare.28.6-15. 10.4067/S0718-33052020000100006, 2020.
[4] M. Yaseen, J. A. Haider, "Modeling and control for a magnetic levitation system based on SIMLAB platform in real time", Volume 8, Pages 153-159, ISSN 2211-3797,2018.
[5] Kar Anirban. OpenCV object tracking by colour detection in python.
[6] Adrian Rosebrock,.Ball tracking with OpenCV ,https://pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/
[7] Araki, M. "PID Control". http://www.eolss.net/ebooks/Sample%20Chapters/C18/E6-43-03-03
[8] Mohsin, M., T., "Development of a Ball and Plate System" ,122nd ASEE Anual Conference & Exposition.Washignton.
[9] OpenCV tutorial on moments. "opencv data sheet," OpenCV tutorial on moments.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY