DOI: 10.51984/SUCP.V4I1.3925

المؤتمر العلمي الأول للتطبيقات الهندسية (ICEA'2024)

The First Scientific Conference on Engineering Applications (ICEA'2024)

Conference homepage: www.icea.ly

Comprehensive Case Study: A Novel Design, Modeling, Simulation, and Experimental Implementation and Evaluation of Control Strategies for a 9-DOF Ball on Plate System

*Ahmed A. Alshareef, Othman E. Aburas, Ausama H. Ahmed

Electrical and Computer Department, College of Engineering, Elmergib University, Al-Khums, Libya

Keywords:

Ball on Plate system
Ball and Beam system
9 Degrees of Freedom
nonlinear mechanical setup
PI controller
PD controller
PID controller
Simulink
Arduino IDE
vision measurement
image processing
embedded system
control algorithms
DIP
DSP
robustness
adptive

ABSTRACT

This comprehensive case study examines the Ball on Plate (BnP) system, a critical component of our prototype involving multiple systems. The BnP system features a complex 9 Degrees of Freedom (DOF) nonlinear mechanical setup, comprising three independent Ball and Beam (BnB) systems that collectively enhance functionality. The primary objective was to model, simulate, and control this system using various control algorithms, including PI, PD, and PID controllers. The methodology involved designing and theoretically modeling each aspect of the BnP system, followed by extensive simulations and practical experiments for performance validation. The study aims to achieve several goals: analyzing hardware components and their mechanisms; modeling the three BnB systems with nonlinear mathematical models and linearizing them; implementing a control loop using Simulink and Arduino IDE; analyzing software components, including vision measurement and image processing; designing controllers with trial-and-error tuning for precise ball positioning; simulating the theoretical system with applied PID; and investigating practical results compared to theoretical outcomes. The embedded system actuates three servos based on webcam feedback with DSP control. Initial PID implementation resulted in high overshooting, which was mitigated by adjusting error gain. The study also explored alternative controllers (PI1, PI2, PD), revealing the PD controller's effective balance of stability and responsiveness. This research validates the efficacy of PID controllers in achieving stable system responses, emphasizing the influence of nonlinear and coupling factors on stability and performance. The complexities of controlling a multi-DOF nonlinear system underscore its significance in automatic control labs, aerospace engineering, and beyond.

دراسة حالة شاملة: تصميم و نمذجة و محاكاة و تنفيذ تجريبي و تقييم إستراتيجيات التحكم لنظام الكرة على اللوح ذو 9 درجات حرية

*أحمد الشريف و عثمان أبوراس و أسامة أحمد

قسم الكهربائية والحاسوب، كلية الهندسة، جامعة المرقب، الخمس، ليبيا

الكلمات المفتاحية:

نظام الكرة على اللوح
نظام الكرة والشعاع
9 درجات من الحرية
إعداد ميكانيكي غير خطي
وحدة تحكم PI
وحدة تحكم PD
وحدة تحكم PID
Simulink
بيئة تطوير Arduino IDE
قياس الرؤية
معالجة الصور
النظام المدمج

الملخص

تستعرض هذه الدراسة الشاملة نظام الكرة على اللوح (BnP) ، وهو مكون حيوي في نموذجنا الأولي الذي يتضمن أنظمة متعددة. يتسم نظام BnP بإعداد ميكانيكي غير خطي مع 9 درجات من الحرية(DOF) ، ويتألف من ثلاثة أنظمة مستقلة من الكرة والشعاع (BnB) التي تعزز الوظائف بشكل جماعي. كان الهدف الرئيسي هو نمذجة هذا النظام ومحاكاته والتحكم فيه باستخدام خوارزميات تحكم متنوعة، بما في ذلك وحدات تحكم PI و PD و .PID تضمنت المنهجية تصميم ونمذجة كل جانب من جوانب نظام BnP من الناحية النظرية، تلاها محاكاة شاملة وتجارب عملية للتحقق من الأداء. تهدف الدراسة إلى تحقيق عدة أهداف: تحليل مكونات الأجهزة وآليات عملها؛ نمذجة الأنظمة الثلاثة BnBباستخدام نماذج رياضية غير خطية ودمجها مع خطية الأنظمة؛ تنفيذ حلقة تحكم باستخدام برنامج Simulinkوبيئة تطويرArduino IDE ؛ تحليل مكونات البرمجيات، بما في ذلك قياس الرؤية ومعالجة الصور؛ تصميم وحدات تحكم مع ضبط تجريبي لتحقيق دقة في تحديد موقع الكرة؛ محاكاة النظام النظري باستخدام وحدة PID؛ ودراسة النتائج العملية مقارنةً بالنتائج النظرية .يعمل النظام المدمج على تحريك ثلاثة محركات سيرفو استنادًا إلى تغذية راجعة من كاميرا الويب مع التحكم .DSP أدى تطبيق PID الأولي إلى حدوث تجاوز كبير

*Corresponding author:
E-mail addresses: ahmed.abdul.hakim.al.shareef@gmail.com, (O. E. Aburas) oeaburas@elmergib.edu.ly,
(A. H. Ahmed) ausama.ahmed@gmail.com

Article History : Received 24 March 2024 - Received in revised form 30 September 2024 - Accepted 15 October 2024

(overshooting)تم معالجته من خلال تعديل مكاسب الخطأ. كما استكشفت الدراسة وحدات تحكم بديلة مثل PI1 وPI2 وPD، مما أظهر فعالية وحدة التحكم PD في تحقيق توازن بين الاستقرار والاستجابة. تؤكد هذه البحث فعالية وحدات تحكم PID في تحقيق استجابة مستقرة للنظام، مع التركيز على تأثير العوامل غير الخطية وعوامل الارتباط على الاستقرار والأداء. كما تسلط الدراسة الضوء على تعقيدات التحكم في الأنظمة غير الخطية متعددة درجات الحرية، مما يبرز أهمية هذا النوع من الأنظمة في مختبرات التحكم الآلي، والهندسة الفضائية، وما بعدها.

خوارزمية التحكم DIP
خوارزمية التحكم DSP
القوة
التكيف

1. Introduction

In robotic applications, vision tasks like object recognition and path planning utilize digital image processing (DIP), a component of digital signal processing (DSP) [8]. DSPs efficiently handle digitized signals, crucial for systems in security and home theaters [9]. Digital controllers are increasingly vital for optimal performance [2]. Embedded systems, typically microcontroller-based, monitor sensors to control devices [7]. Balancing systems, such as the traditional cart-pole and ball and beam (BnB), serve as experimental platforms for control engineers [17]. The ball and plate (BnP) system, an advanced BnB variant, balances a ball on a plate using sensors and servo motors, embodying a complex nonlinear system with multiple Degrees of Freedom (DOF) [1][18].

This study addresses challenges in 9-DOF nonlinear systems by implementing various control algorithms, including PID, to ensure precision and stability. It encompasses modeling, simulation, and practical implementation of control strategies, focusing on tuning PID parameters for enhanced stability. Simulations demonstrate significant performance improvements, with practical experiments underscoring the importance of fine-tuning. Additionally, the study evaluates PI1, PI2, and PD controllers, highlighting their strengths and weaknesses. The findings emphasize the adaptability and robustness of the BnP system, with implications for various applications. The paper is organized as follows: Section 2 reviews previous studies, Section 3 details the BnP system's analysis and control, Section 4 presents simulation and experimental results, and Section 5 discusses conclusions and future research directions.

2. Literature Review

In [1][16][18][3][4][5][6][11][13][14][15], mathematical models describe dynamic systems, while [17][10][12] do not. Systems typically exhibit 2, 4, or 6 degrees of freedom (DOF), with 2 DOF related to plate motion, 4 DOF including ball motion, and 6 DOF representing freely suspended bodies. Each approach informs controllers for decoupled BnB systems. For instance, [1] developed a PID controller using the Zeigler-Nichols method on an ATMega 2560 to actuate servomotors, with feedback via a touchscreen. Their simulation utilized Matlab/Simulink. [16] created a trajectory tracking controller on a DSP board, employing optical encoders and image processing with an FPGA for real-time feedback. [17] explored a reinforcement learning fuzzy controller interfacing with RS-232 for stepper motors, using encoders for feedback and a touchscreen for position sensing, with simulation results derived from extensive experiments. [18] designed a backstepping controller on a C8051F020 microcontroller, sending PWM signals to magnetic suspension actuators, with feedback via potentiometers. The LQR (Linear Quadratic Regulator) controller was implemented in [3] and [4] for trajectory tracking, utilizing digital cameras for position feedback. [5] combined a PID controller with lead compensation in a dSpace environment, while [6] compared multiple controllers using a Texas Instruments MCU. Fuzzy logic controllers were developed in [10] to facilitate real-time feedback, employing machine vision algorithms. Various PID enhancements, such as the IASA-PID and NN-PID in [11], showcased different control strategies. [12] and [13] implemented controllers using Arduino and LabVIEW, respectively, focusing on real-time feedback. Finally, [14] and [15] compared PID with Sliding-Mode Control methods to enhance performance, utilizing microcontrollers for PWM signal generation and position tracking through various feedback mechanisms.

3. Analysis and Modeling the System

3.1 Hardware of the system:

3.1.1 The hardware components:

Ball and socket joint 3 elements (manufactured product); Rod 3 elements (manufactured product); Servomotor 3 elements (manufactured product); Ball (manufactured product); Plate; Ball bearing 3 elements (manufactured product); Upper base; Lower base; One axes joint 3 elements; Servo arm 3 elements; Cable lug 3 elements (manufactured product); Webcam (manufactured product); PC (manufactured product); Power supply 5V (manufactured product); Arduino mega 2560 (manufactured product).

3.1.2 Collecting hardware project components:

The system begins with a plate designed to measure 30cm by 30cm and 0.5cm thick in white (Fig. 1). This plate connects to an upper base shaped like a clipped triangle, centered relative to the plate (Fig. 2). The upper base attaches to a ball and socket joint located on its outer short side (Fig. 3, Fig. 7).

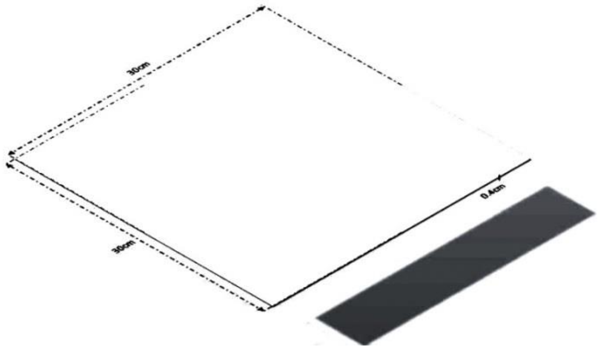

Fig. 1: Plate's dimension

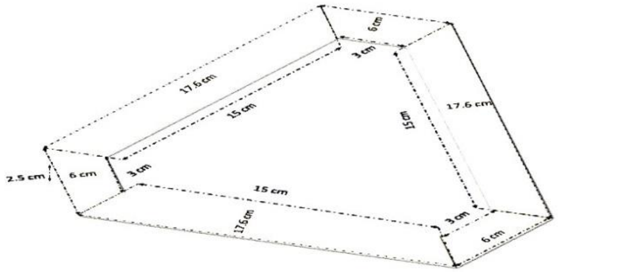

Fig. 2: Upper base's dimension

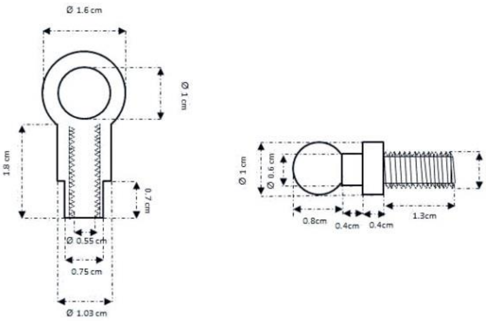

Fig. 3: Ball and socket dimensions

A one-axis fixed joint limits the movement of the ball and socket joint to one axis (Fig. 6). A rod, measuring 21.6 cm, connects via a cable lug to an outer ball bearing circle, enhancing the plate's inclination (Fig. 5). Ball bearings are included to ensure smooth movement. An inner ball bearing circle links to a servo arm (Fig. 4), forming an L-shaped linkage that tilts the plate when driven by the servomotor, allowing the ball to roll on its surface [16].

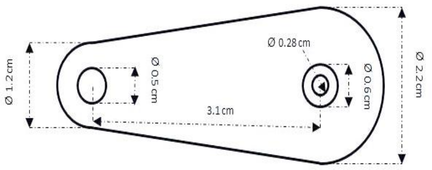

Fig. 4: Servo arm dimension

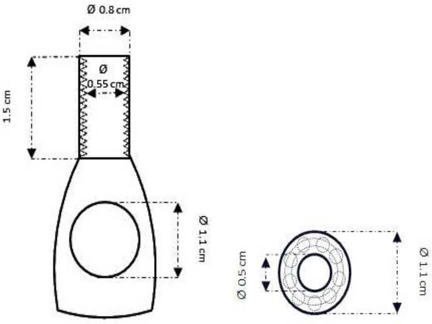

Fig. 5: a) Cable lug dimension ,b) Ball bearing dimension

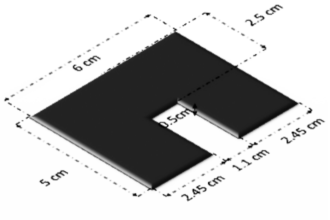

Fig. 6: One axis joint dimension

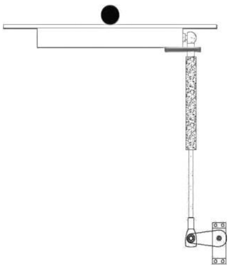

Fig. 7: L-shaped linkage of BnP system actuated with servo motor of one BnB side

The lower base is constructed from three wooden pieces arranged in a "Y" shape with 120° angles (Fig. 8). Each servo motor is fixed to this base (Fig. 9), and three L-shaped linkages connect to the servomotors to maintain the plate's balance.

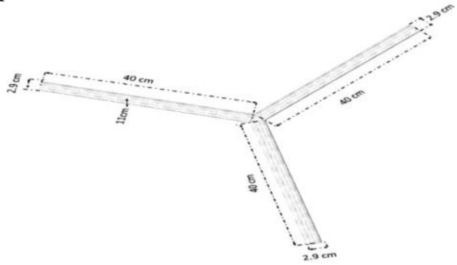

Fig. 8: Lower base's dimension

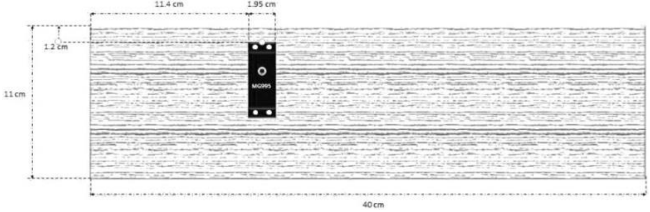

Fig. 9: Positioning servo motor

The control loop consists of three main components: a PC, which manages set points and processes images; an Arduino, which adjusts servomotor angles based on PC data; and a webcam, serving as a ball position sensor that captures real-time images (Fig. 10) [12].

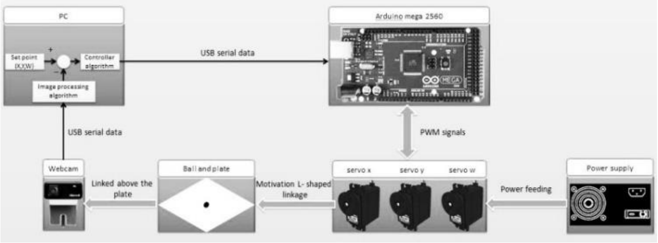

Fig. 10: The overall system control loop

3.1.3 Hardware mechanism analysis:

The BnP system has 9 DOF across three work lines, representing three individual BnB systems (Figs. 11, 12). Each BnB system features three translational and six rotational DOFs. The work lines connect servomotors with L-shaped linkages, forming angles of 120°. Each work line, representing BnB systems A, B, and C, operates to maintain the plate's balance.

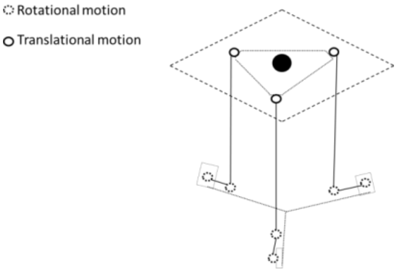

Fig. 11: Degrees of freedom of BnP System

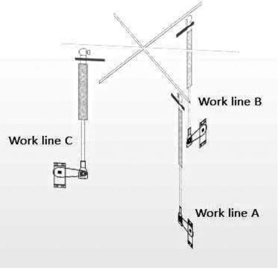

Fig. 12: Work lines meeting its BnB's

The relationship between changes in servo hardware angles and plate angles is analyzed, noting that these angles increase proportionally. The maximum change in servo hardware angle is referenced at $\alpha_H = 45°$,yielding a maximum height d calculated as follows:

$$d = r \sin \alpha_H \tag{1}$$

$$d = 3.65 \sin 45 \ \approx 2.58\, cm$$

The change in servo motor hardware angle α_H was calibrated and compared to the change in servo motor software angle α_S . The differences were documented, as shown in Fig. 13 and tabulated in Table (1).

Table 1: Measure the change of servo motor software angle to meet the change of servo motor hardware angle respect to 0º angle

Software angle ($\boldsymbol{\alpha_S}$)	0	20	45	70	90	110	135	155	180
Hardware angle ($\boldsymbol{\alpha_H}$)	0	25	45	65	85	100	126	145	171

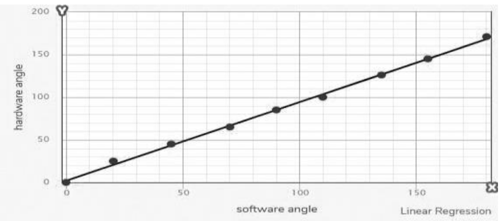

Fig. 13: The relation between servo motor hardware and software angle

The relationship between servo motor hardware and software angles can be approximated by the function:

$$\begin{aligned} &hardware\ angle\ (\alpha_H) \\ &= 0.922482 * software\ angle\ (\alpha_S) \\ &+\ 2.155772 \end{aligned} \tag{2}$$

To refine this relation, the slope between each pair of points was calculated and the mean value was determined, as summarized in Table (2).

Table 2: Slope between each two point of the change of servo motor hardware angle and software angle

M(i)	M1	M2	M3	M4	M5	M6	M7	M8
Slope	1.25	0.8	0.8	1	0.75	1.04	0.95	1.04

The mean slope is calculated as follows:

$$\mu = \frac{\sum_{i=1}^{N} M}{N} = 0.95375 \qquad (3)$$

Thus, the refined approximation function is:

$$\alpha_H \ = \ 0.95375 * \alpha_S \qquad (4)$$

Work line A has dimension as shown in Fig.14 with axis length La=33.2 cm and had β value with law of sine as follow:

$$\sin(\beta) = \frac{d}{la} \qquad (5)$$

$$sin(\beta) = \frac{2.58}{12.3}$$

$$\beta_A \approx 12.11°$$

The relation between plate angle and change of servo software angle in work line A is:

$$\beta_A(radian) = \frac{\beta_A * 0.95375 * \pi}{\alpha * 180°}\alpha_{SA} \qquad (6)$$

$$\approx 0.0045 * \alpha_{SA}(degree)$$

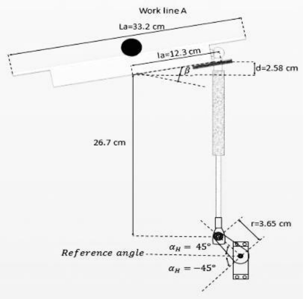

Fig. 14: Work line A dimension

Work line B has dimension as shown in Fig.15 with axis length Lb=33.2 cm which is same length of A and calculated β value as follow:

$$\sin(\beta) = \frac{d}{lb} \qquad (7)$$

$$sin(\beta) = \frac{2.58}{12}$$

$$\beta_B \approx 12.42°$$

The relation between plate angle and change of servo software angle in work line B is:

$$\beta_B(radian) = \frac{\beta_B * 0.95375 * \pi}{\alpha * 180°}\alpha_{SB} \qquad (8)$$

$$\approx 0.0046 * \alpha_{SB}(degree)$$

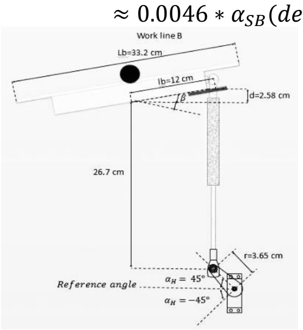

Fig. 15: Work line B dimension

Work line C has dimension as shown in Fig.16 with axis length Lc=30 cm and calculated β value as follow:

$$\sin(\beta) = \frac{d}{lc} \qquad (9)$$

$$sin(\beta) = \frac{2.58}{12.7}$$

$$\beta_C \approx 11.72°$$

The relation between plate angle and change of servo software angle in work line C is:

$$\beta_C(radian) = \frac{\beta_C * 0.95375 * \pi}{\alpha * 180°}\alpha_{SC} \qquad (10)$$

$$\approx 0.0043 * \alpha_{SC}(degree)$$

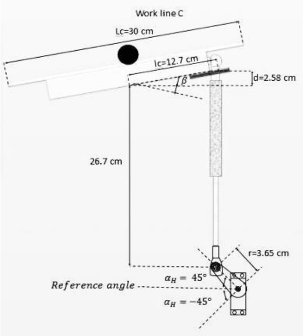

Fig. 16: Work line C dimension

3.2 Mathematical model:

The nonlinear mathematical model for the ball on the beam is represented as [14]:

$$\ddot{X}(t) = -\frac{m_{ball} g \sin(\beta)}{(m_{ball} + \frac{J_{ball}}{R_{ball}{}^2})} \qquad (11)$$

Where the parameter of mathematical model as shown in Table (3) and the moment of inertia of a solid ball is $J_{ball} = \frac{2}{5} m_{ball} R_{ball}$.

Where $X(t)$ is the position of ball.

Table 3: The parameters' values of the mathematical model and its description

Symbol	g	m_{ball}	J_{ball}	R_{ball}
Value / Unit	-9.81 m/s2	0.0027 kg	4.341*10-7 m2kg	0.02005 m
Description	Acceleration due to gravity	Mass of the ball	Moment of inertia of the ball	Radius of the ball

State-space form of the nonlinear mathematical model is:

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix} x_2 \\ 7\sin(x_1)\end{bmatrix} \qquad (12)$$

By linearized the nonlinear mathematical model with Jacobin method by assume:

$$f(x_1,x_2) = \dot{x_1} = x_2, g(x_1,x_2) = \dot{x_2} = 7sin(x_1)$$

$$J|_{(x_1,x_2)} = \begin{pmatrix} \frac{\partial f(x_1,x_2)}{\partial x_1} & \frac{\partial f(x_1,x_2)}{\partial x_2} \\ \frac{\partial g(x_1,x_2)}{\partial x_1} & \frac{\partial g(x_1,x_2)}{\partial x_2} \end{pmatrix} \qquad (13)$$

$$\begin{pmatrix} 0 & 1 \\ 7cos(x_1) & 0 \end{pmatrix}$$

With calculated the equilibrium point to get the Eigen values as:

$$f(x_1,x_2) = \dot{x_1} = 0, g(x_1,x_2) = \dot{x_2} = 0$$

$$equilibrium\ point\ of (x_1,x_2)\ at\ (0,0),(\pi,0)$$

Then

$$J|_{(0,0)} = \begin{pmatrix} 0 & 1 \\ 7 & 0 \end{pmatrix}, Eigenvalue = 2.6458, -2.6458\ is\ saddlepoint$$

$$J|_{(\pi,0)} = \begin{pmatrix} 0 & 1 \\ -7 & 0 \end{pmatrix}, Eigenvalue$$

$$= 2.6458j, -2.6458j\ is\ centerpoint$$

The linearized mathematical model of each work line are:

$$\ddot{X}_A = 7\beta_A(radian) \qquad (14)$$

$$\ddot{X}_B = 7\beta_B(radian) \qquad (15)$$

$$\ddot{X}_C = 7\beta_C(radian) \qquad (16)$$

By substituting equation (6) in (14), (8) in (15) and (10) in (16) as follow:

$$\ddot{X}_A = \ 0.0315 * \alpha_{SA}(degree) \qquad (17)$$

$$\ddot{X}_B = 0.0322 \ * \alpha_{SB}(degree) \qquad (18)$$

$$\ddot{X}_C = 0.0301 \ * \alpha_{SC}(degree) \qquad (19)$$

Laplace transform of each work line model is calculated as:

$$\frac{X_A(s)}{\alpha_A(s)} = \frac{0.0315}{s^2} \qquad (20)$$

$$\frac{X_B(s)}{\alpha_B(s)} = \frac{0.0322}{s^2} \qquad (21)$$

$$\frac{X_C(s)}{\alpha_C(s)} = \frac{0.0301}{s^2} \tag{22}$$

The state space of each work line transfer function are:

$$\begin{bmatrix}\dot{X}_{A1}\\ \dot{X}_{A2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}X_{A1}\\ X_{A2}\end{bmatrix} + \begin{bmatrix}0\\ 0.0315\end{bmatrix}\alpha_{SA}(degree) \tag{23}$$

$$Y_A = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}C_{A1}\\ C_{A2}\end{bmatrix} + [0]\,D_A \tag{24}$$

$$\begin{bmatrix}\dot{X}_{B1}\\ \dot{X}_{B2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}X_{B1}\\ X_{B1}\end{bmatrix} + \begin{bmatrix}0\\ 0.0315\end{bmatrix}\alpha_{SB}(degree) \tag{25}$$

$$Y_B = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}C_{B1}\\ C_{B2}\end{bmatrix} + [0]\,D_B \tag{26}$$

$$\begin{bmatrix}\dot{X}_{C1}\\ \dot{X}_{C2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}X_{C1}\\ X_{C2}\end{bmatrix} + \begin{bmatrix}0\\ 0.0315\end{bmatrix}\alpha_{SC}(degree) \tag{27}$$

$$Y_C = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}C_{C1}\\ C_{C2}\end{bmatrix} + [0]\,D_C \tag{28}$$

The servo motors are calibrated to specific angles to maintain a balanced plate, with defined maximum and minimum angles for effective adjustments. A relationship is established between pixel error (ranging from -30 to 30 pixels) and corresponding servo motor angle adjustments (ranging from -47 to 47 degrees):

$$\alpha_S(degree) = \frac{47°}{30\,(pix)} * e(t)(pix) \tag{29}$$

This ensures that changes in servo angles are proportional to positioning error, minimizing discrepancies efficiently (Fig. 17).

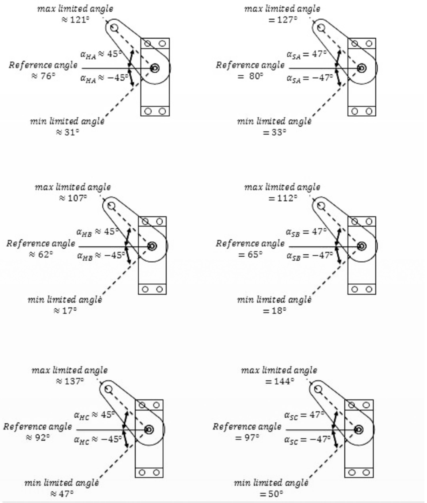

Fig. 17: Difference between each work lines servo motor hardware angel meeting software angle respect to reference

3.3 Software of the system

3.3.1 Software project components:

1. MATLAB R2017a program.
2. MATLAB/Simulink/Image Acquisition Toolbox.
3. Arduino IDE 1.8.12 program.

3.3.2 Collecting software project components:

MATLAB/Simulink facilitates testing and implementing the control loop. The Image Acquisition Toolbox captures 2D images from a USB webcam, enabling adjustments in capture rate and resolution. User-Defined Functions map position and servo angles, while image processing determines the ball's position on the plate. The Instrument Control Toolbox manages data exchange with the Arduino, which actuates the servo motors based on MATLAB signals, providing feedback on current angles.

3.3.3 Software mechanism analysis:

3.3.3.1 Vision measurement and Image processing algorithm:

A machine vision system calculates the ball's displacement across three axes. Using a black ball on a white plate enhances measurement accuracy [16]. The USB webcam captures images at 160x120 pixels in RGB color space for efficient manipulation [10]. The image manipulation steps include:

Thresholding: Converts grayscale images to binary to differentiate the object from the background [16].

Filtering: Reduces noise in binary images.

Morphological: Extracts shape descriptors based on image properties [19].

Clipping: Trims rows and columns beyond the object's edges.

Object Analysis & recognition: Calculates parameters such as area and center, determining the ball's Z-axis dimension from its area in pixels.

Data was collected (Fig. 18) and tabulated in Table (4) to formulate a function relating Z dimension to ball area, represented in equation (30):

Table 4: Measure the Z dimension length above the ball to meet the area of ball in image

z (cm)	Area (pixel)
5.56	510
5.88	480
6.56	430
7.39	370
8.63	320
10.285	265
12.605	224
12.69	217
16.935	171
19.765	143
26.84	106
30.455	94
34.78	82
42.06	66
49.415	54
53.32	49

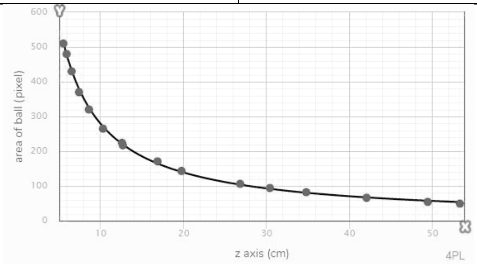

Fig. 18: The relation between z axis length and area of ball

$$Z\,axis = 4.95986 + \frac{1100337000 - 4.95986}{1 + \left(Area\ of\ \frac{ball}{0.000004037671}\right)^{1.03297}} \tag{30}$$

The vision measurement and image process algorithm explained by flow chart as shown in Fig.19.

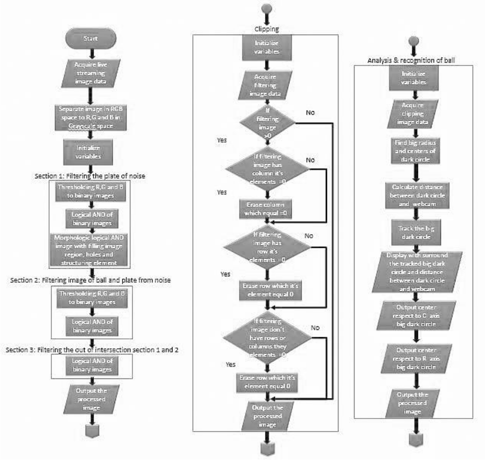
Start
Acquire live streaming image data
Separate image in RGB space to R,G and B in Grayscale space
Initialize variables
Section 1: Filtering the plate of noise
Thresholding R,G and B to binary images
Logical AND of binary images
Morphologic logical AND image with filling image region, holes and structuring element
Section 2: Filtering image of ball and plate from noise
Thresholding R,G and B to binary images
Logical AND of binary images
Section 3: Filtering the out of intersection section 1 and 2
Logical AND of binary images
Output the processed image
Clipping
Initialize variables
Acquire filtering image data
If filtering image >0
No
Yes
If filtering image has column it's elements =0
No
Yes
Erase column which equal =0
If filtering image has row it's elements =0
No
Yes
Erase row which it's element equal 0
If filtering image don't have rows or columns they elements =0
No
Yes
Erase row which it's element equal 0
Output the processed image
Analysis & recognition of ball
Initialize variables
Acquire clipping image data
Find big radius and centers of dark circle
Calculate distance between dark circle and webcam
Track the big dark circle
Display with surround the tracked big dark circle and distance between dark circle and webcam
Output center respect to C axis big dark circle
Output center respect to R axis big dark circle
Output the processed image

Fig. 19: The vision measurement and image process algorithm flow chart

3.3.3.2 Image dividing algorithm according to work lines:

The image dividing algorithm processes images of a ball on a plate by segmenting them into three axes aligned with their respective work lines, aiding in the determination of the ball's position. Initially, the webcam is positioned on the plate and adjusted so that its edges align with the plate's edges. The image is then divided into three axes as illustrated in Fig.20.

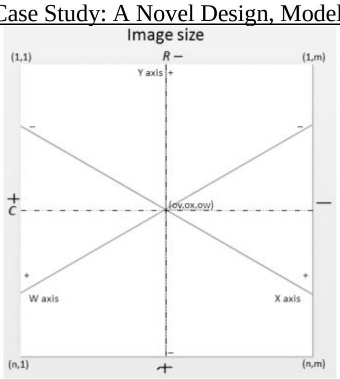

Fig. 20: Dividing an image to three axis respect to each work line

The sines triangle law (Lame’s law) method is used to find either axis position corresponding to ball position on image and related to origin point of the image and then for each axis $(,OC)=(OY,OX,OW)$ respectively.

Assumed position of the ball as shown in Fig.21 as position 1 and position 2 to give some cleared and summarized description of regions by analysis that image frame which has two axis with translated to the three axis framework as follow:

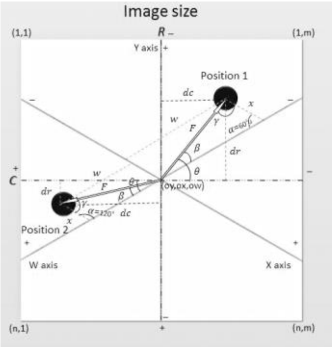

Fig. 21: Analysis the position 1&2 respect to each axis with using vector

1. Found the size of image.
2. Calculated the median of image for rows and columns to specify origin point $(,OC)$.
3. Calculated dr,dc dependent on the position of ball on its coordinate with rows and columns $(rcoord,ccoord)$ in image frame $(n,m)$with subtracted from origin point $(,OC)$.
4. Calculated the axis (Y) from the position of the ball on two axis (R,C) which is minus of dr.
5. Calculated with ball position on two axis (R,C) the inclines angle θ of tendon F which is dependent on dr,dc .
6. Calculated the angle of triangle which are β,α and γ with determination the region which is the ball located in as shown by grayscale gradients of the region in Fig.22. dependent on θ’s value as following:

$$if|\theta|<30then$$
$$\alpha=120°$$
$$\beta=\big||\theta|-30\big|$$
$$\gamma=180-\beta-\alpha$$

$$if|\theta|>30then$$
$$\alpha=60°$$
$$\beta=\big||\theta|-30\big|$$
$$\gamma=180-\beta-\alpha$$

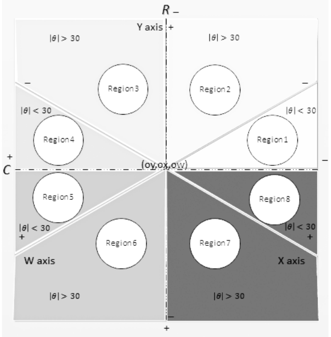

Fig. 22: Explain the Θ's value in each region in the plane

7. Calculated the axis (X,W) from position of the ball on two axis (R,C) with Lame’s law with known F’s value, they’re dependent on region shown before and θ’s values, for example, applied the method on postion1&position 2 shown in Fig.21 as follow:

$$Position1:$$
$$where|\theta|>30,\alpha=60°,\beta=\big||\theta|-30\big|,\gamma=180-\beta-\alpha$$
$$then\qquad \frac{F}{\sin\alpha}=\frac{-w}{\sin\gamma}=\frac{-x}{\sin\beta} \tag{31}$$
$$Position2:$$
$$where|\theta|<30,\alpha=120°,\beta=\big||\theta|-30\big|,\gamma=180-\beta-\alpha$$
$$then\qquad \frac{F}{\sin\alpha}=\frac{w}{\sin\gamma}=\frac{-x}{\sin\beta} \tag{32}$$

Image dividing algorithm explained according to work lines by flow chart as shown in Fig.23.

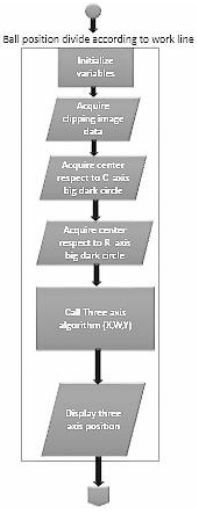
Ball position divide according to work line
Initialize variables
Acquire clipping image data
Acquire center respect to C axis big dark circle
Acquire center respect to R axis big dark circle
Call Three axis algorithm (X,W,Y)
Display three axis position

Fig. 23: Dividing ball position to the three axis according each work line algorithm flow chart

3.3.3.3 Flow control algorithm of sending serial data:

This algorithm used to control the flow of sending states of each servomotor and theirs set angles respectively through one serial port by passed into *to Instrument block* through uploaded Arduino algorithm in Arduino controller which actuating one servo motor and non-energize the others, that actuating and non-actuating states passes through each servomotor, the algorithm explained in flow chart as shown in Fig.24.

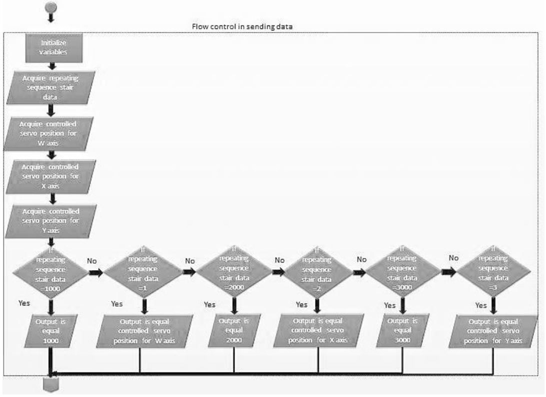
Flow control in sending data
Initialize variables
Acquire repeating sequence stair data
Acquire controlled servo position for W axis
Acquire controlled servo position for X axis
Acquire controlled servo position for Y axis
If repeating sequence stair data =1000
No
If repeating sequence stair data =1
No
If repeating sequence stair data =2000
No
If repeating sequence stair data =2
No
If repeating sequence stair data =3000
No
If repeating sequence stair data =3
Yes
Output is equal 1000
Yes
Output is equal controlled servo position for W axis
Yes
Output is equal 2000
Yes
Output is equal controlled servo position for X axis
Yes
Output is equal 3000
Yes
Output is equal controlled servo position for Y axis

Fig. 24: Flow control sending signal data algorithm flow chart

3.3.3.4 Servomotor actuating algorithm:

This algorithm used to actuate and set servomotor angle to the desired position according to each servomotor states & angles which got them from data on serial from *to Instrument block* and feedbacked data through *Query Instrument block* to check if there's a connection between Matlab/Simulink and the Arduino with properties as in Table (5), the algorithm uploaded to Arduino controller which is used to drive the servomotors and it’s explained in flow chart as shown in Fig.25.

Table 5: Properties of sending and receiving data block

Name	Timeout (s)	Buffer Size	Interface	Port	Bound rate	Block Sample Time
To Instrument	10	512	Serial	COM3	9600	Inherited
Query Instrument	10	512	Serial	COM3	9600	1

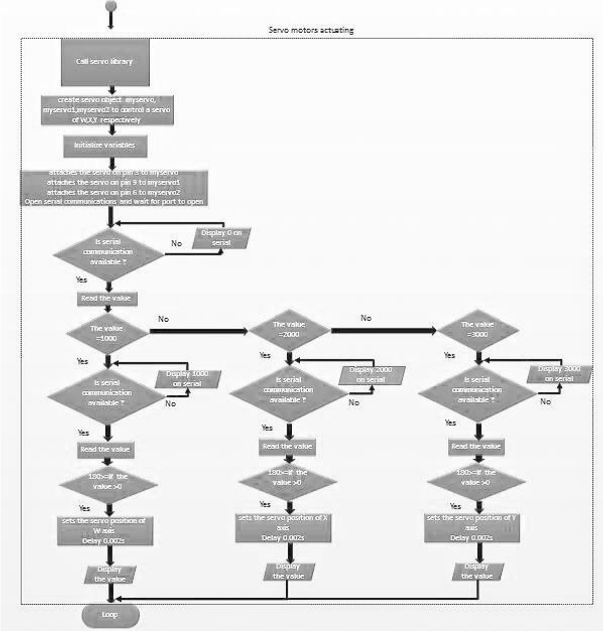
Servo motors actuating
Call servo library
create servo object myservo, myservo1,myservo2 to control a servo of W,X,Y respectively
Initialize variables
attaches the servo on pin 3 to myservo
attaches the servo on pin 9 to myservo1
attaches the servo on pin 6 to myservo2
Open serial communications and wait for port to open
Is serial communication available ?
No
Display 0 on serial
Yes
Read the value
The value =1000
No
The value =2000
No
The value =3000
Yes
Is serial communication available ?
Display 1000 on serial
Display 2000 on serial
Display 3000 on serial
No
Read the value
180>=if the value >0
sets the servo position of W axis Delay 0.002s
sets the servo position of X axis Delay 0.002s
sets the servo position of Y axis Delay 0.002s
Display the value
Loop

Fig. 25: Servo motors actuating according each work line algorithm flow chart

4. Simulation and Experimental Results

This section discusses the simulation and experimental outcomes of a PID controller designed via a trial-and-error method for a ball on plate (BnP) system.

4.1 Simulation results of ball on plate system

Each work line of the BnP system was modeled without a controller (Fig. 26).

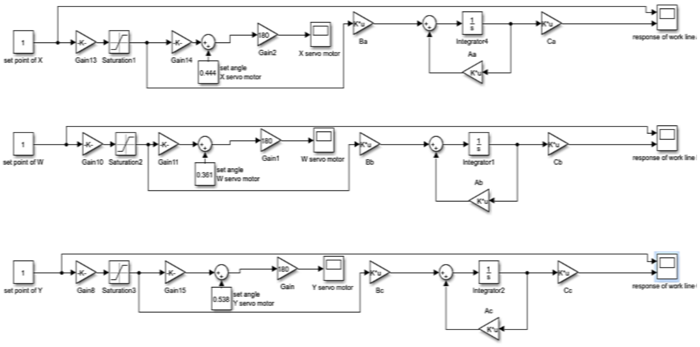

Fig. 26: Block diagram of theoretical system for each work line without controller

The theoretical system's response was simulated to evaluate each work line against its set point (Fig. 27).

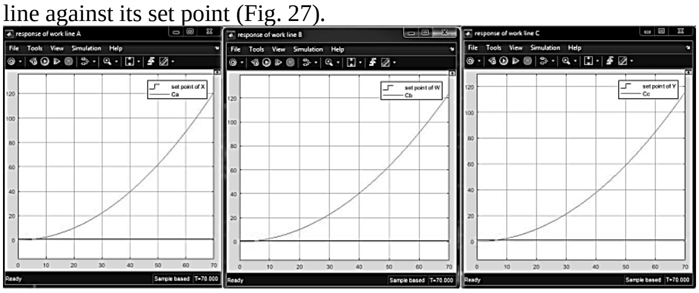

Fig. 27: Response of block diagram for each work line without controller

A PID controller was developed for each work line to minimize error. The application of this controller improved system stability significantly (Fig. 28), as detailed in Table 6, which lists the PID parameters.

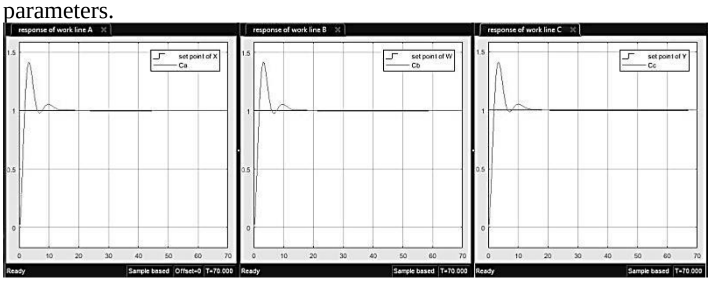

Fig. 28: Response of block diagram for each work line with PID controller

The PID controller also stabilized the servo motors, enhancing performance across all work lines (Fig. 29) in Table (6):

Table 6: PID controller parameters values of each work line of theoretical system

	Proportional (P)	Integral (I)	Derivative (D)	Filter coefficient (N)
W	4.09	0.19	20.27	1
Y	4.09	0.19	20.27	1
X	4.09	0.19	20.27	1

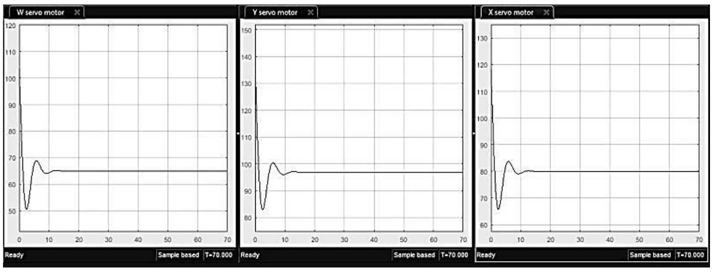

Fig. 29: Response of servo motors for each work line with PID controller

4.2 Experimental result:

Applied a controller on whole system of our prototype with practical way to trying to adapt with our prototype system effectiveness on the BnP system.

4.2.1 Experimental result of ball on plate system:

In practical applications, the PID controller initially caused significant overshoot (Fig. 30). Adjustments were made by reducing the error gain to help the servomotor track the controlled signal effectively.

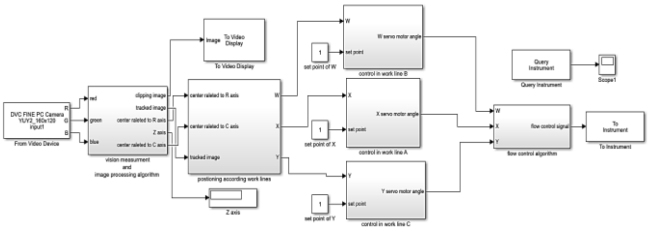

Fig. 30: Block diagram of practical system with implemented controller

The optimal sample time was determined to be 0.002 seconds, with updated PID parameters listed in Table (7).

Table 7: PID controller parameters and Kerror1 values of each work line for practical system

	Proportional (P)	Integral (I)	Derivative (D)	Filter coefficient (N)	Kerror 1
W	4.09	0.19	20.27	1	0.0045
Y	4.09	0.19	20.27	1	0.0095
X	4.09	0.19	20.27	1	0.0025

The response of the system is shown in Fig. 31, which compares the performance of each work line. Notably, work lines B and C exhibited larger steady-state errors compared to line A. The transient domain specifications (e.g., t_p, t_s, M_p) are similar across lines, with some oscillations observed. of each work line it's seems same also there's some oscillation.

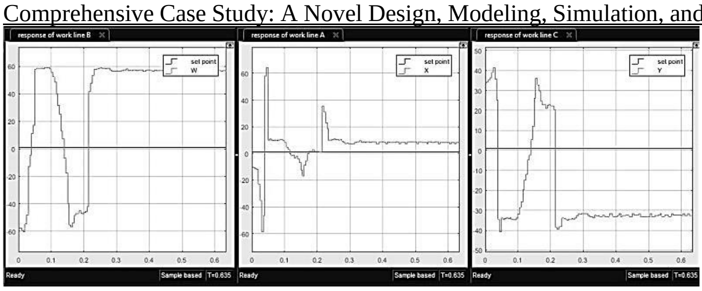

Fig. 31: Response of practical system with implemented PID controller and $K_{error}1$

The responses of the servo motors for each work line are depicted in Fig. 32.

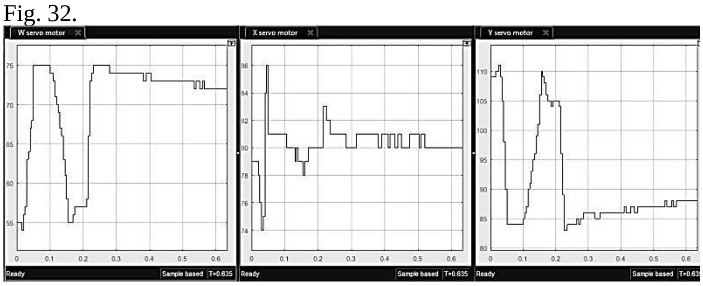

Fig. 32: Response of practical system servo motors with implemented PID controller and $K_{error}1$

The PI1 controller was also implemented, with error reducer gain $K_{error}2$ values listed in Table (8).

Table 8: PI1 controller parameters and Kerror2 values of each work line for practical system

	Proportional (P)	Integral (I1)	Derivative (D)	Filter coefficient (N)	Kerror 2
W	4.09	0.19	0	0	0.0045
Y	4.09	0.19	0	0	0.0086
X	4.09	0.19	0	0	0.0036

The response with the PI1 controller is shown in Fig. 33. While the steady-state error in work line B is still higher than in A and C, the overall error is smaller compared to the previous PID implementation. The transient characteristics are similar across lines, with reduced oscillations.

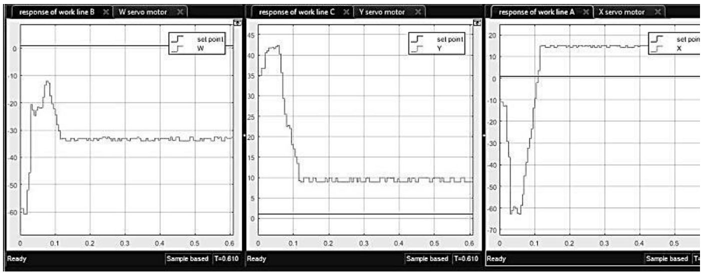

Fig. 33: Response of practical system with implemented PI1 controller and $K_{error}2$

Responses of the servo motors are illustrated in Fig. 34.

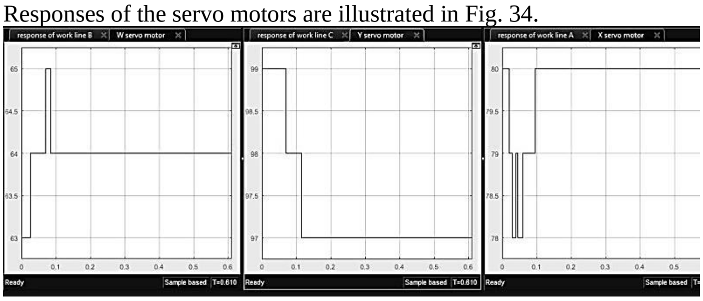

Fig. 34: Response of practical system servo motors with implemented PI1 controller and $K_{error}2$

Next, the PI2 controller was implemented, with $K_{error}3$ values in Table (9).

Table 9: PI2 controller parameters and Kerror3 values of each work line for practical system

	Proportional (P)	Integral (I2)	Derivative (D)	Filter coefficient (N)	Kerror 3
W	4.09	6.19	0	0	0.0046
Y	4.09	5.19	0	0	0.009
X	4.09	4.19	0	0	0.0032

The response of the system with the PI2 controller is shown in Fig. 35. Although work line B still shows a significant steady-state error, it is smaller than in previous implementations. Transient characteristics remain consistent, with minor oscillations.

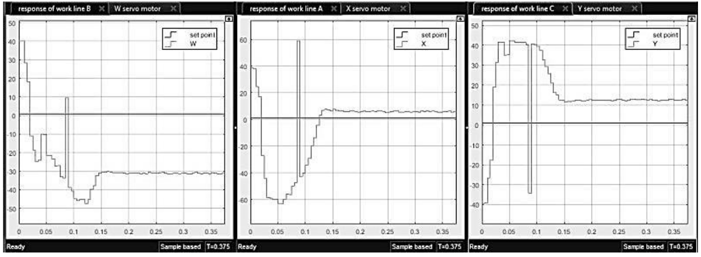

Fig. 35: Response of practical system with implemented PI2 controller and $K_{error}3$

The servo motor responses are depicted in Fig. 36.

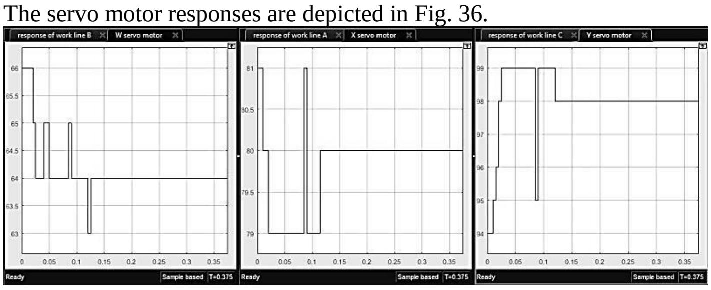

Fig. 36: Response of practical system servo motors with implemented PI2 controller and $K_{error}3$

Finally, a PD controller was implemented, with Kerror4 values in Table (10).

Table 10: PD controller parameters and Kerror4 values of each work line for practical system

	Proportional (P)	Integral (I)	Derivative (D)	Filter coefficient (N)	$K_{error}4$
W	4.09	0	20.27	1	0.0045
Y	4.09	0	20.27	1	0.0092
X	4.09	0	20.27	1	0.003

The system response with the PD controller is shown in Fig. 37. Work line B continues to show a higher steady-state error, and oscillations are more pronounced compared to previous responses.

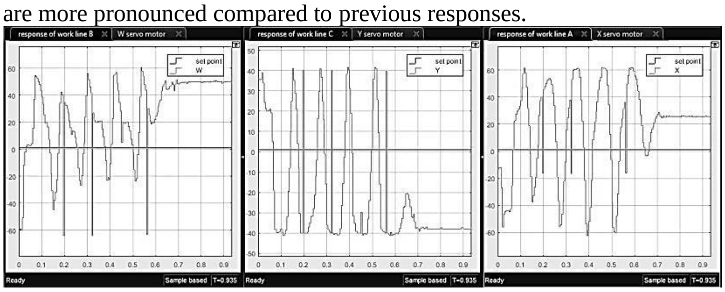

Fig. 37: Response of practical system with implemented PD controller and $K_{error}4$

Responses of the servo motors with the PD controller are illustrated in Fig. 38.

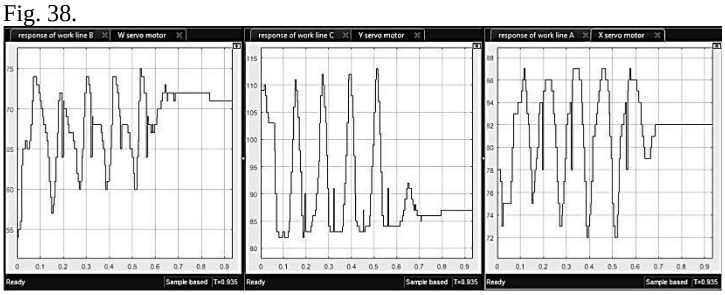

Fig. 38: Response of practical system servo motors with implemented PD controller and $K_{error}4$

4.2.2 Experimental result of applied hand effect on ball on plate system:

PID controller implemented to test the practical system with effect some states such move the ball from stable state with white rod by hand. The other effects that has time simulation start from 2.3 to 6.155 been undisclosed for future development applications. The performance of practical system under these conditions with each work line which shown in Fig.39.

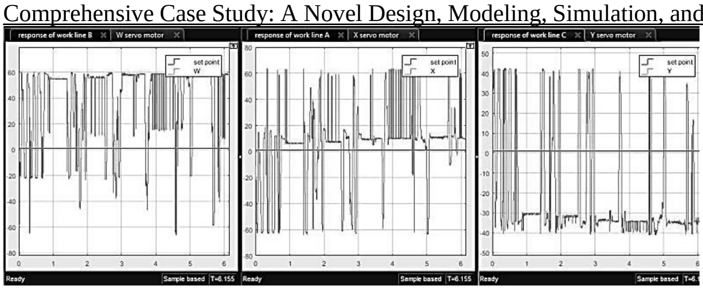

Fig. 39: Response of practical system by effect the states with implemented PID controller and K_{error}1

The responses of the servo motors adapting to these disturbances are depicted in Fig. 40.

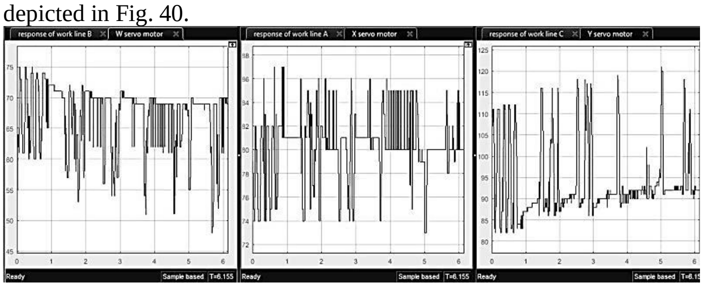

Fig. 40: Response of practical system servo motors by effect the states with implemented PID controller and K_{error}1

The response of each work line was analyzed in relation to simulation time and states, summarized in Table (11).

Table 11: Time of simulation versus the current state which effect on the system performance

Time of simulation (T)			The State
	From	To	
Disclosed	0	1.3	Initial state to be stable
	1.3	2.3	Move the ball from stable state with white rod

5. Conclusion

In this comprehensive study, we explored the dynamics and control strategies of the Ball on Plate (BnP) system, an advanced nonlinear mechanical system with 9 Degrees of Freedom (DOF). The research focused on modeling, simulating, and controlling the BnP system using various control algorithms, particularly Proportional-Integral-Derivative (PID) controllers. Initially, we laid the groundwork by analyzing the theoretical aspects of the BnP system. This involved constructing detailed block diagrams for each work line to understand the system's dynamics and response behaviors. The analysis revealed significant instabilities, highlighting the need for advanced control strategies. The transition from theory to implementation involved designing and fine-tuning PID controllers. Using a trial-and-error method, we adjusted the PID parameters to minimize system errors and enhance stability. Simulations in Simulink and Arduino demonstrated substantial performance improvements, marked by reduced oscillations and quicker settling times. Practical experimentation further validated our findings. Implementing PID controllers in a physical setup revealed challenges like overshooting, which were mitigated by adjusting error gain and optimizing servo motor responses. These insights emphasized the importance of iterative tuning and real-time adjustments. We also explored alternative control strategies, including PI1, PI2, and PD controllers, rigorously tuning their parameters. This comparative analysis highlighted the strengths of each strategy, with the PD controller achieving a balance between stability and responsiveness. The system's robustness under external disturbances, such as manual manipulation of the ball, provided a comprehensive assessment of the control mechanisms' effectiveness. The findings underscore the successful design, modeling, simulation, and control of the BnP system. The detailed analysis validates the effectiveness of PID controllers and emphasizes the system's adaptability for applications in automatic control and engineering.

In conclusion, this paper presents a thorough examination of the BnP system, providing a solid foundation for future research in nonlinear multi-DOF systems. The insights gained contribute significantly to control engineering, offering practical solutions applicable to complex systems.

Key Points: System Analysis: Introduces a novel approach to the BnP system; System Response: Effectively demonstrates pivotal competent of our project concepts; and Controller Performance: Successfully stabilizes servo motor responses.

This case study showcases the BnP system as an innovative setup, utilizing PI, PD, and PID controllers through a trial-and-error tuning approach, effectively demonstrating the impact of these strategies on performance.

6. References

[1] J. Kumar, N. Showme, M. Aravind and R. Akshay*, “Design and control of ball on plate system” in International Science Press, 2016, pp. 765-778.

[2] G. Franklin, J. David and M. Workman, Digital Control of Dynamic Systems, 3rd ed. Pilarcitos Ave, Half Moon Bay, CA: Ellis-Kagle Press, 1998.

[3] F. Pasha, M. S. J, “Asymptotic Stabilization and Trajectory Tracking of 4 DOF Ball Balancer using LQR” in IEEE Region 10 Conference, 2019.

[4] C. Cheng, C. Tsai, “Visual Servo Control for Balancing a Ball-Plate System” in International Journal of Mechanical Engineering and Robotics Research, 2016.

[5] S. Awtar, K. Craig, “Mechatronic Design of a Ball On Plate Balancing System”, 1999.

[6] A. Kastner, J. Inga, T. Blauth, F. Köpf, M. Flad and S. Hohmann , “Model-Based Control of a Large-Scale Ball-on-Plate System With Experimental Validation”, 2019.

[7] K. Wang, Embedded and Real-Time Operating Systems, 1st ed. Pullman, WA, USA: Springer International Publishing, 2017.

[8] F. Groen, P. Jonker, R. Duin, “Hardware Versus Software Implementations of Fast Image Processing Algorithms” in Real-Time Object Measurement and Classification, 1988.

[9] Smith, John. "A Beginner's Guide to Digital Signal Processing (DSP)." Journal of Signal Processing Education, vol. 5, no. 2, 2021, pp. 45-67.

[10] M. Kuncan, Kaplan Kaplan, F. Acar, I. Kundakci and H. Ertunc, “Fuzzy Logic Based Ball on Plate Balancing System Real Time Control by Image Processing” in International Journal of Natural and Engineering Sciences, 2016.

[11] F. Zheng, X. Li, S. Wang and D. Ding, “Position Control of Ball and Plate System Based on Switching Mechanism” in International Conference on Automation and Logistics, 2011.

[12] A. Adiprasetya, A. Wibowo, “Implementation of PID Controller and Pre-Filter to Control Non-Linear Ball and Plate System” in International Conference on Control, Electronics, Renewable Energy and Communications, 2016.

[13] K. Yaovaja, “Balancing on a Stewart Platform using Fuzzy Supervisory PID Visual Servo Control” in IEEE, 2018.

[14] L. Morales, O. Camacho, P. Leica and D. Chávez, “A Sliding-Mode Controller from a Reduced System Model: Ball and Plate System Experimental Application” in Science and Technology Publications, 2017, pp 590-597.

[15] H. Bang and Y. Lee, “Implementation of a Ball and Plate Control System Using Sliding Mode Control” in IEEE, 2018, pp 2169-3536.

[16] M. Ho, Y. Rizal and L. Chu, “Visual Servoing Tracking Control of a Ball and Plate System: Design, Implementation and Experimental Validation” in International Journal of Advanced Robotic Systems, 2013.

[17] N. Mohajerin and M. Menhaj, “A Reinforcement Learning Fuzzy Controller for the Ball and Plate System” in IEEE, 2010.

[18] C. Ker, C. Lin and R. Wang, “Tracking and balance control of ball and plate system” in Journal of the Chinese Institute of Engineers, 2007, pp 459-470.

[19] C. Solomon and T. Breckon, Fundamentals of Digital Image Processing, 1st ed. UK: John Wiley & Sons, 2011.