Proceedings of the 3rd
RSI International Conference on Robotics and Mechatronics
October 7-9, 2015, Tehran, Iran

Time Optimized Digital Image Processing of Ball and Plate System Using Artificial Neural Network

Yasaman Parandian, Hannaneh Z. Arabshahi, Ali Nasr, S. Ali A. Moosavian.
Center of Excellence in Robotics and Control, Advanced Robotics and Automated Systems (ARAS) Lab.
Department of Mechanical Engineering, K.N. Toosi University of Technology, Tehran, Iran.
yasaman.parandian@gmail.com, hananearabshahi@yahoo.com, ANasr@mail.kntu.ac.ir, Moosavian@kntu.ac.ir.

***Abstract*—Regarding robot interaction with moving objects and the necessity for robot controller to react in a timely manner having the objects current position, finding an efficient positioning method has always been a challenge. As ball and plate system is considered to be a standard experimental system in control laboratories. The system is consisted of the following pieces: a horizontal plate which gets tilted along each horizontal axes. The ball, which is located on the plate; is controlled by the system in a way to be placed in any point on the plate. In fact, the ball and plate system is an extension of the classical ball on beam experiment that is often used to study various types of control and stability problems.**

***Index Terms*—Forward Kinematics, Neural Networks, Real-Time Computer Vision, 3RRS Parallel Robot, Ball and Plate.**

I. Introduction

Ball and plate system is a great benchmark to test different controlling algorithms [1]–[5]. This parallel robot is composed of three different sections namely: the feedback method, the system structure and the controller.

Recently, the application of the image processing optimized algorithm for finding an objects position has become a challenging concept and many researches has been performed on the subject. In 2013, M. T. Ho et al. claimed that there are two methods for positioning the ball in a ball and plate system. These methods fall into two categories: the touch screen and the overhead camera [6]. In 2002, Awtar S et al. presented a resistive touch-sensitive glass screen that is actually meant to be a computer touchscreen was used for sensing the ball position [7]. Yuan D et al. illustrated trajectoryTracking of a ball-plate system with a Touch Screen in 2010 [8]. In 2003, Fan X Zhang et al. discussed tracking the ball and plate system with a touch screen and rotary cylinder and the balls position is measured using a CCD came [9]. In 2003, Park J H et al. stated that the center position of the ball is measured with a machine vision system [10]. Moreover, Moarref M et al. discussed that a webcam provides the system with feedback of the ball's position in 2008. The overhead camera to search for the ball has been used by Rad A B et al. in 2003, Yip P T et al in 2004 and MorenoArmendriz M A et al. in 2010 [11]–[14].

The critical challenge in this paper is to find the most efficient practice to determine balls current position on the plate. In general, image processing and touch screen are the most popular methods to achieve the mentioned goal. The ball and plate system presented here is based on 3RRS parallel manipulator and digital camera is used to implement the real time image processing to locate the ball. Digital camera, compared to touch screen method, covers a larger scope and it is more cost effective as well. In this method, in order to compensate the hardware delays, the artificial neural network (ANN) is used for predicting the balls location. The Artificial Neural Network has been created by using a set of sample test data of balls trajectory on the plate in a real world situation. After the training phase, the resulted ANN method can predict the prospective position of the ball in a fraction of a moment.

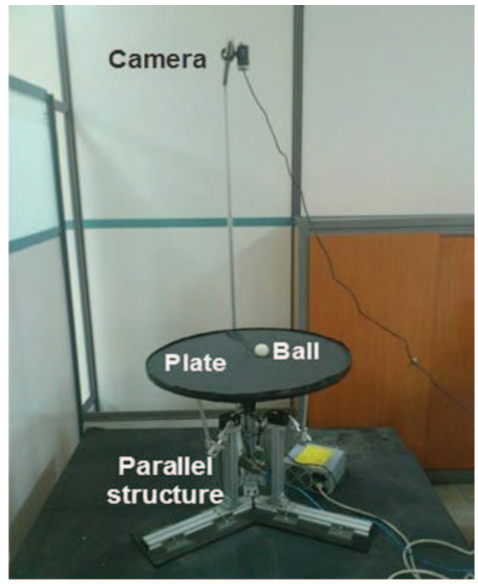

Fig. 1: The ball and plate system based on 3RRS parallel robot.

The present paper is organized as follows: first section is an introduction to the ball and plate system, the other works and reference in this regard and a brief presentation about this papers topic of discussion. The second section gives an overview about image processing and its role in this study. Third section discusses different methods of prediction and of course the prediction method utilized in this paper. Section four presents the description of 3RRS parallel manipulators forward kinematics. It starts with a brief explanation of the

978-1-4673-7234-3/15/$31.00 ©2015 IEEE

Authorized licensed use limited to: Eindhoven University of Technology. Downloaded on May 09,2023 at 07:30:01 UTC from IEEE Xplore. Restrictions apply.

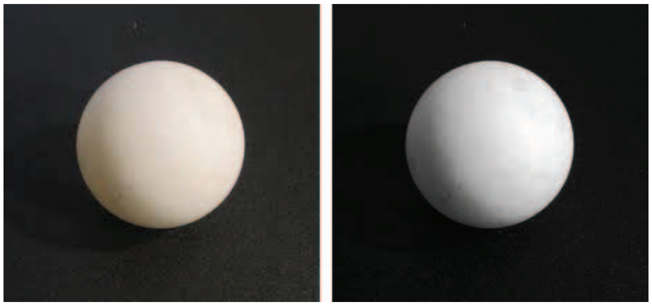

Fig. 2: Colored versus grayscale image.

method used to get input from the camera; then the prediction method is explained. Finally, it is concluded by a description of 3RRS parallel manipulators forward kinematics.

II. Image Processing

The main goal of image processing in this parallel Robot is to inform the system of the balls current position as well as a prediction for the next position of the ball in the next T milliseconds. This prediction is then used by the controller and provides the input for the robot real-time control.

The camera is supposed to capture the trajectory of a moving white ball on a black background. So the resulted film contains several frames; each captured frame has been transformed into a two dimensional array of numbers. In this array the brightest pixels show larger numbers -these are pixels around the ball and the ball itself- while the darker pixels -the background- have values around zero.

Considering the fact that each array was composed of a large spectrum of numbers, the best practice was to somehow minimize this variation of numbers so, each array was, transformed into black and white, Fig.2. Obviously, the grayscale image - the right image- is much easier to process.

The next step was to get the index of the max element in the array. To simplify the process, the whitish gray points were all converted to white; for instance the numbers between 255 and 230 were all changed to number 230. At this point, the array index which contain the larger numbers were the points on the ball. So these bright points show near ball environs on the plate. In order to achieve a single point, the average number of whitish points were used as the balls position.

Fig.3 demonstrates a set of sample data for the ball movement on the plate and Fig.4 shows the changes in the motor angles in an experiment.

III. Prediction

There are several ways of predicting the balls prospective trajectory such as Kalman filter for object tracking which is a fairly precise way of tracking a moving object [15]. The method works properly but it cannot address this robots requirement which is providing live inputs for the motors to determine their next move; moreover when it comes to a still object, it does not work. Another option that can be considered for prediction is creating an Artificial Neural

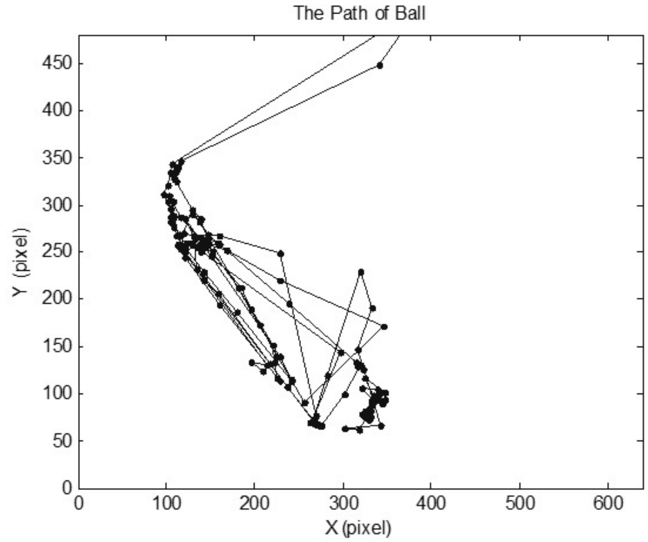

Fig. 3: A demonstration of ball movement on the plate.

Network (ANN). Although an ANN is not always the best prediction mechanism, but when they are chosen diligently; it can lead to perfect results.

In this paper, the prediction mechanism has been implemented using the Artificial Neural Network concept, Fig.5. In order to implement a neural network function, the inputs and the desired outputs are required. In this scenario, five variables have been considered as input data: Table I illustrates the ANN training input data with their corresponding description:

TABLE I: Artificial Neural Network Training Data Description Input Data

1	Input data 1	coordinate x (the ball location on the plate)
2	Input data 2	coordinate y (the ball location on the plate)
3	Input data 3	motor position (the first motor)
4	Input data 4	motor position (the second motor)
5	Input data 5	motor position (the third motor)

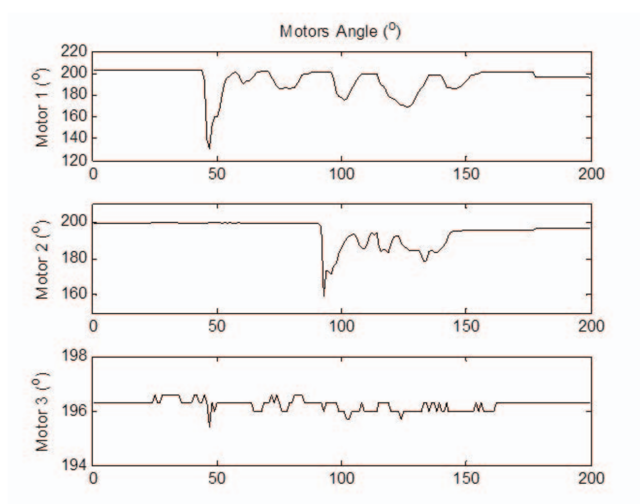

Fig. 4: A demonstration of motor angle changes.

Authorized licensed use limited to: Eindhoven University of Technology. Downloaded on May 09,2023 at 07:30:01 UTC from IEEE Xplore. Restrictions apply.

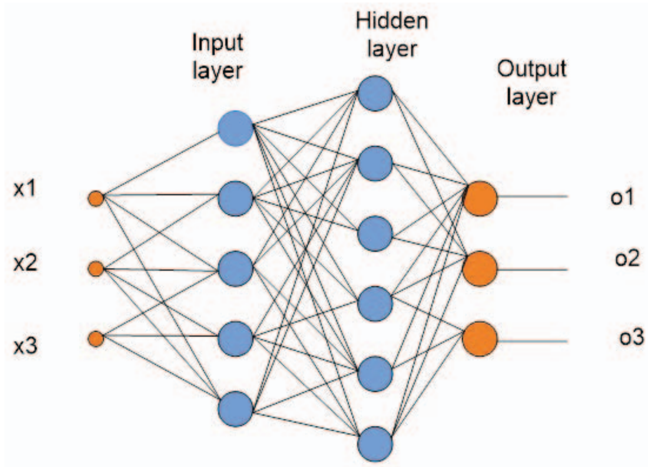

Fig. 5: The artificial Neural Network.

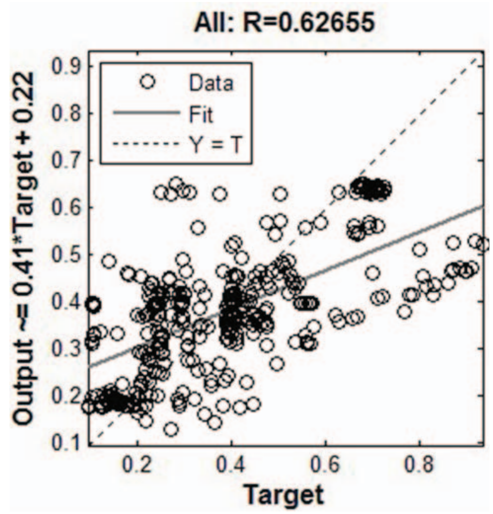

Fig. 6: ANN has been trained with minor errors.

And the output data are the:

TABLE II: Artificial Neural Network Training Data Description Output Data

1	Output data 1	coordinate x (the ball location on the plate after T milliseconds)
2	Output data 2	coordinate y (the ball location on the plate after t milliseconds)

As it is demonstrated in TABLE I and TABLE II, the neural network has been trained with two arrays of data, the input array is of size (5,180) and the size of output array is (2,180). The LevenbergMarquardt algorithm (LMA) has been used for ANN training. This algorithm typically takes more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples. After the learning process, the resulted ANN function can predict the approximate prospective position of the ball in time T with some minor negligible deviations, Fig.6.
The number of network layers for training the Artificial Neural Network have been chosen by a trial and error mechanism; the desirable goal was to achieve an estimated value with a negligible difference from the real value.

IV. Kinematics Analysis of 3-RRS Parallel Manipulator

Although the neural network method is used to calculate the orientation of the mobile platform, this method implicitly contains the kinematic model and the kinematic formulation is discussed in this section.

A. 3-RRS Parallel Manipulator

The structure of 3-DOF spatial parallel manipulator is shown in Fig.1. It is composed of three similar limbs connecting the base platform to the moving platform. Each of these legs consists of a succession of revolute and spherical joints. Therefore, the base is annexed to the moving platform by three identical RRS joining. A 3-RRS mechanism has three DOF, which are one translational and two rotational degrees of freedom, and is driven by three revolute actuators.

B. Kinematics Formulation

Inverse kinematics of the 3RRS parallel robot was extensively presented in [16]. In this section Forward kinematics of 3RRS parallel robot is presented. Forward kinematics of parallel manipulators is generally very complicated, and its solution usually involves systems of nonlinear equations, which are highly coupled and in general have no closed form and unique solution [17].
Forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters [18].
The geometry of the spatial 3-RPS parallel robot is presented in Fig.7. As it is shown, for kinematic analysis, a Cartesian reference coordinate system $O\{x, y, z\}$ is attached on the fixed base platform at point O and another coordinate system $P\{u, v, w\}$ is attached on the moving platform at point p. The frame $A'_i\{x'_{ai}, y'_{ai}, z'_{ai}\}$ (i=1,2,3) which is a local fixed frame and the frame $A_i\{x_{ai}, y_{ai}, z_{ai}\}$ (i=1,2,3) which moves along with the down link, are placed at point A_i. Both axes x_{ai} and x'_{ai} are identical with the revolute axis u_{ai}. $B_i\{x_{bi}, y_{bi}, z_{bi}\}$ (i=1,2,3) is a local frame which moves along with the upper link and its origin is point B_i. Frame $C_i\{x_{ci}, y_{ci}, z_{ci}\}$ (i=1,2,3) is parallel to frame $B_i\{x_{bi}, y_{bi}, z_{bi}\}$. Both of the moving platform environment $A_1A_2A_3$, and the base platform environment $C_1C_2C_3$ are equilateral triangles with $OA_1 = OA_2 = OA_3 = r$ while $PC_1 = PC_2 = PC_3 = R$. Both A_iB_i and B_iC_i are homogenous links, the lengths of which are l_{i1} and l_{i2}, respectively. The rotation angles of A_iB_i is denoted by θ_i whereas α_i is the rotation angles of B_iC_i. The points A_i (i=1,2,3), which are coordinates in fixed frame, is given as:

Authorized licensed use limited to: Eindhoven University of Technology. Downloaded on May 09,2023 at 07:30:01 UTC from IEEE Xplore. Restrictions apply.

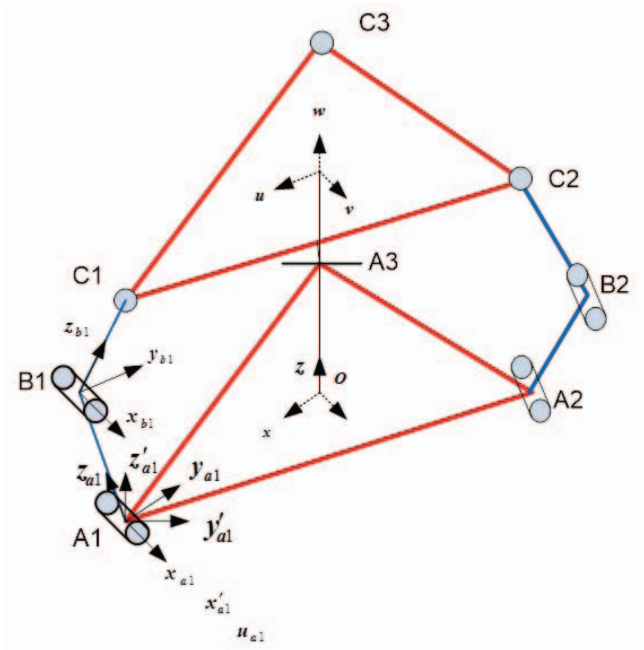

Fig. 7: The geometry of the spatial 3-RRS parallel robot.

$$\begin{aligned} a_1 &= [r, 0, 0] \\ a_2 &= \left[-\frac{1}{2}r, \frac{\sqrt{3}}{2}r, 0\right] \\ a_3 &= \left[-\frac{1}{2}r, -\frac{\sqrt{3}}{2}r, 0\right] \end{aligned} \tag{1}$$

The position vector q_i of the C_i spherical joint, with respect to the fixed coordinate system is obtained by the transformation (2).

$$q_i = a_i + {}^{O}\mathbf{R}_{A'_i}\,{}^{A'_i}\mathbf{R}_{A_i}\,{}^{A_i}L_{i1} + {}^{O}\mathbf{R}_{A'_i}\,{}^{A'_i}\mathbf{R}_{A_i}\,{}^{A_i}\mathbf{R}_{B_i}\,{}^{B_i}L_{i2} \tag{2}$$

where ${}^{O}\mathbf{R}_{A'_i}$, is the constant matrix describing the orientation of the frame $A'_i\,\{x'_{ai}, y'_{ai}, z'_{ai}\}$ with respect to the frame $O\,\{x, y, z\}$ are introduce as:

$$\begin{aligned} {}^{O}\mathbf{R}_{A'_1} &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ {}^{O}\mathbf{R}_{A'_2} &= \begin{bmatrix} -\frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ {}^{O}\mathbf{R}_{A'_3} &= \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3}$$

${}^{A'_i}\mathbf{R}_{A_i}$ is the orientation matrix of frame $A'_i\,\{x'_{ai}, y'_{ai}, z'_{ai}\}$ with respect to frame $A_i\,\{x_{ai}, y_{ai}, z_{ai}\}$ as:

$${}^{A'_i}\mathbf{R}_{A_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i & -\sin\theta_i \\ 0 & \sin\theta_i & \cos\theta_i \end{bmatrix} \tag{4}$$

And ${}^{A_i}\mathbf{R}_{B_i}$ is the orientation matrix of the frame $B_i\,\{x_{bi}, y_{bi}, z_{bi}\}$ with respect to frame $A_i\,\{x_{ai}, y_{ai}, z_{ai}\}$ as:

$${}^{A_i}\mathbf{R}_{B_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{5}$$

The position vector of the second revolute joint with respect to the moving frame A_i, and the position vector of the spherical joint with respect to the moving frame B_i are denoted by ${}^{A_i}L_{i1}$ and ${}^{B_i}L_{i2}$, respectively and introduced as:

$$\begin{aligned} {}^{A_i}L_{i1} &= \begin{bmatrix} 0 \\ 0 \\ l_{i1} \end{bmatrix} \\ {}^{B_i}L_{i2} &= \begin{bmatrix} 0 \\ 0 \\ l_{i2} \end{bmatrix} \end{aligned} \tag{6}$$

Substituting (3) - (6) into (2), obtained:

$$\begin{aligned} q_1 &= \begin{bmatrix} r + l_{21}\sin(\alpha_1+\theta_1) + l_{11}\sin\theta_1 \\ 0 \\ l_{21}\cos(\alpha_1+\theta_1) + l_{11}\cos\theta_1 \end{bmatrix} \\ q_2 &= \begin{bmatrix} -\frac{r}{2} - \frac{l_{22}\sin(\alpha_2+\theta_2)}{2} - \frac{l_{12}\sin\theta_2}{2} \\ \frac{\sqrt{3}}{2}r + \frac{\sqrt{3}}{2}l_{22}\sin(\alpha_2+\theta_2) + \frac{\sqrt{3}}{2}l_{12}\sin\theta_2 \\ l_{22}\cos(\alpha_2+\theta_2) + l_{12}\cos\theta_2 \end{bmatrix} \\ q_3 &= \begin{bmatrix} -\frac{r}{2} - \frac{l_{23}\sin(\alpha_3+\theta_3)}{2} - \frac{l_{13}\sin\theta_3}{2} \\ -\frac{\sqrt{3}}{2}r - \frac{\sqrt{3}}{2}l_{23}\sin(\alpha_3+\theta_3) - \frac{\sqrt{3}}{2}l_{13}\sin\theta_3 \\ l_{23}\cos(\alpha_3+\theta_3) + l_{13}\cos\theta_3 \end{bmatrix} \end{aligned} \tag{7}$$

Since the revolute axis of each joint in the base is parallel to the opposite edge and the two axes of revolute joints on each leg parallel each other, (8) constrains exists:

$$\begin{aligned} &\frac{q_{2y}}{q_{2x}} = \tan(60) \\ &\frac{q_{3y}}{q_{3x}} = \tan(-60) \\ &q_{1y} + q_{2y} + q_{3y} = q_{1x} + q_{2x} + q_{3x} = 0 \end{aligned} \tag{8}$$

By substituting (7) into (8) with some manipulation, the rotation angles of B_iC_i, α_i can be determined as a function of θ_i by (9).

$$\begin{aligned} \alpha_1 &= \sin^{-1}\left\{ \frac{1}{2l_{21}} \begin{bmatrix} r + l_{22}\sin(\alpha_2+\theta_2) \\ + l_{12}\sin\theta_2 \\ + l_{23}\sin(\alpha_3+\theta_3) \\ + l_{13}\sin\theta_3 - l_{11}\sin\theta_1 \end{bmatrix} \right\} - \theta_1 \\ \alpha_2 &= \sin^{-1}\left[\frac{-1}{l_{22}}(r + l_{12}\sin\theta_2)\right] - \theta_2 \\ \alpha_3 &= \sin^{-1}\left[\frac{-1}{l_{23}}(r + l_{13}\sin\theta_3)\right] - \theta_3 \end{aligned} \tag{9}$$

Authorized licensed use limited to: Eindhoven University of Technology. Downloaded on May 09,2023 at 07:30:01 UTC from IEEE Xplore. Restrictions apply.

The orientation of the frame $P\{u, v, w\}$ with respect to frame $O\{x, y, z\}$ can be described by (10).

$$\frac{\vec{N} \bullet \vec{S}}{\left|\vec{N}\right| \left|\vec{S}\right|} = \cos\delta \tag{10}$$

In witch $\vec{N}$ and $\vec{S}$ are normal vectors of moving plan with respect to the fixed and moving frame, respectively, and can be calculated by (11).

$$\begin{aligned} \vec{N} = & \begin{bmatrix} (p_{y2} - p_{y1})(p_{z3} - p_{z1}) \\ -(p_{y3} - p_{y1})(p_{z2} - p_{z1}) \end{bmatrix} i \\ & + \begin{bmatrix} (p_{x2} - p_{x1})(p_{z3} - p_{z1}) \\ -(p_{x3} - p_{x1})(p_{z2} - p_{z1}) \end{bmatrix} j \\ & + \begin{bmatrix} (p_{x2} - p_{x1})(p_{y3} - p_{y1}) \\ -(p_{x3} - p_{x1})(p_{y2} - p_{y1}) \end{bmatrix} k \\ \vec{S} = & 1k \end{aligned} \tag{11}$$

By using (7), (9) and (10), the orientation of moving platform can be determined as a function of θ_i, While the translation of moving platform is calculated as (12).

$$Z = \frac{1}{3}(p_{z1} + p_{z2} + p_{z3}) \tag{12}$$

V. Result

Applying the ANN method to the ball and plate system gives a significant boost to the system response time and vision effectiveness which in turn can compensate the hardware delays and is capable of addressing the requirement for prediction in such a system. Just like any software procedure, the trained neural network method can return the predicted coordinates for the prospective position of the ball in a fraction of a moment. In this way, the controller has the timely correct input to plan the next motor movement and roll the ball to the deliberate next position on the plate. The prediction mechanism of neural network has learned to predict the position of the ball T milliseconds later after the current time, Table III.

TABLE III: Artificial Neural Network Training Result

Result	Samples	MSE	R
Training	125	2.25733e-2	6.55845e-1
Validation	27	2.42405e-2	5.30324e-1
Testing	27	2.61628e-2	5.96871e-1

T is a constant value and in order to change this value, the network should be trained by new training data and create a new method which corresponds to time T.

VI. Validation

The whole estimation time has been considered T=500 (ms). As discussed before, Fig.3 and Fig.4 show an illustration of the ball movement on the plate and the corresponding depiction of changes in motor angles, respectively. The artificial neural network has been trained as discussed and the ANN function has been resulted from the training process. Fig.8 illustrates

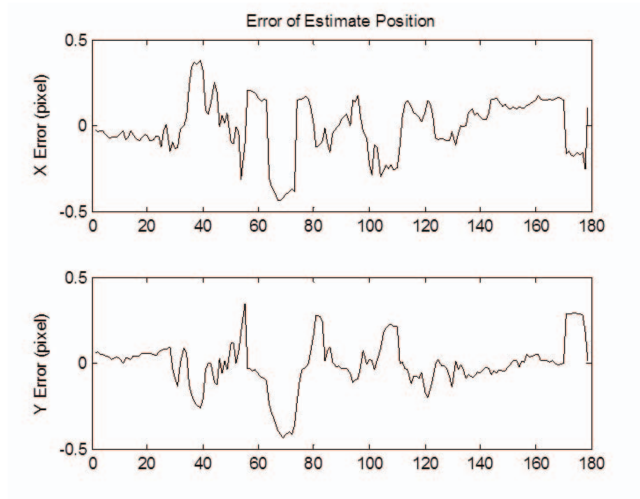

Fig. 8: Error of the estimated position.

the negligible difference between the estimated and the real value. As it is obvious in the Fig.8, the difference between the estimated location and the real location is less that one pixel.

VII. Conclusion

The main challenge of this paper is to introduce an image processing time-optimized method, minimizing the hardware delays side effects. The studied system was a ball and plate manipulator, using the ball tracking mechanism as an image processing method. A camera device captured the images of a ball movements. In order to achieve a time optimized reaction, the artificial neural network method is applied to predict the prospective trajectory of the ball. The neural network had been trained by real tested data and it is capable of predicting the prospective position with an acceptable level of precision. It has been shown that there is a minor error for estimated location and time optimized digital image processing of Ball and Plate System can fully compensate the hardware related delay.

References

[1] Jacob S Glower and Jeffrey Munighan. Designing fuzzy controllers from a variable structures standpoint. *Fuzzy Systems, IEEE Transactions on*, 5(1):138–144, 1997.

[2] John Hauser, Shankar Sastry, and Petar Kokotovic. Nonlinear control via approximate input-output linearization: the ball and beam example. *IEEE transactions on automatic control*, 37(3):392–398, 1992.

[3] Ebrahim H Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13, 1975.

[4] Kim C Ng and Mohan M Trivedi. Neural integrated fuzzy controller (nif-t) and real-time implementation of a ball balancing beam (bbb). In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 2, pages 1590–1595. IEEE, 1996.

[5] Jyh-Shing Roger Jang. Anfis: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(3):665–685, 1993.

[6] Ming-Tzu Ho, Yusie Rizal, and Li-Ming Chu. Visual servoing tracking control of a ball and plate system: Design, implementation and experimental validation. *International Journal of Advanced Robotic Systems*, 10, 2013.

Authorized licensed use limited to: Eindhoven University of Technology. Downloaded on May 09,2023 at 07:30:01 UTC from IEEE Xplore. Restrictions apply.

[7] Shorya Awtar, C Bernard, N Boklund, A Master, D Ueda, and Kevin Craig. Mechatronic design of a ball-on-plate balancing system. *Mechatronics*, 12(2):217–228, 2002.

[8] Dongfeng Yuan and Zhenhao Zhang. Modelling and control scheme of the ball–plate trajectory-tracking pneumatic system with a touch screen and a rotary cylinder. *IET control theory & applications*, 4(4):573–589, 2010.

[9] Xingzhe Fan, Naiyao Zhang, and Shujie Teng. Trajectory planning and tracking of ball and plate system using hierarchical fuzzy control scheme. *Fuzzy Sets and Systems*, 144(2):297–312, 2004.

[10] Jong Hyeon Park and Young Jong Lee. Robust visual servoing for motion control of the ball on a plate. *Mechatronics*, 13(7):723–738, 2003.

[11] Ahmad B Rad, PT Chan, Wai Lun Lo, and CK Mok. An online learning fuzzy controller. *Industrial Electronics, IEEE Transactions on*, 50(5):1016–1021, 2003.

[12] Phillip Thomas Yip. *Symbol-Based Control of a Ball-on-plate Mechanical System*. PhD thesis, 2004.

[13] Marco A Moreno-Armendáriz, César A Pérez-Olvera, Floriberto Ortiz Rodríguez, and Elsa Rubio. Indirect hierarchical fcmac control for the ball and plate system. *Neurocomputing*, 73(13):2454–2463, 2010.

[14] Marco Moreno-Armendáriz, Elsa Rubio, César Pérez-Olvera, et al. Design and implementation of a visual fuzzy control in fpga for the ball and plate system. In *Reconfigurable Computing and FPGAs (ReConFig), 2010 International Conference on*, pages 85–90. IEEE, 2010.

[15] Michael Piovoso and Phillip A Laplante. Kalman filter recipes for real-time image processing. *Real-Time Imaging*, 9(6):433–439, 2003.

[16] Jianfeng Li, Jinsong Wang, Wusheng Chou, Yuru Zhang, Tianmiao Wang, and Qixian Zhang. Inverse kinematics and dynamics of the 3-rrs parallel platform. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2506–2511. IEEE, 2001.

[17] Richard P Paul. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.

[18] Hamid D Taghirad. *Parallel robots: mechanics and control*. CRC press, 2013.

Authorized licensed use limited to: Eindhoven University of Technology. Downloaded on May 09,2023 at 07:30:01 UTC from IEEE Xplore. Restrictions apply.