International Journal of
Science and Engineering Investigations vol. 7, issue 74, March 2018
ISSN: 2251-8843

Implementation of Neural Network Methods in Measurement of the Orientation Variables of Spherical Joints

Kemal Guven[1], Andac T. Samiloglu[2]
[1,2]Mechanical Engineering Department, Baskent University, Ankara, Turkey
([1]kkemalguvenn@gmail.com, [2]andactore@gmail.com)

***Abstract*-** In this study, a method to measure the orientation variables of spherical joints is proposed. First, an experimental setup is designed. In the setup, custom designed sensors consisting infrared transceiver pairs are proposed. Image processing methods are used to determine the true orientation of the joint. The values obtained from the proposed sensors and the values obtained as a result of the image-processing algorithm are combined to generate datasets. The datasets are used to train a single-layer neural network model. Finally, the usability of such sensors are demonstrated.

***Keywords*-** *Spherical Joint Orientation Variables, Infrared Based Sensors, Neural Network, Sensor Network*

I. INTRODUCTION

Spherical joints (also called as ball joints) are used in many mechanical systems. In the automatic control of such systems, the orientation of the spherical joints may be required to be perceived. The measurement of single-axis movements can be done with various sensors; however, there are no special orientation measuring sensors for spherical joints. For one degree of freedom revolute joints, optical and magnetic encoders are frequently used sensors. These sensors are placed on the rotation axis of the joints to measure the rotation angles of the joints. Potentiometers are also used for the same purpose [1].

Another type of sensor used to measure similar movements is inertia sensors. The advantage of these sensors, consisting of accelerometer, gyroscope and magnetometer, is the flexibility of placement. Wang et al. [2] used inertial sensors to predict robot angles. They used Kalman filter for sensor fusion and the robot manipulator's joint angles are measured with 2° resolution. Vikas and Crane [3] measured joint angles using multiple inertial sensors. They proposed an algorithm which is independent of integration errors/drift, does not require knowledge of robot dynamics.

Flexible sensors are also used to detect single-axial rotational movements. These sensors are bent with the motion of the joints. The distortion of sensor changes the resistance between the terminals so that the bending angles can be calculated. Saggio designed a data glove using flex sensors to detect the hand gestures [4]. In his study, non-standard installation of the sensors increased resolution and reduced standard deviation.

In this study, firstly, an experimental setup is designed to measure the orientation of the spherical joints. Dataset generated from this setup is used to predict orientation of a spherical joint. Associating these multiple sensors with orientation is a major problem. This can also be called regression problem. These problems can be solved by statistical methods as well as artificial intelligence methods. The main ones of these methods are fuzzy logic and artificial neural networks. Evolutionary algorithms are also used in these problems.

Ghorbanian used different artificial neural network models to predict the performance of the compressor [5]. They used the general regression neural network for the regression problem in the study and compared it with the rotated general regression network they developed. A multi-layer neural network has been decided. They have achieved successful results with predictions and experiments.

In another study, regression analysis and artificial neural networks were compared in prediction and optimization problems [6]. In this biomedical engineering study, environmental variables such as temperature and humidity were used to estimate xylose consumption and production. The authors used the stability coefficient to evaluate the model. They found that the artificial neural network model they used gave better results than the regression model.

In our study, a single-layer neural network model is utilized to predict the orientation of spherical joints by using custom designed sensor data. In the following sections we will first describe the data collecting setup and the design of infrared sensors. Next, we explain the regression model for the data and propose a kinematic model for three degrees of freedom joint. Lastly, we discuss the performance of single-layer neural network model applied on infrared sensor network by comparing with image processing results.

II. EXPERIMENTAL SETUP

A. The Setup Structure

The setup consists of four parts; a camera used for image acquisition, sensors used to detect position change, a rod and platform with a pair of balls in different colors to detect the orientation of the spherical joint (Figure 1).

Figure 1. Experimental setup

The platform is made of 370x370mm aluminum composite material. The high-resolution USB camera is fixed at 800mm above the platform. RGB LED strips are placed around the camera to reduce the change of light intensity (Figure 2). Six custom made sensors were placed at 60° intervals to detect the change of position (The sensor working principles are explained in the following subsections). 328P microprocessor manufactured by ATMEGA has been used to read the change in these 6 sensors. The sensor data is transferred to the computer program on the computer via a serial port.

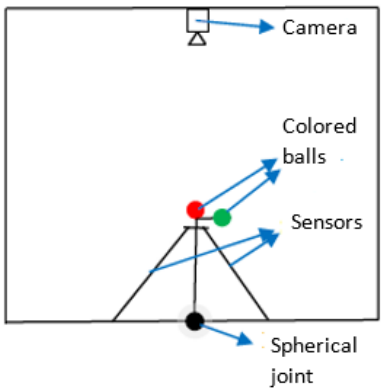

Figure 2. Setup schematic drawing

B. Image Processing

In order to find the positions of the balls in the experimental setup, the balls must be segmented from the images. For this reason, color based segmentation process is performed. The colors of the balls have been chosen differently so that the segmentation process would not get complicated and can be as accurate as possible. Red and green colored balls were placed in the setup (Figure 3). Images were analyzed in L*A*B* color space. In this color space, L represents brightness. A * is increasing from red to green, while B * is increasing from yellow to blue (Figure 4). The biggest advantage of this color space is that it is device independent and is minimally affected by brightness [7].

Figure 3. RGB and L*A*B* Color space

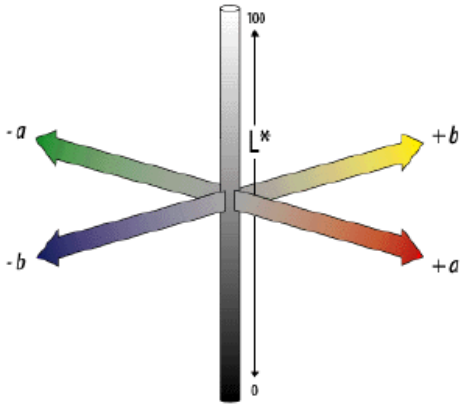

Figure 4. L*A*B Color space

First, the original image in RGB color space from the camera is transferred to L*A*B color space. Then, the red and green balls are segmented from the image by a threshold value algorithm. After finding the coordinates of the center of gravity of the segmented balls, the algorithm is terminated. The flowchart of the image processing algorithm is given in Figure 5.

C. Design and Installation of the Sensors

The designed sensors are manufactured with infrared transceiver pairs operating at 850nm and blue bright silicone hoses with lengths of 180mm (Figure 6). Infrared light is passed through the hose. The measured light intensity varies as the hose changes shape.

The single-axis deformation characteristics of the sensor were investigated before the designed sensor was installed on

the setup. A simple mechanism with a single revolute freedom is designed for this purpose. A potentiometer was placed at the junction of the two links and a sensor was attached to the links (Figure 7). The relationship between potentiometer values and sensor values is examined in the system.

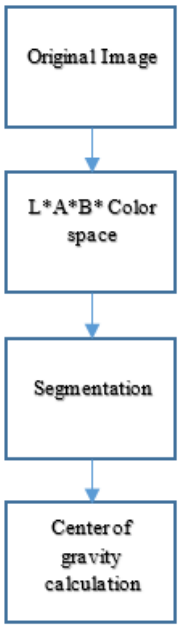

Figure 5. Flowchart of the image processing

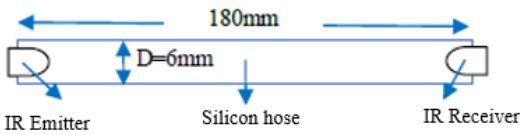

Figure 6. Sensor schematic drawing

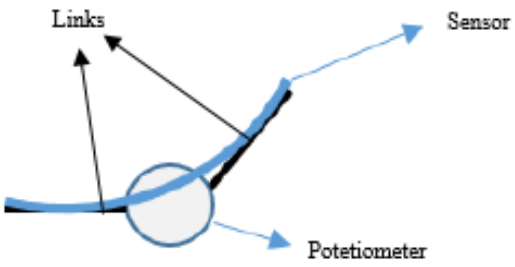

Figure 7. 1-DOF setup structure

At the 90 degree angular movement of the single degree of freedom scale, 2000 samples were collected. The relationship between sensor and potentiometer values is shown in Figure 8.

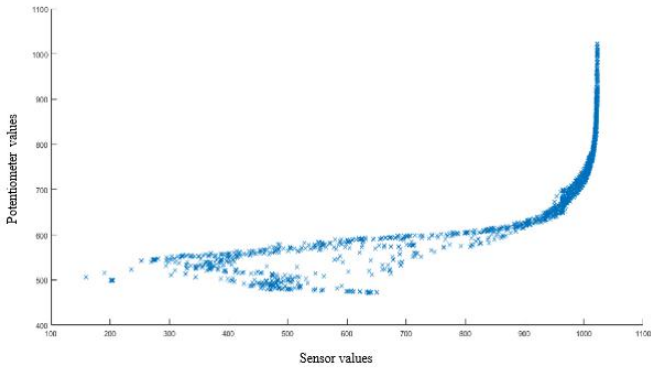

Figure 8. Relation between sensor and potentiometer

The sensors were placed at 60° intervals after the investigation on single-axis setup.

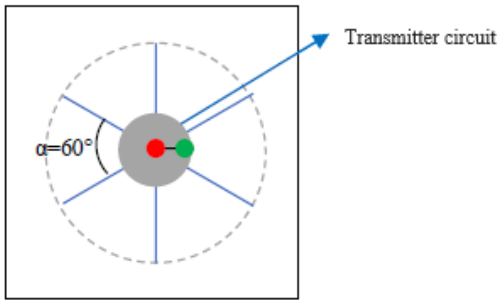

Figure 9. Schematic representation of the setup from top

D. Electronic Design

The setup consists of a receiver and a transmitter circuit. There are 6 infrared transmitters in the transmitter circuit. A card with a diameter of 80 mm was designed for this circuit (Figure 10). In the receiver circuit there are 6 infrared receivers. The data of the sensors are sent from the serial port to the computer by analogue reading between 0-1023 with Arduino UNO card. The block diagram of this system is given in Figure 11.

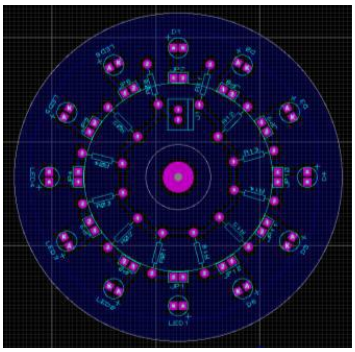

Figure 10. PCB Circuit of the transmitter

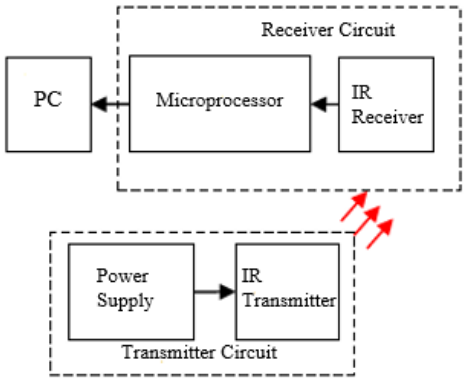

Figure 11. System block diagram

On the computer, the dataset is generated by combining the sensor data and the image data. Each sample in the dataset consists of 6 inputs and 4 outputs. The sensor data represent inputs where the coordinates of the colored balls are the outputs (Figure 12).

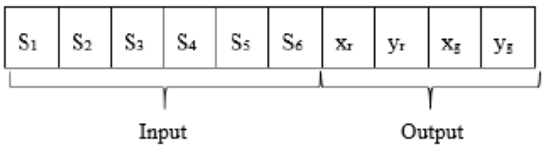

Figure 12. A simple dataset

III. REGRESSION MODEL

A. Dataset

There are 851 samples in the dataset. The sensor data forming the inputs in the data set consists of analog values in the range of 0-1023. The distribution of the inputs and outputs of the data set are shown in Figure 13 and Figure 14, respectively.

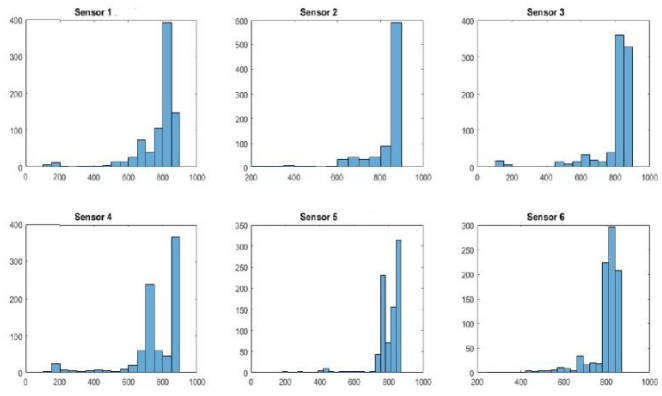

Figure 13. The distribution of the input

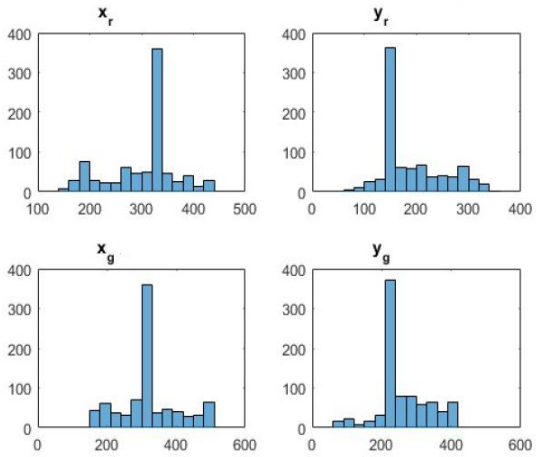

Figure 14. The distribution of the outputs

B. Neural Networks

Artificial Neural Networks (ANN) is a modeling algorithm that mimics the structure of the human brain. ANN consists of artificial nerve cells known as neurons. The simplest ANN structure is called perceptron (Figure 15) [8].

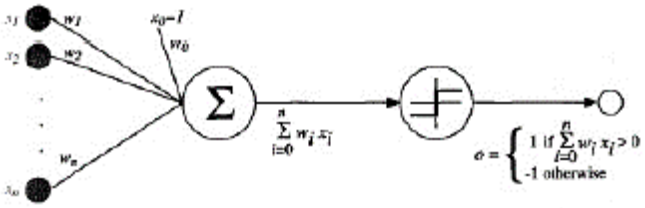

Figure 15. A perceptron [8]

The ANNs can be composed of multiple layers, the input layer, the hidden layers and the output layer (Figure 16).

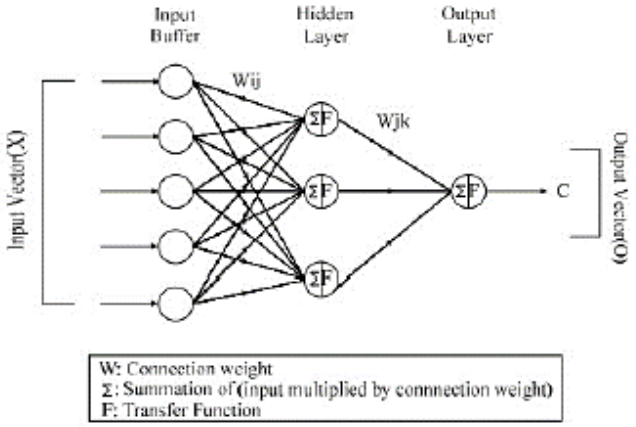

Figure 16. A multilayer neural network

Knowledge learned in ANN is stored in relevant weights. All inputs are given to the system as shown in Figure 16. The model multiplies the inputs by the weights and sends them to the hidden layer. In the hidden layer, the same operations are applied again to the last layer and sent to the activation function. The activation functions produce the final output values depending on the inputs. Activation functions may be linear, sigmoid, and step functions as given in Figure 17 [9].

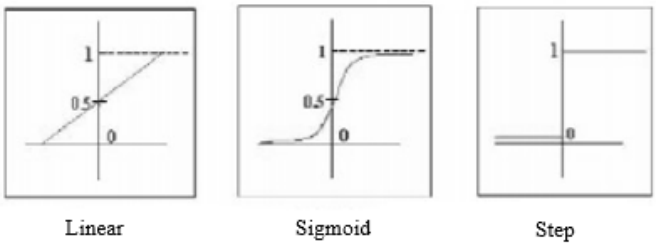

Figure 17. Activation functions

In this study, sigmoid function was chosen as the activation function given in (1).

$$F(x) = \frac{1}{(1+e^{-x})} \quad (1)$$

C. Preprocess of the Dataset

Before the training phase, there should be some arrangements on the dataset. First, the inputs generated by the sensor outputs are examined. The sensor data consists of analog values in the range of 0-1023. Figure 8 shows the outputs of the sensors in single-axis motion. Inputs in the dataset are concentrated in the range of 800-1000 due to the high-sensitivity of the sensors.

Inputs and outputs were normalized before the training phase. Normalization function is given in (2)

$$x_{norm} = \frac{x_{in}-x_{min}}{x_{max}-x_{min}} \quad (2)$$

In the literature, feed forward neural network is used in similar problem types [5] [6]. Therefore, this model is preferred in this study. The ANN model and calculations were performed in the corresponding module of the MATLAB® program.

First of all, a model consisting of a single hidden layer was worked on. The number of neurons in the hidden layer was determined by the inequality presented by Weigend [10].

$$1.1NP \leq 10NH[NI+1] \leq 3NP \quad (3)$$

where:

NP: Number of train data

NH: Number of neurons in hidden layer

NI: Number of neurons in input layer

The dataset consists of 851 samples. 70% of these used in training phase. Inputs have 6 parameters. With these values, NP = 596 and NI = 6 can be used and the inequality in (4) can be calculated.

$$9.4 \leq NH \leq 25.5 \quad (4)$$

The performance of the different neuron numbers was compared taking into account the mean square error (MSE) in (5) and the stability coefficient (R^2) in (6).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (5)$$

Where:

y_i: i[th] output

$\hat{y}_i$: i[th] prediction

n: Number of samples

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}\right) \quad (6)$$

$\bar{y}$: Mean prediction values

IV. Orientation Calculation

To calculate the orientation of the joint, we should handle inverse kinematic problem. The kinematics of the setup can be modeled as shown in Figure 18. To solve inverse kinematic problem, position of the green colored ball should be known. x and y-axis coordinates of the balls are outputs of the image processing phase.

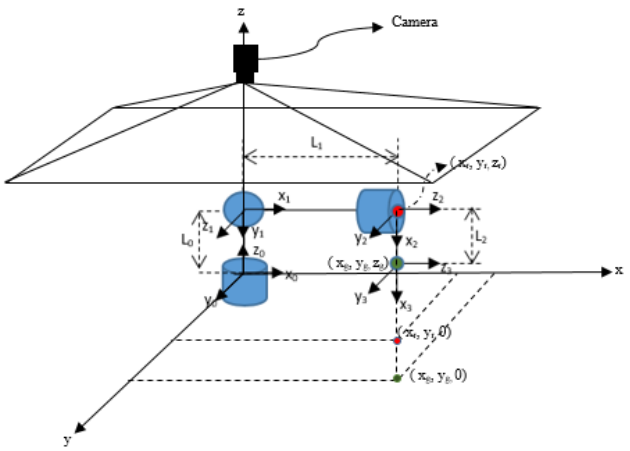

Figure 18. Kinematic model of the setup and camera location

To find z-axis coordinates, some geometric calculations are needed. In the setup, the length of first link is zero (L_0=0). Thus, the distance between red colored ball and origin is equal to L_1. And the distance between red colored ball and green colored ball is equal to L_2. The relations between coordinate variables and the distance variables are given in (7-10).

$$L_1 = \sqrt{x_r^2 + y_r^2 + z_r^2} \quad (7)$$

$$z_r = \sqrt{L_1^2 - x_r^2 - y_r^2} \quad (8)$$

$$L_2 = \sqrt{\left(x_g - x_r\right)^2 + \left(y_g - y_r\right)^2 + \left(z_g - z_r\right)^2} \quad (9)$$

$$z_g = \sqrt{L_2^2 - \left(x_g - x_r\right)^2 - \left(y_g - y_r\right)^2} + z_r \quad (10)$$

After we find the position of the green ball, we can start to solve inverse kinematic problem. Considering Figure 18, D-H parameters of the model can be derived as in Table I [11].

TABLE I. D-H PARAMETERS OF THE SETUP

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	$\pi/2$	0	0	θ_1
2	$\pi/2$	L_1	0	$\pi/2$-θ_2
3	0	L_2	0	θ_3

θ_1 and θ_2 can be calculated considering diagram in Figure 18 using geometrical relations as in (11) and (12).

$$\theta_1 = atan2(y_r, x_r) \quad (11)$$

$$\theta_2 = \sin^{-1}\left(\frac{z_r}{L_1}\right) \quad (12)$$

Transformation matrix of the green colored ball can be written using Table I. Considering position vector, we can find θ_3 as in (13)

$$\theta_3 = \cos^{-1}\left(\frac{x_g^2 + y_g^2 + z_g^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \quad (13)$$

V. RESULTS AND DISCUSSIONS

The *trainlm* command in the MATLAB program is used as the training function in the training phase of the ANN model. This function uses the Levenburg-Marquardt algorithm when updating weights. The dataset is divided into three groups: training, validation and testing. Of the 851 samples, 70% were divided into train, 15% were validation and the remaining 15% were divided into a test set. The number of neurons was taken as 10-15-20-25 respectively.

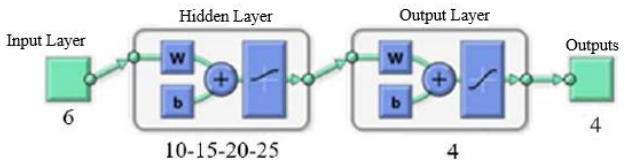

Figure 19. ANN structure

In Table II and Table III, the effect of different neuron numbers on the performance of the ANN model was compared.

TABLE II. PERFORMANCE OF TRAIN AND VALIDATION SETS

Number of Neurons	Train		Validation	
	MSE(e-3)	R^2	MSE(e-3)	R^2
10	7.126	0.93	7.282	0.93
15	5.882	0.95	5.593	0.95
20	6.215	0.94	6.611	0.94
25	6.534	0.94	7.473	0.92

TABLE III. PERFORMANCE OF TEST SET

Number of Neurons	MSE(e-3)	R^2
10	9.813	0.90
15	8.504	0.91
20	7.049	0.93
25	7.177	0.92

When the results, as shown in Table III, are examined the most suitable structure is the neural network of 20 neurons. The errors in this table are computed from the normalized output. The system outputs are the coordinates of the green and red colored balls on the experimental setup. Estimation errors of the coordinates can be calculated from the normalized values given in (14).

$$x_{in} = (x_{max} - x_{min})x_{norm} + x_{min} \quad (14)$$

Coordinates predicted by neural networks are shown by the symbol 'o' in Figure 19. Symbol '*' shows the target output.

Figure 20. Predicted coordinates of the colored balls

Prediction errors for green and red balls are given in Figure 20 and Figure 21, respectively.

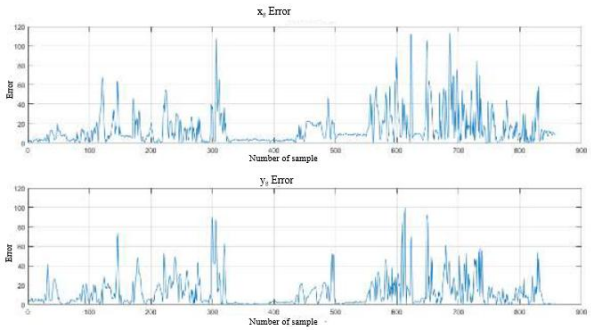

Figure 21. Prediction error of green colored ball

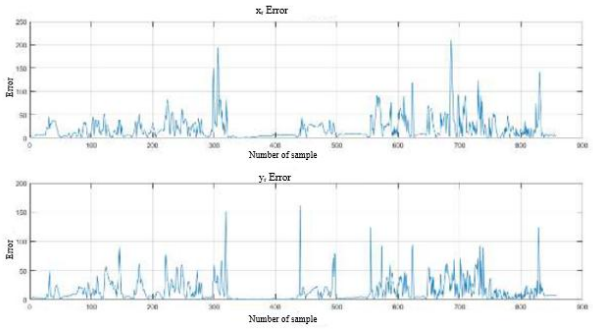

Figure 22. Prediction error of red colored ball

The prediction errors are given in terms of pixels. Table IV shows the mean error values.

TABLE IV. MEAN ERRORS OF PREDICTION

Coordinate	Mean Error	Standard Deviation
x_g	14.378	17.31
y_g	12.189	15.36
x_r	19.909	24.56
y_r	14.590	19.24

VI. CONCLUSION

In this study, we studied on the measurement of the orientation with the custom designed sensors. The generated dataset was used to train a single layer feed forward ANN model. As a result of the reviews, the most suitable model is the one with 20 neurons. The designed model worked at an error rate of 7.049e-3 in the test set.

Errors were suddenly elevated in some places due to noises. Image processing errors, calibration errors of the sensors and the rapid movement of the balls are main reasons of this noisy data. Fast movements in the setup make it difficult to match the images with the sensor data. Despite this, the experiment shows that the proposed sensors can be used for orientation measurements in multi-axis movements.

The increase in the number of the presented sensors will reduce the error rate considerably. It is aimed to improve system performance by the use of different infrared transceiver pairs in the future studies. In addition, the use of materials (flex sensor, velostad, etc.) that change the resistance characteristics depending on flexibility will be studied. The spherical joint angle measurement system developed in this study will also be used to detect shoulder motion of the human operator in our humanoid robot studies.

ACKNOWLEDGMENT

This study is supported by The Scientific and Technological Research Council of Turkey (TUBITAK). (Project number: 2150023).

REFERENCES

[1] A. S. Morris and R. Langari, "Chapter 20 Rotational Motion Transducer," in *Measurement and Instrumentation: Theory and Application*, Academic Press, 2012, pp. 529-560.

[2] Y. Wang, W. Chen and M. Tomizuka, "Extended Kalman Filtering for Robot Joint Angle Estimation Using MEMS Inertial Sensors," *IFAC Proceedings Volumes*, vol. 46, no. 5, pp. 406-413, 2013.

[3] V. Vikas and C. D. Crane, "Joint Angle Measurement Using Strategically Placed Accelerometers and Gyroscope," *ASME Journal of Mechanisms and Robotics*, vol. 8, no. 2, pp. 021003-021003-7, 2015.

[4] G. Saggio, "A novel array of flex sensors for a goniometric glove," *Sensors and Actuators A: Physical*, vol. 205, pp. 119-125, 2014.

[5] K. Ghorbanian and M. Gholamrezaei, "An artificial neural network approach to compressor performance prediction," *Journal of Applied*, vol. 86, no. 7-8, pp. 1210-1221, 2009..

[6] S. M. J. Pappu and S. N. Gummadi, "Artificial neural network and regression coupled genetic algorithm to optimize parameters for enhanced xylitol production by Debaryomyces nepalensis in bioreactor," *Biochemical Engineering Journal*, vol. 120, pp. 136-145, 2017.

[7] [Online].Available:http://dba.med.sc.edu/price/irf/Adobe_tg/models/cielab.html. [Accessed 27 7 2017].

[8] T. M. Mitchell, Machine Learning, McGraw-Hill Science/Engineering/Math, 1997.

[9] M. Aşıkgil, "Conceptual Quantity Modeling of Single Span Highway Bridges by Regression, Neural Networks and Case Based Reasoning Methods," 2012.

[10] A. Weigend, "On overfitting and the effective number of hidden units," *Proceedings of the 1993 connectionist models summer school*, vol. 1, pp. 335-342, 1994.

[11] M. W. Spong, S. Hutchinson and M. Vidyasagar, Robot Dynamics and Control, New York: Wiley, 2004.

Kemal Guven received B.S. and M.S. degrees in mechanical engineering from Baskent University, Turkey, in 2014 and in 2017. He is currently pursuing the Ph.D. degree in mechanical engineering. He is also a research assistant in the Department of Mechanical Engineering, Baskent University, Turkey. His research interests include control theory, machine learning, artificial intelligence and robotics.

Andac Tore Samiloglu received B.S., M.S. and Ph.D. degrees from Middle East Technical University, Turkey, in 2003, 2007 and 2012, respectively. He is currently an Associate Professor with Baskent University. His research interests include control theory, robotics, mechatronics, automotive and multi agent systems.