Downloaded from orbit.dtu.dk on: Mar 29, 2024



DTU Library

Hand-Eye Calibration and Inverse Kinematics of Robot Arm using Neural Network

Wu, Haiyan; Tizzano, Walter; Andersen, Thomas Timm; Andersen, Nils Axel; Ravn, Ole

Published in:
Proceedings of RITA 2013

Link to article, DOI:
10.1007/978-3-319-05582-4_50

Publication date:
2013

Link back to DTU Orbit

Citation (APA):
Wu, H., Tizzano, W., Andersen, T. T., Andersen, N. A., & Ravn, O. (2013). Hand-Eye Calibration and Inverse Kinematics of Robot Arm using Neural Network. In *Proceedings of RITA 2013* (pp. 581-591). Springer. https://doi.org/10.1007/978-3-319-05582-4_50

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Hand-Eye Calibration and Inverse Kinematics of Robot Arm using Neural Network

Haiyan Wu, Walter Tizzano, Thomas Timm Andersen,
Nils Axel Andersen, and Ole Ravn

Automation and Control, Department of Electrical Engineering,
Technical University of Denmark, Elektrovej, 2800, Kgs. Lyngby
{hwua,ttan,naa,or}@elektro.dtu.dk, waltertizzano@gmail.com

Abstract. Traditional technologies for solving hand-eye calibration and inverse kinematics are cumbersome and time consuming due to the high nonlinearity in the models. An alternative to the traditional approaches is the artificial neural network inspired by the remarkable abilities of the animals in different tasks. This paper describes the theory and implementation of neural networks for hand-eye calibration and inverse kinematics of a six degrees of freedom robot arm equipped with a stereo vision system. The feedforward neural network and the network training with error propagation algorithm are applied. The proposed approaches are validated in experiments. The results indicate that the hand-eye calibration with simple neural network outperforms the conventional method. Meanwhile, the neural network exhibits a promising performance in solving inverse kinematics.

Keywords: Neural Network, Calibration, Inverse Kinematics, Robot Arm

1 Introduction

Applying robot arm for dexterous tasks, e.g. object gasping and hanging, challenges both machine vision and control fields. These tasks often requires high-speed sensor feedback as well as fast and precise control of robot arm. Thanks to the advances of camera technology and image processing methodology, visual information has been widely used in various tasks, e.g. for object recognition, 3D localization, scene reconstruction, motion detection and so on [1–3]. In order to utilize the visual feedback in the control loop, it often requires a calibration between the vision system and the robot arm, e.g. for a 'eye to hand' setup, see Fig.1. The hand-eye calibration gives a mapping of 2D image coordinates $[u, v]$ to 3D Cartesian coordinates $[x, y, z]$. Moreover, the control tasks with robot arm are usually performed in Cartesian space with visual feedback providing 3D object information. However, the robot arm is manipulated in joint space. Therefore, it involves a mapping from Cartesian space $[x, y, z]$ to joint space $[q_1, q_2, \ldots, q_n]$. This paper focuses on these two fundamental issues

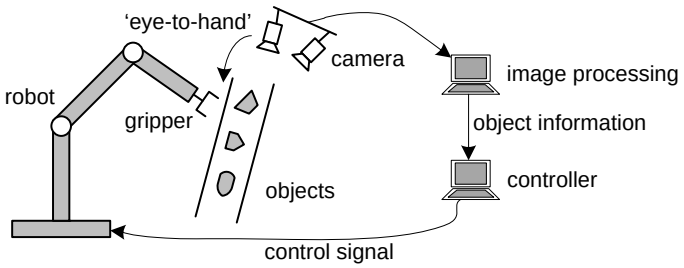


Fig. 1. Eye-to-hand setup.

in robot arm control, namely the hand-eye calibration issue and the inverse kinematics issue, which will be studied in a 6-DOF (degrees of freedom) robot arm control system equipped with a stereo vision system.

The hand-eye calibration problem has fascinated many researchers over the last 30 years. It is traditionally dealt with through triangulation methods [4–6]. However, the triangulation method usually requires camera calibration, calibration from camera frame to gripper frame or robot base frame, and stereo triangulation (if a stereo vision system is applied). Therefore, the calibration procedure is cumbersome and time consuming. Besides, the triangulation involves camera intrinsic and extrinsic parameters. As a result, the performance of the traditional calibration method is susceptible to these parameters. Regarding the inverse kinematics problem, there are several techniques to simplify it and they can be in general grouped into two categories: analytical solution and geometrical solution [7]. For manipulators with higher DOF or with non-spherical wrist, the inverse kinematics problem becomes complex due to the high nonlinearities in the kinematics model, and thus, it is difficult to find a closed-form solution. Inspired by animals' noteworthy abilities in a very wide range of different tasks with their remarkable analogous in biology, the artificial counterpart-neural network is used in robotics for several different purposes [9–14]. In [9] neural network estimators are trained offline for non-linear state estimation assuming non-Gaussian distributions of the states and the disturbances; neural networks can also be used to solve the camera calibration and 3D reconstruction problems [10,11]; the fast inverse kinematics of a 3-DOF planar redundant robot is achieved through neural network in [13]; in order to overcome the problems of singularities and uncertainties in arm configurations, artificial neural network is applied for the motion control of a 6-DOF robot manipulator in [14].

The benefits of neural network with regard to low computational cost and high efficiency inspires neural network based solutions for hand-eye calibration and inverse kinematics of a 6-DOF robot arm presented in this paper. The main purpose of this work is to implement the neural network for a real robot arm equipped with a stereo vision system. Different from some existing works, the robot arm used in this work has an offset wrist which poses difficulties for

solving inverse kinematics in the traditional analytical or geometrical way. Data samples from real robot arm is utilized for performance analysis. Furthermore, this work is implemented as an initial step towards a realtime vision-guided robotic manipulation system based on neural network.

The remainder of this paper is organized as follows: The overall platform consisting of a robot arm and a stereo vision system is described in Section 2. In Section 3, the feedforward neural network is briefly introduced. Then, the training of neural networks for calibration and inverse kinematics is presented. In Section 4, the experimental validation and performance comparison are discussed. Finally, conclusions are given in Section 5.

2 System Overview

As mentioned in the previous section, in the present work the neural network is applied for a 6-DOF robot arm ('hand') with a stereo vision system ('eye'). In this section, these two hardware components will be introduced.

2.1 6-DOF robot arm

Universal robot UR5 is a 6-DOF robot arm with relatively lightweight (18.4 kg), see Fig. 2. It consists of six revolute joints which allows a sphere workspace



Fig. 2. UR5 robot and its joint coordinate system. *Left*: UR5 robot and PolyScope GUI [15]; *Right*: joint coordinate system.

with a diameter of approximately 170 cm. The Movements close to its boundary should be avoided considering the singularity of the arm. The arm is equipped with a graphic interface PolyScope, which allows users to move the robot in a

user-friendly environment through a touch screen. In this work, the connection to the robot controller is realized at script level using TCP/IP socket. Once the TCP/IP connection is established, the robot broadcasts data packets with a rate of 125 Hz.

The UR5 robot arm has a non-spherical wrist, as shown in Fig. 2. The D-H (Denavit-Hartenberg) parameters of the robot arm is given in Table 1. With the D-H parameters the forward kinematics can be calculated through the

Table 1. Denavit-Hartenberg parameters of UR5

Joint	a_i [m]	α_i [rad]	d_i [m]	θ_i [rad]
1	0	$\pi/2$	0.089	q_1
2	-0.425	0	0	q_2
3	-0.392	0	0	q_3
4	0	$\pi/2$	0.109	q_4
5	0	$-\pi/2$	0.095	q_5
6	0	0	0.082	q_6

equation given below

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5, \tag{1}$$

where T_j^i denotes the homogeneous transformation from frame i to frame j, and

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad i = 1, 2, \ldots, 6.$$

2.2 Stereo vision system

The stereo vision system consists of two Guppy Cameras F-036. Each camera has a resolution of 752×480 pixels and runs at a frame rate of 64 fps. Camera calibration is required to determine the intrinsic parameters of each camera, including focal length, principle point, skew coefficients and distortion coefficients. In this work the skew coefficients are not calculated as the pixels are assumed to be rectangular. The calibration is carried out based on Camera Calibration Toolbox in Matlab [16]. The calibration results are given in Table 2. These intrinsic parameters will be applied in geometric stereo triangulation in Section 4.1 for performance comparison.

Table 2. Intrinsic parameters of left and right cameras.

Parameter	Left camera	Right camera
Focal length [pixel]	[683.89±2.49, 685.71±2.51]	[674.03±2.44, 676.18±2.55]
Principle point [pixel]	[307.76±3.26, 236.63±3.08]	[327.05±3.78, 247.40±2.88]
Distortion coef.	[-0.24±0.010, 0.222±0.046]	[-0.210±0.008, 0.168±0.034]

3 Neural network for a 6-DoF robot arm

3.1 Feedforward neural network

According to the different ways of arranging neurons in a network, there are several different architectures of neural network, e.g. *feedforward neural network*, *radial basis function network*, *recurrent neural network*, etc. [8,17]. In this section, the feedforward neural network is introduced.

In feedforward neural network all the neurons are divided into layers, as shown in Fig.3. The neurons in one layer are not connected with each other.

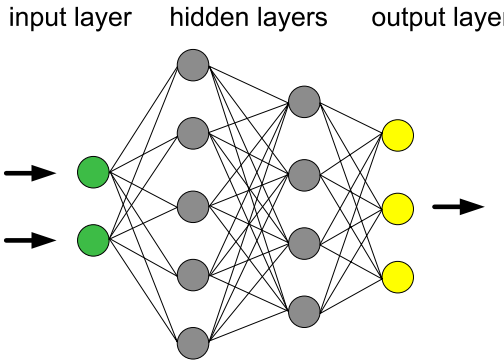


Fig. 3. An example of feedforward neural network.

Moreover, each neuron in a layer receives inputs from all the neurons of the previous layer and has an output connected to all the neurons of the following layer. The first layer is called *input layer* which receives input from external environment, while the last layer is called *output layer* which transmits the results of the network. The layers in between are called *hidden layers* which are invisible to the external environment. The number of neurons in the input/output layer is related with the number of variables that are dealt with. The number of neurons in hidden layers is decided by the complexity of the problem. Generally speaking, a more complex problem requires more hidden layers and neurons. In most cases, one hidden layer will be sufficient [18].

In order to determine the connection weights among the neurons, a network training strategy needs to be carefully selected. In this work, the well-known *error back propagation* algorithm is applied. The back propagation algorithm is a gradient descent procedure that finds the set of weights minimizing the output error [19].

3.2 Training of neural network for hand-eye calibration

In hand-eye calibration, the 2D information is used to retrieve the 3D information with regard to a reference frame. In this work, the aim of hand-eye calibration is to determine the 3D location of an object in the robot base frame, based on

the 2D coordinates in two cameras. In order to build the neural network, the network training is required to set the weights of the connections between the neurons. Therefore, a training sample is demanded which is acquired through the approach described below:

1. First, attach the object to the end-effector of the robot arm;
2. Move the robot arm to random positions within its workspace;
3. Detect the coordinates of the object in image space;
4. Finally, save both the 3D information and 2D information in a file.

With the collected sample data the network is trained based on error back propagation algorithm. After the training, the neural network for hand-eye calibration is built, which transforms 2D image coordinates $[u, v]$ to 3D Cartesian coordinates $[x, y, z]$. Then, the performance of the neural network is evaluated with another set of samples. The *cumulative distribution function* (CDF) and *root mean square error* (RMSE) are selected for results analysis. The CDF describes the probability of a variable X being less than or equal to x

$$CDF_X(x) = P(X \leq x). \tag{2}$$

The RMSE is often used as a measure of the difference between the predicted values and the observed values, defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\text{obs}_i - \text{pre}_i)^2}{n}}, \tag{3}$$

where obs_i is observed value, pre_i is predicted value at time instant i, and n is the number of observed/predicted data.

3.3 Training of neural network for inverse kinematics

Through hand-eye calibration, the object position is determined with regard to a reference base frame, e.g. the robot base frame. For tasks such as object tracking and grasping the inverse kinematics is required to convert the 3D Cartesian coordinates $[x, y, z]$ to joint coordinates $[q_1, q_2, \ldots, q_n]$. It has to be mentioned that in this work the orientation of the object $[\alpha, \beta, \gamma]$ is also considered for inverse kinematics. The orientation information can be obtained through several methods, e.g. through image feature matching algorithm utilized in [20] and particle filter based on one position sensor and one IMU described in [21]. Therefore, the neural network for inverse kinematics has six inputs $[x, y, z, \alpha, \beta, \gamma]$. For a 6-DOF robot-arm, the output of the network is the corresponding six joint values $[q_1, q_2, \ldots, q_6]$.

Similar to the neural network training in Section 3.2, a set of sample data is required. It is obtained through the following approach:

1. Move the end-effector of the robot arm to random positions and orientations within its workspace;
2. Save the joint angles, end-effector positions and end-effector orientations.

The step 1 and step 2 are repeated several times in order to better approximate the uniform distribution of the sample data points. Furthermore, another sample of data is needed for performance evaluation. The CDF and RMSE defined by Eqs. (2)(3) are also applied for results analysis.

4 Experiments

In this part, experiments are carried out to evaluate the performances of the neural networks for hand-eye calibration and inverse kinematics. The experimental setup is shown in Fig. 4. A table-tennis ball is chosen as the object,

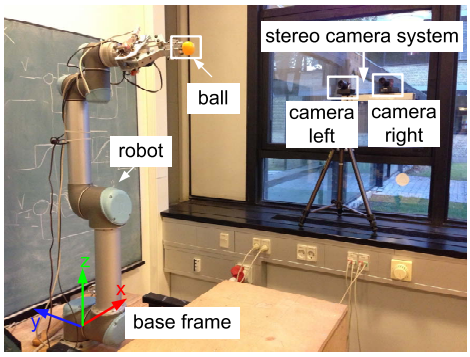
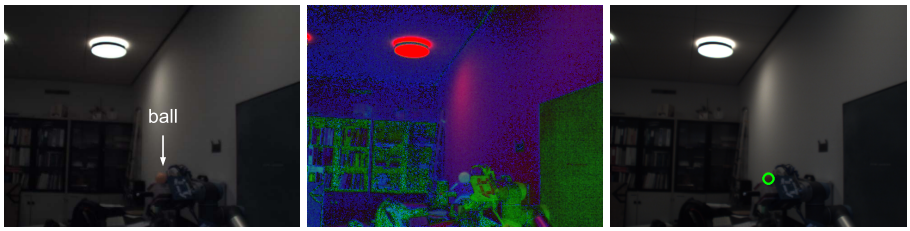


Fig. 4. Experimental setup.

which is mounted on the end-effector of the robot arm. Two Guppy Cameras F-036 are used to build the stereo vision system. Each camera works at 30 fps@640×480 pixels. Before stereo triangulation, image processing algorithm is applied to extract the ball in the 2D image space. First, frames from each camera are converted from RGB colour space to HSV colour space, see Fig. 5(a)(b). Then, the coordinate of the ball is determined based on the colour, contour and



(a) (b) (c)

Fig. 5. Image processing for determining the 2D coordinate of a ball in the image space. (a): RGB image; (b) HSV image converted from RGB image; (c): ball detected (highlighted by green-circle) in image space.

radius information. Fig. 5(c) shows an example of the image processing result.

After the image processing, the ball positions in both images from left and right cameras are found and saved to a file for neural network training.

4.1 Experiment I: Neural Network for hand-eye calibration

In this experiment, neural network is trained for hand-eye calibration, as introduced in Section 3.2. The ball was randomly moved to 2000 different positions by the robot arm. Then, the 2D ball coordinates $[u_l, v_l]$, $[u_r, v_r]$ in the left and right cameras were saved to a file. Besides, the 3D ball position $[x, y, z]$ with regard to the robot base frame was also recorded. The network was trained using 80% of the sample, while the remaining 20% was utilized to test the performance.

A network with 4 input neurons, 3 output neurons and 2 hidden layers with 8 neurons each was heuristically selected. The results are in shown in Fig. 6, which gives the CDF (*left*) and histogram (*right*) of the RMSE and the mean absolute error $mean(|x|, |y|, |z|)$ along different axis. Here, the RMSE describes the absolute distance between the measured 3D ball position and the predicted 3D ball position. The results are also summarized in Table 3. It is noticed that

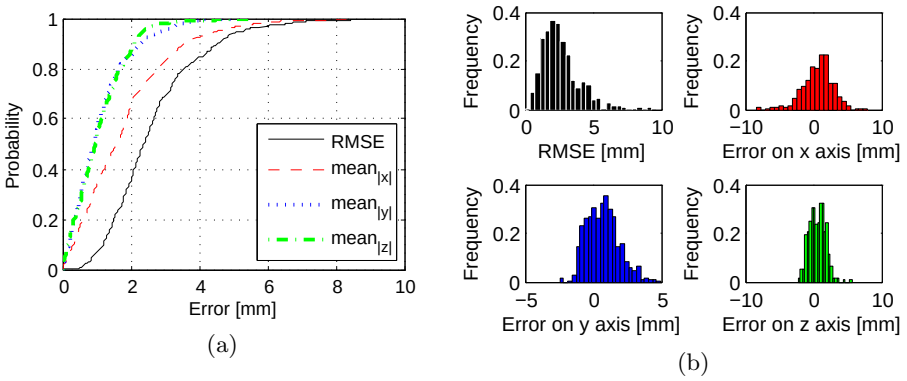


Fig. 6. Cumulative distribution (left) and histogram (right) of the calibration error based on the neural network approach. The neural network has 4 input neurons, 3 output neurons and 2 hidden layers with 8 neurons each.

the error is not homogeneously distributed along the different directions: the error on the x-direction is considerably bigger. The mean absolute error on the x-axis is $mean_{|x|} = 1.81$ mm. It is due to the fact that the x-direction of the base frame is roughly parallel to the optical axis of the cameras (see Fig. 4), which is more susceptible to the estimation uncertainty.

Moreover, the performance of the neural network based approach is compared with the one of the standard geometrical triangulation approach. As shown in table 3, the hand-eye calibration error is reduced dramatically by the proposed approach. For example, the RMSE of the proposed approach is 2.65 mm, while that of the geometrical approach is 19.10 mm. In addition, a deduction of approximate 80% of the mean absolute error on axis x, y, z is gained through the

Table 3. Results of the neural network for hand-eye calibration and against that of the geometrical triangulation.

Parameter	Neural Network	Geometric Triang.
RMSE	2.65 (−86.1%)	19.10
95th percentile	5.29	32.42
99.5th percentile	8.44	41.41
Mean absolute error $[\|x\|, \|y\|, \|z\|]$	[1.81, 1.10, 1.07] (−86.7%, −88.0%, −78.8%)	[13.56, 9.193, 5.039]
95th percentile $[\|x\|, \|y\|, \|z\|]$	[4.49, 3.01, 2.38]	[27.46, 21.74, 11.61]
99.5th percentile $[\|x\|, \|y\|, \|z\|]$	[8.02, 4.20, 4.56]	[38.70, 25.75, 16.09]

*All the values are expressed in [mm].

neural network approach. Based on the experimental results it is concluded that hand-eye calibration with the neural network can achieve a good performance and thus, can be implemented for different applications such as object tracking and grasping.

4.2 Experiment II: Neural Network for inverse kinematics

In this experiment, the neural network is trained for solving inverse kinematics as proposed in Section 3.3. The robot arm is moved randomly in its workspace. A sample of data is recorded, including the six joint values $[q_1, q_2, \ldots, q_6]$ and the position and the orientation of the end-effector $[x, y, z, \alpha, \beta, \gamma]$. Similarly to the experiment in Section 4.1, 80% of the sample data is used for training the neural network, while the remaining 20% is utilized for performance evaluation.

A network with 6 input neurons, 6 output neurons and 1 hidden layer with 12 neurons was trained to solve the inverse kinematics problem. The experimental results are shown in Fig. 7 and Table 4. The maximum mean error of the joints is 8.57×10^{-3} rad with a standard deviation of 10.95×10^{-3} rad. In general the error is smaller than 0.04 rad. It is also noticed that the neural network gives comparatively more imprecise results on the last joint, which needs to be further investigated in the future.

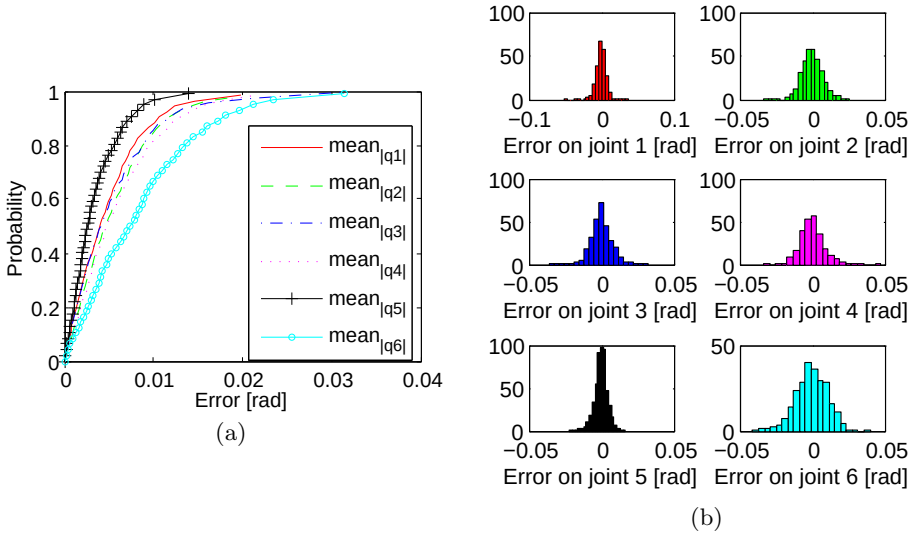
Table 4. Results of the neural network for inverse kinematics.

Joint	1	2	3	4	5	6
RMSE	5.11	5.73	5.50	6.16	3.42	8.57
Std. deviation	7.02	7.54	7.63	8.14	4.53	10.95
95th percentile	13.03	14.46	14.29	15.95	9.01	21.29
99.5th percentile	28.18	27.29	30.91	29.99	15.34	36.73

*All the values are expressed in [10^{-3} rad].

5 Conclusion

In order to overcome the high complexity and high computational cost of traditional technologies, the neural networks are applied for hand-eye calibration and



(a) (b)

Fig. 7. Cumulative distribution (left) and histogram (right) of the error of inverse kinematics based on the neural network approach. The neural network has six input neurons, six output neurons and one hidden layer with 12 neurons.

inverse kinematics in this paper. The feedforward neural network is selected and trained through the well-known error back propagation algorithm. The hand-eye calibration and inverse kinematics problems solved through neural network are investigated in a 6-DOF robot-arm control system equipped with a stereo vision system. The neural network is trained to transform the 2D image coordinates from the stereo vision system to the 3D Cartesian coordinates. Furthermore, the 3D coordinates combined with orientation information is converted to joint coordinates through inverse kinematics based also on neural network.

The experimental results indicate a significant improvement (a 80% deduction in the mean error) in the performance of the hand-eye calibration compared to the traditional method. A promising performance of inverse kinematics (mean error smaller than 9×10^{-3} rad) is achieved through neural network based approach. The proposed approaches can be directly extended to different hand-eye setups and robot-arms.

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[2] H. Wu, K. Zou, T. Zhang, A. Borst, and K. Kühnlenz, "Insect-inspired high-speed motion vision system for robot control," *Biological cybernetics*, vol. 106, no. 8-9, pp. 453–463, 2012.

[3] J. Smisek, M. Jancosek, and T. Pajdla, "3d with kinect," in *Consumer Depth Cameras for Computer Vision*, pp. 3–25, Springer, 2013.

[4] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.
[5] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.
[6] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *the Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4647–4653, IEEE, 2006.
[7] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.
[8] M. Norgaard, *Neural networks for modelling and control of dynamic systems: A practitioner's handbook*. Springer, 2000.
[9] E. Bayramoglu, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Pre-trained neural networks used for non-linear state estimation," in *the Proceedings of the 10th International Conference on Machine Learning and Applications and Workshops (ICMLA)*, vol. 1, pp. 304–310, IEEE, 2011.
[10] Q. Memon and S. Khan, "Camera calibration and three-dimensional world reconstruction of stereo-vision using neural networks," *International Journal of Systems Science*, vol. 32, no. 9, pp. 1155–1159, 2001.
[11] M. T. Ahmed, E. E. Hemayed, and A. A. Farag, "Neurocalibration: a neural network that can tell camera calibration parameters," in *the Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 463–468, IEEE, 1999.
[12] S. Tejomurtula and S. Kak, "Inverse kinematics in robotics using neural networks," *Information Sciences*, vol. 116, no. 2, pp. 147–164, 1999.
[13] R. V. Mayorga and P. Sanongboon, "Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: An artificial neural network approach," *Robotics and Autonomous Systems*, vol. 53, no. 3, pp. 164–176, 2005.
[14] A. T. Hasan, N. Ismail, A. M. S. Hamouda, I. Aris, M. H. Marhaban, and H. Al-Assadi, "Artificial neural network-based kinematics jacobian solution for serial manipulator passing through singular configurations," *Advances in Engineering Software*, vol. 41, no. 2, pp. 359–367, 2010.
[15] *http://www.universal-robots.com/*.
[16] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.
[17] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
[18] J. Heaton, *Introduction to neural networks with Java*, vol. 200. Heaton Research St Louis, 2005.
[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 1, p. 213, 2002.
[20] H. Wu, L. Lou, C.-C. Chen, S. Hirche, and K. Kolja, "Cloud-based networked visual servo control," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 554–566, 2012.
[21] S.-h. Won, W. W. Melek, F. Golnaraghi, *et al.*, "A kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1787–1798, 2010.