Configuration Space Control of a Parallel Delta Robot with a Neural Network Based Inverse Kinematics

Tarik Uzunovic[1], Edin Golubovic[2], Eray A. Baran[3], and Asif Sabanovic[4]

[1, 2, 3, 4]Sabanci University, Istanbul, Turkey
{tuzunovic, edin, eraybaran, asif}@sabanciuniv.edu

Abstract

This paper describes configuration space control of a Delta robot with a neural network based kinematics. Mathematical model of the kinematics for parallel Delta robot used for manipulation purposes in microfactory was validated, and experiments showed that this model is not describing "real" kinematics properly. Therefore a new solution for kinematics mapping had to be investigated. Solution was found in neural network utilization, and it was used to model robot's inverse kinematics. It showed significantly better mapping between task space coordinates and configuration (joint) space coordinates than the mathematical model, for the workspace of interest. Consequently positioning accuracy improvement is expected. Neural network is then used as a part of the control system. Applied control strategy was configuration space acceleration control with disturbance observer.

1. Introduction

In the recent period micromanufacturing is attracting significant interest from researchers around the world. Devices that provide high precision accuracies are one of the crucial parts in production of miniaturized parts and devices. Therefore different robotic manipulators have been investigated for utilization in microproduction applications.

Generally robotic manipulators are divided in two main groups, being serial and parallel manipulators. When these two groups are compared, parallel manipulators provide higher stiffness, higher accuracies, and higher payload-to-weight ratio. Main disadvantage of the parallel mechanisms is limited workspace. However in micromanufacturing area this is usually not one of the key requirements, thus parallel robots are being more frequently used.

Due to their utilization capabilities in high precision applications, parallel robots have been investigated a lot in the past. One of the widely used types of these mechanisms is famous Delta robot, introduced in the late 1980's by Clavel [1]. This is 3-DOF fully parallel robot designed especially for high speed applications. Delta robot design is being very intensively used in the production, as many companies like ABB [2], Festo [3], Bosch [4] are offering manipulators based on this design.

Mechanism that is being analyzed in this paper is a miniaturized Delta robot designed for positioning purposes in the microfactory [5]. Experimental measurements were done to validate conventional kinematics model, and they showed that this mathematical model of the mapping between end effector Cartesian coordinates (task space coordinates) and active joint positions (configuration space coordinates), taken from [1], is not describing properly this relationship for the real mechanism. Coordinate transformation between configuration space and task space (and vice versa) is known as direct (inverse) kinematics. There are few reasons for this mismatch between mathematically derived kinematics and "real" kinematics of the Delta robot. Some of them are manufacturing tolerances, assembly errors, wear and transmission errors.

In order to keep mechanism usable for manipulation in microfactory it was necessary to find new kinematics model. Chosen solution was application of the feedforward neural network (NN).

There are multiple examples in the literature where neural networks were used in robots kinematics derivation. In [6] artificial neural network was used to model difference between actual inverse kinematics of the parallel manipulator and computationally obtained one. Neural network was actually trained to model kinematics error. Neural network based inverse kinematics for a 2-DOF planar robotic arm was introduced in [7]. Authors used Kohonen's self-organizing mapping algorithm which uses Widrow-Hoff type error correction rule. Multilayer backpropagation network was proposed in [8] to model inverse kinematics of a 6R robot manipulator with offset wrist, for which closed form solution does not exist. Widrow-Hoff neural NN, with adaptive learning algorithm derived using Lyapunov stability theory, was used in [9] to obtain kinematics inversion of redundant manipulator. As this inverse kinematics has infinite number of solutions, additional fuzzy neural network was used as hint generator to limit searching space and to improve quality of the solution. Some physical constraints, like joint velocity limits and joint angle limits, are incorporated into this approach. In [10] radial basis function NN modeled inverse kinematics of a 7-DOF manipulator. Genetic algorithm was used to provide sample data for network training. NN usage in kinematics modeling was also discussed in [11-13]. In [14, 15] authors were using neural networks for kinematic calibration of parallel manipulators.

In this paper feedforward NN was used to model Delta robot's "real kinematics". Experimental setup was made to provide necessary set of data for network training. Backpropagation algorithm is then used to train the network. Its mapping performance is compared with the one of mathematical model of the inverse kinematics. Comparison results justified NN utilization in kinematics modeling. Neural network based kinematics model is then used as a part of a novel positioning control system for a Delta robot.

This paper is organized as follows. Delta robot structure and operation principle are briefly presented in Section 2. Section 3 explains undertaken kinematics validation and neural network design. Control system structure is given in Section 4. Section 5 brings experimental results of the control performance, and finally conclusion and future work are provided.

2. Delta Robot Description

Delta robot was invented in the 1980's by Reymond Clavel [1]. Its main purpose was to manipulate light and small objects on very high speeds. Main limitation of this mechanism was small workspace and disability to operate with heavy objects. If micromanufacturing applications are considered these disadvantages actually disappear. Therefore parallel Delta robot was chosen as the manipulation mechanism for the microfactory application [16]. Delta robot that is used in this work is presented in [5] in detail and it is shown in Fig. 1.

Delta robot consists of the fixed base, travelling plate and three kinematic chains that connect fixed base and travelling plate which is actually robot's end effector. Each kinematic chain consists of upper arm that is actuated by revolute brushless DC servomotor, and lower arm. Lower arms have formation of parallelogram formed by the links and spherical joints. Lower arms transmit motion from actuated upper links to the moving plate. Their parallelogram formation ensures parallelism between fixed base and travelling plate, enforcing the necessary mechanical constraints to confine the motion into 3D space.

Kinematics of the Delta robot is derived according to parameters and procedure presented in [1]. Kinematic parameters shown in Fig. 2 had these values: L_A = 40 (mm), L_B = 68 (mm), R_A = 40 (mm), R_B = 30 (mm). Design of the Delta robot was initially optimized to cover workspace of 40 (mm) cube, but in the current application 10 (mm) cube is of the interest.

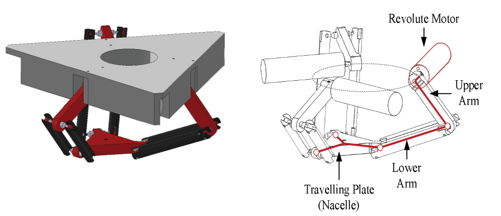

Fig. 1. Delta robot structure

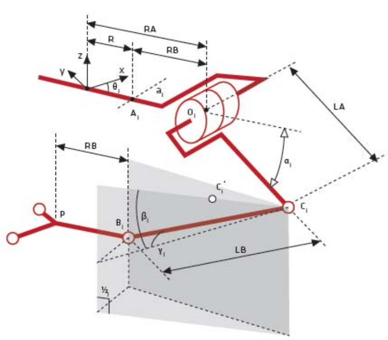

Fig. 2. Delta robot kinematic model

3. Kinematics Validation and NN Design

3.1. Kinematics Validation

Decision was made to validate kinematics mapping of the Delta robot. Therefore experimental setup was build to collect necessary measurements. Inverse kinematics was of the particular interest, as it is used in robot control.

Experimental setup was made for proper measurement. Delta robot end effector was connected to the end effector of a high resolution XYZ stage. Stage position is measured with three 6.87 (nm) resolution encoders. This measurement error is negligible for discussed application. The XYZ stage is mounted on the rotational stage that is used for axes alignment between XYZ stage and Delta robot.

Delta robot was driven in its workspace by the XYZ stage, and angles between its base and upper arms (they will be called joint angles) and Cartesian coordinates of its effector were measured. At the beginning of the experiment Delta robot was positioned so that all joint angles were equal 30 (degrees) and then it was moved along its workspace by the stage. Measured data are used to validate used kinematics of the Delta robot.

Position of the end effector was calculated from the joint angles using forward kinematics, and it was compared with the actual position measured by the XYZ positioning stage, that was moved in 10 (mm) cube along the trajectory given in Fig. 3. Totally 13333 samples were recorded along this trajectory. Positioning stage was controlled to track the reference

$$\begin{aligned} x^{ref}(t) &= -0.005\sin(2\pi\cdot 0.0075t)(\mathrm{m}), \\ y^{ref}(t) &= 0.005\sin(2\pi\cdot 0.03t)(\mathrm{m}), \\ z^{ref}(t) &= -0.07084 - t\cdot 0.01/1333.3(\mathrm{m}). \end{aligned} \tag{1}$$

From the collected joint angles data positions of the Delta robot were calculated according to forward kinematics, and they are compared to the measured ones. Significant deviation was shown, as it can be seen from the Fig. 4, especially on the workspace borders in *x-y* plane. Only part of the measured results is shown in these figures for the better view. Error is the difference between value calculated by direct kinematics and the measured one. For the *x*-coordinate error range was from -695.9 (μm) to 873.05 (μm), for the *y*-coordinate -881.7 (μm) to 1.16 (mm), while range in the *z*-direction was 29.7 (μm) to 826.27 (μm). Error is not shown in the *z*-coordinate graph as it would not be visible properly. Errors are intentionally shown for the forward kinematics to demonstrate how big positioning errors can occur as the consequence of the wrong kinematics mapping.

It is obvious that direct kinematics mathematical model does not represent "real direct kinematics" of the mechanism, and inverse kinematics model (as its functional inversion) is not describing "real inverse kinematics". In order to have satisfactory positioning, better model of the inverse kinematics had to be found, and neural network was chosen to model it, as it will be used as a part of the control system.

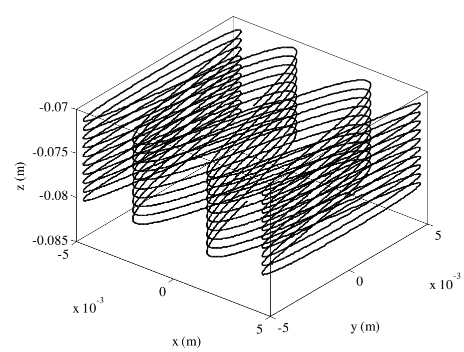

Fig. 3. Trajectory for kinematics validation

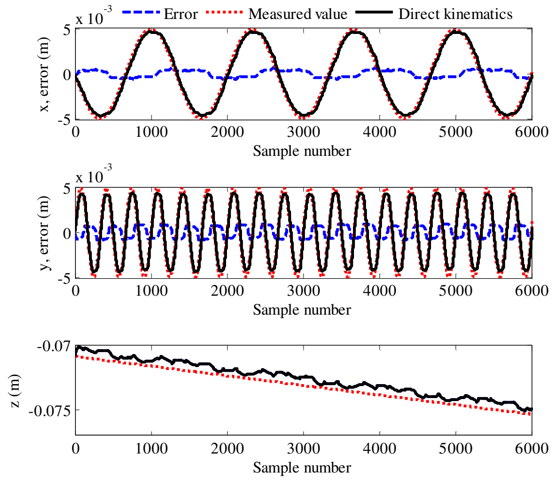

Fig. 4. Kinematics validation results

3.2. NN Design

Feedforward neural network with input layer of 3 neurons, one hidden layer with 25 tansig neurons, and output layer with 3 purelin neurons was trained to model "real inverse kinematics" of the Delta robot. Experimental data collected during kinematics validation were used for its training. Backpropagation Levenberg-Marquardt method was applied. Recorded Cartesian positions of the end effector were presented to the network input, while recorded joint angles were target output. Network was trained to minimize performance function

$$F=\sum_{i=1}^{N}e_i^2=\sum_{i=1}^{N}\left(t_i-a_i\right)^2, \tag{2}$$

where t is target output, a is network output and N is the total number of recorded output data, including all network outputs. Trained NN output was compared with the output of the mathematical model of the inverse kinematics for the recorded data (Fig. 5). For the better visualization results are shown for only one part of the recorded data. Joint angles are marked as α_{1s}, α_{2s}, and α_{3s}.

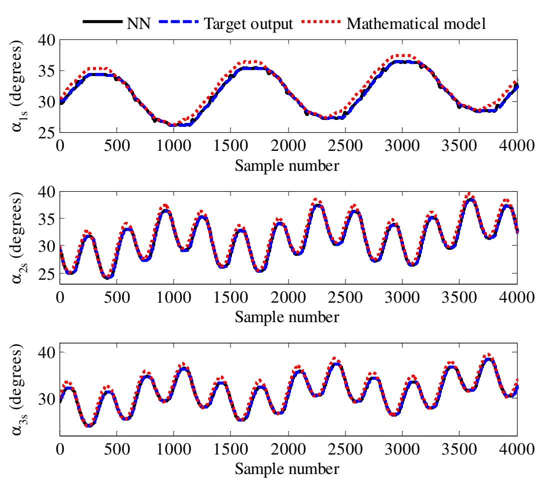

Fig. 5. NN vs. mathematical model of inverse kinematics

It is obvious from the Fig. 5 that NN provides better mapping with the experimental data. In order to prove the improvement mean square error (MSE) and mean absolute value (MAE) were calculated for the NN and mathematical model of the inverse kinematics, for all data from the recorded set, and for each joint. Results are shown in Table 1. From these results and responses given from Fig. 1 it is obvious that improvement is significant, and the least one is for the first joint. Probably manufacturing and tear problems are the biggest for this joint. From these results it is expected that created NN will improve positioning, so it is used in robot positioning control.

Table 1. NN vs. mathematical model of inverse kinematics

	Mathematical model		NN	
	RMSE (degrees)	MAE (degrees)	RMSE (degrees)	MAE (degrees)
α_{1s}	0.8270	0.6593	0.1655	0.0969
α_{2s}	0.8845	0.7142	0.0616	0.0476
α_{3s}	0.8083	0.6255	0.0576	0.0431

4. Control System Design

Dynamical model of the Delta robot can be described with the following equation

$$\mathbf{M}(\boldsymbol{\alpha}_r)\ddot{\boldsymbol{\alpha}}_r+\mathbf{b}(\boldsymbol{\alpha}_r,\dot{\boldsymbol{\alpha}}_r)+\mathbf{g}(\boldsymbol{\alpha}_r)=\boldsymbol{\tau}\,. \tag{3}$$

In (3) $\boldsymbol{\alpha}_r=[\alpha_{1r}\ \alpha_{2r}\ \alpha_{3r}]^T$ is the vector of active joints angles on the rotor side. Control was done on the rotor side, and gear box transforms rotor position to the shaft position. With $\mathbf{M}$ inertia matrix is marked, $\mathbf{b}$ is the vector of nonlinear friction and Coriolis forces, vector $\mathbf{g}$ represents gravitational forces, and $\boldsymbol{\tau}$ is vector of the torques applied to the joints. Using the approach discussed in [17] dynamic equations for each active joint can be written as

$$\begin{aligned}
&m_{jj\,n}\ddot{\alpha}_{jr}=\tau_j-\tau_{dis\,j}=K_{Tj\,n}\left(i_j-i_{dis\,j}\right),\\
&\tau_{dis\,j}=K_{Tj\,n}\cdot i_{dis\,j},\\
&\tau_{dis\,j}=\Delta m_{jj}\cdot\ddot{\alpha}_{jr}+\sum_{\substack{k=1\\k\neq j}}^{3}m_{jk}\ddot{\alpha}_{kr}+b_j(\boldsymbol{\alpha}_r,\dot{\boldsymbol{\alpha}}_r)+g_j(\boldsymbol{\alpha}_r)-\Delta K_{Tj}\cdot i_j,
\end{aligned} \tag{4}$$

where $m_{jj\,n}$ is the constant nominal value of the j-th joint inertia (j = 1, 2, 3), and Δm_{jj} is its change that depends on the configuration, K_{Tjn} and ΔK_{Tj} are nominal torque constant of the j-th joint motor and its change, respectively, and $\tau_{dis\ j}$ is disturbance torque. For the dynamics (4) acceleration control [17] with asymptotic convergence can be applied. To track the reference α_{jr}^{ref} j-th joint control current is formed as

$$i_j=m_{jj\,n}/K_{Tj\,n}\left[\hat{\ddot{\alpha}}_{jr}^{ref}-10^2\left(\hat{\dot{\alpha}}_{jr}-\hat{\dot{\alpha}}_{jr}^{ref}\right)-10^3\left(\alpha_{jr}-\alpha_{jr}^{ref}\right)\right]+\hat{i}_{dis\,j}. \tag{5}$$

In (5) $\hat{\dot{\alpha}}_{jr}^{ref}$ and $\hat{\ddot{\alpha}}_{jr}^{ref}$ stand for filtered reference velocity and acceleration calculated as in (6) with g_{ref} = 1200

$$\hat{\dot{\alpha}}_{jr}^{ref}=\left(g_{ref}-\frac{g_{ref}^{\ 2}}{s+g_{ref}}\right)\alpha_{jr}^{ref},\ \hat{\ddot{\alpha}}_{jr}^{ref}=\left(g_{ref}-\frac{g_{ref}^{\ 2}}{s+g_{ref}}\right)\hat{\dot{\alpha}}_{jr}^{ref}. \tag{6}$$

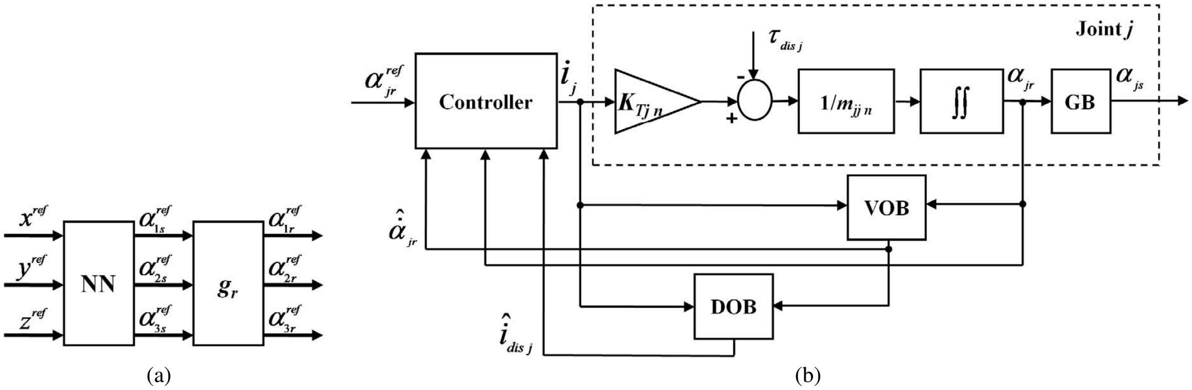

Fig. 6. Reference transformation (a), control system structure (b)

With α_{jr} j-th joint measured angle is marked. Velocity observer (VOB) [18] is used to get estimated joint velocity on the rotor side $\hat{\dot{\alpha}}_{jr}$. Disturbance observer (DOB) [19] based on control current i_j and estimated velocity $\hat{\dot{\alpha}}_{jr}$ is used to estimate disturbance current, i.e. to calculate $\hat{i}_{dis\,j}$. Control (5) will enforce asymptotic convergence to the reference angle.

In the control system, reference Cartesian coordinates are transformed by the neural network to the reference joint angles on the shaft side α_{js}^{ref}. Using known gear box (GB) ratio g_r, reference angles on the rotor side are calculated (Fig. 6a), and control (5) is applied. Control system structure for j-th joint (j = 1, 2, 3) is given in Fig. 6b.

5. Experimental Results

Two different experiments were done to validate neural network utilization in inverse kinematics modeling and designed control system performance. Control algorithms are executed on dSPACE1005 control platform with 10 (kHz) control loop frequency. Joints are actuated by the Faulhaber 1628024B brushless DC servomotors driven by Faulhaber MCBL 3006S drivers. In these experiments reference Cartesian coordinates of the robot's end effector were transformed to shaft side joint references using designed NN. Joint angles were measured with encoders that provide resolution of 0.01 (degrees) on shaft side.

In the first experiment reference trajectory in Cartesian coordinates, in meters, were described as $x^{ref} = 0.0045\sin(2\pi ft)$, $y^{ref} = 0.003\cos(2\pi ft)$, $z^{ref} = -0.075+0.002\sin(2\pi ft)$, $f = 0.2$ (Hz), and it is depicted in Fig. 7. Responses are given in Fig. 8.

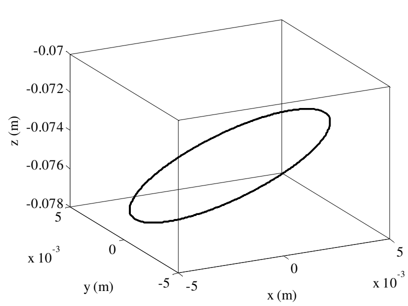

Fig. 7. Reference trajectory for the first experiment

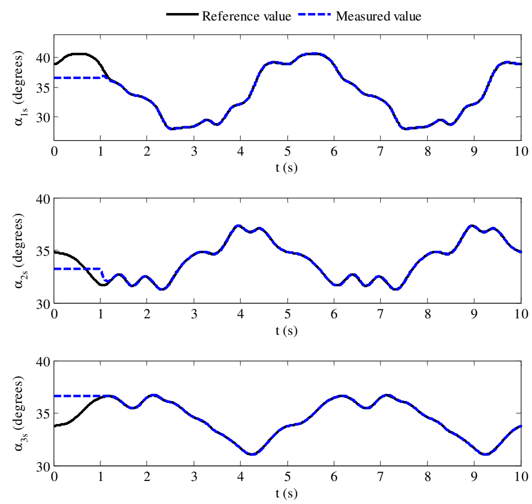

Fig. 8. Angle responses for the first experiment

Reference signals are not simply harmonic signals, as they would be if the mathematical model of inverse kinematics was applied. Due to the discussed reasons, that mapping is not proper, and NN is applied and it provided shown reference signals. Measured signals asymptotically converge to reference after control is activated at time instant t = 1 (s), with no overshoot. Tracking error is within 1.1 % of peak-to-peak reference signal for all joints.

In the second experiment reference trajectory was described as $x^{ref} = 0.003\sin(2\pi ft)$, $y^{ref} = 0.003\cos(4\pi ft)$, $z^{ref} = -0.075+$, $f =$ 0.2 (Hz), and it is depicted in Fig. 9. Angle responses are shown in Fig. 10. Similar conclusions can be made for second experiment, as for the first one. Control is activated at time instant t = 1 (s), and angles asymptotically converge to its references. Tracking is satisfactory, as tracking errors are within 0.7 % of peak-to-peak references.

6. Conclusion and Future Work

In this paper it was shown how neural networks can provide significantly better mapping between manipulator's task space and joint space, compared to the mathematical model of the kinematics. They were used in combination with acceleration control framework, and control performance was satisfactory.

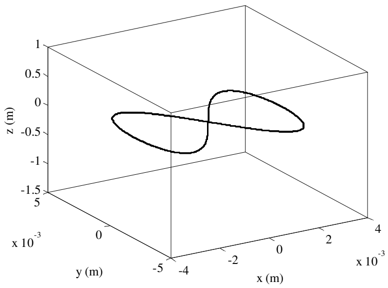

Fig. 9. Reference trajectory for the second experiment

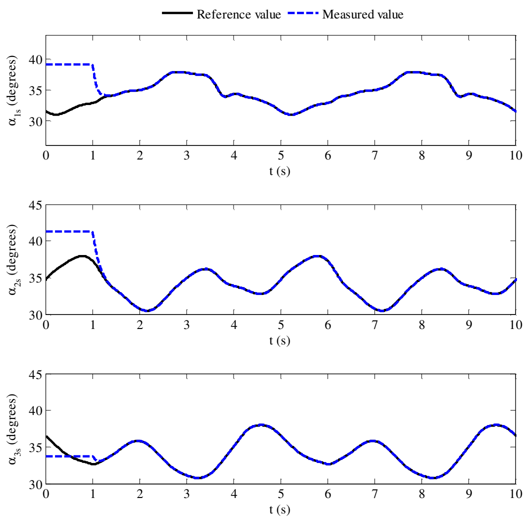

Fig. 10. Angle responses for the second experiment

Mathematical model of the kinematics is always based on the assumptions that manufacturing, assembly, and long operation time did not cause any differences between designed and real operating physical object. Unfortunately this is rarely the case, and neural networks can be used to deal with these things.

Further investigation of the neural networks application in kinematics modeling will be part of our future work. Neural networks implementation on FPGA platform could also be part of our future work, as neural network are suitable for execution on parallel hardware.

Acknowledgment

Authors acknowledge Yousef Jameel Scholarship Fund, TUBITAK Bideb 2215, TUBITAK Bideb, TUBITAK Project number 111M359, TUBITAK Project number 110M425 for partial financial support.

References

[1] R. Clavel, "Une nouvelle structure de manipulateur parallèle pour la robotique légère", *APII 23*, pp. *501-519,* 1989.

[2] IRB 360 FlexPicker, ABB, October 2013, [online] http://new.abb.com/products/robotics/industrial-robots/irb-360

[3] Tripod EXPT, FESTO, June 2013, [online] http://www.festo.com/cms/nl-be_be/17009.htm

[4] Paloma D2, Bosch, June 2013, [online] http://www.boschpackaging.com/en/pa/products/industries/pd/product-detail/paloma-d2 12225.php?ind=1674&mt=14030&tg=17538

[5] E. D. Kunt, A. T. Naskali, and A. Sabanovic, "Miniaturized modular manipulator design for high precision assembly and manipulation tasks", in *The 12th IEEE International Workshop on Advanced Motion Control*, Sarajevo, B&H, 2012.

[6] D. Yu, "Parallel robots pose accuracy compensation using back propagation network", *Int. Journ. of the Phy. Sci.*, vol. 6(21)*,* pp. 5005-5011, Sep., 2011.

[7] S. Kieffer, V. Morellas, and M. Donath, "Neural network learning of the inverse kinematic relationships for a robot arm", in *International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 2418-2425.

[8] Z. Bingul, H. M. Ertunc, and C. Oysu, "Comparison of inverse kinematics solutions using neural network for 6R robot manipulator with offset", in *2005 ICSC Congress on Computational Intelligence Methods and Applications*, Istanbul, Turkey, 2005.

[9] S. F. M. Assal, K. Watanabe, and K. Izumi, "Neural network-based kinematic inversion of industrial redundant robots using cooperative fuzzy hint for the joint limits avoidance", *IEEE/ASME Trans. on Mech.,* vol. 11, no. 5, pp. 593-603, Oct., 2006.

[10] Y. Yang, G. Peng, Y. Wang, H. Zhang, "A new solution for inverse kinematics of 7-DOF manipulator based on neural network", in *IEEE International Conference on Automation and Logistics*, Jinan, CHN, 2007.

[11] S. Lee, R. M. Kil, "Robot kinematic control based on bidirectional mapping neural network", in *IJCNN International Joint Conference on Neural Networks*, San Diego, CA, 1990.

[12] D. Huang, L. Chen, "Inverse kinematic control of free floating space robot system based on a mutual mapping neural network", in *7th World Congress on Intelligent Control and Automation*, Chongqing, CHN, 2008.

[13] Y. Zhang, J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network", *IEEE/ASME Trans. on Syst. Man and Cybernetics – part B: Cybernetics*, vol. 34, no. 1, pp. 752-759, Feb., 2004.

[14] D. Zhang, Z. Gao, P. Jiang, "A novel calibration method of parallel kinematic manipulators based on multi-population coevolutionary neural network", in *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, Paris, FR, 2011, pp. 205-211.

[15] D. Zhang, Z. Gao, "Optimal kinematic calibration of parallel manipulators with pseudoerror theory and cooperative coevolutionary network", *IEEE Trans. on Indust. Electr.,* vol. 59, no. 8, pp. 3221-3231, Aug., 2012.

[16] E.D. Kunt, "Microfactory concept with bilevel modularity", PhD thesis, Sabanci Univ., Ist., TR, 2012.

[17] A. Sabanovic and K. Ohnishi, "Motion control systems", John Wiley & Sons (Asia), Singapore, 2011.

[18] E. A. Baran, E. Golubovic, and A. Sabanovic, "Functional observers for motion control systems", *Automatika*, vol. 54, no. 2, pp. 231-241, May, 2013.

[19] T. Murakami, K. Ohnishi, "Observer-based motion control applications to robust control and parameter identification", in *Asia-Pacific Workshop on Advances in Motion Control*, 1993, pp. 1-6.